

Mobile Link クイックスタート

バージョン 16.0

2013 年 2 月

バージョン 16.0 2013 年 2 月

© 2013 SAP AG or an SAP affiliate company.All rights reserved.

次の条件に従うかぎり、このマニュアルの全部または一部を使用、印刷、再生、配布することができます。1)マニュアルの全部または一部にかかわらず、すべてのコピーにこの情報またはマニュアル内のその他の版権と商標の表示を含めること。2)マニュアルに変更を加えないこと。3) SAP 以外の人間がマニュアルの著者または情報源であるかのように示す行為をしないこと。ここに記載された情報は事前の通知なしに変更されることがあります。

SAP AG およびディストリビュータが販売しているソフトウェア製品には、他のソフトウェアベンダー独自のソフトウェアコンポーネントが含まれているものがあります。国内製品の仕様は変わることがあります。

これらの資料は SAP AG および関連会社 (SAP グループ) が情報のみを目的として提供するものであり、いかなる種類の表明ま たは保証も行うものではなく、SAP グループはこの資料に関する誤りまたは脱落について責任を負わないものとします。SAP グループの製品およびサービスに関する保証は、かかる製品およびサービスに付属している明確な保証文書がある場合、そこ で明記されている保証に限定されます。ここに記載されているいかなる内容も、追加保証を構成するものとして解釈されるも のではありません。

ここに記載された SAP および他の SAP 製品とサービス、ならびに対応するロゴは、ドイツおよび他の国における SAP AG の商 標または登録商標です。http://www.sap.com/corporate-en/legal/copyright/index.epx#trademark を参照してください。

目次

はじめに	v
Mobile Link テクノロジ	1
Mobile Link 同期	1
Sybase Central の Mobile Link プラグイン	22
Mobile Link の CustDB サンプル	57
Mobile Link Contact サンプル	72
Mobile Link チュートリアル	85
チュートリアル:Mobile Link の概要	85
チュートリアル : SQL Anywhere 統合データベースでの Mobile Link の使	用 105
チュートリアル:Oracle Database 10g での Mobile Link の使用	118
チュートリアル:Adaptive Server Enterprise 統合データベースと Mobile	Э
Link の使用	136
チュートリアル : カスタムユーザ認証用の Java と .NET の使用	155
チュートリアル:ダイレクトローハンドリングの使用	164
チュートリアル:Microsoft Excel との同期	190
チュートリアル:XML との同期	210
チュートリアル:リモートデータベースの集中管理の使用	231
チュートリアル:スクリプトバージョン句を使用したスキーマの変更	257
チュートリアル:ScriptVersion 拡張オプションを使用したスキーマの変更	更.265
チュートリアル:Mobile Link リプレイユーティリティを使った複数の Mobile Link クライアントのシミュレート	271

索引		28	7
----	--	----	---



このマニュアルでは、セッションベースのリレーショナルデータベース同期システムである Mobile Link について説明します。Mobile Link テクノロジは、双方向レプリケーションを可能に し、モバイルコンピューティング環境に非常に適しています。

Mobile Link テクノロジ

この項では、Mobile Link 同期テクノロジの概要と、Mobile Link 同期テクノロジを使用して2つ 以上のデータベース間でデータをレプリケートする方法について説明します。

Mobile Link 同期

Mobile Link は、Ultra Light と SQL Anywhere のリモートデータベースを統合データベースと同期 させるために設計された、セッションベース同期テクノロジです。

Mobile Link アプリケーションの各部分

Mobile Link 同期では、多くのクライアントが Mobile Link サーバを介して中央のデータソースと 同期します。



- Mobile Link クライアント クライアントは、モバイルデバイス、サーバ、デスクトップコン ピュータ、またはスマートフォンにインストールできます。Ultra Light データベースと SQL Anywhere データベースの2種類のクライアントがサポートされています。1つの Mobile Link インストール環境では、これらのうちのどちらか1つまたは両方を使用できます。 「Mobile Link クライアント」『Mobile Link クライアント管理』を参照してください。
- **ネットワーク** Mobile Link サーバと Mobile Link クライアント間の接続では、複数のプロト コルを使用できます。 次の項を参照してください。
 - Mobile Link サーバ: 「-x mlsrv16 オプション」『Mobile Link サーバ管理』
 - O Ultra Light と SQL Anywhere クライアント:「Mobile Link クライアントネットワークプロ トコルオプション」『Mobile Link クライアント管理』
- Mobile Link サーバ 同期処理を管理し、すべての Mobile Link クライアントと統合データ ベースサーバ間のインタフェースを提供します。「Mobile Link サーバ」『Mobile Link サーバ 管理』を参照してください。
- 統合データベース 統合データベースには通常、同期システムのアプリケーション情報の中 核となるコピーが収められています。また、通常は、Mobile Link 同期に必要なシステムテー

ブルとプロシージャ、および同期するために必要なステータス情報も格納されます。 「Mobile Link 統合データベース」『Mobile Link サーバ管理』を参照してください。

● ステータス情報 Mobile Link サーバは通常、統合データベース内のシステムテーブルの同期 情報を管理します。情報の管理は、ODBC 接続を介して行われます。

また、ステータス情報を別のデータベースに格納することもできます。「Mobile Link システ ムデータベース」『Mobile Link サーバ管理』を参照してください。

- SQL ローハンドリング Mobile Link サーバ用に SQL スクリプトを作成すると、サーバでは これらのスクリプトを使用し、ODBC 接続を介して、統合データベースとの間でローが転送 されます。「サーバ側の同期論理の作成オプション」11 ページを参照してください。
- **ダイレクトローハンドリング** Mobile Link のダイレクトローハンドリングを使用して、統合 データベース以外にオプションで他のデータソースと同期することもできます。「ダイレク トローハンドリング」『Mobile Link サーバ管理』を参照してください。
- 同期スクリプト リモートデータベースの各テーブルに対して同期スクリプトを記述し、これらのスクリプトを統合データベースの Mobile Link システムテーブルに保存してください。これらのスクリプトは、アップロードデータに対して行う処理や、ダウンロードするデータを決定します。2 種類のスクリプトがあります。テーブルスクリプトと接続レベルスクリプトです。次の項を参照してください。
 - ○「Mobile Link イベントの概要」『Mobile Link サーバ管理』
 - ○「同期スクリプトの作成」『Mobile Link サーバ管理』
 - ○「同期イベント」『Mobile Link サーバ管理』
 - ○「サーバ側の同期論理の作成オプション」11ページ

Mobile Link の機能

Mobile Link 同期には高い適応性と柔軟性があります。以下に、主な機能の一部を示します。

機能

- 使い始めるのが簡単 [同期モデル作成ウィザード] を使用すると、簡単に同期アプリケーションを作成できます。このウィザードによって、複雑な同期システムに伴う多くの難解な実装作業が処理されます。Sybase Central を使用すると、同期モデルをオフラインで表示し、 簡単なインタフェースで変更を行い、展開オプションを使用してモデルを統合データベースに展開できます。
- モニタとレポート Mobile Link には、同期をモニタする3つのメカニズムが用意されていま す。Mobile Link プロファイラ、Mobile Link用 SQL Anywhere モニタ、統計スクリプトです。
- パフォーマンスチューニング Mobile Link のパフォーマンスをチューニングするための複数 のメカニズムがあります。たとえば、競合レベル、アップロードのキャッシュサイズ、デー タベース接続数、ロギングの冗長性、または BLOB のキャッシュサイズを調整できます。
- スケーラビリティ Mobile Link はスケーラビリティに優れ、堅牢な同期プラットフォームで す。Mobile Link の単一のサーバが数千の同期を同時に処理したり、負荷分散を使用して複数

の Mobile Link サーバを同時に稼働したりできます。Mobile Link サーバはマルチスレッド化 されており、統合データベースで接続プールを使用します。

- セキュリティ Mobile Link には、既存の認証に統合できるユーザ認証、暗号化、安全な証明 書の交換によって機能するトランスポートレイヤセキュリティなど、豊富なセキュリティオ プションがあります。Mobile Link には、FIPS 認定のセキュリティオプションもあります。
- Relay Server と Sybase Relay Server のホスティングサービス Relay Server は、Web サーバ を通じて通信するモバイルデバイスとバックエンドサーバ間で、安全で負荷分散された通信 を実現します。「Relay Server の概要」『Relay Server』を参照してください。

Sybase Relay Server のホスティングサービスは、Sybase をホストとする Relay Server のファー ムです。特に公共無線ネットワークを使用してデータを送信する場合に、Mobile Link データ 同期を使用するモバイルアプリケーションの開発と評価が従来より容易になります。 「Sybase ホストのリレーサービス」『Relay Server』を参照してください。

次の図は、Mobile Link 環境への Relay Server の組み込み方を示しています。



アーキテクチャ

● データ調整 Mobile Link によって、データの特定部分を同期対象として選択できます。また、Mobile Link 同期では、異なるデータベースで行われた変更内容の競合を解決できます。同期処理は、SQL、Java または.NET アプリケーションとして作成できる同期論理によって制御されます。この論理の各部は、「スクリプト」と呼ばれます。スクリプトを使用すると、たとえば、アップロードされたデータを統合データベースに適用する方法の指定やダウンロード内容を取得するデータベースの指定を行ったり、統合データベースとリモートデータベースとで異なるスキーマや名前を処理したりできます。イベントベースのスクリプト機能により、競合解決、エラーレポート、ユーザ認証などの機能を含め、同期処理の設計がきわめて柔軟になります。

● 双方向の同期 すべてのロケーションでデータベースを変更できます。

- **アップロード専用の同期またはダウンロード専用の同期** デフォルトでは、同期は双方向で、 アップロードとダウンロードの両方が行われます。ただし、アップロード専用の同期または ダウンロード専用の同期を選択することもできます。
- ファイルベースのダウンロード ダウンロード内容はファイルとして配布することが可能で あり、同期の変更をオフラインで配布できます。これには、適切なデータの適用を保証する 機能が含まれます。
- サーバ起動同期 Mobile Link 同期は、統合データベースから開始できます。これは、データの更新をリモートデータベースにプッシュし、リモートデータベースによってデータが統合データベースにアップロードされるようにできることを意味しています。 Mobile Link サーバ起動同期を参照してください。

サーバ起動同期への代替機能としてサーバ起動リモートタスク (SIRT) を使用できます。詳細については、「リモートデータベースの集中管理」『Mobile Link サーバ管理』と「サーバ起動リモートタスク (SIRT)」『Mobile Link サーバ管理』を参照してください。

- 複数のネットワークプロトコル 同期は TCP/IP、HTTP、HTTPS 経由で実行できます。 Windows Mobile デバイスは、Microsoft ActiveSync を使用して同期できます。
- セッションベース すべての変更内容は、単一のトランザクションでアップロードし、単一のトランザクションでダウンロードできます。同期が成功するたびに、統合データベースとリモートデータベースが一貫した状態になります。(トランザクションの順序を保持する場合は、リモートデータベースの各トランザクションを別個のトランザクションとしてアップロードすることもできます。)

トランザクション全体が同期されるか、トランザクション全体がまったく同期されないかの どちらかになります。これにより、各データベースでトランザクション単位の整合性が確保 されます。

- データの一貫性 Mobile Link は、緩やかな一貫性方式を使用しています。つまり、変更内容 はすべて一貫性が保たれるように各サイトで同期されますが、時間的にはわずかなズレがあ るため、ある瞬間だけを見ると、各サイトに存在するデータのコピーが異なる場合もありま す。
- **多様なハードウェアとソフトウェアのプラットフォーム** Mobile Link の統合データベースと して、各種の一般的なデータベース管理システムを使用できます。また、Mobile Link サー バ API を使用して任意のデータソースへの同期を定義することもできます。リモートデー タベースには、SQL Anywhere または Ultra Light を使用できます。Mobile Link サーバは、 Windows、UNIX、Linux、Mac OS X 上で動作します。SQL Anywhere は、Windows、Windows Mobile、UNIX、Linux、Mac OS X 上で動作します。Ultra Light は、Windows Mobile または BlackBerry 上で動作します。「サポートされるプラットフォーム」『SQL Anywhere 16 紹介』 を参照してください。
- Mobile Link 監視サーバ Mobile Link 監視サーバは、サーバファーム内の1つの Mobile Link サーバだけがプライマリサーバとして動作するようにします。これにより、サーバ起動同期 環境での冗長な通知が回避されます。次の図は、サーバファーム環境の Mobile Link 監視サー バを示しています。



Mobile Link のクイックスタート

Mobile Link は、1 つまたは複数の中央のデータソースと断続的に接続する多数のリモートアプリ ケーション間でデータを同期するように設計されています。基本的な Mobile Link アプリケー ションでは、リモートクライアントは SQL Anywhere または Ultra Light のデータベースで、中央 のデータソースはサポートされている ODBC 準拠のリレーショナルデータベースのいずれかで す。Mobile Link サーバ API を使用してこのアーキテクチャを拡張すると、サーバ側の同期先の 制限を事実上なくすことができます。

すべての Mobile Link アプリケーションで、Mobile Link サーバが同期処理の主要要素です。通常 は、Mobile Link リモートサイトで Mobile Link サーバへの接続を開くと、同期が開始されます。 同期中に、リモートサイト側の Mobile Link クライアントは、前回の同期後にリモートデータベー スに対して行われたデータベースの変更をアップロードできます。Mobile Link サーバは、この データを受信すると、統合データベースを更新し、変更内容を統合データベースからリモート データベースにダウンロードできます。

Mobile Link アプリケーションの開発を始める最も簡単な方法は、[同期モデル作成ウィザード] を使用することです。このウィザードを使用すると、以下で説明している手順の大部分がウィ ザードにより処理されます。 「同期モデル」29 ページを参照してください。 ただし、Mobile Link モデルを使用する場合でも、Mobile Link 同期の処理とコンポーネントは理解しておく必要があります。

Mobile Link アプリケーションの概要

1. 統合データベースを設定

○データベースに対して設定スクリプトを実行し、Mobile Link 同期に必要なシステムオブ ジェクトを追加します。または、これらのオブジェクトを格納するためのシステムデータ ベースを別途作成します。

「Mobile Link 統合データベース」『Mobile Link サーバ管理』を参照してください。

2. リモートデータベースを設定

- ○リモートデータベースとしては SQL Anywhere と Ultra Light のどちらも使用できます。また、両方を組み合わせて使用することもできます。
- ○リモートデータベースで Mobile Link ユーザを作成します。 「Mobile Link ユーザ」『Mobile Link クライアント管理』を参照してください。
- ○SQL Anywhere のリモートデータベースでアップロードを特定するには、パブリケーション とサブスクリプションを作成します。「パブリケーション」『Mobile Link クライアント管 理』を参照してください。

Ultra Light のリモートデータベースでアップロードを特定するには、パブリケーションを 作成します。「Ultra Light でのデータのパブリッシュ」『Ultra Light データベース管理とリ ファレンス』を参照してください。

- 3. サーバ同期論理を作成して、アップロードの適用形式を特定 「同期スクリプトの作成」『Mobile Link サーバ管理』を参照してください。
- タイムスタンプベースの同期を設定して、前回のダウンロード以降に変更されたデータをダウン ロード 「タイムスタンプベースのダウンロードの実装」『Mobile Link サーバ管理』を参照 してください。
- 5. **Mobile Link サーバを起動** 「Mobile Link サーバ」『Mobile Link サーバ管理』を参照してく ださい。
- 6. クライアントの同期を開始

○SQL Anywhere リモートデータベースの詳細については、「同期の開始」『Mobile Link クラ イアント管理』を参照してください。

○Ultra Light リモートデータベースの詳細については、「Ultra Light クライアントの同期設計」 『Ultra Light データベース管理とリファレンス』を参照してください。

入門情報

- ●「Mobile Link 同期」1 ページ
- ●「同期の方法」『Mobile Link サーバ管理』

チュートリアル

- 「Mobile Link の CustDB サンプル」 57 ページ
- 「Mobile Link Contact サンプル」72 ページ
- ●「チュートリアル: Mobile Link の概要」85ページ
- ●「チュートリアル: SQL Anywhere 統合データベースでの Mobile Link の使用」105ページ
- ●「チュートリアル: Ultra Light の CustDB サンプルアプリケーションの構築」『Ultra Light デー タベース管理とリファレンス』
- ●「チュートリアル: Oracle Database 10g での Mobile Link の使用」118 ページ
- ●「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」 136 ページ
- ●「チュートリアル:カスタムユーザ認証用の Java と.NET の使用」155ページ
- ●「チュートリアル:ダイレクトローハンドリングの使用」164ページ
- ●「チュートリアル: Microsoft Excel との同期」190ページ
- ●「チュートリアル: XML との同期」210ページ
- ●「チュートリアル:リモートデータベースの集中管理の使用」231ページ
- ●「チュートリアル:スクリプトバージョン句を使用したスキーマの変更」257ページ
- ●「チュートリアル: ScriptVersion 拡張オプションを使用したスキーマの変更」265ページ
- ●「チュートリアル: Mobile Link リプレイユーティリティを使った複数の Mobile Link クライア ントのシミュレート」271ページ

クイックスタートのためのその他の資料

- ●Mobile Link には、Mobile Link 機能を確かめるために調べたり実行したりできるサンプルが数 多く用意されています。Mobile Link のサンプルは、製品とともに %SQLANYSAMP16% ¥MobiLink にインストールされます。
- ●Mobile Link のコード交換サンプルは、http://www.ianywhere.jp/dl/dl_evl.html にあります。ダウ ンロードにはアイエニウェアホームページのユーザ登録が必要です。

Mobile Link アプリケーションの設計

データベースアプリケーションには、次の2つの基本的なアーキテクチャがあります。

- オンラインアプリケーション オンラインアプリケーションユーザが統合データベースに直 接接続してデータを更新します。接続できない場合、ユーザによる作業はできません。
- 随時接続スマートクライアントアプリケーション 各ユーザがローカルデータベースを保有しています。各ユーザは接続状態にかかわらず常に自分のデータベースアプリケーションを使用できます。また、このデータベースアプリケーションはシステム内の他のデータベースと同期されます。

Mobile Link では随時接続スマートクライアントアプリケーションを作成できます。スマートク ライアントアプリケーションにより、アプリケーションの利便性、効率性、スケーラビリティが 大幅に向上します。しかし、アプリケーションの開発に関する新しい問題も発生します。このセ クションでは、スマートクライアントアプリケーションの開発に関する主な問題について説明し ます。また、Mobile Link 同期環境でソリューションを実装する方法についても説明します。

必要な同期のみを実行

大部分のアプリケーションでは、リモートデバイスのデータを一部だけでも更新する場合に毎回 統合データベース全体をダウンロードすると、大変なことになります。必要な時間と帯域が膨大 なものとなり、システム全体の動作が停止してしまいます。各ユーザにとって必要なアップロー ドとダウンロードのみを行うようにするための方法は複数あります。

まず、各リモートデータベースには統合データベースのテーブルとカラムのサブセットのみが格納されるようにします。たとえば、地域 A の販売担当者が必要とするテーブルやカラムは地域 B の販売担当者や管理職とは異なる場合があります。

リモートデータベースで作成したテーブルやカラムのうち、同期が必要なものだけを同期対象と して指定します。Mobile Link アプリケーションでは、データ型が一致していれば、名前が異なっ ていてもテーブルやカラムをマッピングすることができます。デフォルトではデータのアップ ロードとダウンロードの両方が可能ですが、Mobile Link では特定のカラムをアップロード専用 またはダウンロード専用にすることもできます。

同期時には、各ユーザに関連するリモートデータベースにのみローをダウンロードするようにします。ダウンロードをリモートデータベース別、ユーザ別、またはその他の基準別に分割して行うこともできます。たとえば、地域 A の販売担当者が地域 A のデータのみを更新する必要がある場合を考えてみます。

更新する必要があるのは変更があるデータのみです。Mobile Link アプリケーションでは、アッ プロードはトランザクションログに基づいて行われるため、デフォルトではリモートデータベー スで変更されているデータのみがアップロードされます。ダウンロードも同様に行うには、同期 形式をタイムスタンプベースに指定して、データが正常にダウンロードされた日時をシステムが 記録するようにします。これにより、データのダウンロードはその日時以降に変更があった場合 に限られます。

また、高優先度同期方式の実装が必要になる場合もあります。緊急のデータは更新頻度が高くな るようスケジュールし、緊急でないデータは夜間やデバイスがクレードルにある間に更新するよ うスケジュールします。高優先度同期を実装するには、それぞれ異なる時刻に実行するようスケ ジュールされた複数のパブリケーションを作成します。

この他に、プッシュ同期により、必要に応じてデータを効果的にリモートデバイスにプッシュダ ウンすることもできます。たとえば、トラック運送会社の配送係が交通の混乱を知らされた場合 に、該当地域に向かっているトラック運転手の更新情報をダウンロードできます。Mobile Link では、これをサーバ起動同期と呼んでいます。

アップロードの競合の処理

倉庫を例にとって考えてみます。各従業員はモバイルデバイスを保有しており、箱の搬入出時に 在庫情報を更新します。最初に100個の箱が搬入されています。そのため各従業員のリモート データベースと統合データベースには100と登録されています。デービッドが20個の箱を搬出 しました。彼は自分のデータベースを更新して同期します。これで彼のデータベースと統合 データベースの両方に80と登録されます。次にスーザンが10個の箱を搬出します。ここで スーザンは自分のデータベースを更新して同期しようとしますが、スーザンのアプリケーション は統合データベースに登録されている箱の個数が80ではなく100であると想定しています。こ のため、アップロードの競合が発生します。 この倉庫アプリケーションの例では、この問題を解決するには、「デービッドによる更新値 - (最 初の値 - スーザンによる更新値) = 正しい値」となるように、次のような競合解決論理を作成す る必要があります。

80 - (100 - 90) = 70

この競合解決論理は倉庫などの在庫ベースのアプリケーションでは有効ですが、すべてのビジネ スアプリケーションで適しているわけではありません。Mobile Link では、次の競合解決論理を 定義できます。

- **在庫モデル** ローを更新してユニット数を修正します。
- 日付 最新の更新が適用されます(値がデータベースで変更された日付が基準です。値が同 期された日付ではありません)。
- ●操作者 たとえば、管理者を常に優先したり、レコードの所有者を常に優先したりします。
- **カスタム** その他実装が必要なビジネス用論理です。

場合によっては、アップロードの競合が発生しないようにシステムを設計することができます。 重複を避けるためにデータがリモートで分割されている場合は、競合を回避できる可能性があり ます。それでも競合が発生する場合は、競合の検出と解決を行うためのプログラムを作成する必 要があります。

ユニークなプライマリキー

データをアップロードし、アップロードの競合を検出して、削除されたローを統合データベース で同期するには、データベースシステム内で同期されているすべてのテーブルにユニークなプラ イマリキーが必要です。各ローには、データベース内だけでなく、データベースシステム全体で ユニークなプライマリキーが必要です。プライマリキーは更新できないようにする必要があり ます。

Mobile Link では、ユニークなプライマリキーを設定するための方法が複数あります。方法の1つとして、プライマリキーのデータ型を GUID に設定することがあげられます。GUID (グローバルユニーク識別子) は 16 バイトの 16 進数です。Mobile Link の NEWID 関数を使用すると、新しいローに対して自動的に GUID を作成できます。

また、別の解決方法として、複合キーを使用することがあげられます。Mobile Link では各リモー トデータベースにリモート ID と呼ばれるユニークな値が設定されています。リモート ID と通 常のプライマリキー (順序数など)を組み合わせたものをプライマリキーとして使用することが できます。

SQL Anywhere ではグローバルオートインクリメント方式による解決方法もあります。あるカラ ムを GLOBAL AUTOINCREMENT として宣言すると、ローの追加時に、これまでのプライマリ キーの最後の値を増分することにより、プライマリキーが自動的に作成されます。この解決方法 は、統合データベースが SQL Anywhere の場合に最適です。

最後に、リモートデータベースに配布されるプライマリキー値のプールを作成することもできま す。

同期ソリューションの開発時における各種の決定と同様に、どのプライマリキー方式を選択する かは、統合データベースとリモートデータベースに対する制御のレベルによって異なります。多 くの場合、リモートデータベースは管理なしで動作できる必要があります。また、統合データ ベースのスキーマを変更することが困難な場合もあります。さらに、統合データベースとして RDBMS を選択した場合、すべての RDBMS ですべての機能がサポートされているわけではない ため、使用できるオプションが限られることがあります。

削除の処理

同期方式に関する別の問題として、統合データベースから削除されたローの処理があげられま す。たとえば、統合データベースからあるローを削除したとします。次にデービッドが自分のリ モートデータベースを同期すると、この削除データがダウンロードされ、デービッドのデータ ベースからローが削除されます。しかし統合データベースでこの後どうすればよいでしょうか。 実際にはスーザンに対しても削除データのダウンロードが必要になるため、ローを削除できませ ん。

ダウンロード削除を処理する方法は2通りあります。1つ目の方法は、ローが削除されているか どうかを示すステータスカラムを各テーブルに追加することです。この場合、ローは実際には削 除されず、削除するようマークが付けられるだけです。削除するようマーク付けされたローは、 後ですべてのリモートデータベースが更新されていることを確認できた時点で消去できます。 また、各テーブルのシャドウテーブルを作成することもできます。シャドウテーブルには、削除 されたローのプライマリキーの値が格納されています。ローを削除すると、トリガによりシャド ウテーブルに値が入力されます。シャドウテーブルの値により、リモートデータベースで削除す るローが決定されます。

トランザクション

同期データベースシステムでは、コミットされたデータベーストランザクションのみが同期され ます。さらに、同期するデータが関わっている、コミットされたすべてのトランザクションを同 期する必要があります。この処理が正しく行われないと、エラーが発生します。これは Mobile Link でのデフォルトの動作です。

また、統合データベースへの接続の独立性レベルも考慮する必要があります。データの一貫性を 確保できる範囲で最高のパフォーマンスを実現できる独立性レベルを使用する必要があります。 通常、独立性レベル0(READ UNCOMMITTED)は同期には不適切で、データの不整合を引き起 こす可能性があります。

デフォルトでは、Mobile Link はアップロードには独立性レベル SQL_TXN_READ_COMMITTED を使用し、可能な場合には、ダウンロードにはスナップショットアイソレーションを使用しま す (不可能な場合には、SQL_TXN_READ_COMMITTED を使用します)。スナップショットアイ ソレーションは、トランザクションが統合データベースで閉じられるまでダウンロードがブロッ クされる問題を解消します。ただし、すべての RDBMS がこの機能をサポートしているわけでは ありません。

夏時間

毎年、夏時間から通常時間への移行時にデータベースの同期に関する問題が発生する可能性があります。秋に時刻が1時間戻ると、午前2時が午前1時になります。午前1時と午前2時の間に同期を実行しようとすると、同期のタイムスタンプが、たとえば移行前の午前1時15分なのか、移行後の午前1時15分なのかわからなくなります。

この問題を解決するには、秋になり時間が移行するときに、1時間の間シャットダウンするか、 統合データベースサーバを協定世界時(UTC)にします。 参照

●「同期の方法」『Mobile Link サーバ管理』

Mobile Link アプリケーションの開発オプション

Mobile Link では、アプリケーション開発のためのさまざまな方法が用意されています。それぞれの方法を単独で使うことも、組み合わせて使うこともできます。

 ● 同期モデル作成ウィザード このウィザードを使用すると、アプリケーションの開発作業を 順を追って簡単に実行できます。まずスキーマがある中央データベースを作成し、その後で リモートデータベースと同期に必要なスクリプトを作成できます。また、このウィザードで はダウンロード削除などの処理を行うためのシャドウテーブルを統合データベース上に作成 できます。ウィザードが完了したら、モデルをさらにカスタマイズできます。同期モデル展 開ウィザードでは、データベースとテーブルの作成、Mobile Link システムテーブルの更新、 Mobile Link ユーティリティを実行するためのスクリプトの作成ができます。

展開した Mobile Link モデルがさらにカスタマイズを必要とする場合は、次のいずれかの方法 で変更を加えることができます。

- Sybase Central Sybase Central の Mobile Link 16 プラグインを使用して、Mobile Link アプリ ケーションのすべての要素を更新できます。
- システムプロシージャ 中央データベースが統合データベースとして動作するよう設定する と、Mobile Link 同期で使用するシステムオブジェクトが作成されます。この中には Mobile Link システムテーブルも含まれます。ここにはサーバ側の Mobile Link アプリケーションの 大部分が格納されます。また、この中には、Mobile Link スクリプトへの Mobile Link システ ムテーブルの挿入やリモートユーザの登録などの作業を行うためのシステムプロシージャや ユーティリティも含まれます。
- Mobile Link システムテーブルの直接操作 上級ユーザの場合は、Mobile Link システムテーブ ルのデータの追加、削除、更新を直接行うこともできます。この操作を行うには、Mobile Link の仕組みを詳細に理解している必要があります。

参照

- 「Sybase Central の Mobile Link プラグイン」 22 ページ
- ●「同期モデルタスク」32ページ
- ●「Mobile Link サーバシステムプロシージャ」『Mobile Link サーバ管理』
- ●「Mobile Link ユーティリティ」『Mobile Link サーバ管理』
- ●「Mobile Link サーバのシステムテーブル」『Mobile Link サーバ管理』

サーバ側の同期論理の作成オプション

Mobile Link 同期スクリプトは、SQL で記述することも、Java (Java 用 Mobile Link サーバ API を 使用) または .NET (.NET 用 Mobile Link サーバ API を使用) で記述することもできます。

サポートされている統合データベースと同期する場合は、通常は SQL 同期論理が最適です。

サポート対象外の統合データベースと同期する場合は、Java や.NET が便利です。また、SQL 言語の制限事項やデータベース管理システムの機能によって設計が制限されている場合や、異なる RDBMS タイプ間での移植性が必要な場合も、Java や.NET が便利です。

Java と.NETの同期論理は、SQL 論理と同様に機能します。Mobile Link サーバは、Mobile Link イベントの発生時に SQL スクリプトにアクセスするのと同様に、Java メソッドや.NET メソッ ドを呼び出すことができます。Java または.NET を使用している場合は、一部の追加処理を実行 するイベントを使用できます。ただし、アップロードローやダウンロードローを直接処理するイ ベントのスクリプトを処理するときは、Java または.NET の実装が SQL 文字列を返す必要があ ります。ダイレクトローハンドリングで使用される2つのイベントを除き、Java と.NET の同期 論理では、アップロードとダウンロードに直接アクセスできません。Java または.NET から SQL 文字列で返された内容を Mobile Link が実行します。

ダイレクトローハンドリングでは、handle_UploadData イベントと handle_DownloadData イベント を使用して、データソースと同期します。この操作により、アップロードローとダウンロード ローが「直接操作されます」。

Java または .NET でのスクリプトの作成を検討する場合のシナリオを以下に示します。

- ダイレクトローハンドリング Java と.NET の同期論理では、Mobile Link を使用して、統合 データベース以外のデータソース (アプリケーションサーバ、Web サーバ、ファイルなど) に アクセスできます。
- 認証 ユーザ認証プロシージャを Java または .NET で記述し、Mobile Link 認証を企業のセキュリティポリシーに組み込みます。
- **ストアドプロシージャ** RDBMS でユーザ定義のストアドプロシージャを使用できない場合 は、Java や.NET でメソッドを作成できます。
- **外部呼び出し** プログラムが同期イベント中に外部サーバへの接続を必要とする場合は、 Java または.NET 同期論理を使用して、同期イベントによりトリガされるアクションを実行 できます。Java または.NET 同期論理は、複数の接続間で共有できます。
- 変数 データベースに変数を処理する機能がない場合は、接続または同期の間持続する変数 を Java や .NET で作成できます。また、SQL スクリプトでユーザ定義の名前付きパラメータ を使用することもできます。この方法は、すべてのタイプの統合データベースに使用できま す。「ユーザ定義の名前付きパラメータ」『Mobile Link サーバ管理』を参照してください。

Mobile Link サーバ API

Java と .NET 同期論理は、Mobile Link サーバ API を介して使用できます。Mobile Link サーバ API は、Mobile Link 同期用のクラスとインタフェースのセットです。

Java 用 Mobile Link サーバ API には、次の利点があります。

●統合データベースへの既存の ODBC 接続に JDBC 接続としてアクセスできます。

●JDBC、Web サービス、JNI などのインタフェースを使用して、別のデータソースにアクセス できます。

- ●統合データベースへの新規 JDBC 接続を作成し、現在の同期接続の外部でデータベースを変更 できます。たとえば、同期接続でロールバックを行う場合でも、これをエラーログや監査に使 用できます。
- ●統合データベースと同期する場合は、Java コードを作成してデバッグしてから Mobile Link サーバで実行できます。多くのデータベース管理システムの SQL 開発環境は、Java アプリ ケーションが使用可能な環境に比べると初歩的です。
- ●SQL ローハンドリングとダイレクトローハンドリングの両方を使用できます。
- ●高度で豊かな Java 言語が提供する多数の既存のコードやライブラリを使用できます。

「Mobile Link サーバ Java API リファレンス」『Mobile Link サーバ管理』を参照してください。

.NET 用 Mobile Link サーバ API には、次の利点があります。

- ●.NET から ODBC を呼び出す iAnywhere クラスを使用して、統合データベースへの既存の ODBC 接続にアクセスできます。
- ●ADO.NET、Web サービス、OLE DB などのインタフェースを使用して、別のデータソースに アクセスできます。
- ●統合データベースと同期する場合は、.NET コードを作成してデバッグしてから、Mobile Link サーバで実行できます。多くのデータベース管理システムの SQL 開発環境は、.NET アプリ ケーションが使用可能な環境に比べると初歩的です。
- ●SQL ローハンドリングとダイレクトローハンドリングの両方を使用できます。
- ●.NET Common Language Runtime (CLR) 内でコードが実行されるため、すべての.NET ライブラ リ (SQL ローハンドリングとダイレクトローハンドリングの両方を含む) にアクセスできま す。

「Mobile Link サーバ.NET API リファレンス」『Mobile Link サーバ管理』を参照してください。

参照

- ●「同期スクリプトの作成」『Mobile Link サーバ管理』
- ●「同期の方法」『Mobile Link サーバ管理』
- ●「Java による同期スクリプトの作成」『Mobile Link サーバ管理』
- ●「.NET の同期スクリプト」『Mobile Link サーバ管理』
- ●「ダイレクトローハンドリング」『Mobile Link サーバ管理』

同期処理

「同期」とは、Mobile Link クライアントと中央データソースの間で行われるデータ交換処理で す。この処理の間、クライアントは Mobile Link サーバとのセッションを確立して維持します。 同期に成功した場合、セッションによってリモートデータベースと統合データベースは互いに一 貫した状態に保たれます。 クライアントは同期処理を正常に開始します。この処理は、Mobile Link サーバとの接続を確立 することから始まります。

アップロードとダウンロード

ローをアップロードするために、Mobile Link クライアントが「アップロード」を準備し送信します。このアップロードは、リモートデータベース上で前回の同期以後に更新、挿入、または削除されたすべてのローのリストを含みます。同様に、ローをダウンロードするために、挿入、更新、削除のリストを含む「ダウンロード」を Mobile Link サーバが準備し送信します。

● アップロード デフォルトでは、Mobile Link クライアントは、前回成功した同期以後にリ モートデータベースで挿入、更新、または削除されたローを自動的に追跡します。接続が確 立すると、Mobile Link クライアントはこれらのすべての変更を記載したリストを Mobile Link サーバにアップロードします。

アップロードは、リモートデータベースで変更されたローに対する新旧のロー値のセットで 構成されます(更新には新旧のロー値があります。削除には古い値のみ、挿入には新しい値の みがあります)。ローが更新されたり削除されたりしていれば、前回成功した同期直後に存在 していた値が古い値になります。ローが挿入または更新されていれば、現在のローの値が新 しい値です。現在の状態に至るまでローが複数回変更されていても、その途中の値は送信さ れません。

Mobile Link サーバは、アップロードを受信して、定義されたアップロードスクリプトを実行 します。デフォルトでは、1回のトランザクションですべての変更が適用されます。処理が 完了すると、Mobile Link サーバはトランザクションをコミットします。

● ダウンロード Mobile Link サーバは、ユーザが作成した同期論理を使用して、Mobile Link ク ライアント側で挿入、更新、または削除されるローのリストを収集します。これらのローを Mobile Link クライアントにダウンロードします。このリストを収集するために、Mobile Link サーバは統合データベースで新しいトランザクションを開きます。

Mobile Link クライアントは、ダウンロードを受信します。Mobile Link クライアントは、ダウ ンロードの着信を、アップロードしたすべての変更内容が統合データベースで正常に適用さ れたことの確認とみなします。確認後、Mobile Link クライアントはこれらの変更内容が統合 データベースに再送されないようにします。

次に、Mobile Link クライアントは、ダウンロードを自動的に処理して、古いローの削除、新 しいローの挿入、変更されたローの更新を行います。これらの変更はすべて、リモートデー タベース内の1つのトランザクションで適用されます。終了すると、トランザクションをコ ミットします。

Mobile Link 同期中に情報が明確に交換されることはほとんどありません。クライアントは完全 なアップロードを構築してアップロードします。これに応答して、Mobile Link サーバは完全な ダウンロードを構築してダウンロードします。電話回線または公共無線ネットワークを使用し ている場合など、通信が低速で遅延時間が長い場合は、プロトコルの冗長性の制限が重要になり ます。

注意

Mobile Link は、統合データベースのデフォルトの独立性レベルとして ODBC 独立性レベル SQL_TXN_READ_COMMITTED を使用して動作します。統合データベースで使用される RDBMS がスナップショットアイソレーションをサポートし、スナップショットがデータベース に対して有効である場合、Mobile Link はデフォルトで、スナップショットアイソレーションを ダウンロードに使用します。「Mobile Link 独立性レベル」『Mobile Link サーバ管理』を参照し てください。

参照

- ●「Mobile Link イベントの概要」『Mobile Link サーバ管理』
- ●「アップロード中のイベント」『Mobile Link サーバ管理』
- ●「ダウンロード中のイベント」『Mobile Link サーバ管理』

Mobile Link イベント

Mobile Link クライアントが同期を開始すると、複数の同期イベントが発生します。同期イベントが発生すると、Mobile Link はその同期イベントに対応するスクリプトを探します。このスクリプトには、実行する作業の詳細を示す指示が含まれています。イベント用のスクリプトが定義され、Mobile Link システムテーブルに格納されている場合は、そのスクリプトが呼び出されます。

Mobile Link スクリプト

イベントに関連するスクリプトが作成されている場合、イベントの発生時に Mobile Link サーバ がそのスクリプトを実行します。スクリプトが存在しなければ、次の順位のイベントが発生しま す。

注意

[同期モデル作成ウィザード] を使用して Mobile Link アプリケーションを作成すると、必要な Mobile Link のスクリプトがすべて自動的に作成されます。ただし、デフォルトのスクリプトを カスタマイズしたり、新しいスクリプトを作成することもできます。

テーブルに対する一般的なアップロードスクリプトを以下に示します。最初のイベント upload_insert は、upload_insert スクリプトの実行をトリガします。このスクリプトにより、emp_id カラムと emp_name カラムのすべての変更が emp テーブルに挿入されます。upload_delete スク リプトと upload_update スクリプトは、emp テーブルでの削除と更新アクションに対して同様の 機能を実行します。

イベント	スクリプトの内容例
upload_insert	INSERT INTO emp (emp_id,emp_name) VALUES ({ml r.emp_id}, {ml r.emp_name})
upload_delete	DELETE FROM emp WHERE emp_id = {ml r.emp_id}

イベント	スクリプトの内容例
upload_update	UPDATE emp SET emp_name = {ml r.emp_name} WHERE emp_id = {ml r.emp_id}

ダウンロードスクリプトはカーソルを使用します。次に、download_cursor スクリプトの例を示します。

SELECT order_id, cust_id FROM ULOrder WHERE last_modified >= {ml s.last_table_download} AND emp_name = {ml r.emp_id}

イベントとスクリプトの詳細については、次の項を参照してください。

●「同期スクリプトの作成」『Mobile Link サーバ管理』

●「同期イベント」『Mobile Link サーバ管理』

SQL、Java、または.NET でスクリプトを作成可能

統合データベースのネイティブ SQL ダイアレクトを使用するか、Java または.NET の同期論理 を使用して、スクリプトを記述できます。Java と.NET の同期論理を使用すると、Mobile Link サーバによって呼び出されるコードを記述して、データベースへの接続、変数の操作、アップ ロードされたローハンドリングの直接操作、またはダウンロードへのローハンドリングの追加が 可能です。同期の要件に適したクラスとメソッドを持つ Mobile Link サーバ API (Java 用と.NET 用) があります。

「サーバ側の同期論理の作成オプション」11ページを参照してください。

RDBMS 依存のスクリプト記述については、「Mobile Link 統合データベース」『Mobile Link サー バ管理』を参照してください。

スクリプトの格納

SQL スクリプトは統合データベースの Mobile Link システムテーブルに格納されます。Mobile Link サーバ API を使用して記述されたスクリプトの場合は、完全に修飾されたメソッド名をス クリプトとして格納します。スクリプトを統合データベースに追加する方法は複数あります。

●[同期モデル作成ウィザード] を使用する場合は、プロジェクトを展開するときにスクリプトが Mobile Link システムテーブルに格納されます。

●統合データベースを設定するときにインストールされたストアドプロシージャを使用して、スクリプトを手動でシステムテーブルに追加できます。

●Sybase Central を使用して、スクリプトをシステムテーブルに手動で追加できます。

「スクリプトの追加と削除」『Mobile Link サーバ管理』を参照してください。

同期処理のトランザクション

Mobile Link サーバは、各 Mobile Link クライアントからアップロードされた変更を、1 回のトラ ンザクションで統合データベースに組み込みます。Mobile Link サーバは、新しいローの挿入、 古いローの削除、更新の実行、競合の解決が完了した後で、これらの変更をコミットします。

警告

SQL 同期スクリプト、または SQL 同期スクリプトから呼び出されるプロシージャやトリガで、 暗黙的または明示的なコミットまたはロールバックを実行しないでください。SQL スクリプト 内に COMMIT 文または ROLLBACK 文があると、同期手順のトランザクションの性質が変化し てしまいます。これらの文を使用すると、Mobile Link では、障害が発生した場合にデータの整 合性を保証できません。

ダウンロードした情報の追跡

Mobile Link では、リモートデータベースに格納されている最終ダウンロードタイムスタンプを 使用して、ダウンロードの作成方法が簡素化されます。

ダウンロードトランザクションの主な役割は、統合データベースのローを選択することです。ダ ウンロードに失敗しても、リモートデータベースが同じ最終ダウンロードタイムスタンプを繰り 返しアップロードするため、データが失われることはありません。

開始時と終了時のトランザクション

Mobile Link クライアントはダウンロードの情報を1回のトランザクションで処理します。ローを挿入、更新、削除して、リモートデータベースを統合データベースの最新の状態にします。

Mobile Link サーバは、他にトランザクションを2つ使用します。1つは同期の開始時に、もう1 つは同期の終了時に使用します。これらのトランザクションは、各同期とその処理時間に関する 情報を記録します。したがって、試行された同期、成功した同期、同期にかかった時間について の統計を記録できます。データは処理のさまざまな時点でコミットされるので、これらのトラン ザクションによって、データを失敗した同期の分析に役立てられるようにコミットできます。

参照

- ●「スクリプトでの最終ダウンロード時刻」『Mobile Link サーバ管理』
- ●「Mobile Link イベントの概要」『Mobile Link サーバ管理』
- ●「アップロード中のイベント」『Mobile Link サーバ管理』
- ●「ダウンロード中のイベント」『Mobile Link サーバ管理』

同期の障害処理の方法

Mobile Link は、フォールトトレラントになっています。たとえば、同期中に通信リンクに障害 が起きた場合は、リモートデータベースと統合データベースの両方が同じ状態のままになりま す。

クライアントでは、障害はリターンコードで示されます。

同期障害の処理方法は、発生したタイミングによって異なります。次に示すケースは、それぞれ 異なる方法で処理されます。 ● **アップロード中の障害** アップロードの構築中や適用中に障害が起きた場合は、リモート データベースは同期の起動時とまったく同じ状態のままになります。サーバ側では、適用さ れたアップロードのすべての部分がロールバックされます。

● アップロードとダウンロード間の障害 アップロードの完了後、Mobile Link クライアントが ダウンロードを受信する前に障害が発生した場合、クライアントはアップロードした変更が 統合データベースに適切に適用されたかどうかを確認できません。アップロードが完全に適 用されコミットされているか、サーバがアップロード全体を適用する前に障害が起きていま す。Mobile Link サーバは、統合データベースにある不完全なトランザクションを自動的に ロールバックします。

Mobile Link クライアントは、アップロードされたすべての変更を記録します。Mobile Link ク ライアントは、次に同期したときに前回のアップロードの状態を要求してから、新しいアッ プロードを構築します。前回のアップロードがコミットされていない場合は、新しいアップ ロードに前回のアップロードからの変更がすべて含まれます。

● ダウンロード中の障害 ダウンロードの適用中にリモートデバイスで障害が起きた場合は、 適用されたダウンロードはすべての部分がロールバックされ、リモートデータベースはダウ ンロード前と同じ状態のままになります。

非ブロッキングダウンロード確認を使用している場合、ダウンロードトランザクションはす でにコミットされていますが、nonblocking_download_ack スクリプトと publication nonblocking download ack スクリプトは呼び出されません。

ダウンロード確認を使用していない場合、ダウンロード中に障害が発生しても、サーバ側に は影響はありません。

再起動可能なダウンロード機能

Mobile Link は、ダウンロードの失敗からのリカバリを支援する機能を備えています。この機 能を使用すると、ダウンロード全体の再送を防ぐこともできます。この機能は、SQL Anywhere と Ultra Light リモートデータベースでそれぞれ別に実装されています。「-ds mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

障害が発生しても、データは失われません。Mobile Link サーバと Mobile Link クライアントが障 害時のデータ管理を行います。開発者やユーザは、アプリケーション内のデータが一貫性を保持 しているかどうか心配する必要はありません。

アップロードの処理方法

Mobile Link サーバが Mobile Link クライアントからアップロードを受信すると、同期が完了する までアップロード全体が格納されます。このような処理が行われるのは、次の理由によります。

● ダウンロードローのフィルタ ダウンロードするローを決定する方法で最も一般的なのは、前回のダウンロード以後に修正されたローをダウンロードすることです。同期中は、ダウンロードよりアップロードが優先されます。アップロード中に挿入または更新されたローが、前回のダウンロード以後に修正されたローになります。

アップロードの一部として送られたダウンロードローから除外する download_cursor スクリ プトを記述するのは困難です。このため、Mobile Link サーバがダウンロードからこのような ローを自動的に取り除きます。

- 挿入と更新の処理 デフォルトでは、アップロード内のテーブルは、参照整合性に違反しな い順序で統合データベースに適用されます。アップロード内のテーブルは、外部キー関係に 基づいて並べられます。たとえば、テーブルAとテーブルCの両方がテーブルBのプライマ リキーカラムを参照する外部キーを持っている場合は、テーブルBのローの挿入や更新が先 にアップロードされます。
- 挿入と更新後の削除の処理 削除は、すべての挿入と更新が適用された後で統合データベースに適用されます。削除が適用されると、テーブルはアップロードの処理とは逆の順序で処理されます。削除されるローが、削除される別のテーブルのローを参照している場合、この操作の順序では、参照元ローが参照先ローより先に削除されることになります。
- デッドロック アップロードを統合データベースに適用すると、他のトランザクションとの 同時実行性が原因でデッドロックが発生することがあります。そのトランザクションは、他 の Mobile Link サーバのデータベース接続からのアップロードトランザクションの場合や、統 合データベースを使用している他のアプリケーションからのトランザクションの場合があり ます。アップロードトランザクションがデッドロックされている場合は、そのトランザク ションはロールバックされ、Mobile Link サーバが自動的にアップロードをもう一度最初から 適用し始めます。

注意

パフォーマンスに関するヒント 競合をできるだけ避けるように同期スクリプトを書くこと が重要です。複数のユーザが同時に同期しているときに競合が起きると、パフォーマンスに 大きな影響があります。

参照整合性と同期

Ultra Light Java Edition を除くすべての Mobile Link クライアントは、ダウンロードをリモート データベースに組み込むときに参照整合性を確保します。

参照整合性に違反するローがあった場合、デフォルトでは、Mobile Link クライアントはダウン ロードトランザクションを失敗させずに、参照整合性に違反するすべてのローを自動的に削除し ます。

この機能には次のような利点があります。

- ●同期スクリプトの間違いから保護します。スクリプトに柔軟性があると、リモートデータベー スの整合性をそこなうローを誤ってダウンロードしてしまうことがあります。Mobile Link ク ライアントは、介入を要求せずに参照整合性を自動的に管理します。
- ●この参照整合性のメカニズムを使用して、リモートデータベースから情報を効率的に削除でき ます。親レコードに削除データを送信するだけで、Mobile Link クライアントはすべての子レ コードを自動的に削除します。これにより、Mobile Link がリモートデータベースに送信する トラフィックの量を大幅に減らすことができます。

Mobile Link クライアントは、参照整合性を維持するためにローを明示的に削除する必要がある 場合、次のような通知を行います。

●SQL Anywhere クライアントの場合は、dbmlsync によってログにエントリが書き込まれます。 dbmlsync イベントフックも使用できます。 次の項を参照してください。

- ○「sp hook dbmlsync download ri violation」『Mobile Link クライアント管理』
- ○「sp hook dbmlsync download log ri violation」『Mobile Link クライアント管理』

●Ultra Light クライアントの場合は、

SQLE_ROW_DELETED_TO_MAINTAIN_REFERENTIAL_INTEGRITY 警告が発生します。この警告には、テーブル名のパラメータが含まれます。参照整合性を維持するため、削除されるすべてのローで警告が発生します。同期をそのまま進める場合は、警告を無視してかまいません。警告を明示的に処理する場合は、エラーコールバック関数を使用して警告をトラップすることで、たとえば、削除されたローの数を取得することができます。

警告が発生したときに同期を失敗させるには、同期 observer を実装し、observer に (グローバル変数などを使用して) エラーコールバック関数から信号を送信する必要があります。この場合、同期は observer への次回の呼び出しで失敗します。

トランザクション終了時にチェックされる参照整合性

Mobile Link クライアントは、1 つのトランザクション内のダウンロードから変更を組み込みま す。柔軟性をより向上させるために、参照整合性のチェックはこのトランザクションの終了時に 行われます。チェックが遅いため、参照整合性に違反する状態を一時的にパスすることがありま す。しかし、参照整合性に違反するローは、ダウンロードがコミットされる前に自動的に削除さ れます。

例

Ultra Light 販売アプリケーションに、次の2つのテーブルが含まれているとします。1つのテーブルには、販売注文が含まれています。もう1つのテーブルには、各注文で販売された商品情報 が含まれています。この2つのテーブルは、次のような関係です。



1 つの注文を削除するために sales_order テーブルに download_delete_cursor を使用すると、削除 された受注を示す sales_order_items テーブルのすべてのローが、デフォルトの参照整合性メカニ ズムによって自動的に削除されます。

この方法には、次のような利点があります。

[●]sales_order_items テーブルからのローは自動的に削除されるので、sales_order_items テーブルス クリプトは必要ありません。

●同期の効率が向上します。sales_order_items テーブルから削除するローをダウンロードする必要がありません。各販売注文に項目がたくさんある場合は、ダウンロードが小さくなるのでパフォーマンスが向上します。この方法は、速度の遅い通信方法を使用しているときに特に役立ちます。

デフォルトの動作の変更

SQL Anywhere クライアントの場合は、sp_hook_dbmlsync_download_ri_violation クライアントイ ベントフックを使用して、参照整合性違反を処理できます。Dbmlsync も、ログにエントリを書 き込みます。

次の項を参照してください。

- 「sp hook dbmlsync download log ri violation」 『Mobile Link クライアント管理』
- ●「sp_hook_dbmlsync_download_ri_violation」『Mobile Link クライアント管理』

セキュリティ

Mobile Link のインストール環境のように、広範囲の分散システム全体のデータの安全を確保するには、いくつかの要素があります。

● 統合データベースのデータ保護 統合データベース内のデータは、データベースのユーザ認証 システムとその他のセキュリティ機能で保護できます。

詳細については、使用しているデータベースのマニュアルを参照してください。SQL Anywhere 統合データベースを使用している場合は、「データのセキュリティ」『SQL Anywhere サーバデータベース管理』を参照してください。

● リモートデータベースのデータ保護 SQL Anywhere リモートデータベースを使用している場合、SQL Anywhere のセキュリティ機能を使用してデータを保護できます。このセキュリティ 機能は、デフォルトではクライアント/サーバ通信での不正なアクセスを防止するように設計されていますが、データベースファイルから直接情報を抽出するという悪質な攻撃を回避 する確実な手段とはなりません。

クライアント上のファイルは、クライアントのオペレーティングシステムのセキュリティ機能によって保護されます。

SQL Anywhere リモートデータベースを使用している場合は、「データのセキュリティ」『SQL Anywhere サーバ データベース管理』を参照してください。

Ultra Light データベースを使用している場合は、「データベースセキュリティ」『Ultra Light データベース管理とリファレンス』を参照してください。

- 同期中のデータ保護 Mobile Link クライアントから Mobile Link サーバへの通信は、Mobile Link トランスポートレイヤセキュリティ機能で保護できます。「トランスポートレイヤセキュリティ」『SQL Anywhere サーバ データベース管理』を参照してください。
- 権限のないユーザからの同期システムの保護 Mobile Link 同期は、パスワードによるユーザ認 証システムで保護できます。このメカニズムにより、権限のないユーザはデータを同期でき

なくなります。 「Mobile Link ユーザ」『Mobile Link クライアント管理』を参照してください。

Sybase Central の Mobile Link プラグイン

Sybase Central の Mobile Link プラグインはバージョン 12 で再設計されています。以前のバー ジョンでは、プラグインに 2 つのモードがありました。モデルモードと管理モードです。Mobile Link の機能は 2 つのモードに分割されていたため、どちらのモードを使用しているのかを常に 認識する必要がありました。次に示すように、バージョン 12 以降では、これらのモードはなく なりました。



Mobile Link プラグインの [フォルダ] ウィンドウ枠から使用できる最上位レベルの機能は、次の とおりです。

●Mobile Link プロジェクトの操作。「Mobile Link プロジェクトの作成」23 ページを参照して ください。

●統合データベースの操作。「統合データベースの追加」25ページを参照してください。

- ●Mobile Link サーバのコマンドラインの操作。「Mobile Link サーバのコマンドラインを作成」 27 ページを参照してください。
- ●リモートスキーマ名の操作。「Sybase Central の Mobile Link エージェント」『Mobile Link サー バ管理』を参照してください。
- ●グループの操作。「Sybase Central の Mobile Link エージェント」『Mobile Link サーバ管理』を 参照してください。
- ●リモートタスクの操作。「リモートタスク」『Mobile Link サーバ管理』を参照してください。

●同期モデルの操作。「同期モデル」29ページを参照してください。

リモートスキーマ名、グループ、リモートタスクはすべて、リモートデータベース機能の集中管理の一部です。「リモートデータベースの集中管理」『Mobile Link サーバ管理』を参照してください。

Sybase Central で Mobile Link を操作するには、まず Mobile Link プロジェクトを定義する必要があります。

「Mobile Link プロジェクト」は、モバイルアプリケーションに関連する同期モデル、統合データ ベース、リモートタスクで構成されるフレームワークです。

Mobile Link プロジェクトは、次の要素から成る名前付きコレクションです。

- ●同期モデルのリスト
- ●設計済みで展開されていないリモートタスクのリスト
- ●統合データベースへの接続のリスト
- ●Mobile Link ユーザのユーザ定義グループのリスト

Mobile Link プロジェクトの作成

Sybase Central の Mobile Link を操作する前に、Mobile Link プロジェクトを作成します。

前提条件

この作業を実行するための前提条件は、ありません。

内容と備考

サンプルの Mobile Link プロジェクトは %SQLANYSAMP16%¥MobiLink¥CustDB¥project.mlp にあります。

◆タスク

1. Sybase Central で、[ツール]» [Mobile Link 16]» [新しいプロジェクト] をクリックします。

- (新しいプロジェクトの名前を指定してください。) フィールドに、プロジェクトの名前を入力 します。
- (新しいプロジェクトの保存場所を指定してください。)フィールドにプロジェクトフォルダ のロケーションを入力するか、(参照)をクリックしてプロジェクトファイルのフォルダを選 択します。
- 4. [次へ] をクリックします。
- 5. [統合データベースを指定] ページで、次のタスクを実行します。
 - a. [データベースの表示名] フィールドに、統合データベースに使用する表示名を入力しま す。この名前は、プロジェクトの統合データベースリストに表示されます。
 - b. [接続文字列] フィールドに、統合データベースへの接続に使用するデータベース接続パ ラメータを入力するか、[編集] をクリックして ODBC データソースに接続するための ウィンドウを開きます。
 - c. [パスワードを記憶] を選択し、データベースへの接続に使用するパスワードを保存します。 このオプションを選択すると、パスワードは難読化した形式でプロジェクトファイルに
 - d. [次へ] をクリックします。

保存されます。

- 6. [新しいリモートデータベーススキーマ] ページで、リモートデータベースに表示する統合 テーブルとカラムを選択します。[次へ] をクリックします。
- [リモートスキーマ名をプロジェクトに追加] を選択し、スキーマが同じであるリモートデー タベースのグループを識別します。または後から追加することもできます。リモートスキー マ名は、リモートタスクを作成する場合にのみ役に立ちます。

リモートスキーマ名を追加する場合は、次のように指定します。

● [新しいリモートスキーマ名を指定してください。] 同じスキーマ名を共有するリモート データベースのグループを特定するために使用する名前を入力します。バージョン番号 を含めることをおすすめします。

[次へ] をクリックします。

- 8. [リモートデータベースタイプを指定] ページで、使用するリモートデータベースのタイプとして、[SQL Anywhere] または [Ultra Light] のいずれかを選択します。この設定は、後からプロジェクトの [プロパティ] ページで変更できます。
- 9. [完了] をクリックして新しいプロジェクトを保存します。
- 10. Mobile Link システム設定のインストールを求められたら、[はい] をクリックしてから [OK] をクリックします。

結果

Mobile Link プロジェクトが作成され、指定したデータベースへの接続が確立され、同期モデル が作成されます。

次の手順

これで、同期モデルを編集または展開したり、Mobile Link プロジェクトの他のオブジェクト(リ モートタスクなど)を操作したりできます。

参照

●「集中管理の概念」『Mobile Link サーバ管理』

統合データベースの追加

Sybase Central で、Mobile Link プロジェクトに1つまたは複数の統合データベースを追加します。

前提条件

Mobile Link プロジェクトは、定義されている必要があります。

内容と備考

リモートタスクを展開するためには、少なくとも1つの統合データベースを割り当てておく必要 があります。

◆タスク

- 1. Mobile Link プロジェクトを選択します。
- プロジェクト名をダブルクリックし、[ファイル]»[新規]»[統合データベース] をクリックします。
- 3. 必要なデータベース接続パラメータを入力して [次へ] をクリックします。
- (表示名) フィールドに、プロジェクトでこのデータベースに使用する名前を入力します。デフォルトの表示名は ODBC データソース名です。データベースの説明を指定するには、[説明] フィールドに入力します。
- 5. [パスワードを記憶]を選択し、データベースへの接続に使用するパスワードを保存します。

このオプションを選択すると、パスワードは難読化した形式でプロジェクトファイルに保存されます。

- 6. [完了] をクリックして、統合データベースをプロジェクトに追加します。
- 7. Mobile Link システム設定のインストールを求められたら、[はい] をクリックしてから [OK] をクリックします。

結果

統合データベースが、Mobile Link プロジェクトに追加されます。

参照

●「集中管理の概念」『Mobile Link サーバ管理』

Mobile Link システム設定

テーブル、カラム、トリガなどの同期に必要なオブジェクトを追加してから、データベースを Mobile Link 統合データベースとして使用する必要があります。これらのオブジェクトの追加 は、データベースに対して設定スクリプトを実行することによって行われます。サポートされて いる各 RDBMS 用に個別の設定スクリプトがあります。これらのスクリプトは、すべて %SQLANY16%¥MobiLink¥setup フォルダにあります。スクリプトによる処理の内容を詳細に確認 するには、テキストエディタでスクリプトを開きます。

統合データベースを Mobile Link プロジェクトに追加すると、Mobile Link システム設定がチェッ クされます。Mobile Link システム設定が存在しない場合は、インストールを求めるプロンプト が表示されます。古いバージョンが検出された場合にも、アップグレードを求めるプロンプトが 表示されます。また、同期モデルを展開する場合(まだ行っていない場合)にも、Mobile Link シ ステム設定のインストールを求めるプロンプトが表示されます。

参照

- ●「Mobile Link システム設定のチェック」26ページ
- ●「統合データベースの設定」『Mobile Link サーバ管理』
- ●「同期モデルの展開」48ページ

Mobile Link システム設定のチェック

同期に必要なオブジェクトをチェックします。

前提条件

Mobile Link プロジェクトが定義されており、そのプロジェクトに少なくとも1つの統合データベースが存在する必要があります。

◆タスク

- 1. Mobile Link オブジェクトをダブルクリックします。
- 2. [統合データベース] をダブルクリックし、チェックする統合データベースを選択し、[Mobile Link システム設定のチェック] をクリックします。
- 3. 設定がインストール済みでないか、または最新でない場合は、[はい] をクリックしてインス トールまたは更新し、次に、[OK] をクリックします。

結果

システム設定は、指定されたとおり、インストールまたは更新されます。

Mobile Link システム設定の削除

Mobile Link システム設定によってインストールされている、同期に必要なオブジェクトを削除 します。

前提条件

Mobile Link プロジェクトが定義されており、その統合データベースに Mobile Link システム設定 がインストールされている必要があります。

◆タスク

- 1. Mobile Link オブジェクトをダブルクリックします。
- [統合データベース] をダブルクリックし、チェックする統合データベースを選択し、[Mobile Link システム設定を削除] をクリックします。
- 3. すべての同期モデルがデータベースから削除され、同期状態が失われ、データベースがプロ ジェクトから削除されることを示す警告が表示されます。[はい]をクリックします。

結果

Mobile Link システム設定がデータベースから削除され、データベースが Mobile Link プロジェクトから削除されます。

Mobile Link サーバのコマンドラインを作成

[Mobile Link サーバのコマンドラインのプロパティ] ウィンドウを使用して、Mobile Link プロ ジェクトに取り込む Mobile Link サーバのコマンドラインを保存します。

前提条件

Mobile Link プロジェクトは、定義されている必要があります。

◆タスク

- 1. Mobile Link オブジェクト名をダブルクリックします。
- 2. [Mobile Link サーバのコマンドライン] をダブルクリックします。
- 3. [ファイル]»[新規]»[Mobile Link サーバのコマンドライン] をクリックします。
- 4. コマンドラインに対してわかりやすい名前を入力します。
- 5. [統合データベース] ドロップダウンリストから統合データベースを選択します。
- 6. [追加] をクリックして、同期要求を Mobile Link サーバで受信する方法を決定するネットワー クオプションを指定します。オプションが指定されていない場合、Mobile Link サーバはポー ト 2439 で TCP/IP 接続を受信します。

- [冗長性] ドロップダウンリストから冗長性レベルを選択します。[カスタム] オプションを選 択するとウィンドウが開き、使用できる冗長性オプションの中から必要なものを選択できる ようになります。
- 8. その他の必要なオプションは [詳細] ページで設定します。必要なオプションがリストにない場合は、それを [その他のオプション] フィールドに入力できます。
- 9. [コマンドライン] フィールドに表示されたコマンドラインを確認します。必要に応じて上 記オプションを編集するか、または [OK] をクリックしてコマンドラインを保存します。

結果

指定したオプションと一緒にコマンドラインが保存されます。

LDAP サーバの追加

ユーザを認証する場合に Mobile Link サーバによって使用される信頼できる LDAP サーバを追加します。

前提条件

Mobile Link プロジェクトが定義されており、そのプロジェクトに少なくとも1つの統合データベースが存在する必要があります。

◆タスク

- 1. Mobile Link プロジェクト名をダブルクリックしてから、使用する統合データベースをダブル クリックします。
- 2. [LDAP サーバ] をダブルクリックします。
- 3. [ファイル]»[新規]»[LDAP サーバ] をクリックします。
- 4. LDAP サーバ作成ウィザードの手順に従います。
- 5. [完了] をクリックします。

結果

LDAP サーバが作成され、[LDAP サーバ] タブに表示されます。

次の手順

ユーザ認証ポリシーを設定することで、LDAP サーバによる Mobile Link ユーザの認証の方法を 定義できます。「ユーザ認証ポリシーの追加」28 ページを参照してください。

ユーザ認証ポリシーの追加

ユーザ認証ポリシーを作成し、LDAP サーバに対して Mobile Link ユーザを認証します。

前提条件

統合データベースに少なくとも1つの LDAP サーバが定義されている必要があります。

◆ タスク

- 1. Mobile Link プロジェクト名をダブルクリックしてから、使用する統合データベースをダブル クリックします。
- 2. [ユーザ認証ポリシー] をダブルクリックします。
- 3. [ファイル]»[新規]»[ユーザ認証ポリシー] をクリックします。
- 4. ユーザ認証ポリシー作成ウィザードの手順に従います。
- 5. [完了] をクリックします。

結果

ユーザ認証ポリシーが作成され、[ユーザ認証ポリシー] タブに表示されます。

次の手順

新しい Mobile Link ユーザを作成するか、または認証ポリシーを使用するように既存の Mobile Link ユーザを変更することができます。

同期モデル

「同期モデル」は、Mobile Link アプリケーションを簡単に作成できるツールです。同期モデル は、Sybase Central のプロジェクト作成ウィザードまたは同期モデル作成ウィザードによって作 成されるファイルです。

同期モデルが作成されたら、引き続きモデルのカスタマイズを行うことができます。統合データベースまたはリモートデータベースは、モデルを展開するまで変更されません。モデルは拡張子.mlsmのモデルファイルに保存され、このファイルの参照は Mobile Link プロジェクトファイルに保存されます。

モデルが完成したら、同期モデル展開ウィザードを使用してモデルを展開します。同期モデル展 開ウィザードでは、選択した展開オプションを使用して、Mobile Link サーバとクライアントを 実行するスクリプトファイルを作成できます。展開時に既存のデータベースを変更するか、ウィ ザードを使用して後で実行する SQL スクリプトを作成するかを選択できます。リモートデータ ベース用に作成されたファイルは、リモートタスクで使用できます。

展開後に、同期モデルまたはデータベースを引き続きカスタマイズし、再展開することができま す。必要に応じて、Mobile Linkのマニュアル全体に記載されている技術を使用して、展開した 同期システムを Sybase Central 外で変更することもできます。

[同期モデル作成ウィザード] を使用した Mobile Link アプリケーション の設定

Sybase Central の [同期モデル作成ウィザード] を使用して、Mobile Link アプリケーションの同期 論理を設定します。

前提条件

Mobile Link プロジェクトは、定義されている必要があります。

◆ タスク

- 1. Mobile Link プロジェクト名をダブルクリックし、[同期モデル] をダブルクリックします。
- 2. [ファイル]»[新規]»[同期モデル] をクリックし、[同期モデル作成ウィザード] を起動します。
- 3. [ようこそ] ページで、同期モデルの名前を選択します。モデルは、プロジェクトディレクト リに.*mlsm* ファイルとして保存されます。[次へ] をクリックします。
- [プライマリキー要件]ページで、システムがプライマリキー要件を満たしていることを確認し、3つのチェックボックスをオンにし、[次へ]をクリックします。プライマリキーの詳細については、「ユニークなプライマリキー」『Mobile Link サーバ管理』を参照してください。
- 5. [統合データベーススキーマ] ページで、統合データベーススキーマを取得するための統合 データベースを選択し、[次へ] をクリックします。

Oracle データベースを選択した場合、全所有者のスキーマをロードするには時間がかかるため、所有者のサブセットの選択を求めるプロンプトが表示されることがあります。

- リモートデータベーススキーマは、統合データベーススキーマまたは既存のリモートデータ ベースに基づいて作成できます。既存のリモートデータベースには、SQL Anywhere または Ultra Light を使用できます。決定方法の詳細については、「リモートスキーマ」31ページを 参照してください。
- 7. [同期モデル作成ウィザード]の残りの指示に従います。可能な場合はベストプラクティスに 基づいたデフォルトの推奨事項が使用されます。

8. [完了] をクリックします。

結果

[完了] をクリックすると、Sybase Central の右ウィンドウ枠に同期モデルが表示され、モデルの 表示、編集、または展開が行えるようになります。 「同期モデルタスク」32ページを参照し てください。

次の手順

同期モデルを作成したら、次のいずれかを実行します。
●[テスト] ウィンドウを使用して同期モデルをテストします。

- ●マッピング、イベント、認証情報を編集します。「テーブルとカラムのマッピング」33ページ を参照してください。
- ●同期モデル展開ウィザードを使用して、完了したモデルを展開します。「同期モデルの展開」 48ページを参照してください。

同期モデルの削除

Mobile Link プラグインを使用して同期モデルを追加してある場合は、Mobile Link プラグインを 使用して統合データベースから同期モデルを削除することができます。

前提条件

Mobile Link プラグインを使用して同期モデルが追加されている必要があります。

内容と備考

この機能は、DB2 データベースでは使用できません。

◆タスク

- 1. 削除する同期モデルが含まれているプロジェクトをダブルクリックします。
- [同期モデル] をダブルクリックし、削除する同期モデルを右クリックし、[統合データベース から削除] を選択します。
- 3. 該当する同期モデルが複数の統合データベースにある場合は、その中でも同期モデルを削除 する統合データベースを選択します。
- 4. [OK] をクリックします。

結果

選択した統合データベースから同期モデルが削除されます。

リモートスキーマ

同期モデルには、リモートデータベースのスキーマが含まれています。このスキーマは、既存の リモートデータベースまたは統合データベースから取得できます。

既存のリモートデータベースは、次のような場合に使用します。

●すでにリモートデータベースがある場合、特にスキーマが統合データベーススキーマのサブ セットではない場合。

●統合カラムとリモートカラムのタイプが異なっている必要がある場合。

●リモートテーブルと統合データベースのテーブルの所有者が異なっている必要がある場合。 統合データベースから作成された新しい SQL Anywhere リモートスキーマでは、リモートテー ブルの所有者は、統合データベース内の対応するテーブルの所有者と同じになります。別の所 有者にするには、設定するテーブル所有者により所有される既存の SQL Anywhere リモート データベースを使用します。

注意

既存のデータベーススキーマを手動で変更し、スキーマ更新ウィザードを実行すると、Mobile Link プロジェクトで同期モデルを更新できます。「スキーマの更新」47ページを参照してください。

モデルの展開時は、モデルでのリモートスキーマの作成方法にかかわらず、リモートデータベースは3つのオプションから選択して作成できます。展開時のリモートデータベースのオプションは次のとおりです。

- 新しいリモートデータベースを作成する。 展開時に、同期モデルのスキーマが使用され、新 しいリモートデータベースが作成されます。データベースはデフォルトオプションで作成さ れます。
- **ユーザテーブルがない既存のリモートデータベースを更新する。** 展開中に、同期するユーザ テーブルを作成するか、または再作成するかを選択できます。このオプションは、特定の照 合など、デフォルトでないデータベース作成オプションを使用する場合に便利です。

SQL Anywhere データベースの場合、一部のオプションはデータベース作成後には設定できません。備考、初期化ユーティリティ (dbinit) 『SQL Anywhere サーバ データベース管理』を参照してください。

Ultra Light データベースの場合は、データベース作成後にデータベースのプロパティを変更 することはできません。「Ultra Light データベース作成パラメータ」『Ultra Light データベー ス管理とリファレンス』を参照してください。

● モデルと同じスキーマを持つ既存のリモートデータベースを更新する。 このオプションは、 同期する既存のリモートデータベースがある場合に便利です。既存のリモートデータベース に直接展開する場合は、同期するテーブルが作成または再作成されないように選択できます。 既存のテーブルとその内容は変更されません。

SQL Anywhere リモートデータベースでは、テーブルと元のデータベースの所有者は同じです。 Ultra Light データベーステーブルには、所有者はありません。

参照

●「同期モデルの展開」48ページ

同期モデルタスク

[同期モデル作成ウィザード] で同期モデルを作成したら、それを使用していくつかのタスクを実行できます。変更内容は、同期モデルファイルにのみ保存されます。同期モデルを展開するまで、変更内容は統合データベースまたはリモートデータベースには保存されません。

Sybase Central 外で同期モデルを変更することはできますが、変更内容をリバースエンジニアリ ングでモデルに戻すことはできません。たとえば、システムプロシージャを使用して Mobile Link スクリプトの追加や変更を行うことができます。 「Mobile Link サーバシステムプロシー ジャ」『Mobile Link サーバ管理』を参照してください。

テーブルとカラムのマッピング

テーブルマッピングは、どのテーブルを同期するか、テーブルをどのように同期するか、および 統合データベースとリモートデータベースとの間で同期データをどのようにマッピングするか を指定します。

アップロード専用、ダウンロード専用、非同期テーブルまたはカラムデフォルトでは、Mobile Link は完全に双方向の同期を行います。各テーブルは、アップロード専用またはダウンロード 専用に変更できます。また、テーブルを同期しないことを選択することもでき、この場合は、 テーブルマッピングは削除されます。

同期モデルでは、テーブルをダウンロード専用として指定することのみができ、ダウンロード専 用パブリケーションを作成することはできません。「ダウンロード専用のパブリケーション」 『Mobile Link クライアント管理』を参照してください。

テーブルマッピングの方向の変更

テーブルマッピングの方向を変更して、テーブルを同期する方法を示すか、またはテーブルを同期しないことを示します。

前提条件

同期モデルが必要です。

◆ タスク

- 1. Mobile Link オブジェクトをダブルクリックします。
- 2. [同期モデル] をダブルクリックし、同期モデル名をダブルクリックします。
- 3. [マッピング] タブをクリックします。
- 4. [テーブルマッピング] ウィンドウ枠で、統合テーブルを選択します。
- 5. [方向] ドロップダウンリストから、次のいずれかを選択します。
 - [同期しない] このオプションを選択することは、テーブルマッピングを削除することと 同じです。
 - [双方向] データベース操作は、リモートデータベースから統合データベースの方向へ、 および統合データベースからリモートデータベースの方向へ同期されます。
 - [リモートにのみダウンロード] 変更は統合データベースからリモートデータベースの方 向へのみ同期されます。
 - [統合にのみアップロード] 変更はリモートデータベースから統合データベースの方向へのみ同期されます。

結果

テーブルマッピングが更新されます。

テーブルマッピングの削除

テーブルマッピングを削除すると、テーブルは同期しなくなります。

前提条件

同期モデルが必要です。

◆ タスク

- 1. Mobile Link オブジェクト名をダブルクリックします。
- 2. [同期モデル] をダブルクリックし、同期モデル名をダブルクリックします。
- 3. [マッピング] タブをクリックします。
- 4. [テーブルマッピング] ウィンドウ枠で、テーブルマッピングを選択します。
- 5. [方向] ドロップダウンリストから、[同期しない] をクリックします。

結果

マッピングは、次に同期モデルを保存するときに削除されます。

テーブルマッピングの変更

テーブルとカラムのマッピングを確認しカスタマイズします。既存のリモートデータベースに 基づいてモデルを作成すると、テーブルとカラムのマッピングは推測に基づいて設定されます。

前提条件

同期モデルが必要です。

◆タスク

- 1. Mobile Link オブジェクト名をダブルクリックします。
- 2. [同期モデル] をダブルクリックし、同期モデル名をダブルクリックします。
- 3. [マッピング] タブをクリックします。
- 4. [テーブルマッピング] ウィンドウ枠で、変更するマッピングのリモートテーブルをクリック します。
- 5. リモートテーブル名の横にある省略記号 (ピリオド3つ) ボタンをクリックし、同期されてい ないリモートテーブルのリストから別のテーブルを選択します。

- ●統合テーブルにまだマッピングされていないリモートテーブルのみを選択できます。
- ●リモートスキーマにテーブルを追加するには、「スキーマの更新」47 ページ を参照して ください。

結果

テーブルマッピングが更新されます。

マッピングされていないテーブルをマッピングに追加

統合データベース内のテーブルのうち、同期できるようにマッピングされていないテーブルを同 期モデルに追加します。

前提条件

同期モデルが必要です。

◆タスク

- 1. Mobile Link オブジェクト名をダブルクリックします。
- 2. [同期モデル] をダブルクリックし、同期モデル名をダブルクリックします。
- 3. [マッピング] タブをクリックします。
- [ファイル]»[新規]»[テーブルマッピング] をクリックして、[新しいテーブルマッピングの作 成] ウィンドウを開きます。このウィンドウでは、追加する統合テーブルを選択します。
- 1つまたは複数の統合テーブルを選択し、リモートスキーマに追加します。新しいリモート テーブルの名前とカラムセットは対応する統合テーブルと同じであり、リモートテーブルと 統合テーブルとの間のマッピングは自動的に作成されます。

次のオプションを使用できます。

- [選択したテーブルが存在しない場合はリモートスキーマに追加] リモートスキーマに変更 を加えない場合は、このオプションを無効にします。
- [新しいリモートテーブルを選択する] リモートスキーマにテーブルを追加するが、すべて のカラムを追加するわけではない場合に、このオプションを選択します。このオプション を有効にすると、選択したテーブルごとにカラムを選択できます。
- [同期モデルのシャドーテーブルのような名前のテーブルを非表示] デフォルトで、同期モ デルシャドウテーブル名のような名前の統合データベーステーブルは表示されません。 これは、そのようなシャドウテーブルは同期してはならないためです。

結果

同期モデルのマッピングに新しいテーブルが追加されます。

テーブルマッピングのカラムマッピングの変更

必要に応じてテーブルとカラムのマッピングを確認しカスタマイズします。既存のリモート データベースに基づいてモデルを作成すると、テーブルとカラムのマッピングは推測に基づいて 設定されます。

前提条件

同期モデルが必要です。

内容と備考

同期された統合テーブルのカラムをリモートテーブルカラムにマッピングできます。カラムを 同期するか同期から除外すると、値が決まります。値にマッピングするとき、Mobile Link ユー ザ名、リモートデータベース ID、または SQL 式 (Mobile Link 名前付きパラメータを含むことが できる)を使用できます。プライマリキーカラムを値にマッピングし、テーブルマッピングが双 方向の場合は、リモートデータベースにダウンロードするときにプライマリキーが重複しないよ うにしてください。

◆ タスク

- 1. [テーブルマッピング] タブで、テーブルマッピングを選択します。
- 2. [詳細] ウィンドウ枠で、[カラムマッピング] タブを開きます。
- 3. 変更するカラムマッピングを右クリックし、コンテキストメニューから次のいずれかのオプ ションを選択します。

● [なし]

- [Mobile Link ユーザ名]
- [リモート ID]
- [カスタム]
- マッピングされていないリモートカラム

統合カラムをリモートカラムと統合するには、メニューの一番下のグループからマッピング されいないリモートカラムを選択します。マッピングされていないリモートカラムのみが表 示されます。

統合カラムを同期から除外するには、**[なし]**をクリックします。方向アイコンに、統合カラムが同期されないことが示されます。

統合カラムを値にマッピングするには、[Mobile Link ユーザ名]、[リモート ID]、または [カ スタム] を選択して、同期中にリモートテーブルの upload_insert、upload_update、upload_delete 同期スクリプトが実行されるときに評価される SQL 式を入力します。[方向] アイコンには 値がアップロード専用であることが示されます。すなわち、統合カラムはリモートデータ ベースにダウンロードされません。

結果

カラムマッピングが更新されます。

ダウンロードタイプの変更

[詳細] ウィンドウ枠の [ダウンロードタイプ] タブでダウンロードタイプを変更します。テーブ ルマッピングのダウンロードタイプには、タイムスタンプ、スナップショット、カスタムのいず れかを設定できます。

前提条件

同期モデルが必要です。

内容と備考

ダウンロードタイプは次のとおりです。

- [タイムスタンプベースのダウンロード] このオプションを選択すると、タイムスタンプベー スのダウンロードがデフォルトとして使用されます。最後の同期後に変更されたローのみが ダウンロードされます。「タイムスタンプベースのダウンロードの実装」『Mobile Link サー バ管理』を参照してください。
- [スナップショットダウンロード] このオプションを選択すると、スナップショットダウン ロードがデフォルトとして使用されます。最後の同期後に変更されていない場合でも、同期 のたびにすべてのローがダウンロードされます。次の項を参照してください。

○「スナップショットを使った同期」『Mobile Link サーバ管理』
 ○「スナップショットを使った同期をいつ行うか」『Mobile Link サーバ管理』

- [カスタムダウンロードロジック] download_cursor スクリプトと download_delete_cursor スク リプトを自動的に生成せず、独自に作成する場合は、このオプションを選択します。 参照:
 - ○「同期スクリプトの作成」『Mobile Link サーバ管理』
 - ○「download cursor スクリプト」『Mobile Link サーバ管理』
 - ○「download_delete_cursor スクリプト」『Mobile Link サーバ管理』

◆ タスク

- 1. Mobile Link オブジェクト名をダブルクリックします。
- 2. [同期モデル] をダブルクリックし、同期モデル名をダブルクリックします。
- 3. [マッピング] タブを開きます。
- 4. [テーブルマッピング] ウィンドウ枠で、テーブルマッピングを選択します。
- 5. [詳細] ウィンドウ枠で、[ダウンロードタイプ] タブを選択します。
- 6. [ダウンロードタイプ] ドロップダウンリストで、[タイムスタンプ]、[スナップショット]、[カ スタム] のいずれかを選択します。
- 7. [カスタム] を選択した場合は、上ウィンドウ枠の [イベント] タブをクリックして、 download_cursor と download_delete_cursor の各スクリプトを入力します。

- 8. [タイムスタンプ] を選択した場合は、次のタスクを実行します。
 - a. [タイムスタンプカラム名] フィールドにカラム名を入力します。
 - b. 次のうちの1つを選択してください。
 - i. [統合テーブルでタイムスタンプカラムを使用する]
 - [トリガの代わりにカラムデフォルトを使用する]
 - ii. [シャドウテーブルを使用してタイムスタンプカラムを保持する]

結果

ダウンロードタイプが変更されます。

削除の記録方法の変更

統合データベースでの削除をリモートにダウンロードするかどうか、およびこのような削除に関 する情報を統合データベースにどのように保存するかを制御する場合には、[削除のダウンロー ド] タブを使用します。

前提条件

同期モデルが必要です。さらに、テーブルマッピングのダウンロードタイプが [タイムスタン プ] に設定されている必要があります。

内容と備考

スナップショットダウンロードを使用している場合は、リモートデータベースのすべてのローが 削除されてからスナップショットがダウンロードされます。このため、削除操作を追跡する必要 はありません。タイムスタンプベースのダウンロードを使用している場合は、統合データベース での削除をリモートデータベースへのダウンロードに記録する方法を指定できます。

統合データベースからローを削除したときに、リモートデータベースからもローを削除する場合 は、削除された各ローを記録しておく必要があります。この情報は、シャドウテーブルを使用す るか、論理削除によって記録することができます。

◆ タスク

- 1. Mobile Link オブジェクト名をダブルクリックします。
- 2. [同期モデル] をダブルクリックし、同期モデル名を選択します。
- 3. [マッピング] タブをクリックします。
- 4. [テーブルマッピング] ウィンドウ枠で、テーブルマッピングを選択します。
- 5. [詳細] ウィンドウ枠で、[削除のダウンロード] タブを開きます。
- [削除のダウンロード] チェックボックスをオンにして、統合データベースから削除をダウン ロードします。統合データベースから削除のダウンロードを行わない場合は、チェックボッ クスをオフにします。

削除を記録するには、シャドウテーブルまたは論理削除の使用を設定します。

結果

削除の処理方法が更新されます。

論理削除

Mobile Link 同期モデルでの論理削除のサポートは、論理削除カラムが統合データベースのみに あり、リモートデータベースにはないことを前提としています。統合スキーマを新しいリモート スキーマにコピーする場合、同期モデル設定の論理削除カラムに対応するすべてのカラムを除外 します。

デフォルトの論理削除カラム名と一致するカラムは、新しいリモートスキーマに自動的にコピー されません。

リモートデータベースで論理削除を使用しない場合は、削除をダウンロードしないことを選択 し、必要に応じて、論理削除カラムが含まれるようにリモートスキーマを更新します。

注意

リモートスキーマでの論理削除カラムのカラムマッピングを、統合スキーマでの論理削除カラム に設定してください。

参照

- ●「削除」『Mobile Link サーバ管理』
- ●「download delete cursor スクリプト」『Mobile Link サーバ管理』
- ●「download_cursor テーブルイベント」『Mobile Link サーバ管理』

サブセットのダウンロード

各 Mobile Link リモートデータベースでは、統合データベースに格納されているデータのサブ セットを受信できます。テーブルごとにダウンロードサブセットをカスタマイズできます。

ダウンロードサブセットのオプションは、次のとおりです。

● [ユーザ] このオプションを選択すると、Mobile Link ユーザ名によってデータが分割され、 登録済みの Mobile Link ユーザごとに異なるデータがダウンロードされます。

このオプションを使用するには、Mobile Link ユーザ名を保持するカラムが各ローに含まれて いる必要があります。Mobile Link ユーザ名は展開時に選択するため、統合データベースの既 存の値と一致する名前を選択できます。(Mobile Link ユーザ名用に使用するカラムは、ユー ザ名として使用する値を格納できるタイプである必要があります。)サブセットのテーブルと は異なるテーブルに Mobile Link ユーザ名がある場合は、そのテーブルにジョインする必要が あります。

● [リモート ID] このオプションを選択すると、リモート ID によってデータが分割され、リ モートデータベースごとに異なるデータがダウンロードされます。

このオプションを使用するには、リモート ID を保持するカラムが各ローに含まれている必要 があります。リモート ID はデフォルトでは GUID として作成されますが、統合データベース の既存の値に一致するようにリモート ID を設定できます。(リモート ID 用に使用するカラムは、リモート ID として使用する値を格納できるタイプである必要があります。)サブセットのテーブルとは異なるテーブルにリモート ID がある場合は、そのテーブルにジョインする必要があります。

注意

リモート ID は、リモートコンピュータがリセットされるか、置き換えられたときに変更され る場合があるため、通常、リモート ID 別ではなく、ユーザ別、または認証パラメータ別に分 割することをおすすめします。

● [カスタム] ダウンロードするローを指定する SQL 式を使用するには、このオプションを選 択します。各同期は、SQL 式が true のローのみをダウンロードします。この SQL 式は、生成 された download_cursor スクリプトの WHERE 句に追加されます。Mobile Link 名前付きパラ メータを式で使用できます。また、他のテーブルを参照することもできます。他のテーブル を参照する場合は、式の上にあるフィールドで他のテーブルをリストし、式にジョイン条件 を含める必要があります。

参照

- ●「Mobile Link ユーザ」『Mobile Link クライアント管理』
- ●「リモート ID」『Mobile Link クライアント管理』
- ●「リモートデータベース間でローを分割する」『Mobile Link サーバ管理』
- ●「スクリプトでのリモート ID と Mobile Link ユーザ名」『Mobile Link クライアント管理』
- ●「スクリプトでの最終ダウンロード時刻」『Mobile Link サーバ管理』
- ●「download_cursor スクリプト」『Mobile Link サーバ管理』

ダウンロードサブセットの変更

テーブルごとにダウンロードサブセットをカスタマイズします。各 Mobile Link リモートデータ ベースでは、統合データベースに格納されているデータのサブセットを同期できます。

前提条件

同期モデルが必要です。テーブルマッピングのダウンロードタイプは **[カスタム]** に設定されて いない必要があります。

◆ タスク

- 1. Mobile Link オブジェクト名をダブルクリックします。
- 2. [同期モデル] をダブルクリックし、同期モデル名を選択します。
- 3. [マッピング] タブをクリックします。
- 4. [テーブルマッピング] ウィンドウ枠で、リモートテーブルを選択します。
- 5. [詳細] ウィンドウ枠で、[サブセットのダウンロード] タブを開きます。
- 6. [サブセットのダウンロード] ドロップダウンリストから次のダウンロードサブセットのいず れかを選択します。[なし]、[ユーザ]、[リモート]、または [カスタム]。

7. [ユーザ] または [リモート] を選択した場合は、ユーザ名とリモート ID を保持するカラムが 置かれている場所を識別します。

同期されている統合テーブルにカラムがある場合は、[統合テーブルでカラムを使用する] を 選択し、[カラム名] ドロップダウンリストからユーザ名またはリモート ID を保持するカラム を選択します。

カラムが別のテーブルにある場合は、[共有されている関係テーブル内のカラムを使用する] を選択します。[ジョインするテーブル] ドロップダウンリストからカラムが含まれている テーブルを選択します。[一致させるカラム] ドロップダウンリストで、ユーザ名またはリ モート ID を保持しているカラムを選択します。ジョイン条件を使用して、同期テーブルを シャドウテーブルにジョインするためのジョイン条件を定義します。

- [カスタム] を選択した場合は、2 つのテキストボックスが表示されます。このテキストボックスには download_cursor スクリプトの作成に使用する情報を追加できます。 download_cursor をすべて自分で作成する必要はありません。ダウンロードサブセットのジョインやその他の制限を指定するために追加情報を指定する必要があるだけです。
 - ●download_cursor にその他のテーブルへのジョインが必要な場合は、最初のテキストボック ス ([ダウンロードカーソルの FROM 句に追加するテーブル]) にテーブル名を入力します。 ジョインで複数のテーブルが必要な場合は、カンマで区切ります。
 - ●2 つ目のボックス ([ダウンロードカーソルの WHERE 句で使用する SQL 式]) に、ダウン ロードサブセット条件とジョイン条件を指定する、生成された WHERE 句に追加する SQL 式を入力します。認証パラメータを含む Mobile Link 名前付きパラメータを式で使用でき ます。デフォルトでは、これと同じ式とジョインされたテーブルがダウンロード削除サブ セット用に使用されます。削除の追跡用にシャドウテーブルを使用しており、同じ式を使 用する場合、式にベーステーブル名は使用しないようにします。それができない場合は、 カスタムダウンロード削除サブセットを使用します。

結果

ダウンロードサブセットが変更されます。

例

「ユーザの例」

たとえば、CustDBのULOrder テーブルは、ユーザ間で共有できます。デフォルトで、注文は作成した従業員に割り当てられていますが、別の従業員によって作成された注文を確認したい場合があります。たとえば、マネージャは、部署の従業員が作成したすべての注文を確認する必要があるかもしれません。CustDBデータベースでは、この状況に備えるために、ULEmpCustテーブルを使用します。これにより、顧客を従業員に割り当てることができます。マネージャは、ある従業員と顧客の関係における注文をダウンロードします。

ダウンロードサブセットなしで、ULOrder の download_cursor スクリプトを表示します。次に、 [マッピング] タブで ULEmpCust テーブルを選択します。[ダウンロードタイプ] で [タイムスタ ンプ]、[サブセットのダウンロード] で [なし] をそれぞれ選択します。テーブルを右クリックし、 [イベントに移動] をクリックします。テーブルの download cursor は、次のようになります。 SELECT "DBA"."ULOrder"."order_id", "DBA"."ULOrder"."cust_id", "DBA"."ULOrder"."prod_id", "DBA"."ULOrder"."emp_id", "DBA"."ULOrder"."disc", "DBA"."ULOrder"."quant", "DBA"."ULOrder"."notes", "DBA"."ULOrder"."status" FROM "DBA"."ULOrder"."last modified" >= {ml s.last table download}

[マッピング] タブに戻ります。[詳細] ウィンドウ枠の [サブセットのダウンロード] タブで、 ULOrder の [サブセットのダウンロード] ドロップダウンリストを [ユーザ] に変更します。[ジョ インされている関係テーブル内のカラムを使用する] を選択します。ジョインするテーブルで、 ULEmpCust を選択します。一致させるカラムで、emp_id を選択します。ジョイン条件で、 DBA.ULOrder.cust_id=DBA.ULEmpCust.cust_id を選択します。

一番上のウィンドウ枠でテーブルを右クリックし、[イベントに移動] をクリックします。テーブ ルの download cursor は、次のようになります (新しい行を太字で示しています)。

SELECT "DBA"."ULOrder"."order_id", "DBA"."ULOrder"."cust_id", "DBA"."ULOrder"."prod_id", "DBA"."ULOrder"."disc", "DBA"."ULOrder"."disc", "DBA"."ULOrder"."quant", "DBA"."ULOrder"."notes", "DBA"."ULOrder"."status" FROM "DBA"."ULOrder"."IBA"."ULEmpCust" WHERE "DBA"."ULOrder"."last_modified" >= {ml s.last_table_download} [AND "DBA"."ULOrder".cust_id" = "DBA"."ULEmpCust"."cust_id" AND "DBA"."ULEmpCust"."cust_id" = {ml s.username}]

「カスタムの例」

Mobile Link ユーザによる Customer というテーブルのダウンロードをサブセットし、active=1 で あるローのみをダウンロードするとします。Mobile Link ユーザ名はサブセットするテーブルに 存在しないため、ユーザ名が含まれる SalesRep というテーブルへのジョインを作成する必要が あります。

[マッピング] タブで、Customer テーブルのマッピングを選択します。[詳細] ウィンドウ枠の [ダ ウンロードタイプ] タブを開きます。ダウンロードタイプを [タイムスタンプ] に設定します。 [詳細] ウィンドウ枠の [サブセットのダウンロード] タブを開きます。[サブセットのダウンロー ド] ドロップダウンリストの [カスタム] を選択します。最初のテキストボックス ([ダウンロー ドカーソルの FROM 句に追加するテーブル]) に、次のように入力します。

SalesRep

2 つ目のテキストボックス (**|ダウンロードカーソルの WHERE 句で使用する SQL 式]**) に、次の ように入力します。

SalesRep.ml_username = {ml s.username} AND Customer.active = 1 AND Customer.cust_id = SalesRep.cust_id

cust_id カラムが両方のテーブルにあるので、これらのカラムへの参照は、式でテーブル名をプレフィクスとして付ける必要があります。削除の追跡をダウンロードするためにシャドウテー

ブルを使用する場合、シャドウテーブルが Customer ではなく Customer_del と呼ばれるので、 Customer テーブルマッピングの [ダウンロード削除サブセット] カラムにある [なし] または [カ スタム] を使用する必要があります。

一番上のウィンドウ枠でテーブルを右クリックし、[イベントに移動] をクリックします。テーブ ルの download_cursor は、次のようになります。

SELECT "DBA"."Customer"."cust_id", "DBA"."Customer"."cust_name" FROM "DBA"."Customer", SalesRep WHERE "DBA"."Customer"."last_modified" >= {ml s.last_table_download} AND SalesRep.ml_username = {ml s.username} AND Customer.cust_id = SalesRep.cust_id

WHERE 句の最後の行により、Customer から SalesRep へのキージョインが作成されます。

ダウンロード削除サブセットの変更

ダウンロードサブセットを使用して統合データベースでデータのサブセットを同期している場 合は、デフォルトによりダウンロード削除サブセットが [同じ] に設定され、ダウンロードサブ セットとまったく同じになります。この設定は、[なし] または [カスタム] に変更できます。

ダウンロードサブセットが [カスタム] であり、削除の追跡にシャドウテーブルを使用している 場合は、ローを削除するときに、ダウンロードサブセット論理によって使用されるカラム値をい ずれもシャドウテーブルにコピーする必要があります。プライマリキーカラム値は、シャドウ テーブルに自動的にコピーされます。他の任意のカラム値、具体的には [ダウンロード削除サブ セット] タブ上のカラム値へのアクセスが必要な場合は、[シャドウテーブルのカラムを削除] リ ストで必要なカラムをチェックします。

余計なカラムを指定してシャドウテーブルに追加することを除けば、カスタムダウンロードサブ セットを定義することは、カスタムダウンロード削除サブセットを設定することと同じです。

参照

●「ダウンロードサブセットの変更」40ページ

競合の検出と解決

リモートデータベースと統合データベースの両方でローが更新された場合は、次にデータベース を同期するときに競合が発生します。

競合の検出には、次のオプションがあります。

● [**ローベースの競合検出**] 最後の同期後に、リモートデータベースと統合データベースの両方 でローが更新されていた場合に競合が検出されます。

このオプションは、upload_fetch スクリプトと upload_update スクリプトを定義します。 「upload_fetch または upload_fetch_column_conflict スクリプトによる競合の検出」『Mobile Link サーバ管理』を参照してください。 ● [カラムベースの競合検出] リモートデータベースと統合データベースの両方で、ローの同じ カラムが更新されていた場合に競合が検出されます。

このオプションは、upload_fetch_column_conflict スクリプトを定義します。 「upload_fetch ま たは upload_fetch_column_conflict スクリプトによる競合の検出」『Mobile Link サーバ管理』を 参照してください。

テーブルに BLOB があり、カラムベースの競合検出を選択した場合は、ローベースの競合検 出が使用されます。

競合の解決には、次のオプションがあります。

- [統合] 先入れ勝ちです。アップロードされた更新が競合する場合は破棄されます。
- **[リモート]** 後入れ勝ちです。アップロードされた更新が常に適用されます。
- [タイムスタンプ] 最新の更新が適用されます。このオプションを使用するには、テーブル の TIMESTAMP カラムを作成し、維持する必要があります。この TIMESTAMP カラムに、 ローが最後に変更された時刻が記録されます。統合データベースとリモートデータベースの 両方にカラムが存在し、タイムスタンプベースのダウンロードで使用されるカラムと異なっ ている必要があります。これを機能させるには、リモートデータベースと統合データベース で同じタイムゾーン (UTC を推奨) を使用し、かつクロックが同期されている必要がありま す。
- [カスタム] 独自の resolve_conflict スクリプトを作成します。スクリプトは [イベント] タブ で作成できます。「resolve_conflict スクリプトによる競合の解決」『Mobile Link サーバ管理』 を参照してください。

参照

- ●「競合の検出と解決の変更」44ページ
- ●「競合の解決」『Mobile Link サーバ管理』

競合の検出と解決の変更

Sybase Central を使用して競合の検出および解決の設定をカスタマイズします。

前提条件

同期モデルが必要です。

◆ タスク

- 1. Mobile Link オブジェクト名をダブルクリックします。
- 2. 同期モデル名をダブルクリックします。
- 3. [マッピング] タブをクリックします。
- 4. [テーブルマッピング] ウィンドウ枠で、テーブルマッピングを選択します。

- 5. [詳細] ウィンドウ枠で、[競合の処理] タブを開きます。
- 6. [競合の検出] ドロップダウンリストで、[ローベース] または [カラムベース] のいずれかをク リックします。
- 7. [競合の解決] ドロップダウンリストから、[統合]、[リモート]、[タイムスタンプ]、または [カ スタム] を選択します。
- 8. [タイムスタンプ] 競合解決を選択した場合は、使用する [TIMESTAMP カラム] の名前を入力 します。
- 9. [カスタム] を選択した場合は、[イベント] タブを開き、テーブルの resolve_conflict スクリプトを作成します。

結果

競合の検出と解決のための設定が更新されます。

同期モデル内のスクリプトの変更

同期モデル内のスクリプトは、[イベント]タブを使用して変更します。

[イベント] タブでは、次のタスクを実行できます。

●現在のテーブルマッピングに基づいて生成されたスクリプトを表示および変更します。

●新しいスクリプトを作成する。

[イベント] タブの上部には、選択したスクリプトが属するグループが表示されます。単一テーブルのスクリプトはすべて1つにまとめられています。[イベント] タブの上部には、選択したスクリプトの名前が表示されます。また、このスクリプトが Mobile Link プラグインによって生成されたものかどうか、ユーザによって定義されたものかどうか、または生成したスクリプトが上書きされているかどうかも表示されます。また、同期論理が SQL、.NET、Java のどれを使用して作成されているかも表示されます。

このスクリプトは、生成されたスクリプトを上書きするスクリプトを追加すると、完全に制御で きるようになります。したがって、関連する設定を変更しても、このスクリプトが自動的に変更 されることはありません。たとえば、モデルの download_delete_cursor を変更した後に [マッピン グ] タブの [テーブルマッピング] ウィンドウ枠で [削除] カラムの選択を解除しても、カスタマイ ズした download_delete_cursor に影響はありません。

[ファイル] メニューのオプションを使用すると、上書き生成スクリプトをリストアしたり、無視 されるように設定したスクリプトをリストアしたり、追加した新しいスクリプトを削除したりす ることができます。リストアまたは削除するスクリプトを選択してから[ファイル]をクリック して、オプションを表示します。

特定のテーブルのスクリプトの検索

Sybase Central を使用して、特定のテーブルに対して定義されたスクリプトを検索します。

前提条件

同期モデルが必要です。

◆タスク

- 1. Mobile Link オブジェクトをダブルクリックします。
- 2. 同期モデル名をダブルクリックします。
- 3. [イベント] タブを開きます。
- 4. [グループ] ドロップダウンで、確認するイベントが含まれているテーブルを選択します。
- 5. [イベント] フィールドで、検索するスクリプト名を選択します。既存のスクリプトは太字で 強調表示されます。

結果

選択したスクリプトにカーソルが移動します。

次の手順

選択したスクリプトに変更を加えます。

POP3、IMAP、LDAP の各外部サーバに対する認証

[認証] タブを使用して、同期モデルの外部サーバに対して同期認証を有効にします。

前提条件

同期モデルが必要です。

◆タスク

- 1. Mobile Link オブジェクト名をダブルクリックします
- 2. [同期モデル] をダブルクリックし、同期モデル名をダブルクリックします。
- 3. 右ウィンドウ枠で、[認証] タブを開きます。
- 4. [この同期モデルのカスタム認証を有効にする] を選択します。
- 5. 認証するサーバのタイプを選択します。
- 6. 適切なホスト、ポート、URL 情報をそれぞれのフィールドに入力します。

これらのフィールドの詳細については、「外部認証識別符号プロパティ」『Mobile Link クライ アント管理』を参照してください。

結果

認証が有効になります。

スキーマの更新

スキーマ更新ウィザードを使用して、同期モデル内の統合データベースとリモートデータベー スのスキーマを更新します。

前提条件

同期モデルが必要です。

内容と備考

スキーマ更新ウィザードは、モデルを展開した後と、次の場合に最適です。

- ●モデルに追加する必要があるリモートデータベースのスキーマを変更した場合。
- ●モデルに追加する必要がある統合データベースのスキーマを変更した場合。

たとえば、1 つまたは複数のテーブルにタイムスタンプベースのダウンロードを作成したモデルを再展開する前に、[スキーマの更新]を実行したとします。この場合は、以前に展開したときに、TIMESTAMPカラムまたはシャドウテーブルを追加して、統合データベースのスキーマを変更したために、スキーマを更新する必要があります。

◆タスク

- 1. Mobile Link オブジェクト名をダブルクリックします。
- 2. [同期モデル] をダブルクリックします。
- 3. 同期モデル名をダブルクリックし、[ファイル] » [スキーマ更新] をクリックします。
- 4. 次のオプションのうちの1つを選択してください。
 - [統合データベーススキーマ] モデル内の統合スキーマは更新されます。モデル内のリ モートスキーマは変更されません。
 - [リモートデータベーススキーマ] モデル内のリモートスキーマは更新されます。モデル 内の統合スキーマは変更されません。
 - [統合データベーススキーマとリモートデータベーススキーマ] 統合およびリモートのス キーマの両方が、既存のデータベースのスキーマに一致するようにモデル内で更新されま す。
- 5. スキーマ更新ウィザードの指示に従います。

[完了] をクリックしても、同期モデルを展開するまでモデル以外での変更は行われません。 そのときまで統合データベースは変更されず、リモートデータベースは作成も変更もされま せん。 6. [マッピング] タブで、新しいリモートテーブルをマッピングします。

結果

スキーマが更新されます。

参照

●「テーブルとカラムのマッピング」33ページ

同期モデルの展開

同期モデルは、同期モデル展開ウィザードを使用して展開します。

次の項目を展開できます。

- ●統合データベースの変更内容
- ●SQL Anywhere または Ultra Light リモートデータベース (データベースを作成するか、既存の 空のデータベースにテーブルを追加するか、リモートテーブルがすでに格納されている既存の データベースを使用)
- ●モデルを展開するバッチファイル(生成されるバッチファイルの先頭には変数宣言があり、 バッチファイルの実行前にこの変数宣言を編集できます)

●Mobile Link サーバと Mobile Link クライアントを実行するためのバッチファイル

Mobile Link 16 プラグインでは、同期モデルを展開すると、スキーマの比較が行われます。デー タベースオブジェクトの中でも展開対象のオブジェクトと異なるもの(たとえば、テーブルやイ ンデックスなど)のみが変更されます。新しいオブジェクトを展開することにより既存の同期シ ステムが壊れる場合は警告が表示されます。操作は中止することができます。この機能は、IBM DB2 LUW 統合データベースでは利用できません。

同じスクリプトバージョンに更新を再展開すると、新しいスクリプトバージョンで必要なくなったスキーマは自動的に削除されます。たとえば、ダウンロードをサポートするために last_modified カラムがテーブルに追加されている場合、同期しかアップロードしないように変更 すると、last modified カラムは展開中に削除されます。

また、ml_model_drop システムプロシージャを使用すれば、統合データベースから同期モデルと そのスキーマを削除することができます。このシステムプロシージャは、同期スクリプトを削除 するほか、同期モデルが展開されたときに作成されたスキーマ(シャドウテーブル、トラッキン グカラム、トリガ、インデックスなど)も削除します。別の script_version と共有されているス キーマは削除されません。Mobile Link 16 プラグインを使用してインストールしたスキーマしか 削除できません。この機能は、IBM DB2 LUW 統合データベースでは利用できません。 「ml_model_drop システムプロシージャ」『Mobile Link サーバ管理』を参照してください。

統合データベースへの配備

同期モデル展開ウィザードでは、次の2つの方法で統合データベースに展開できます。

● Mobile Link システムテーブルにデータを追加し、必要なシャドウテーブル、カラム、トリガ、ストアドプロシージャを作成することで、同期モデルを統合データベースに直接適用します。

● 同じ変更内容をすべて含む UTF-8 でエンコードされたファイルとバッチファイルを作成して、統合データベースに対して SQL ファイルを実行します。このファイルは、いつでも確認、変更、実行することができます。結果は変更を直接適用する場合と同じです。

展開前の同期モデルのテスト

[テスト] ウィンドウを使用し、展開する前に同期モデルをテストします。

前提条件

Mobile Link プロジェクトで定義された同期モデルが必要です。

内容と備考

テスト機能を使用すると、統合データベースに変更が加えられます。

◆タスク

- 1. Mobile Link オブジェクト名をダブルクリックします
- 2. [同期モデル] をダブルクリックし、同期モデル名をダブルクリックします。
- [展開] ウィンドウ枠で、[テスト] をクリックします。同期モデルをテストすると統合データ ベースに変更が加えられテーブル内のデータが変更されることを示す警告が表示されます。
 [OK] をクリックします。
- 4. [同期] をクリックします。

結果

同期モデルがテストされ、同期が成功したかどうかを示します。

次の手順

テストが完了したら、次のタスクを実行できます。

- ●[データ] タブに関する情報を確認し、統合データベースのデータとリモートデータベースの データを比較します。
- ●テスト中の失敗に関する情報については、[クライアントログ] タブの情報を確認してください。これは、SQL Anywhere クライアントでのみ使用できます。
- ●テスト中の失敗に関する情報については、[Mobile Link ログ] タブの情報を確認してください。

- ●同期をもう一度実行する前に、リモートデータベースと統合データベース、またはそのいずれ かを変更します。データベースを変更するには、いくつかのオプションを使用できます。
 - [データ] タブを使用して、同期されているテーブル内のデータを直接変更する。
 - **[アクション]** メニューを使用して、いずれかのデータベースで dbisql ユーティリティを開く。

SQL ファイルからの統合データベースの展開

同期モデル展開ウィザード を使用して、統合データベースを展開します。

前提条件

展開時にシャドウテーブルを作成した場合は、シャドウテーブルが作成されるベーステーブルの 所有者または管理者として、統合データベースに接続する必要があります。

◆タスク

● 同期モデル展開ウィザードの実行時に、([データベースの同期を準備する方法を選択します] ページで)後で実行できるようにファイルを作成することを選択した場合は、モデルの project-name_deploy サブフォルダにあるバッチファイルを実行する必要があります。この ファイルによって、同期スクリプト、シャドウテーブル、トリガなど、統合データベースに 作成することを選択したオブジェクトがすべて作成されます。また、Mobile Link ユーザが統 合データベースに登録される場合もあります。

このファイルを実行するには、project-name_deploy ディレクトリに移動し、cons_setup.bat または cons_setup.sh ファイルを実行します。このとき、接続情報を指定する必要があります。たとえば、次のように入力します。

cons_setup.bat "DSN=my_odbc_datasource;UID=myuserid;PWD=mypassword"

一部のドライバでは、ODBC データソースにユーザ ID とパスワードがあるため、これらの指定は不要です。

結果

統合データベースが展開されます。

リモートデータベースの展開

既存のリモートデータベースを使用するか、ウィザードでリモートデータベースを作成できま す。ウィザードでは、リモートデータベースを直接作成するか、または、リモートデータベース を作成または更新するために実行する、UTF-8 でエンコードされた SQL ファイルとバッチファ イルを作成できます。

ウィザードでは、モデルで指定したデータベース所有者を使用して、デフォルトのデータベース オプションでリモートデータベース (SQL Anywhere または Ultra Light) が作成されます。また、 同期モデル展開ウィザードを使用しないでカスタム設定でリモートデータベースを作成し、ウィ ザードを使用して必要なリモートテーブルを追加するか、すでにリモートテーブルがある既存の リモートデータベースに展開することもできます。

SQL ファイルからのリモートデータベースの展開

同期モデル展開ウィザード を使用して、リモートデータベースを展開します。

前提条件

この作業を実行するための前提条件はありません。

◆ タスク

 同期モデル展開ウィザードの実行時に、([データベースの同期を準備する方法を選択します] ページで)後で実行できるようにファイルを作成することを選択した場合は、projectname_deploy ディレクトリにある SQL ファイルで作成されたバッチファイルを実行する必要 があります。このファイルによって、テーブル、パブリケーション、サブスクリプション、 Mobile Link ユーザなど、リモートデータベースに作成することを選択したオブジェクトがす べて作成されます。

このファイルを実行するには、project-name_deploy ディレクトリに移動し、remote_setup.bat または remote setup.bat.sh ファイルを実行します。たとえば、次のように入力します。

remote_setup.bat

既存のリモートデータベースを使用している場合は、パスワードの入力を要求されます。

結果

リモートデータベースが展開されます。

バッチファイルを展開して同期ツールを実行

ウィザードでは、次のバッチファイルを作成できます。

- 指定するオプションで Mobile Link サーバを実行するバッチファイル
- SQL Anywhere のリモートデータベースの場合、指定するオプションで dbmlsync を実行する バッチファイル
- Ultra Light のリモートデータベースの場合、指定するオプションで ulsync を実行するバッチ ファイル。ulsync は、同期のテストに使用されるので、正常に機能する Ultra Light アプリケー ションがない場合に初めて同期するときに役立ちます。

同期モデルの再展開

同期モデルを展開した後、同期モデルに変更を加えて再展開すると、同期モデルを変更できま す。また、システムプロシージャや他のメソッドを使用して変更することもできます。ただし、 展開後のモデルを Sybase Central 外で変更した場合、変更内容をリバースエンジニアリングで同 期モデルに戻すことはできません。Sybase Central 外で加えられた変更は、モデルを再展開する ときに上書きされます。 展開することでスキーマが変更されてしまう場合があるので、その他の変更を行っていないときでも、スキーマを更新する必要があります。たとえば、統合データベース内の各同期テーブルにTIMESTAMPカラムを追加するモデルを展開する場合は(モデル作成時のデフォルトの動作)、モデル内の統合スキーマを更新してから再展開してください。同様に、統合データベースにテーブルを追加してから再展開する場合は、モデル内の統合スキーマを更新してから、新しいリモートテーブルを作成する必要があります。

「スキーマの更新」47ページを参照してください。

展開した同期モデル

同期モデルファイルは、Mobile Link プロジェクトディレクトリにあります。同期ファイルのファイル拡張子は、.*mlsm*です。同期モデルを展開すると、該当するモデル名で始まり_*deploy*で終わるディレクトリが作成されます。

この場合、次のようなファイルが作成されます。作成されるファイルは選択した展開オプション によって異なります。

ファイル	説明
cons_setup.bat	SQL ファイルを実行して統合データベースを設定す るバッチファイル。
cons_setup.sql	統合データベースの設定に使用する SQL ファイル。
mlsrv.bat	Mobile Link サーバを実行するためのバッチファイル。
remote_setup.bat	SQL ファイルを実行してリモートデータベースを設 定するバッチファイル。
remote_setup.sql	統合データベースの設定に使用する SQL ファイル。
summary.txt	同期モデルの詳細を要約するファイル。
sync.bat	SQL Anywhere リモートデータベースを展開した場合の、SQL Anywhere データベースを dbmlsync と同期するバッチファイル。
modelname_remote.db	新しい SQL Anywhere リモートデータベースを作成 する場合のデータベースファイル。
modelname_remote.udb	新しい Ultra Light リモートデータベースを作成する 場合のデータベースファイル。

ファイル	説明
modelname_ulsync.bat	Ultra Light データベースを展開した場合の、ulsync ユーティリティを使用して Ultra Light リモートデー タベースとの同期をテストするためのバッチファイ ル。

バッチファイルの実行

同期モデル展開ウィザードで作成されたバッチファイルをコマンドラインから実行して、同期モ デルを同期します。

前提条件

同期モデルが必要です。

内容と備考

統合データベースで Mobile Link 設定スクリプトを実行していない場合は、実行してから展開します。 「統合データベースの設定」『Mobile Link サーバ管理』を参照してください。

バッチファイルの多くは実行時に接続情報の指定を必要とします。これらのバッチファイルを 実行する前に ODBC データソースを作成する必要がある場合もあります。「ODBC データソー ス」『SQL Anywhere サーバ データベース管理』を参照してください。

◆タスク

 同期モデル展開ウィザードの実行時に、([データベースの同期を準備する方法を選択します] ページで)後で実行できるようにファイルを作成することを選択した場合は、モデルの project-name_deploy サブフォルダにあるバッチファイルを実行する必要があります。この ファイルによって、同期スクリプト、シャドウテーブル、トリガなど、統合データベースに 作成することを選択したオブジェクトがすべて作成されます。また、Mobile Link ユーザが統 合データベースに登録される場合もあります。

このファイルを実行するには、project-name_deploy ディレクトリに移動し、cons_setup.bat または cons_setup.sh ファイルを実行します。コマンドラインには、接続情報を含めてください。たとえば、次のように入力します。

cons_setup.bat "DSN=MY_ODBC_DATASOURCE"

 同期モデル展開ウィザードの実行時に、([データベースの同期を準備する方法を選択します] ページで)後で実行できるようにファイルを作成することを選択した場合は、projectname_deploy ディレクトリにあるバッチファイルを実行する必要があります。このファイル によって、テーブル、パブリケーション、サブスクリプション、Mobile Link ユーザなど、リ モートデータベースに作成することを選択したオブジェクトがすべて作成されます。

このファイルを実行するには、project-name_deploy ディレクトリに移動し、remote_setup.bat または remote_setup.sh ファイルを実行します。たとえば、次のように入力します。

remote_setup.bat

既存のリモートデータベースを使用している場合は、パスワードの入力を要求されます。

3. *mlsrv.bat* を実行して Mobile Link サーバを起動します。コマンドラインには、統合データベースの接続情報を含めてください。たとえば、次のように入力します。

mlsrv.bat "DSN=MY_ODBC_DATASOURCE"

4. 同期を実行します。

SQL Anywhere リモートデータベースの場合

●dbmlsync を実行できるようにユーザに SYS_RUN_REPLICATION_ROLE システムロール を付与します。たとえば、Interactive SQL で次のコマンドを実行します。

GRANT ROLE SYS_RUN_REPLICATION_ROLE TO userid

●ロールを付与されたユーザとしてデータベースに接続します。

●*project-name_deploy* ディレクトリにあるリモートデータベースを起動します。たとえば、 次のように入力します。

dbsrv16 MyModel_remote.db

●SQL Anywhere Mobile Link クライアント dbmlsync を起動します。project-name_deploy ディ レクトリにあるファイル sync.bat を実行します。コマンドラインには、接続情報を含めて ください。たとえば、次のように入力します。

sync.bat "UID=userid;PWD=password;SERVER=MyModel_remote"

Ultra Light リモートデータベースの場合

●同期をテストするには、remote ディレクトリに存在する_ulsync.bat で終わるファイルを実行します。

●Ultra Light アプリケーションを実行することもできます。

結果

同期モデルが再展開されます。

参照

- ●「GRANT ROLE SYS_RUN_REPLICATION_ROLE 文 [Mobile Link] [SQL Remote]」 『SQL Anywhere サーバ SQL リファレンス』
- ●「GRANT ROLE SYS_REPLICATION_ADMIN_ROLE 文 [Mobile Link] [SQL Remote]」 『SQL Anywhere サーバ SQL リファレンス』
- ●「dbmlsync の権限」『Mobile Link クライアント管理』

同期モデルの制限事項

次に、同期モデルの制限事項をいくつか示します。

● モデル以外で行われた変更の再展開はできない 同期モデルを展開し、その後でモデル以外で 変更を加えた場合、この変更はモデルに保存されません。モデルを開始ポイントとして使用 して展開し、すべての変更をモデル以外で行う場合はこれで問題ありません。ただし、モデ ルを再展開する場合は、Mobile Link プロジェクトで変更を行い、変更の保存と再展開ができ るようにする方が適切です。

- バージョン 1つの同期モデルに設定できるバージョンは1つのみです。「スクリプトバー ジョン」『Mobile Link サーバ管理』を参照してください。
- Mobile Link システムデータベース 同期モデルを展開する場合、統合データベースとは別の Mobile Link システムデータベースを使用することはできません。 「Mobile Link システムデータベース」『Mobile Link サーバ管理』を参照してください。
- 複数のパブリケーション 複数のパブリケーションを作成することはできません。モデルの 展開後、CREATE PUBLICATION 文などの非モデルメソッドを使用して、パブリケーション をさらに追加することはできます。ただし、この方法で追加したパブリケーションをリバー スエンジニアリングでモデルに戻すことはできません。「パブリケーション」『Mobile Link クライアント管理』を参照してください。
- **ビュー** テーブルマッピング用に統合データベースのテーブルを選択する場合、ビューは選 択できません。
- ●計算カラム 統合データベースのテーブルで、計算カラムはアップロードできません。計算 カラムがある同期モデルを展開すると、タイムスタンプベースのダウンロードに使用するト リガの作成に失敗する場合があります。カラムを同期対象から除外するか、テーブルをダウ ンロード専用に設定します(このとき、スナップショットダウンロードを使用するか、生成済 み統合 SQL ファイルを編集して、計算カラムをトリガ定義から削除します)。

計算カラムをコピーすると、新しいリモートスキーマを展開して新しいリモートデータベー スを作成する際に構文エラーが発生します。計算カラムを扱う際は、次のいずれかを行いま す。

○同期モデルを既存のリモートデータベースに展開する。

○リモートスキーマから計算カラムを除外する。計算カラムがある統合データベースのテー ブルを同期する場合、テーブルにアップロードすることはできません。

Microsoft SQL Server AdventureWorks サンプルデータベースには、計算カラムが含まれていま す。このデータベースを使用してモデルを作成する場合、カラムをダウンロード専用に設定 するか、カラムを同期対象から除外します。

- Oracle の XMLTYPE データ型 Sybase Central の Mobile Link プラグインでは、Oracle XMLTYPE が次のようにサポートされています。
 - 新しいリモートスキーマに XMLTYPE カラムが選択された場合、またはカラムマッピン グを編集するとき、表示される [データタイプ] 名は XMLTYPE となります。
 - XMLTYPE カラムは、新しい SQL Anywhere リモートデータベースの XML タイプのカラ ムにマッピングされます。「Oracle データのマッピング」『Mobile Link サーバ管理』を参 照してください。
 - 生成された同期スクリプトは、XMLTYPE カラム値が4KB 未満の場合に動作します。 値 がそれよりも大きい場合は、同期モデル内の生成されたスクリプトを、Oracle の

XMLTYPE データ型『Mobile Link サーバ管理』 に記載されている技法を使用したスクリ プトで上書きしてください。

配備に関する考慮事項

- 空間カラム 空間カラムはコピーされます。ただし、空間サブタイプとSRID については、 それらを取得するためのメタデータサポート (SQL/MM 標準の ST_GEOMETRY_COLUMNS ビューなど) が統合 RDBMS にない場合は、コピーされないことがあります。Ultra Light での 空間サポートは、SRID=0 または SRID=4326 のポイント値とカラム SRID 制約のみをサポー トするタイプ (ST_GEOMETRY) に制限されています。このため、互換性のない空間タイプを 新しい Ultra Light データベースに展開するときに警告またはエラーが表示されることがあり ます。
- 長いオブジェクト名 配備時に作成されるデータベースオブジェクトには、データベースで サポートされている長さより長い名前が割り当てられる場合があります(新しいオブジェク ト名がベーステーブル名にサフィックスを追加して作成されるため)。この場合、(データ ベースに直接ではなく)ファイルのみに配備し、生成された SQL ファイルを編集して、長す ぎる名前をすべて置換します。
- 新しいリモートスキーマ [同期モデル作成ウィザード] で新しいリモートスキーマを作成す る場合、新しいリモートデータベースのカラムには統合データベース内のカラムのインデッ クスは格納されません。外部キーとデフォルトのカラム値は新しいリモートデータベースに コピーされます。ただし、このサポートは ODBC ドライバによって返されるデータベースの メタデータに依存しており、ドライバの問題が原因で構文エラーやその他のエラーが生じる 可能性があります。たとえば、ドライバがレポートしたデフォルトのカラム値が、SQL Anywhere または Ultra Light リモートデータベースではデフォルト値の宣言に使用できない フォーマットだった場合、エラーが起こる可能性があります (展開時の構文エラーなど)。

Ultra Light では、NCHAR(n)、NVARCHAR(n)、LONG NVARCHAR の各カラムタイプはサポートされていません。同期モデルを新しい Ultra Light データベースに展開する場合、リモートスキーマ内のこのようなカラムは CHAR(4n)、VARCHAR(4n)、または LONG VARCHAR に変換されます。4n が CHAR と VARCHAR の最大長を超える場合は、最大長が使用され、警告が表示されます。

既存のリモートデータベースを使用して、同期モデルを作成するか、モデル内でリモートス キーマを更新できます。

- プロキシテーブル プロキシテーブルである統合データベーステーブルを別のデータベース と同期することはできますが、タイムスタンプベースのダウンロードのために TIMESTAMP カラムを使用する場合は、ベーステーブルとプロキシテーブルの両方に TIMESTAMP カラム を追加してください。同期モデル展開ウィザードではカラムをプロキシテーブルやそのベー ステーブルに追加することはできません。そのため、ベーステーブルとプロキシテーブルの 両方に存在するカラムを使用するか、シャドウテーブルまたはスナップショットダウンロー ドを使用する必要があります。
- ●マテリアライズドビュー タイムスタンプベースのダウンロードを使用していて、 TIMESTAMPカラムを統合テーブルに追加するように選択した場合は、テーブルに依存する マテリアライズドビューを無効にしてから展開を開始します。こうしないと、テーブルの変 更時にエラーが発生する場合があります。SQL Anywhere 統合データベースでは、 sa dependent viewsシステムプロシージャを使用して、テーブルに依存するマテリアライズド

ビューがあるかどうかを判別します。 「sa_dependent_views システムプロシージャ」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

その他の考慮事項

● Oracle 統合データベースに基づくリモートデータベースの作成 Oracle 統合データベースに基づいて SQL Anywhere または Ultra Light リモートデータベースを作成する場合、統合データベースの DATE カラムを TIMESTAMP に変更することが必要になる場合があります。この 処理を行わないと、秒未満の情報が更新時に失われます。

Mobile Link の CustDB サンプル

CustDB は販売管理アプリケーションです。CustDB サンプルは、Mobile Link 開発者にとって貴 重なリソースです。このサンプルを使用して、Mobile Link アプリケーションの開発時に必要な さまざまな方法の実装例を紹介します。

このアプリケーションを使用して、いくつかの一般的な同期方法を説明します。この項で説明することを最大限に活用するために、サンプルアプリケーションのソースコードを読んで理解してください。

サポートされるオペレーティングシステムとデータベースの種類ごとに、CustDB が用意されています。

CustDB 統合データベースを使用する Mobile Link プロジェクトは、%SQLANYSAMP16% ¥MobiLink¥CustDB¥project.mlp ディレクトリにあります。このプロジェクトを Sybase Central で 開いて、CustDB プロジェクトとともに使用し、データベーススクリプトを表示できます。

CustDB シナリオ

統合データベースは、本社に配置されています。以下のデータが統合データベースに格納されま す。

- ●同期されるメタデータを格納する Mobile Link システムテーブル。同期論理を実装する同期ス クリプトが含まれています。
- ●すべての顧客、製品、注文の情報を含み、ベーステーブルのローに格納される CustDB のデータ
- リモートデータベースは、モバイル管理者用と営業担当者用の2種類があります。

各モバイル営業担当者のデータベースにはすべての製品とその営業担当者に対応する注文のみ が格納されていますが、モバイル管理者のデータベースには特定の顧客セットに対するすべての 製品と注文が格納されています。

同期の設計

CustDB サンプルアプリケーションでは、以下の機能を使用して同期を行います。

● 完全なテーブルのダウンロード ULProduct テーブルのすべてのローとカラムが、リモート データベースでも完全に共有される。

- **カラムのサブセット** ULCustomer テーブルの一部のカラムのすべてのローが、リモートデー タベースでも共有される。
- **ローのサブセット** ULOrder テーブルから、リモートユーザごとに異なるローセットを取得 する。

ローのサブセットの詳細については、「リモートデータベース間でローを分割する」『Mobile Link サーバ管理』を参照してください。

● **タイムスタンプベースの同期** 最後のデバイス同期以降に実行された統合データベースに対 する変更を識別する方法。ULCustomer テーブルと ULOrder テーブルは、タイムスタンプ ベースの方法で同期される。

「タイムスタンプベースのダウンロードの実装」『Mobile Link サーバ管理』を参照してください。

● スナップショットを使った同期 同期を実行するたびにすべてのローをダウンロードする単純な同期方法。ULProduct テーブルは、この方法で同期される。

「スナップショットを使った同期」『Mobile Link サーバ管理』を参照してください。

● プライマリキープール (ユニークなプライマリキー管理用) プライマリキーの値を、Mobile Link インストール環境全体で確実にユニークにする必要がある。このアプリケーションで 使われるプライマリキープールメソッドは、プライマリキーをユニークにする方法の1つで ある。

「プライマリキープール」『Mobile Link サーバ管理』を参照してください。

プライマリキーをユニークにする他の方法については、「ユニークなプライマリキー」『Mobile Link サーバ管理』を参照してください。

参照

- ●「チュートリアル: Ultra Light の CustDB サンプルアプリケーションの構築」『Ultra Light デー タベース管理とリファレンス』
- ●「チュートリアル: Ultra Light の CustDB サンプルアプリケーションの構築」『Ultra Light デー タベース管理とリファレンス』
- ●「CustDB サンプルデータベースアプリケーション」『SQL Anywhere 16 紹介』
- ●「CustDB 統合データベースの設定」59ページ

CustDB ファイル

この項では、CustDB サンプルアプリケーションとサンプルデータベースを作成するコードを部分的に説明します。これらのコードを以下に示します。

●サンプル SQL スクリプト。%SQLANYSAMP16%¥MobiLink¥CustDB にあります。

・ ●アプリケーションのコード。%SQLANYSAMP16%¥UltraLite¥CustDBにあります。

●プラットフォーム固有のユーザインタフェースのコード。%*SQLANYSAMP16%*¥UltraLite ¥CustDB の各オペレーティングシステムの名前が付いたサブディレクトリにあります。

CustDB 統合データベースの設定

MobileLink でサポートされている任意の統合データベースを CustDB 統合データベースとして 使用できます。

SQL Anywhere 16 CustDB

SQL Anywhere 16 CustDB 統合データベースは、*%SQLANYSAMP16%¥UltraLite¥CustDB¥custdb.db* にあります。SQL Anywhere 16 CustDB という DNS はインストール環境に含まれています。

このデータベースは、%SQLANYSAMP16%¥UltraLite¥CustDB¥makedbs.cmd ファイルを使用して再構築できます。

CustDB サンプルの構造を詳しく調べるには、%*SQLANYSAMP16%¥MobiLink¥CustDB¥custdb.sql* ファイルを参照してください。

その他の RDBMS 用の CustDB

次の SQL スクリプトを使用すると、サポートされている RDBMS のいずれかで、CustDB 統合 データベースを構築できます。これらのスクリプトは、%SQLANYSAMP16%¥MobiLink¥CustDB にあります。

RDBMS	CustDB 設定スクリプト
Adaptive Server Enterprise	custase.sql
SQL Server	custmss.sql
Oracle	custora.sql
DB2 LUW	custdb2.sql
MySQL	custmys.sql

統合データベースとして使用するデータベースを準備する方法の詳細については、「統合データ ベースの設定」『Mobile Link サーバ管理』を参照してください。

CustDB を統合データベースとして構築 (Adaptive Server Enterprise、MySQL、 Oracle、または SQL Server の場合)。

次の手順を実行すると、サポートされている任意の RDBMS 用の CustDB 統合データベースが作成されます。

前提条件

サポートされている RDBMS のいずれであっても、CustDB 統合データベースを構築する場合に 使用される SQL スクリプトへのアクセス権が必要です。SQL スクリプトは、 %SQLANYSAMP16%¥MobiLink¥CustDB¥ にあります。

◆ タスク

- 1. 使用している RDBMS でデータベースを作成します。
- 以下の SQL スクリプトのいずれかを実行して Mobile Link システムオブジェクトを追加します。これらのスクリプトは、SQL Anywhere 16 インストール環境の MobiLink¥setup サブフォルダにあります。
 - ●Adaptive Server Enterprise 統合データベースの場合は、*syncase.sql* を実行します。
 - ●MySQL 統合データベースの場合は、syncmys.sql を実行します。
 - ●Oracle 統合データベースの場合は、syncora.sql を実行します。
 - ●SQL Server 統合データベースの場合は、syncmss.sql を実行します。
- 3. 以下の SQL スクリプトのいずれかを実行して、サンプルユーザテーブル、ストアドプロシー ジャ、Mobile Link 同期スクリプトを CustDB データベースに追加します。これらは、 %SQLANYSAMP16%¥MobiLink¥CustDB にあります。
 - ●Adaptive Server Enterprise 統合データベースの場合は、*custase.sql* を実行します。
 - ●MySQL 統合データベースの場合は、custmys.sql を実行します。
 - ●Oracle 統合データベースの場合は、custora.sql を実行します。
 - ●SQL Server 統合データベースの場合は、custmss.sql を実行します。
- 4. クライアントコンピュータ上で、データベースを参照する CustDB という ODBC データソー スを作成します。
 - a. [スタート]»[プログラム]»[SQL Anywhere 16]»[管理ツール]»[ODBC データソースア ドミニストレータ] (32 ビットまたは 64 ビット)を選択します。
 - b. [追加] をクリックします。
 - c. リストから適切なドライバを選択します。
 [完了] をクリックします。
 - d. この ODBC データソースに CustDB という名前を付けます。
 - e. [ログイン] タブをクリックします。データベースの [ユーザ ID] と [パスワード] を入力 します。
- 5. [OK] をクリックし、もう一度 [OK] をクリックします。

結果

選択した RDBMS 用に CustDB 統合データベースが作成されます。

CustDB を統合データベースとして構築 (DB2 LUW の場合)

この手順を使用して、DB2 LUW 統合データベースを設定します。

前提条件

この作業を実行するための前提条件はありません。

◆ タスク

- 1. DB2 LUW サーバ上に CustDB という統合データベースを作成します。
- 2. デフォルトのテーブル領域 (通常は USERSPACE1) が 8 KB ページを使用することを確認します。

デフォルトのテーブル領域が8KBページを使用しない場合は、次の手順を行います。

- a. 1つ以上のバッファプールに8KBページがあることを確認します。ない場合は、8KB ページのバッファプールを作成してください。
- b. 8 KB ページのある新しいテーブル領域とテンポラリテーブル領域を作成します。 詳細については、DB2 LUW のマニュアルを参照してください。
- 3. 以下のように、*MobiLink¥setup¥syncdb2.sql*ファイルを使用して、Mobile Link システムオブ ジェクトを DB2 LUW 統合データベースに追加します。
 - a. syncdb2.sql ファイルの先頭にある接続コマンドを変更します。DB2Database を、お使い のデータベース名 (またはそのエイリアス) に置き換えます。この例では、このデータ ベースを CustDB と呼びます。以下に示すように、DB2 のユーザ名とパスワードを追加 することもできます。

connect to CustDB user userid using password ~

b. サーバまたはクライアントコンピュータで、DB2 LUW コマンドウィンドウを開きます。 次のコマンドを入力して *syncdb2.sql* を実行します。

db2cmd db2 -c -ec -td~ +s -v -f syncdb2.sql

- 4. データテーブル、ストアドプロシージャ、Mobile Link 同期スクリプトを CustDB データベー スに追加します。
 - a. 必要に応じて、custdb2.sqlの接続コマンドを変更します。たとえば、以下に示すように、 ユーザ名とパスワードを追加できます。*userid と password*を、使用するユーザ名とパス ワードに置き換えてください。

connect to CustDB user userid using password

- b. サーバまたはクライアントコンピュータで、DB2 コマンドウィンドウを開きます。
- c. 次のコマンドを入力して custdb2.sql を実行します。

db2cmd db2 -c -ec -td~ +s -v -f custdb2.sql

d. 処理が完了したら、次のコマンドを入力してコマンドウィンドウを閉じます。

exit

- 5. DB2 LUW クライアント上で、DB2 LUW データベースを参照する CustDB という ODBC デー タソースを作成します。
 - a. ODBC データソースアドミニストレータを起動します。

[スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [ODBC データソースア ドミニストレータ] を選択します。

[ODBC データソースアドミニストレータ] が表示されます。

- b. [ユーザ DSN] タブで、[追加] をクリックします。
- c. [データソースの新規作成] ウィンドウで、DB2 LUW データベース用の ODBC ドライバ を選択します。たとえば、IBM DB2 UDB ODBC ドライバを選択します。[完了] をクリッ クします。

ODBC ドライバの設定方法については、次のマニュアルを参照してください。

- DB2 LUW のマニュアル
- http://www.sybase.com/detail?id=1011880

結果

DB2 LUW 統合データベースが設定されます。

参照

●「IBM DB2 LUW 統合データベース」『Mobile Link サーバ管理』

Ultra Lite のリモートデータベースの設定

以下の例では、CustDB のリモートデータベースを作成します。CustDB リモートデータベース は、Ultra Light データベースでなければなりません。

リモートデータベースのアプリケーション論理は %SQLANYSAMP16%¥UltraLite¥CustDB にあります。これには、以下のファイルが含まれています。

- Embedded SQL の論理 ファイル custdb.sqc には、Ultra Light データベースの情報を問い合わせて修正するための SQL 文と、統合データベースとの同期を開始するために必要な呼び出しが格納されています。
- **C++ API の論理** ファイル *custdbcpp.cpp* には、C++ API の論理が格納されています。
- **ユーザインタフェース機能** この機能は、*Samples¥UltraLite¥CustDB* のプラットフォーム固有 のサブフォルダに、別々に格納されています。

Ultra Light が実行されているリモートデバイスにサンプルアプリケーションをインストールするには、次の手順を実行します。

- 1. 統合データベースを起動します。
- 2. Mobile Link サーバを起動します。
- 3. サンプルアプリケーションをクライアントデバイスにインストールして、起動します。

4. サンプルアプリケーションを同期します。

例

次の例では、DB2 統合データベースを対象として動作している Windows デスクトップに CustDB サンプルをインストールする方法を示します。

1. 次のようにして、統合データベースが稼働していることを確認します。

DB2 LUW データベースの DB2 コマンドウィンドウを開きます。次のコマンドを入力しま す。ここでは、*userid* と *password* は DB2 LUW データベースに接続するためのユーザ ID と パスワードです。

db2 connect to CustDB user userid using password

2. Mobile Link サーバを起動します。

DB2 LUW データベースと同期できるようにするため、コマンドプロンプトで次のコマンド を実行します。

mlsrv16 -c "DSN=CustDB" -zp

- 3. CustDB サンプルアプリケーションを起動します。
 - a. [スタート] » [プログラム] » [SQL Anywhere 16] » [Mobile Link] » [同期サーバのサン プル] をクリックします。
 - b. Employee ID に値 50 を入力し、[OK] をクリックします。

アプリケーションは自動的に同期を実行し、一連の顧客、製品、注文情報が CustDB 統 合データベースからアプリケーションにダウンロードされます。

4. リモートアプリケーションを統合データベースと同期します。

[ファイル]»[同期]を選択します。

この手順は、データベースを変更したときのみ行ってください。

CustDB データベース内のテーブル

CustDB データベースのテーブル定義は、%SQLANYSAMP16%¥MobiLink¥CustDB にあるプラット フォーム固有のファイル内に格納されています。

CustDB テーブルの ER (実体関連) 図については、「CustDB サンプルデータベースアプリケー ション」『SQL Anywhere 16 紹介』を参照してください。

次の5つのテーブルは統合データベースとリモートデータベースの両方にありますが、その定義 は両者で少し異なります。

ULCustomer

ULCustomer テーブルには、顧客リストがあります。

リモートデータベースの ULCustomer には、次のカラムがあります。

● cust_id 顧客を識別するユニークな整数を保持するプライマリキーカラム。

● cust_name 顧客の名前を保持する 30 バイトの文字列。

統合データベースの ULCustomer には、次のカラムもあります。

● last_modified ローが最後に変更されたときのタイムスタンプ。このカラムは、タイムスタ ンプベースの同期に使用されます。

ULProduct

ULProduct テーブルには、製品リストがあります。

リモートデータベースと統合データベースの両方の ULProduct に、次のカラムがあります。

- prod_id 製品を識別するユニークな整数を保持するプライマリキーカラム。
- price 単価を識別する整数。
- prod_name 製品の名前を保持する 30 バイトの文字列。

ULOrder

ULOrder テーブルには受注リストがあります。注文した顧客の情報、受注した従業員、注文された製品についての詳細が含まれています。

リモートデータベースの ULOrder には、次のカラムがあります。

- order_id 注文を識別するユニークな整数を保持するプライマリキーカラム。
- cust_id ULCustomer を参照する外部キーカラム。
- prod_id ULProduct を参照する外部キーカラム。
- emp_id ULEmployee を参照する外部キーカラム。
- disc 注文に適用される値引きを保持する整数。
- quant 注文された製品の数を保持する整数。
- **notes** 注文に関する注記を保持する 50 バイトの文字列。
- status 注文のステータスが記述された 20 バイトの文字列。

統合データベースの ULOrder には、次のカラムもあります。

● last_modified ローが最後に変更されたときのタイムスタンプ。このカラムは、タイムスタ ンプベースの同期に使用されます。

ULOrderIDPool

ULOrderIDPool テーブルは、ULOrder のプライマリキープールです。

リモートデータベースの ULOrderIDPool には、次のカラムがあります。

● pool order id 注文 ID を識別するユニークな整数を保持するプライマリキーカラム。

統合データベースの ULOrderIDPool には、次のカラムもあります。

- pool_emp_id 注文 ID が割り当てられたリモートデータベースの所有者の従業員 ID を保 持する整数カラム。
- last_modified ローが最後に変更されたときのタイムスタンプ。

ULCustomerIDPool

ULCustomerIDPool テーブルは、ULCustomer のプライマリキープールです。

リモートデータベースの ULCustomerIDPool には、次のカラムがあります。

● pool_cust_id 顧客 ID を識別するユニークな整数を保持するプライマリキーカラム。

統合データベースの ULCustomerIDPool には、次のカラムもあります。

- pool_emp_id リモートデータベースで生成された新しい従業員に使用される従業員 ID を 保持する整数カラム。
- last_modified ローが最後に変更されたときのタイムスタンプ。

以下のテーブルは、統合データベースにのみ存在します。

ULIdentifyEmployee_nosync

ULIdentifyEmployee_nosync テーブルは、統合データベースとリモートデータベースの両方に存在します。これには、次の1つのカラムがあります。

● emp_id このプライマリキーカラムには、リモートデータベースの従業員 ID を示す整数が 保持されています。

ULEmployee

ULEmployee テーブルは、統合データベースにのみ存在します。これには、営業担当者リストが 格納されています。

ULEmployee には、次のカラムがあります。

- emp_id 従業員を識別するユニークな整数を保持するプライマリキーカラム。
- emp_name 従業員の名前を保持する 30 バイトの文字列。

ULEmpCust

ULEmpCust テーブルは、どの顧客の注文をダウンロードするかを制御します。従業員が新しい 顧客の注文を必要とする場合は、従業員 ID と顧客 ID を挿入すると、その顧客の注文がダウン ロードされます。

- emp_id ULEmployee.emp_id の外部キー。
- **cust_id** ULCustomer.cust_id の外部キー。プライマリキーは、emp_id と cust_id で構成されています。
- action 従業員のレコードをリモートデータベースから削除するかどうかを決定するのに使 用される文字。従業員が顧客の注文を必要としなくなった場合は、D(削除)に設定します。 注文が必要な場合、action は NULL に設定してください。

この場合は、ULOrder テーブルから削除するローを統合データベースが識別できるようにす るため、論理削除を使用する必要があります。削除情報がダウンロードされると、action が Dに設定されたその従業員のすべてのレコードは統合データベースからも削除できます。

● last_modified ローが最後に変更されたときのタイムスタンプ。このカラムは、タイムスタ ンプベースの同期に使用されます。

ULOIdOrder と ULNewOrder

これらのテーブルは、統合データベースにのみ存在します。また、競合を解決するために使用され、ULOrder と同じカラムが含まれています。SQL Anywhere と Microsoft SQL Server では、これらはテンポラリテーブルです。Adaptive Server Enterprise の場合、これらのテーブルは通常のテーブルであり、@@spid です。DB2 LUW と Oracle にはテンポラリテーブルがないため、同期ユーザに属するローを Mobile Link が識別できるようになっている必要があります。これらのテーブルはベーステーブルであるため、5人のユーザが同期している場合は、これらのテーブルのローを各ユーザが同時に保持することがあります。

@@spid の詳細については、「変数」『SQL Anywhere サーバ SQL リファレンス』を参照してください。

CustDB サンプル内のユーザ

CustDB サンプルのユーザには、営業担当者とモバイル管理者の2つのタイプがあります。相違 点は次のとおりです。

● 営業担当者 営業担当者に関連付けられたリモートデータベースは、ユーザ ID 50、51、52 で識別されます。営業担当者は、次のタスクを実行できます。

○顧客と製品のリスト表示

○新規顧客の追加

○注文の追加や削除

○未処理の注文リストのスクロール

○変更内容に関する統合データベースとの同期

● モバイル管理者 モバイル管理者に関連付けられたリモートデータベースは、ユーザ ID 53 で識別されます。モバイル管理者は、営業担当者と同じタスクを実行できます。このほか、 モバイル管理者は次のタスクを実行できます。
○注文の承認や拒否

同期論理のソースコード

Sybase Central を使用すると、統合データベース内の同期スクリプトを調べることができます。

スクリプトタイプとイベント

*custdb.sql*ファイルは、ml_add_connection_script または ml_add_table_script を呼び出して、各同期 スクリプトを統合データベースに追加します。

例

*custdb.sql*の次の行は、ULProduct テーブル用のテーブルレベルのスクリプトを追加します。この スクリプトは、download_cursor イベントで実行されます。スクリプトには、SELECT 文が1つ あります。

```
call ml_server.ml_add_table_script(
'CustDB 16.0',
'ULProduct', 'download_cursor',
'SELECT prod_id, price, prod_name FROM ULProduct' )
go
```

CustDB サンプルの注文の同期

ビジネスルール

ULOrder テーブルのビジネスルールは、次のとおりです。

- ●注文は、承認されていないか、ステータスが NULL である場合にかぎりダウンロードされる。
- ●注文は、統合データベースでもリモートデータベースでも修正できる。
- ●各リモートデータベースには、従業員に対応する注文のみが保持される。

ダウンロード

統合データベースでは、注文を挿入、削除、更新できます。これらの操作に対応するスクリプト は、次のとおりです。

download_cursor download_cursor スクリプトの最初のパラメータは、最終ダウンロードタイムスタンプです。これは、最後の同期以後にリモートデータベースまたは統合データベースのいずれかで修正されたローのみをダウンロードするために使用されます。2番目のパラメータは従業員 ID です。この ID は、ダウンロードするローを判断するために使用されます。

CustDBの download_cursor スクリプトを次に示します。

CALL ULOrderDownload({ml s.last_table_download}, {ml s.username})

CustDB の ULOrderDownload プロシージャを次に示します。

CREATE PROCEDURE ULOrderDownload (IN LastDownload timestamp, IN EmployeeID integer) BEGIN

```
SELECT o.order_id, o.cust_id, o.prod_id, o.emp_id, o.disc, o.quant, o.notes, o.status
FROM ULOrder o, ULEmpCust ec
WHERE o.cust_id = ec.cust_id
AND ec.emp_id = EmployeeID
AND ( o.last_modified >= LastDownload
OR ec.last_modified >= LastDownload)
AND ( o.status IS NULL OR o.status != 'Approved' )
AND ( ec.action IS NULL )
END
```

download_delete_cursor CustDB の download_delete_cursor スクリプトを次に示します。

SELECT o.order_id FROM ULOrder o, dba.ULEmpCust ec WHERE o.cust_id = ec.cust_id AND ((o.status = "Approved" AND o.last_modified >= {ml s.last_table_download}) OR (ec.action = "D")) AND ec.emp_id = {ml s.username}

アップロード

リモートデータベースでは、注文を挿入、削除、更新できます。これらの操作に対応するスクリプトは、次のとおりです。

upload_insert CustDB の upload insert スクリプトを次に示します。

INSERT INTO ULOrder (order_id, cust_id, prod_id, emp_id, disc, quant, notes, status) VALUES({ml r.order_id, r.cust_id, r.prod_id, r.emp_id, r.disc, r.quant, r.notes, r.status })

upload_update CustDB の upload_update スクリプトを次に示します。

UPDATE ULOrder SET cust_id = {ml r.cust_id}, prod_id = {ml r.prod_id}, emp_id = {ml r.emp_id}, disc = {ml r.disc}, quant = {ml r.quant}, notes = {ml r.notes}, status = {ml r.status} WHERE order_id = {ml r.order_id}

upload_delete CustDB の upload delete スクリプトを次に示します。

DELETE FROM ULOrder WHERE order_id = {ml r.order_id}

upload_fetch CustDB の upload fetch スクリプトを次に示します。

SELECT order_id, cust_id, prod_id, emp_id, disc, quant, notes, status FROM ULOrder WHERE order_id = {ml r.order_id}

upload_old_row_insert CustDB の upload_old_row_insert スクリプトを次に示します。

INSERT INTO ULOIdOrder (order_id, cust_id, prod_id, emp_id, disc, quant, notes, status) VALUES({ml r.order_id, r.cust_id, r.prod_id, r.emp_id, r.disc, r.quant, r.notes, r.status })

upload_new_row_insert CustDB の upload new row insert スクリプトを次に示します。

INSERT INTO ULNewOrder (order_id, cust_id, prod_id, emp_id, disc, quant, notes, status) VALUES({ml r.order_id, r.cust_id, r.prod_id, r.emp_id, r.disc, r.quant, r.notes, r.status })

競合解決

resolve_conflict CustDBの resolve_conflict スクリプトを次に示します。

CALL ULResolveOrderConflict

CustDB の ULResolveOrderConflict プロシージャを次に示します。

CREATE PROCEDURE ULResolveOrderConflict() BEGIN -- approval overrides denial IF 'Approved' = (SELECT status FROM ULNewOrder) THEN UPDATE ULOrder o SET o.status = n.status, o.notes = n.notes FROM ULNewOrder n WHERE o.order_id = n.order_id; END IF; DELETE FROM ULOIdOrder; DELETE FROM ULNewOrder; END

CustDB サンプルの顧客の同期

ビジネスルール

顧客を規定するビジネスルールは、次のとおりです。

●顧客情報は、統合データベースでもリモートデータベースでも修正できる。

●リモートデータベースと統合データベースの両方に、完全な顧客リストがある。

ダウンロード

統合データベースでは、顧客情報を挿入または更新できます。これらの操作に対応するスクリプ トは、次のとおりです。

download_cursor 次の download_cursor スクリプトは、ユーザが最後に情報をダウンロードした後で情報が変更された顧客をすべてダウンロードします。

SELECT cust_id, cust_name FROM ULCustomer WHERE last_modified >= {ml s.last_table_download}

アップロード

リモートデータベース側で顧客情報を挿入、更新、または削除できます。これらの操作に対応するスクリプトは、次のとおりです。

upload_insert CustDB の upload_insert スクリプトを次に示します。

INSERT INTO ULCustomer(cust_id, cust_name) VALUES({ml r.cust_id, r.cust_name })

upload_update CustDB の upload_update スクリプトを次に示します。

UPDATE ULCustomer SET cust_name = {ml r.cust_name} WHERE cust_id = {ml r.cust_id}

このテーブルに対する競合検出は実行されません。

upload_delete CustDB の upload_delete スクリプトを次に示します。

DELETE FROM ULCustomer WHERE cust_id = {ml r.cust_id}

CustDB サンプルの製品の同期

ビジネスルール

ULProduct に関するすべてのローがダウンロードされます。これはスナップショット同期と呼ばれます。

「スナップショットを使った同期」『Mobile Link サーバ管理』を参照してください。

ULProduct テーブルのビジネスルールは、次のとおりです。

●統合データベースでは、製品の修正のみが可能。

●各リモートデータベースには、すべての製品が含まれている。

ダウンロード

統合データベースでは、製品情報を挿入、削除、更新できます。これらの操作に対応するスクリ プトは、次のとおりです。

download_cursor 次の download_cursor スクリプトは、同期が行われるたびに ULProduct テーブルのすべてのローとカラムをダウンロードします。

SELECT prod_id, price, prod_name FROM ULProduct

顧客と注文のプライマリキープールの管理

CustDB サンプルデータベースでは、ULCustomer テーブルと ULOrder テーブルのユニークなプ ライマリキーを管理するために、プライマリキープールが使用されます。 プライマリキープール は、ULCustomerIDPool テーブルと ULOrderIDPool テーブルです。

ULCustomerIDPool

以下のスクリプトは、ULCustomerIDPoolテーブルで定義されています。

ダウンロード

begin_download CustDB の begin_download スクリプトを次に示します。

CALL ULCustomerIDPool_maintain({ml s.username})

CustDB の ULCustomerIDPool maintain プロシージャを次に示します。

CREATE PROCEDURE ULCustomerIDPool_maintain (IN syncuser_id INTEGER) BEGIN DECLARE pool_count INTEGER; -- Determine how many ids to add to the pool

```
SELECT COUNT(*) INTO pool_count
FROM ULCustomerIDPool
WHERE pool_emp_id = syncuser_id;
-- Top up the pool with new ids
WHILE pool_count < 20 LOOP
INSERT INTO ULCustomerIDPool ( pool_emp_id )
VALUES ( syncuser_id );
SET pool_count = pool_count + 1;
END LOOP;
END
```

download_cursor

```
SELECT pool_cust_id FROM ULCustomerIDPool
WHERE last_modified >= {ml s.last_table_download}
AND pool_emp_id = {ml s.username}
```

アップロード

upload_delete CustDB の upload_delete スクリプトを次に示します。

```
DELETE FROM ULCustomerIDPool
WHERE pool_cust_id = {ml r.pool_cust_id}
```

ULOrderIDPool

以下のスクリプトは、ULOrderIDPool テーブルで定義されています。

ダウンロード

begin_download CustDB の begin_download スクリプトを次に示します。

CALL ULOrderIDPool_maintain({ml s.username})

CustDB の ULOrderIDPool maintain プロシージャを次に示します。

```
ALTER PROCEDURE ULOrderIDPool_maintain ( IN syncuser_id INTEGER )
BEGIN
DECLARE pool_count INTEGER;
-- Determine how many ids to add to the pool
SELECT COUNT(*) INTO pool_count
FROM ULOrderIDPool
WHERE pool_emp_id = syncuser_id;
-- Top up the pool with new ids
WHILE pool_count < 20 LOOP
INSERT INTO ULOrderIDPool ( pool_emp_id )
VALUES ( syncuser_id );
SET pool_count = pool_count + 1;
END LOOP;
END
```

download_cursor CustDBの download_cursor スクリプトを次に示します。

SELECT pool_order_id FROM ULOrderIDPool WHERE last_modified >= {ml s.last_table_download} AND pool_emp_id = {ml s.username}

アップロード

upload_delete CustDBの upload_delete スクリプトを次に示します。

DELETE FROM ULOrderIDPool WHERE pool_order_id = {ml r.pool_order_id}

CustDB データベースのリストア

サンプルをリストアするには、%SQLANYSAMP16%¥UltraLite¥CustDB ディレクトリから次のコマンドを実行します。

makedbs

Mobile Link Contact サンプル

Contact サンプルは、Mobile Link 開発者にとって貴重なリソースです。このサンプルを使用して、Mobile Link アプリケーションの開発時に必要なさまざまな方法の実装例を紹介します。

Contact サンプルアプリケーションは、1 つの SQL Anywhere 統合データベースと、2 つの SQL Anywhere リモートデータベースから構成されています。このサンプルでは、同期の一般的な方法をいくつか説明します。この項で説明することを最大限に活用するために、サンプルアプリケーションのソースコードを読んで理解してください。

統合データベースは SQL Anywhere データベースですが、同期スクリプトは他のデータベース管理システムの最小限の変更を処理する SQL 文で構成されています。

Contact サンプルは %SQLANYSAMP16%¥MobiLink¥Contact にあります。

同期の設計

Contact サンプルアプリケーションの同期の設計では、以下の機能を使用します。

- **カラムのサブセット** 統合データベース上の Customer、Product、SalesRep、Contact の各テー ブルのカラムのサブセットが、リモートデータベースで共有される。
- **ローのサブセット** 統合データベース上の SalesRep テーブルのローのうち1つのみについ て、すべてのカラムが各リモートデータベースで共有される。「リモートデータベース間で ローを分割する」『Mobile Link サーバ管理』を参照してください。
- **タイムスタンプベースの同期** 最後のデバイス同期以降に実行された統合データベースに対 する変更を識別する方法。Customer、Contact、Product の各テーブルは、タイムスタンプベー スの方法を使用して同期される。「タイムスタンプベースのダウンロードの実装」『Mobile Link サーバ管理』を参照してください。

権限

次のロールおよび権限が必要です。

●SYS REPLICATION ADMIN ROLE システムロール

●SYS RUN REPLICATION ROLE システムロール

●CREATE ANY TRIGGER システム権限

Contact サンプルの設定

Contact サンプルデータベースを構築できるように、Windows バッチファイル *build.bat* が用意されています。UNIX システムでは *build.sh* です。

前提条件

この作業を実行するための前提条件はありません。

内容と備考

このバッチファイルの内容を調べることができます。このバッチファイルによって次のアクションが実行されます。

- ●統合データベースと2つのリモートデータベース用に、ODBC データソース定義が作成されま す。
- ●*consol.db* という統合データベースが作成され、Mobile Link システムテーブル、データベース スキーマ、一部のデータ、同期スクリプト、Mobile Link ユーザ名がデータベースにロードさ れます。
- ●remote.db という名前の2つのリモートデータベースが、サブフォルダ remote_1 と remote_2 に 作成されます。両方のデータベースに共通の情報がロードされ、カスタマイズ内容が適用され ます。カスタマイズ内容には、グローバルデータベース識別子、Mobile Link ユーザ名、2つの パブリケーションに対するサブスクリプションが含まれます。

◆ タスク

- 1. コマンドプロンプトで、%SQLANYSAMP16%¥MobiLink¥Contact に移動します。
- 2. build.bat (Windows) または build.sh (UNIX) を実行します。

結果

Contact サンプルデータベースが構築されます。

次の手順

「Contact サンプルの実行」73ページに進みます。

Contact サンプルの実行

Contact サンプルには最初の同期を実行するバッチファイルが含まれており、Mobile Link サーバ と dbmlsync のコマンドラインが例示されています。

前提条件

Contact サンプルデータベースを構築するための build.bat が実行されていることを確認します。

内容と備考

%SQLANYSAMP16%¥MobiLink¥Contact に次のバッチファイルがあり、その内容はテキストエディタで確認できます。

• step1.bat

- *step2.bat*
- step3.bat

◆ タスク

- 1. Mobile Link サーバを起動します。
 - a. コマンドプロンプトで、%SQLANYSAMP16%¥MobiLink¥Contact に移動します。
 - b. 次のコマンドを実行します。

step1

このコマンドは、Mobile Link サーバを冗長モードで起動するバッチファイルを実行しま す。このモードは、開発中やトラブルシューティング中には便利ですが、パフォーマン スが低下するため、通常の運用環境では使用しません。

- 2. 両方のリモートデータベースを同期します。
 - a. コマンドプロンプトで、%SQLANYSAMP16%¥MobiLink¥Contact に移動します。
 - b. 次のコマンドを実行します。

step2

これは、両方のリモートデータベースを同期するバッチファイルです。

- 3. Mobile Link サーバを停止します。
 - a. コマンドプロンプトで、%SQLANYSAMP16%¥MobiLink¥Contact に移動します。
 - b. 次のコマンドを実行します。

step3

これは、Mobile Link サーバを停止させるバッチファイルです。

結果

Contact サンプルが実行され、リモートデータベースと統合データベースが同期されます。

次の手順

Contact サンプルで同期の動作方法を調べるには、Interactive SQL を使用してリモートデータベースと統合データベースでデータを修正し、バッチファイルを使用して同期を行います。

Contact データベース内のテーブル

Contact データベースのテーブル定義は、次のファイルにあります。これらのファイルはすべて サンプルのフォルダにあります。

●*MobiLink*¥*Contact*¥*build_consol.sql*

MobiLink¥Contact¥build_remote.sql

次の3つのテーブルは統合データベースとリモートデータベースの両方にありますが、その定義 は両者で少し異なります。

SalesRep

SalesRep テーブルには、営業担当者ごとに1つのローがあります。リモートデータベースは、それぞれ1人の営業担当者が所持します。

各リモートデータベースの SalesRep には、次のカラムがあります。

● rep_id 営業担当者の識別番号が格納されるプライマリキーカラム。

● name 担当者の名前。

統合データベース側には、この他に営業担当者の Mobile Link ユーザ名を保持する ml_username カラムもあります。

Customer

このテーブルには、顧客ごとに1つのローがあります。顧客は、それぞれ1人の営業担当者が担当する会社です。SalesRep テーブルと Customer テーブルは1対多の関係になっています。

各リモートデータベースの Customer には、次のカラムがあります。

- cust_id 顧客の識別番号を保持するプライマリキーカラム。
- name 顧客名。これは会社名です。
- rep_id SalesRep テーブルを参照する外部キーカラム。顧客に割り当てられた営業担当者を 識別します。

統合データベースには、この他に last modified カラムと active カラムがあります。

- last_modified ローを最後に変更した時刻。このカラムは、タイムスタンプベースの同期に 使用されます。
- active 顧客が現在アクティブであるか(1)、またはこの顧客との取引がなくなったか(0)を 示すビットカラム。このカラムに非アクティブ(0)のマークが付いている場合は、この顧客 に対応するすべてのローがリモートデータベースから削除されます。

Contact

このテーブルには、窓口ごとに1つのローがあります。窓口担当者は、顧客の会社の従業員です。Customer テーブルと Contact テーブルは1対多の関係になっています。

各リモートデータベースの Contact には、次のカラムがあります。

- contact_id 窓口担当者の識別番号を保持するプライマリキーカラム。
- name 各窓口担当者の氏名。
- cust_id 窓口担当者が所属する顧客の識別子。

統合データベースでは、このテーブルに次のカラムもあります。

- last_modified ローを最後に変更した時刻。このカラムは、タイムスタンプベースの同期に 使用されます。
- active 窓口が現在アクティブであるか(1)、またはこの窓口との取引がなくなったか(0)を 示すビットカラム。このカラムに非アクティブ(0)のマークが付いている場合は、この窓口 に対応するローがリモートデータベースから削除されます。

Product

Product テーブルには、会社で販売される製品ごとに1つのローがあります。Product テーブルは 別のパブリケーションに保持されるため、リモートデータベースはこのテーブルを別途同期でき ます。

各リモートデータベースの Product には、次のカラムがあります。

- id 製品を識別するユニークな数値を保持するプライマリキーカラム。
- name 品目の名前。
- size 品目のサイズ。
- quantity 品目の在庫数量。営業担当者が注文を受け取った時点で、このカラムは更新され ます。

● unit_price 製品の単価。

統合データベースの Product テーブルには、次のカラムもあります。

- supplier 製品を製造している会社。
- last_modified ローを最後に変更した時刻。このカラムは、タイムスタンプベースの同期に 使用されます。
- active 製品が現在アクティブ(1)であるかどうかを示すビットカラム。このカラムに非ア クティブ(0)のマークが付いている場合は、この製品に対応するローがリモートデータベー スから削除されます。

統合データベースには、これらのテーブルに加えてテーブルセットが作成されます。これには、 競合解決中に使用されるテンポラリテーブル product_conflict と、ユーザ mlmaint が所有する Mobile Link アクティビティをモニタリングするためのテーブルセットが含まれます。Mobile Link モニタリングテーブルを作成するためのスクリプトは、%SQLANYSAMP16%¥MobiLink ¥Contact¥mlmaint.sql ファイルにあります。

Contact サンプル内のユーザ

Contact サンプルには、複数のデータベースユーザ ID と Mobile Link ユーザ名が含まれています。

データベースユーザ ID

2 つのリモートデータベースは、営業担当者 Samuel Singer (rep_id 856) と Pamela Savarino (rep_id 949) に割り当てられます。

この2人のユーザはどちらも、それぞれのリモートデータベースへの接続時に、デフォルトの SQL Anywhere ユーザ ID dba とパスワード SQL を使用します。

また、各リモートデータベースには、ユーザ ID sync_user (パスワードは sync_user) もあります。 このユーザ ID は、dbmlsync コマンドラインでのみ使用します。sync_user は SYS_RUN_REPLICATION_ROLE システムロールを持っている必要があります。すなわち、 dbmlsync からの接続時にはあらゆる操作を実行できますが、他のアプリケーションからの接続 時には何の権限もありません。そのため、sync_user の ID およびパスワードを使用しても問題に はなりません。

統合データベースには、mlmaintというユーザが存在します。このユーザは Mobile Link 同期統計とエラーのモニタリングに使用されるテーブルの所有者です。mlmaint ユーザは接続権限を持っていません。テーブルを個々のユーザ ID に割り当てるには、スキーマ内でオブジェクトを他のオブジェクトから分離するだけであり、Sybase Central や他のユーティリティで管理しやすくなっています。

Mobile Link ユーザ名

Mobile Link ユーザ名は、データベースユーザ ID とは異なります。各リモートデバイスには、 データベースへの接続時に使用するユーザ ID の他に、Mobile Link ユーザ名があります。Samuel Singer の Mobile Link ユーザ名は SSinger です。Pamela Savarino の Mobile Link ユーザ名は PSavarino です。Mobile Link ユーザ名は、次のロケーションで格納または使用されています。

- ●リモートデータベース。Mobile Link ユーザ名が、CREATE SYNCHRONIZATION USER 文を使用して追加されています。
- ●統合データベース。Mobile Link ユーザ名とパスワードが、mluser ユーティリティを使用して 追加されています。
- ●*MobiLink¥Contact¥step2.bat*内の dbmlsync コマンドライン。同期時に、接続ユーザの Mobile Link パスワードが指定されます。
- ●Mobile Link サーバ。同期時、Mobile Link ユーザ名がパラメータとして多数のスクリプトに指定されます。
- ●統合データベース側の SalesRep テーブル。ml_username カラムがあります。同期スクリプトは、このカラムの値と Mobile Link ユーザ名パラメータを比較します。

Contact サンプルの営業担当者の同期

SalesRep テーブルの同期スクリプトは、「スナップショットを使った同期」の例を示しています。 営業担当者の情報は、変更されたかどうかに関係なくダウンロードされます。 「スナップショットを使った同期」『Mobile Link サーバ管理』を参照してください。

ビジネスルール

SalesRep テーブルのビジネスルールは、次のとおりです。

- ●リモートデータベース側ではテーブルを修正しない。
- ●営業担当者の Mobile Link ユーザ名と rep id 値を変更しない。
- ●各リモートデータベースの SalesRep テーブルには、リモートデータベース所有者の Mobile Link ユーザ名に対応するローが 1 つだけ存在する。

ダウンロード

● download_cursor 各リモートデータベースの SalesRep テーブルには、ローが1つだけ存在 します。単一のローのダウンロードに伴うオーバヘッドはほとんどないため、単純なスナッ プショットの download cursor スクリプトを使用します。

SELECT rep_id, name FROM SalesRep WHERE ? IS NOT NULL AND ml_username = ?

このスクリプトの最初のパラメータは、最終ダウンロードタイムスタンプですが、これは使用されません。IS NOT NULL は、パラメータを使用するために指定されたダミー式です。2 番目のパラメータは Mobile Link ユーザ名です。

アップロード

このテーブルはリモートテーブル側では更新しないため、アップロードスクリプトはありません。

Contact サンプルの顧客の同期

Customer テーブルの同期スクリプトは、「タイムスタンプベースの同期」とローの分割の例を示しています。これらの方法では、同期中に転送されるデータの量が最小限になり、テーブルデータの整合性が保持されます。

次の項を参照してください。

- ●「タイムスタンプベースのダウンロードの実装」『Mobile Link サーバ管理』
- ●「リモートデータベース間でローを分割する」『Mobile Link サーバ管理』

ビジネスルール

顧客を規定するビジネスルールは、次のとおりです。

- ●顧客情報は、統合データベースでもリモートデータベースでも修正できる。
- ●営業担当者間で、顧客の再割り当てを定期的に変更できる。このプロセスは、一般に領域の再 編成と呼ばれる。

●各リモートデータベースには、割り当てられている顧客のみが保持される。

ダウンロード

● download_cursor 次の download_cursor スクリプトは、最後の正常なダウンロード以後に情報が変更されたアクティブな顧客のみをダウンロードします。また、営業担当者別に顧客をフィルタリングします。

SELECT cust_id, Customer.name, Customer.rep_id FROM Customer key join SalesRep WHERE Customer.last_modified >= ? AND SalesRep.ml_username = ? AND Customer.active = 1

download_delete_cursor 次の download_delete_cursor スクリプトは、最後の正常なダウン ロード以後に情報が変更された顧客のみをダウンロードします。また、非アクティブのマー クが付いているか、指定された営業担当者に割り当てられていない顧客を、すべて削除しま す。

SELECT cust_id FROM Customer key join SalesRep WHERE Customer.last_modified >= ? AND (SalesRep.ml_username != ? OR Customer.active = 0)

統合データベースにある Customer テーブルからローが削除されると、この結果セットには表 示されないため、リモートデータベースからは削除されません。代わりに、顧客には非アク ティブのマークが付きます。

領域が再編成されると、このスクリプトは営業担当者への割り当てから外れた顧客を削除し ます。また、他の営業担当者に移された顧客も削除します。このような追加の削除にはフラ グとして SQLCODE 100 が設定されますが、同期の妨げにはなりません。より複雑なスクリ プトを作成すれば、現在の営業担当者から外された顧客のみを識別できます。

Mobile Link クライアントはリモートデータベースでカスケード削除を実行するため、このスクリプトによって、他の営業担当者に割り当てられた顧客のすべての窓口が削除されます。

アップロード

リモートデータベース側で顧客情報を挿入、更新、または削除できます。これらの操作に対応するスクリプトは、次のとおりです。

● upload_insert 次の upload_insert スクリプトは、Customer テーブルにローを1つ追加して、 顧客にアクティブのマークを付けます。

INSERT INTO Customer(cust_id, name, rep_id, active) VALUES (?,?,?,1)

● upload_update 次の upload_update スクリプトは、統合データベースにある顧客情報を修正 します。このテーブルでは競合検出は実行されません。

UPDATE Customer SET name = ?, rep_id = ? WHERE cust_id = ? ● upload_delete 次の upload_delete スクリプトは、統合データベースで顧客に非アクティブのマークを付けます。ローは削除されません。

UPDATE Customer SET active = 0 WHERE cust id = ?

Contact サンプルの顧客窓口の同期

Contact テーブルには、顧客の会社の社員名、顧客を参照するための外部キー、窓口を識別する ユニークな整数が含まれています。また、last_modified タイムスタンプと、窓口がアクティブで あるかどうかを示すマーカもあります。

ビジネスルール

このテーブルのビジネスルールは、次のとおりです。

●窓口情報は、統合データベースでもリモートデータベースでも修正できる。

●各リモートデータベースには、営業担当者が割り当てられている顧客の窓口のみが含まれる。

●営業担当者間で顧客を再割り当てした場合は、窓口の再割り当てもする。

トリガ

Customer テーブルのトリガは、顧客情報に変更があったときに窓口が確実に選択されるように するために使用されます。トリガは、各窓口の last_modified カラムを、その窓口に対応する顧客 の情報が変更されるたびに明示的に変更します。

CREATE TRIGGER UpdateCustomerForContact AFTER UPDATE OF rep_id ORDER 1 ON DBA.Customer REFERENCING OLD AS old_cust NEW as new_cust FOR EACH ROW BEGIN UPDATE Contact SET Contact.last_modified = new_cust.last_modified FROM Contact WHERE Contact.cust_id = new_cust.cust_id END

顧客が修正されるたびにすべての窓口レコードを更新することで、トリガは顧客とその関連窓口 を結合します。そのため、顧客が修正されるとすべての関連窓口も修正され、次回の同期時に顧 客とその関連窓口が一括してダウンロードされます。

ダウンロード

● download_cursor Contact の download cursor スクリプトを次に示します。

SELECT contact_id, contact.name, contact.cust_id FROM (contact JOIN customer) JOIN salesrep ON contact.cust_id = customer.cust_id AND customer.rep_id = salesrep.rep_id WHERE Contact.last_modified >= ? AND salesrep.ml_username = ? AND Contact.active = 1 このスクリプトは、アクティブな窓口、営業担当者が最後にダウンロードした後に(明示的 に、または対応する顧客の修正によって)変更された窓口、営業担当者に割り当てられている 窓口をすべて取り出します。この営業担当者に関連付けられている窓口を識別するには、 Customer テーブルと SalesRep テーブルのジョインが必要です。

download_delete_cursor Contact の download delete cursor スクリプトを次に示します。

SELECT contact_id FROM (Contact JOIN Customer) JOIN SalesRep ON Contact.cust_id = Customer.cust_id AND Customer.rep_id = SalesRep.rep_id WHERE Contact.last_modified >= ? AND Contact.active = 0

リモートデータベースから顧客が削除されると、Mobile Link クライアントでは、カスケード 参照整合性が自動的に使用され、対応する窓口が削除されます。このため、 download_delete_cursor スクリプトは、非アクティブのマークが付いている窓口のみを削除し ます。

アップロード

リモートデータベース側で窓口情報を挿入、更新、または削除できます。これらの操作に対応す るスクリプトは、次のとおりです。

● upload_insert 次の upload_insert スクリプトは、Contact テーブルにローを1つ追加して、 窓口にアクティブのマークを付けます。

INSERT INTO Contact (contact_id, name, cust_id, active) VALUES (?,?,?,1)

● upload_update 次の upload_update スクリプトは、統合データベースにある窓口情報を修正 します。

UPDATE Contact SET name = ?, cust_id = ? WHERE contact_id = ?

このテーブルでは競合検出は実行されません。

● upload_delete 次の upload_delete スクリプトは、統合データベースで窓口に非アクティブのマークを付けます。ローは削除されません。

UPDATE Contact SET active = 0 WHERE contact id = ?

Contact サンプルの製品の同期

Product テーブル用のスクリプトは、競合の検出と解決の例を示しています。

Product テーブルは他のテーブルとは別のパブリケーションに保管されているため、別個にダウンロードできます。たとえば、価格変更と営業担当者が低速リンク経由で同期している場合は、それぞれ顧客と窓口の変更をアップロードしなくても、製品変更をダウンロードできます。

ビジネスルール

リモートデータベース側で可能な変更は、 注文を受けた時点で quantity カラムの値を変更するこ とだけです。

ダウンロード

● download_cursor 次の download cursor スクリプトは、最後のリモートデータベースの同期 以後に変更されたすべてのローをダウンロードします。

SELECT id, name, size, quantity, unit price FROM product WHERE last modified >= ? AND active = 1

● download delete cursor 次の download delete cursor スクリプトは、会社が販売を中止し た製品をすべて削除します。これらの製品には、統合データベース内で非アクティブのマー クが付きます。

SELECT id, name, size, quantity, unit price FROM product WHERE last modified >= ? AND active = 0

アップロード

リモートデータベースからは UPDATE 操作のみがアップロードされます。これらのアップロー ドスクリプトの主な機能は、競合の検出と解決のためのプロシージャです。

2人の営業担当者が注文を受けて同期を行うと、それぞれの注文数が Product テーブルの quantity カラムから減算されます。たとえば、Samuel Singer が野球帽 (製品 ID 400) 20 個の注文を受ける と、数量は 90 から 70 に変化します。Pamela Savarino が Samuel Singer の変更を受け取る前に野 球帽 10 個の注文を受けると、自分のデータベース内のカラムの値が 90 から 80 に変化します。

Samuel Singer が自分の変更を同期すると、統合データベース内の quantity カラムは 90 から 70 に 変更されます。Pamela Savarino が自分の変更を同期した場合の正しい値は 60 です。この設定 は、競合を検出することで行われます。

競合検出スキーマには、次のスクリプトが含まれています。



● upload_update 次の upload update スクリプトは、統合データベース側で単純な UPDATE を実行します。

UPDATE product SET name = ?, size = ?, quantity = ?, unit_price = ? WHERE product.id = ?

● upload fetch 次の upload fetch スクリプトは、Product テーブルから単一のローをフェッチ して、アップロードされるローの古い値と比較します。2つのローが異なる場合は、競合が 検出されます。

SELECT id, name, size, quantity, unit price FROM Product WHERE id = ?

● upload_old_row_insert 競合が検出されると、古い値が product_conflict テーブルに挿入さ れます。これは resolve_conflict スクリプトによって使用されます。row_type カラムに、Old を表す値 O を持つローが追加されます。

INSERT INTO DBA.product_conflict(id, name, size, quantity, unit_price, row_type) VALUES(?, ?, ?, ?, ?, 'O')')

upload_new_row_insert 次のスクリプトは、アップロードされるローの新しい値を product_conflict テーブルに追加します。これは、resolve_conflict スクリプトによって使用され ます。

```
INSERT INTO DBA.product_conflict(
id, name, size, quantity, unit_price, row_type )
VALUES(?,?,?,?,?, 'N')
```

競合解決

● resolve_conflict 次のスクリプトは、統合データベース内の数量値に新しい値と古い値の差 を加算して、競合を解決します。

UPDATE Product SET p.quantity = p.quantity - old_row.quantity + new_row.quantity FROM Product p, DBA.product_conflict old_row, DBA.product_conflict new_row WHERE p.id = old_row.id AND p.id = new_row.id AND p.id = new_row.id AND old_row.row_type = 'O' AND new_row.row_type = 'N'

Contact サンプルの統計とエラーのモニタリング

Contact サンプルには、単純なエラーレポートスクリプトとモニタリングスクリプトがいくつか 用意されています。これらのスクリプトを作成するための SQL 文は、*MobiLink¥Contact ¥mlmaint.sql* ファイルにあります。

各スクリプトは、値を保持するように作成されたテーブルにローを挿入します。便宜上、各テーブルの所有者は個別ユーザ mlmaint となっています。

Mobile Link チュートリアル

この項には、Mobile Link テクノロジの設定方法と使用方法を示すチュートリアルがあります。 チュートリアルは、新しいユーザのための入門用チュートリアルから、高度な機能の使用方法の 説明にまで及びます。

追加の Mobile Link チュートリアルは、オンラインで提供されています。http://www.sybase.com/ detail?id=1081144 を参照してください。

注意

オンラインチュートリアルは、バージョン 12.0.0 の SQL Anywhere に基づいています。図やプロ シージャのいくつかは SQL Anywhere 16.0 とは異なります。

チュートリアル:Mobile Link の概要

このチュートリアルでは、同期スクリプトの作成、Mobile Link ログの解釈、Mobile Link プロファ イラを使用した統合データベースと2つのリモートデータベース間での同期のモニタリングに 関する基本手順について説明します。Sybase Central を使用した、データベースと同期の設定手 順を説明します。

必要なソフトウェア

•SQL Anywhere 16

権限

統合データベースで次のロールおよび権限を持つ必要があります。

●SYS_AUTH_DBA_ROLE 互換ロール

リモートデータベースで次のロールおよび権限を持つ必要があります。

●SYS AUTH DBA ROLE システムロール

●SYS_RUN_REPLICATION_ROLE システムロール

前提知識と経験

次の知識と経験が必要です。

●Mobile Link イベントスクリプトの基本的な知識

概要

このチュートリアルでは、次の作業の方法について説明します。

●統合データベースのスキーマのリモートデータベースへの移行

●Sybase Central を使用した、同期に必要な基本スクリプトの作成と統合データベースへの保存

- ●Mobile Link サーバの起動
- ●ログファイルと Mobile Link プロファイラを使用した同期のモニタ

参照

- ●「Interactive SQL」『SQL Anywhere サーバ データベース管理』
- ●「Sybase Central」『SQL Anywhere サーバ データベース管理』
- ●「Mobile Link 同期」1 ページ
- ●「同期スクリプトの作成」『Mobile Link サーバ管理』
- ●「同期の方法」『Mobile Link サーバ管理』
- ●「同期イベント」『Mobile Link サーバ管理』
- ●「Mobile Link プロファイラ」『Mobile Link サーバ管理』
- ●「競合の解決」『Mobile Link サーバ管理』
- http://www.sybase.com/detail?id=1058600#319(このページを表示するには、Sybase.com アカウ ントが必要です。)
- http://sqlanywhere-forum.sybase.com

レッスン1: Mobile Link 統合データベースの設定

このレッスンでは、SQL Anywhere 統合データベースを作成し、ODBC データソースを定義して、 SQL Anywhere 統合データベースを設定します。

前提条件

この作業を実行するための前提条件はありません。

◆ タスク

- 1. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [Sybase Central] をクリック します。
- 2. [ツール] » [SQL Anywhere 16] » [データベースの作成] をクリックします。
- 3. [次へ] をクリックします。
- [このコンピュータにデータベースを作成] をデフォルトのままにし、[次へ] をクリックします。
- 5. [メインデータベースファイルを保存] フィールドに、*c:*¥MLintro¥MLconsolidated.db と入力し ます。[次へ] をクリックします。このディレクトリが存在しない場合は、作成するよう求め られます。[はい] をクリックします。
- 6. データベース作成ウィザードの残りの指示に従い、デフォルト値をそのまま使用します。 DBA ユーザのユーザ ID とパスワードを指定するように求められたら、それぞれ DBA と sql を入力します。

[データベースへの接続] ページで、[最終切断後にデータベースを停止] オプションをオンに します。 7. [完了] をクリックします。

MLconsolidated データベースが作成されます。

- 8. ウィンドウが自動的に閉じない場合は、[データベースの作成] ウィンドウの [閉じる] をク リックします。
- 9. [接続]»[切断] をクリックして、パーソナルデータベースサーバを停止します。
- 10. [ツール] » [SQL Anywhere 16] » [ODBC アドミニストレータを開く] をクリックします。
- 11. [ユーザ DSN] タブをクリックし、[追加] をクリックします。
- [データソースの新規作成] ウィンドウで、[SQL Anywhere 16] をクリックし、[完了] をクリックします。
- 13. [SQL Anywhere の ODBC 設定] ウィンドウで、次の操作を行います。
 - a. [ODBC] タブをクリックします。
 - b. [データソース名] フィールドに mlintro consdb と入力します。
 - c. [ログイン] タブをクリックします。
 - d. [認証] ドロップダウンリストで、デフォルトの [データベース] のままにして、ユーザ ID とパスワードを使用して接続します。
 - e. [ユーザ ID] フィールドに DBA と入力します。
 - f. [パスワード] フィールドに sql と入力します。
 - g. **[アクション]** ドロップダウンリストで、**[このコンピュータのデータベースを起動して接** 続] をデフォルトのままにします。
 - h. [データベースファイル] フィールドに c:¥MLintro¥MLconsolidated.db と入力します。
 - i. [サーバ名] フィールドに、ML consolidated と入力します。
 - j. [開始行] フィールドに、dbsrv16 と入力します。
 - k. [ODBC] タブをクリックし、[接続テスト] ボタンをクリックして、接続が成功したことを 確認します。[OK] をクリックします。
 - [OK] をクリックし、[OK] をもう一度クリックして、[ODBC データソースアドミニスト レータ] ウィンドウを閉じます。
- 14. [接続] » [SQL Anywhere 16 に接続] をクリックします。
- 15. [アクション] ドロップダウンリストで、[ODBC データソースを使用した接続] を選択します。
- 16. [ODBC データソース名] フィールドに mlintro consdb と入力します。
- 17. **[接続]** をクリックします。

結果

SQL Anywhere 統合データベースと ODBC データソースが作成されます。

次の手順

「レッスン2: Mobile Link 統合データベースでのテーブルの作成と移植」88ページに進みます。

参照

- ●「Mobile Link 統合データベース」『Mobile Link サーバ管理』
- ●「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』

レッスン 2: Mobile Link 統合データベースでのテーブルの作成 と移植

このレッスンでは、Mobile Link 統合データベースで **Product** テーブルを作成し、サンプルデー タを挿入します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Linkの概要」 85ページ

◆ タスク

1. Interactive SQL で統合データベースに接続します。コマンドプロンプトで次のコマンドを実行します。

Interactive SQL は、Sybase Central またはコマンドプロンプトから起動できます。

●Sybase Central から、[MLconsolidated - DBA] データベースを右クリックし、[Interactive SQL を開く] をクリックします。

- dbisql -c "DSN=mlintro_consdb"
- 2. Interactive SQL で次の SQL 文を実行し、Product テーブルを作成します。

```
CREATE TABLE Product (
name VARCHAR(128) NOT NULL PRIMARY KEY,
quantity INTEGER
):
```

Product テーブルには次のカラムがあります。

カラム	説明
name	製品の名前です。
quantity	品目の販売数です。

テーブルを作成したら、サンプルデータ付きで Product テーブルを移植します。

3. Interactive SQL で次の SQL 文を実行して、Product テーブルにサンプルデータを移植します。

 INSERT INTO Product(name, quantity) VALUES ('Screwmaster Drill', 10);
 INSERT INTO Product(name, quantity) VALUES ('Drywall Screws 10lb', 30);
 INSERT INTO Product(name, quantity) VALUES ('Putty Knife x25', 12);

COMMIT;

4. Product テーブルが、前の手順で挿入したデータを含んでいるかどうかを確認します。

次の SQL 文を実行して、内容を確認します。

SELECT * FROM Product

Product テーブルの内容は、Interactive SQL に表示されます。

5. Interactive SQL を閉じます。SQL 文を保存する必要はありません。

結果

統合データベースに Products テーブルが作成されます。

次の手順

「レッスン3: Mobile Link プロジェクトと同期モデルの作成」89ページに進みます。

参照

- ●「Interactive SQL」『SQL Anywhere サーバ データベース管理』
- ●「CREATE TABLE 文」『SQL Anywhere サーバ SQL リファレンス』

レッスン3: Mobile Link プロジェクトと同期モデルの作成

このレッスンでは、プロジェクト作成ウィザードを使用して新しい Mobile Link プロジェクトを 作成します。また、プロジェクト作成ウィザードでは、後で編集可能なデフォルトを使用して同 期モデルも作成されます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページ を参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。 「チュートリアル: Mobile Linkの概要」 85ページ

◆タスク

- Sybase Central で、[ツール]» [Mobile Link 16]» [新しいプロジェクト] をクリックします。
 [プロジェクト作成ウィザード] が表示されます。
- 2. [新しいプロジェクトの名前を指定してください。] フィールドに mlintro_project と入力します。
- 3. [ロケーション] フィールドに、C:¥MLintro と入力します。[次へ] をクリックします。
- 4. [データベースの表示名] フィールドに mlintro_consdb と入力します。
- 5. [接続文字列] フィールドに dsn=mlintro consdb と入力し、[次へ] をクリックします。
- [リモートデータベースに含める統合データベースのテーブルとカラムを指定してください。] リストから [Product] テーブルが選択されていることを確認し、[次へ] をクリックします。
- 7. [リモートスキーマ名をプロジェクトに追加] オプションを選択します。
- 8. [新しいリモートスキーマ名を指定してください。] フィールドに sync_mlintro と入力し、[次 へ] をクリックします。
- 9. [使用するリモートデータベースのタイプを指定してください。] オプションの [SQL Anywhere] を選択し、[完了] をクリックします。
- 10. [はい] をクリックして Mobile Link システムテーブルをインストールしてから、[OK] をク リックします。

リモートスキーマと同じ名前を持つ同期モデルが作成されます。

- 11. [マッピング] タブから [Product] テーブルローを選択します。
- 12. [詳細] ウィンドウ枠の [ダウンロードタイプ] タブで、[ダウンロードタイプ] として [タイムス タンプ] を選択します。
- 13. [シャドウテーブルを使用してタイムスタンプカラムを保持する] を選択します。

シャドウテーブルを使用すると既存のテーブルを変更する必要がないため、推奨されること が多くあります。反対に、デフォルト設定の場合は、テーブルにタイムスタンプカラムが追 加され、通常、パフォーマンスが向上します。

14. 同期モデルに加えた変更を保存するには、[ファイル]»[保存]を選択します。

結果

Mobile Link プロジェクト、リモートスキーマ名、同期モデルがすべて作成されます。

次の手順

「レッスン4:同期モデルのテスト」91ページに進みます。

参照

- ●「同期モデル」29ページ
- ●「統合データベースの設定」『Mobile Link サーバ管理』
- ●「Mobile Link サーバのシステムテーブル」『Mobile Link サーバ管理』
- ●「Mobile Link サーバシステムプロシージャ」『Mobile Link サーバ管理』
- ●「download_delete_cursor スクリプト」『Mobile Link サーバ管理』
- ●「競合の解決」『Mobile Link サーバ管理』
- ●「競合解決」『Mobile Link サーバ管理』
- ●「パブリケーション」『Mobile Link クライアント管理』
- ●「同期モデルタスク」32ページ
- ●「ダウンロードタイプの変更」37ページ
- ●「競合の検出と解決の変更」44ページ
- ●「テーブルとカラムのマッピング」33ページ

レッスン4:同期モデルのテスト

このレッスンでは、同期モデルを迅速にテストする方法を説明します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Linkの概要」 85ページ

◆ タスク

- 1. Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、mlintro_project » [同期モデル] » sync mlintro の順に展開します。
- 2. [ファイル]»[テスト] をクリックします。
- 3. [OK] をクリックして、統合データベースが変更されることを示す警告を解除します。

同期モデルが統合データベースに展開され、テストのためにリモートデータベースが作成されます。Mobile Link サーバが起動します。

4. [データ] タブをクリックします。上部ウィンドウ枠に、統合データベースの Product テーブル からのローが表示されます。下部ウィンドウ枠からは、リモートデータベースに Product テーブルのローが現在何も含まれていないことがわかります。

複数のテーブルが定義されている場合は、[表示] ドロップダウンリストで、表示するテーブ ルを選択します。

5. [同期] をクリックします。これで、リモートデータベースに3つのローが表示されます。

- 6. [**クライアントログ**] タブを選択します。エラーメッセージまたは警告のログをスキャンしま す。[Mobile Link ログ] タブでも同じ操作を実行します。
- 7. [データ] タブをクリックします。下ウィンドウ枠で、Drywall Screws 10lb という名前を持つ ローを右クリックし、[ローの編集] を選択し、数量を 99 に変更します。
- 8. [同期] をクリックします。リモートデータベースで行った変更と、統合データベースとの同期が完了します。
- 9. [アクション] ボタンをクリックし、[統合データベースで Interactive SQL を開く] を選択しま す。統合データベースに接続する Interactive SQL ウィンドウが開きます。
- 10. 次の SQL 文を実行して、Interactive SQL を閉じます。

update product set quantity = quantity + 1; commit;

- 11. 再び Sybase Central の [データ] タブで、[アクション]»[データの再表示] タブをクリックしま す。統合データベース内の更新されたローが表示されます。
- 12. [同期] をクリックします。リモートデータベースが新しい数量値に更新されます。

13. [テスト] ウィンドウを閉じます。

結果

同期モデルのテストが正常に完了しました。

次の手順

「レッスン5:同期モデルの調整」92ページに進みます。

参照

●「同期モデルの展開」48ページ

レッスン5:同期モデルの調整

このレッスンでは、同期モデルに変更を加える方法を説明するとともに、モデルを展開するとき に使用できる選択肢の一部を解説します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Linkの概要」 85ページ

◆タスク

- 1. Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、mlintro_project » [同期モデル] » sync_mlintro の順に展開します。
- 2. [マッピング] タブで、Product テーブルのローを選択します。
- 3. 下ウィンドウ枠で、[競合の処理] タブを選択します。
- (競合の解決)で、(統合)を選択します。このオプションを選択すると、あるローが統合デー タベースとリモートデータベースの両方で変更された場合に統合データベースの値が正しい 値であるとみなされます。
- 5. [展開] タブで、[テスト] をクリックします。同期モデルを保存するように求めるプロンプト が表示された場合は、[はい] をクリックします。
- 6. [データ] タブを選択し、[同期] をクリックしてリモートデータベースを更新します。
- 7. [Screwmaster Drill] という名前のローを編集し、統合データベースでの数量を 20 に設定し、 リモートデータベースでの数量を 10 に設定します。
- 8. [同期] をクリックします。リモートデータベースと統合データベースの両方で、数量は 20 と 表示されるはずです。
- 9. [テスト] ウィンドウを閉じます。
- sync_mlintro モデルの [イベント] タブを選択します。このページは、この同期モデルのために Mobile Link サーバによって実行される SQL 文を表示します。左側のマージンにある緑色のバーは、[マッピング] タブで選択した内容に基づいて SQL 文が自動的に生成されることを示しています。

[マッピング] タブで使用できるオプションが、シナリオの実現に十分でない場合もあります。 このような場合は、[イベント] タブを使用して同期スクリプトをさらにカスタマイズするこ とができます。

 [イベント] タブでは、Product テーブルの upload_insert イベントを探します。これは、Mobile Link サーバがリモートデータベースから新しいローを受信するときに実行する SQL 文を示 します。このイベントに変更を加えて、新しい注文の最大数量を 50 に制限します。テキス ト {ml r."quantity"} は、アップロードされたリモート quantity カラムを示します。これを次 のように変更します。

If {ml r."quantity"} < 50 then {ml r."quantity"} else 50 end if

完全な upload insert イベントは、次のようになります。

Product (DBA): upload_insert /* Insert the row into the consolidated database. */ INSERT INTO "DBA"."Product" ("name", "quantity") VALUES ({ml r."name"}, lf {ml r."quantity"} < 50 then {ml r."quantity"} else 50 end if)

upload_insert イベントに対するマージンのバーが黄色になったことに注目してください。こ れは、**[マッピング]** タブからの設定が上書きされたことを示します。

- 12. [展開] タブで、[テスト] をクリックします。[データ] タブを選択し、[同期] をクリックします。次に、リモートデータベースの下ウィンドウ枠で右クリックします。[ローの追加] を選択し、Hammer という名前と数量 200 が含まれているローを追加します。[同期] をクリックします。統合データベースとリモートデータベースは両方とも Hammer ローの値が 50 になります。
- 13. [テスト] ウィンドウを閉じて、[イベント] タブに戻ります。upload_insert イベントスクリプト を右クリックし、['Product (DBA): upload_insert' スクリプトのリストア] を選択します。 マージン内のバーの色が再び緑になり、前に行ったカスタマイズが取り消されました。

結果

同期モデルに変更が加えられます。

次の手順

「レッスン6: Mobile Link サーバオプションの選択」 94 ページに進みます。

参照

●「同期モデルの展開」48ページ

レッスン 6: Mobile Link サーバオプションの選択

このレッスンでは、同期モデルを展開するときに Mobile Link サーバを実行するために後で使用 されるオプションの選択方法を説明します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link の概要」 85ページ

◆タスク

- Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、mlintro_project » [Mobile Link サー バのコマンドライン] の順に展開します。同期モデルを保存するように求めるプロンプトが 表示された場合は、[はい] をクリックします。
- 2. 右ウィンドウ枠で [デフォルト] を右クリックし、[プロパティ] を選択します。
- 3. [デフォルト Mobile Link サーバのコマンドラインのプロパティ] ウィンドウで、次のタスク を実行します。

●[一般] タブを選択します。[冗長性] ドロップダウンリストから、[High (-v+)] を選択しま す。 ●[詳細] タブを選択します。

●-dl オプションを true に設定します。

●-o オプションを mlsrv.mls に設定します。

●-zf オプションを true に設定します。

●-zu オプションを true に設定します。

注意

-zfオプションは、デバッグと開発の目的でのみ使用してください。後のレッスンで新しいス クリプトを統合データベースに追加するときに Mobile Link サーバを停止する必要をなくす るため、このチュートリアルでは -zfオプションが必要です。-zu+オプションは、新しい Mobile Link ユーザを同期環境に自動的に追加します。

4. [OK] をクリックします。

結果

同期モデルの展開時に Mobile Link サーバを実行するためのオプションが設定されます。

次の手順

「レッスン7:同期モデルの展開」95ページに進みます。

参照

●「同期モデルの展開」48ページ

レッスン7:同期モデルの展開

このレッスンでは、同期モデル展開ウィザードを使用して同期モデルを展開して同期のために統 合データベースを設定し、リモートデータベースを作成して展開します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Linkの概要」 85ページ

◆タスク

- 1. Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、mlintro_project » [同期モデル] » sync_mlintro の順に展開します。
- 2. [ファイル]»[展開] をクリックします。

- 3. [ウィザードによって生成されたファイルを保管するフォルダを選択します。] フィールドの デフォルト設定を使用し、[次へ] をクリックします。
- 4. [クライアントネットワークオプション] ページのデフォルト設定をそのまま使用し、[次へ] をクリックします。
- 5. [どのような Mobile Link のユーザとパスワードを使用しますか]の下の [Mobile Link ユーザ とパスワード]で [これらの使用] を選択し、次のタスクを実行します。
 - a. [Mobile Link ユーザ] フィールドに、mlintro user と入力します。
 - b. [Mobile Link パスワード] フィールドに、sql と入力します。
 - c. [このユーザを統合データベースに登録します。]をオンにします。登録されたユーザに同 期が許可されるので、[次へ] をクリックします。
- [同期プロファイル] ページで、[同期プロファイル名] フィールドに mlintro_remote_syncprofile と入力します。[参照] をクリックし、[冗長性] を [高] に設定しま す。[OK] をクリックしてから、[次へ] をクリックします。
- 7. [データベースの同期を準備する方法を選択します] ページで次のタスクを実行します。
 - a. [統合データベースの同期を準備するために作成された SQL スクリプトの処理方法を指定してください] で [統合データベースに対して実行します。] を選択します。
 - b. [リモートデータベースの同期を準備するために作成された SQL スクリプトの処理方法 を指定してください] で [新しいリモートデータベースに対して実行します。] を選択します。
 - c. [次へ] をクリックします。
- 8. [選択内容の確認] ページでは、ウィザードで選択した内容を確認できます。また、[ビュー] ボタンを使用すれば、データベースに対して実行される SQL 文を確認できます。

[完了] をクリックします。

sync mlintro remote.db と呼ばれるリモートデータベースが作成されます。

結果

リモートデータベースが正常に作成され、展開されました。

次の手順

「レッスン8: Mobile Link サーバの起動」97ページに進みます。

参照

- ●「同期モデルの展開」48ページ
- ●「リモートデータベースとしての SQL Anywhere データベースの使用」『Mobile Link クライア ント管理』

レッスン 8: Mobile Link サーバの起動

このレッスンでは、mlsrv16 - c オプションを使って Mobile Link サーバを起動し、統合データベースに接続します。追加のオプションを使用して、Mobile Link サーバの動作を設定します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link の概要」 85ページ

◆ タスク

- 1. コマンドプロンプトで、*c*:¥*MLintro*¥*mlintro_project*¥*sync_mlintro_deploy* ディレクトリに移動し ます。
- 2. 次のコマンドを実行します。

mlsrv.bat

注意

-zu オプションを設定すると、任意のユーザが接続できるようになるので、サーバのセキュリティが低下します。運用環境で -zu オプションを使用しないでください。

結果

Mobile Link サーバが起動します。

次の手順

「レッスン9: Mobile Link クライアントの起動」97ページに進みます。

レッスン9: Mobile Link クライアントの起動

このレッスンでは、同期の準備をするためにリモートデータベースを起動します。このレッスンでは、リモートデータベース、統合データベース、Mobile Link サーバが同一のコンピュータ上に存在することを前提としています。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link の概要」 85ページ

◆ タスク

- 1. コマンドプロンプトで、*c*:*¥MLintro¥mlintro_project¥sync_mlintro_deploy* ディレクトリに移動します。
- 2. 次のコマンドを実行して、sync mlintro remote データベースを起動します。

dbsrv16 sync_mlintro_remote

結果

リモートデータベースが起動します。

次の手順

「レッスン 10: Mobile Link プロファイラの起動」98 ページに進みます。

レッスン 10: Mobile Link プロファイラの起動

このレッスンでは、同期の発生を監視するために、Mobile Link プロファイラを開始して設定します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Linkの概要」 85ページ

内容と備考

Mobile Link プロファイラは、同期の統計情報を収集するために使用できます。グラフィック チャートでは、水平軸に時間の経過が示され、垂直軸にはタスクが示されます。Mobile Link プ ロファイラを使用することによって、エラーを引き起こしたり、特定の条件を満たしたりする同 期を迅速に突き止めることができます。Mobile Link プロファイラによってパフォーマンスが大 幅に低下することはないので、開発環境でも運用環境でもこのプロファイラを使用することをお すすめします。

◆ タスク

- 1. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [Mobile Link プロファイラ] をクリックします。
- 2. Mobile Link サーバに Mobile Link プロファイラを接続します。[ファイル]»[プロファイリン グセッションの開始] を選択します。

[Mobile Link サーバへの接続] ウィンドウが表示されます。

3. [ユーザ] フィールドでは、デフォルトユーザ名 monitor_user をそのまま使用します。[パス ワード] フィールドに、sql と入力します。[OK] をクリックします。

結果

前のセッションで Mobile Link サーバを -zu+ オプション付きで起動したので、このユーザは自動 的に追加されています。プロファイラによって同期データの収集が開始されます。

次の手順

「レッスン11:同期」99ページに進みます。

参照

●「Mobile Link プロファイラ」『Mobile Link サーバ管理』

レッスン 11:同期

このレッスンでは、dbmlsync ユーティリティを使用して、リモートデータベースを統合データベースと同期します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link の概要」 85ページ

◆ タスク

- 1. コマンドプロンプトで、*c*:¥*MLintro*¥*mlintro_project*¥*sync_mlintro_deploy* ディレクトリに移動し ます。
- 2. 次のコマンドを実行して、sync mlintro remote データベースを同期します。

sync.bat "server=sync_mlintro_remote;UID=DBA;PWD=sql"

sync_mlintro_remote クライアントの統合データベースとの同期に関連するすべての情報を 表示するウィンドウが表示されます。この情報は remote.dbs ファイルに保存されます。この ファイルはクライアント同期ウィンドウを閉じた後でアクセス可能になります。

3. クライアント同期ウィンドウを閉じます。

[シャットダウン] をクリックします。

4. Mobile Link プロファイラで、[一時停止] ボタンをクリックして水平方向のスクロールを止め、下部ウィンドウ枠を使用して最新の同期にスクロールし戻します。

5. 最新の同期のプロパティを確認します。

同期プロパティを表示するには、色つきの縦線をダブルクリックします。

結果

リモートデータベースが統合データベースと同期されます。

次の手順

「レッスン 12: Mobile Link サーバログファイルビューアを使用したエラーと警告の確認」 100ページに進みます。

参照

●「Mobile Link SQL Anywhere クライアントユーティリティ (dbmlsync)」『Mobile Link クライア ント管理』

レッスン 12: Mobile Link サーバログファイルビューアを使用 したエラーと警告の確認

テーブルの同期が完了したら、各コマンドラインを使用してメッセージログファイル、つまり mlsrv.mlsと remote.dbs をそれぞれ使用して同期の経過を表示できます。これらのファイルのデ フォルトロケーションは、コマンドが実行されたディレクトリです。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Linkの概要」 85ページ

◆タスク

- 1. Sybase Central で、[ツール] » [Mobile Link 16] » [Mobile Link サーバログファイルビューア] をクリックします。
- 2. ログファイルをテキストエディタで開きます。

c:¥MLintro¥mlintro project¥sync mlintro deploy¥mlsrv.mls を参照し、[**開く**] をクリックします。

[Mobile Link サーバログファイルビューア] ウィンドウが表示されます。

- 3. [同期] タブをクリックして、同期中に発生したエラーや警告を調べます。
- 4. [メッセージ] タブをクリックすると、Mobile Link サーバから報告されたエラーと警告を検索 できます。

[情報を表示] オプションをクリアし、[適用] をクリックします。

エラーと警告を含む同期だけが [メッセージ] ウィンドウ枠に表示されます。たとえば、次の 内容を示す警告が表示されることがあります。

[10093] Mobile Link サーバは現在、パフォーマンスを低下させる -zf で実行中です。

- 5. [概要] タブをクリックすると、ログファイルに示される全体の統計情報を検索できます。
- 6. テキストエディタで、remote.dbs などのクライアントログファイルを開きます。
- 7. ファイルの左側を下へスキャンします。「E.」 で始まる行が表示された場合、エラーが発生 したことを示します。ログファイルにエラーがない場合、同期は正常に完了しています。

結果

メッセージログファイルを使用して同期の経過を確認しました。

次の手順

「レッスン 13: SQL Anywhere モニタでの Mobile Link リソースのモニタ」101 ページに進みます。

参照

●「Mobile Link サーバのロギング」『Mobile Link サーバ管理』

レッスン 13: SQL Anywhere モニタでの Mobile Link リソー スのモニタ

このレッスンでは、Mobile Link サーバと Mobile Link サーバファームのモニタリングを設定しま す。このレッスンでは、モニタ Developer Edition を使用しています。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Linkの概要」 85ページ

◆タスク

1. モニタを起動します。次の手順では、モニタが現在バックグラウンドで実行されていないこ とを前提としています。

モニタ Developer Edition を起動するには、次の手順に従います (Windows の場合)。 [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [SQL Anywhere モニタ] をクリック します。 モニタ Developer Edition を起動するには、次の手順に従います (Linux の場合)。 モニタのインストールディレクトリの bin32 または bin64 ディレクトリから samonitor.sh スクリプトを実行します。

samonitor.sh launch

モニタによってメトリックの収集が開始され、モニタにログインするためのデフォルト URL がブラウザに表示されます。*http://localhost:4950* です。

注意

ネットワークを経由してモニタにアクセスしている場合は、http://computer-name:4950 をブラ ウズします。computer-name は、モニタが実行されているコンピュータの名前です。

2. デフォルトの管理者ユーザとしてモニタにログインします。

[ユーザ名] フィールドに admin と入力し、[パスワード] フィールドに admin と入力します。

注意

次の手順を実行するには、モニタに管理者としてログインしてください。読み込み専用ユー ザとオペレータユーザの持つ権限では、一部のタスクしか実行できません。

◆モニタユーザタイプの確認

1. モニタにログインします。

2. [ツール]»[ユーザ設定]を選択し、[ユーザのタイプ] 設定を確認します。

「モニタのユーザ」『SQL Anywhere サーバ データベース管理』を参照してください。

- 3. Mobile Link サーバリソースをモニタに追加するには、次の手順に従います。
 - a. 左のナビゲーションメニューで [ツール]»[管理] をクリックします。
 - b. [リソース] をクリックして、[追加] をクリックします。
 - c. [Mobile Link サーバ] をクリックして [次へ] をクリックします。
 - d. [名前] フィールドに MobiLinkServerSample と入力し、[次へ] をクリックします。
 - e. [ホスト] フィールドに localhost と入力し、[次へ] をクリックします。
 - f. 必要な認証情報が要求されたら、[ユーザ ID] フィールドに monitor_user などのユーザ名 を入力し、[パスワード] フィールドに sql などのパスワードを入力します。 これらのクレデンシャルは、Mobile Link サーバにユーザを作成するために使用されます。 モニタでは、このユーザ ID とパスワードを保存し、Mobile Link サーバへの接続とモニ タリングに使用します。
 - g. [作成] をクリックします。
 - h. 新しいリソース MobiLinkServerSample が作成され、モニタリングが開始されます。
 - i. [閉じる] をクリックします。
 - j. [閉じる] をクリックします。
- k. [概要]»[リソースリスト]をクリックします。MobiLinkServerSample をクリックしてリ ソースのダッシュボードを作成し、開きます。
- 4. 2 つの Mobile Link サーバをモニタリングするための Mobile Link サーバファームリソースを 追加するには、次の手順に従います。
 - a. 2 つの Mobile Link サーバをモニタリング対象のリソースとして追加します。最初のリ ソースとして、前の手順で追加した MobiLinkServerSample リソースを使用します。 2 番目の Mobile Link サーバリソースを追加します。
 - i. コマンドプロンプトで次のコマンドを実行し、ポート 8039 で受信する Mobile Link サーバを起動します。

mlsrv16 -vcrs -zu+ -c "DSN=mlintro_consdb" -ot ml_tcpip.txt -zs ml_tcpip -x tcpip{port=8039}

- ii. [ツール]»[管理] をクリックします。
- iii. [リソース] をクリックして、[追加] をクリックします。
- iv. [Mobile Link サーバ] をクリックして [次へ] をクリックします。
- v. [名前] フィールドに ml tcpip と入力し、[次へ] をクリックします。
- vi. [ホスト] フィールドに、localhost と入力します。

[ポート] フィールドに 8039 と入力し、[次へ] をクリックします。

vii. 必要な認証情報が要求されたら、[ユーザ ID] フィールドに monitor_user などのユー ザ名を入力し、[パスワード] フィールドに sql などのパスワードを入力します。

これらのクレデンシャルは、Mobile Link サーバにユーザを作成するために使用され ます。モニタでは、このユーザ ID とパスワードを保存し、Mobile Link サーバへの接 続とモニタリングに使用します。

viii. [作成] をクリックします。

[概要] ダッシュボードの [リソースリスト] に ml_tcpip リソースが追加されます。

- ix. [閉じる] をクリックします。
- x. [閉じる] をクリックします。
- b. Mobile Link サーバファームリソースを追加します。 Mobile Link サーバファームの詳細 については、「サーバファーム内の Mobile Link サーバ」『Mobile Link サーバ管理』を参 照してください。
 - i. [管理] ウィンドウを開きます。

[ツール]»[管理]をクリックします。

- ii. [リソース] をクリックして、[追加] をクリックします。
- iii. [Mobile Link サーバのファーム] をクリックして [次へ] をクリックします。
- iv. [名前] フィールドに MobiLink Test Farm と入力し、[次へ] をクリックします。
- v. MobiLinkServerSample と ml_tcpip をクリックして、[作成] をクリックします。

- vi. [閉じる] をクリックします。
- vii. [閉じる] をクリックします。
- c. [ダッシュボード]»[概要] をクリックします。 [リソースリスト] に MobiLink_Test_Farm リソースが表示されます。 Mobile Link サーバリソースは [リソースリスト] に残ります。
- d. MobiLink_Test_Farm の左にある矢印をクリックすると、ファームに含まれている Mobile Link サーバリソースのリストが表示されます。
- e. MobiLink_Test_Farm をクリックすると、MobiLink_Test_Farm ダッシュボードが開き、 収集されたメトリックが表示されます。

[警告リスト]、[リソースウィジェット]、[サーバ情報] ウィジェットが表示されます。

5. 警告のテストまたはその他のモニタ機能については、「チュートリアル:モニタによるリソー スのモニタリング」『SQL Anywhere サーバデータベース管理』を参照してください。

結果

SQL Anywhere モニタを使用して、Mobile Link サーバと Mobile Link サーバファームをモニタリ ングしました。

次の手順

「クリーンアップ」104ページに進みます。

クリーンアップ

すべてのチュートリアルをコンピュータから削除します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」86ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link の概要」 85ページ

◆タスク

1. 以下のアプリケーションのインスタンスをすべて閉じます。

●Mobile Link プロファイラ

●Mobile Link 用 SQL Anywhere モニタ

•Sybase Central

•Interactive SQL

- 2. SQL Anywhere、Mobile Link、同期クライアントの各ウィンドウを閉じます。
- 3. 次の手順で、チュートリアルに関連するすべてのデータソースを削除します。
 - a. ODBC データソースアドミニストレータを起動します。
 - b. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [ODBC データソースア ドミニストレータ] をクリックします。
 - c. [ユーザデータ ソース] リストから mlintro_consdb を選択し、[削除] をクリックします。
- 4. 統合データベースとリモートデータベースが保存されているディレクトリを削除します。

結果

チュートリアルがコンピュータから削除されます。

チュートリアル:SQL Anywhere 統合データベースでの Mobile Link の使用

このチュートリアルでは、Mobile Link を使用して SQL Anywhere データベースを活用する方法に ついて説明します。ここでは、SQL Anywhere 統合データベースと Ultra Light リモートデータ ベースの間の同期を設定します。SQL Anywhere リモートデータベースを使用することもできま す。

このチュートリアルの目的は、多くの地域で動作する携帯電話会社のデータを活用することで す。このシナリオで、各地域には次のような特徴があります。

- リモート同期環境である。
- Mobile Link を使って、中央の SQL Anywhere 統合データベースと同期するローカルの Ultra Light データベースがある。
- ●新しい顧客がアカウントをアクティブにしたり、既存の顧客が新しいモバイルデバイスをア クティブにしたときに、そのロケーションから製品情報にアクセスしたり、リモートデータ ベースからデータを操作することができる。

必要なソフトウェア

•SQL Anywhere 16

権限

統合データベースで次のロールおよび権限を持つ必要があります。

●SYS_AUTH_DBA_ROLE 互換ロール

概要

このチュートリアルでは、次の作業の方法について説明します。

●リモートスキーマの設計時に、リモートテーブルの同期方向などの重要な考慮事項を決定す る。 ●統合データベースとリモートデータベースにユニークなプライマリキーを追加する。

●[同期モデル作成ウィザード]を使用して、統合データベースとリモートデータベースの間の同期を設定する。

●Sybase Central を使用して同期設定をカスタマイズする。

●同期モデル展開ウィザードを使用して統合データベースとリモートデータベースを展開する。

●リモートクライアントを統合データベースと同期する。

参照

●「Mobile Link 同期」1 ページ

レッスン1:スキーマの設計

このチュートリアルでは、SQL Anywhere が動作しているコンピュータにサンプルデータベース がインストールされていることを前提としています。

サンプルデータベースは統合データベースとして使用されます。次の表は、SQL Anywhere 統合 データベースの各テーブルの説明です。

テーブル	説明
Customers	レコードに情報が記録されている顧客。
SalesOrders	アカウントアクティブ化レコード。
Products	購入可能なすべての製品のレコード。
CustomerProducts	それぞれの顧客が所有している製品のリス ト。

リモートスキーマの設計

地域ごとに統合データベース全体をコピーしておくことは不要であり、非効率的です。リモート スキーマでは同じテーブル名を使用しますが、各地域に関する情報だけが格納されます。この設 定を実現するため、リモートスキーマは統合データベースのサブセットとして次のように設計さ れています。

統合テーブル	リモートテーブル
Customers	地域によるフィルタ。
SalesOrders	適切な地域にいる顧客の顧客 ID によるフィ ルタ。
Products	すべてのローを含みます。

統合テーブル	リモートテーブル
CustomerProducts	適切な地域にいる顧客の顧客 ID によるフィ ルタ。

それぞれの営業担当者は、すべての顧客に提供された製品の情報と同様、担当の地域の顧客に関 する情報を維持する必要があります。しかし、営業担当者は別の地域の顧客に関する情報は必要 としないので、この情報は各地域のオフィスとは同期されません。そのため、ローは地域の識別 子に基づいてフィルタされます。

注意

リモートデータベースで不要になるカラムがある場合は、テーブルからカラムのサブセットを取得することもできます。

次に、各テーブルの同期の方向を選択します。リモートデータベースで読み込む情報と、リモー トデータベースで作成、変更、または削除する情報について考慮する必要があります。この例で は、地域は、顧客に提供される製品のリストにアクセスする必要がありますが、新製品をシステ ムに入力することはありません。これは、製品情報の入力は必ず中央の統合データベースから行 うという制限を生み出します。一方で、営業担当者が、新しいアカウントのアクティブ化を常時 記録できるようにする必要があります。これらの要因から、各テーブルの同期の方向は次のよう になります。

テーブル	同期
Customers	統合データベースへのアップロードのみ。
SalesOrders	統合データベースへのアップロードのみ。
Products	リモートデータベースへのダウンロードの み。
CustomerProducts	統合データベースへのアップロードのみ。

「レッスン2:統合データベースの準備」107ページに進みます。

レッスン2:統合データベースの準備

このレッスンでは、統合データベースに接続し、CustomerProducts テーブルを作成し、地域情報 を含めるように Customers テーブルを変更します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」106ページを参照してください。 このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル: SQL Anywhere 統合デー タベースでの Mobile Link の使用」

内容と備考

同期システムでは、テーブルのプライマリキーは、異なるデータベース内の同じローを識別する 唯一の方法であり、競合を検出する唯一の方法です。使用する各テーブルには、プライマリキー が必要です。プライマリキーが更新されることはありません。また、1つのデータベースに挿入 されたプライマリキーの値が別のデータベースに挿入されないようにする必要があります。

あとのレッスンでは、統合スキーマからリモートスキーマを作成するため、リモートスキーマは 統合スキーマと同じプライマリキーを持つことになります。

プライマリキーがすべてのデータベースに対してユニークになるようにカラムが選択されてい ます。Customers テーブルでは、プライマリキーは ID カラムで構成されています。リモートの Customers テーブルに挿入されるすべての値には、ユニークな顧客 ID 番号が必要です (地域の値 は常に同じです)。これにより、リモートの各 Customers テーブルで一意性が確保されます。統 合データベースの Customers テーブルのプライマリキーは、複数の販売担当者がデータがアップ ロードした場合の競合を防止する役割があります。ある地域からの各アップロードは、地域値が 異なるので、他の地域とは異なっています。

◆タスク

- 1. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [Sybase Central] をクリッ クします。
- 2. [接続] » [SQL Anywhere 16 に接続] をクリックします。
- 3. [接続] ウィンドウで次のタスクを実行します。
 - a. [アクション] ドロップダウンリストで、[ODBC データソースを使用した接続] を選択し ます。
 - b. [ODBC データソース名] フィールドに SQL Anywhere 16 Demo と入力します。
 - c. [接続] をクリックします。
- 4. Interactive SQL の統合データベースに接続します。

コマンドプロンプトで次のコマンドを実行します。

dbisql -c "DSN=SQL Anywhere 16 Demo"

5. Interactive SQL で、次の文を実行し、CustomerProducts テーブルを作成してデータを挿入しま す。

CREATE TABLE CustomerProducts (ID int default AUTOINCREMENT PRIMARY KEY, SalesOrderID int NOT NULL, CustomerID int NOT NULL, ProductID int);

INSERT INTO CustomerProducts (SalesOrderID,CustomerID,ProductID)

SELECT SalesOrders.ID, SalesOrders.CustomerID, SalesOrderItems.ProductID FROM SalesOrders, SalesOrderItems WHERE SalesOrders.ID = SalesOrderItems.ID;

6. Interactive SQL で、次の文を実行し、Customers テーブルのそれぞれの顧客に地域情報を追加 します。

```
ALTER TABLE Customers
ADD Region VARCHAR(255);
UPDATE Customers
SET Region = (SELECT TOP 1 SalesOrders.Region
FROM SalesOrders
WHERE Customers.ID = SalesOrders.CustomerID
ORDER BY Region);
COMMIT;
```

結果

SQL Anywhere Demo データベースとの接続が確立され、CustomerProducts という名前のテーブルが作成され、地域情報を含むように Customers テーブルが変更されます。

次の手順

「レッスン3:同期モデルの作成」109ページに進みます。

参照

- ●「Mobile Link 統合データベース」『Mobile Link サーバ管理』
- ●「ユニークなプライマリキー」『Mobile Link サーバ管理』

レッスン3:同期モデルの作成

このレッスンでは、プロジェクト作成ウィザードを使用して新しい Mobile Link プロジェクトを 作成します。また、プロジェクト作成ウィザードでは、後で編集可能なデフォルトを使用して同 期モデルも作成されます。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル: SQL Anywhere 統合デー タベースでの Mobile Link の使用」

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」106ページを参照してください。

◆ タスク

- 1. Sybase Central で、[ツール] » [Mobile Link 16] » [新しいプロジェクト] をクリックします。
- 2. [プロジェクト作成ウィザード] が表示されます。

- 3. [新しいプロジェクトの名前を指定してください。] フィールドに mlsqla_project と入力します。
- (新しいプロジェクトの保存場所を指定してください。) フィールドに C:¥mlsqla と入力し、[次 へ] をクリックします。
- 5. [データベースの表示名] フィールドに demo と入力します。
- 6. [編集] をクリックします。
- 7. [汎用 ODBC データベースに接続] ページで次のタスクを実行します。
 - a. [ODBC データソース名] フィールドで、[参照] をクリックして SQL Anywhere 16 Demo を選択します。
 - b. [OK] をクリックし、[保存] をクリックします。
- 8. [パスワードを記憶] オプションを選択し、[次へ] をクリックします。
- 9. [新しいリモートデータベーススキーマ] ページの [リモートデータベースに含める統合デー タベースのテーブルとカラムを指定してください。] リストで、次のテーブルを選択します。
 - CustomerProducts
 - Customers
 - Products
 - SalesOrders

[次へ] をクリックします。

- 10. [リモートスキーマ名をプロジェクトに追加] オプションを選択します。
- 11. [新しいリモートスキーマ名を指定してください。] フィールドに mlsqla_remote_schema と入力し、[次へ] をクリックします。
- 12. [使用するリモートデータベースのタイプを指定してください。] オプションの [Ultra Light] を選択し、[完了] をクリックします。
- 13. Mobile Link 設定スクリプトをインストールするように要求されたら [はい] をクリックします。
- 14. リモートスキーマをインポートするように要求された場合は、[はい]をクリックします。
- 15. [OK] をクリックします。
- 16. 新しい同期モデルを右クリックして、[プロパティ]を選択します。
 - a. 最初のフィールドに sync_mlsqla と入力します。
 - b. [パブリケーション名] フィールドに sync_mlsqla_publication と入力します。
 - c. [スクリプトバージョン] に sync_mlsqla_scriptversion と入力します。 パブリケーションは、同期するデータを指定するリモートデータベース上のオブジェクトです。Mobile Link サーバのスクリプトにより、リモートデータベースからアップロードされたデータを統合データベースに適用する方法と、スクリプトバージョンによりス

クリプトをグループ化する方法が定義されます。アプリケーションごとに異なるスクリ プトバージョンを使用できるため、1 つの Mobile Link サーバを管理するだけで複数のア プリケーションを同期できます。

d. [適用] をクリックしてから [OK] をクリックします。

17. Sybase Central の右ウィンドウ枠で次のタスクを実行します。

- a. [イベント] タブをクリックします。
- b. CustomerProducts ダウンロードカーソルを更新して、東部地域の顧客向けの顧客製品だけ をダウンロードするようにします。

CustomerProductss テーブル用の download_cursor イベントの既存の SQL スクリプトを次のクエリに置き換えます。

SELECT "DBA"."CustomerProducts"."ID", "DBA"."CustomerProducts"."SalesOrderID", "DBA"."CustomerProducts"."CustomerID", "DBA"."CustomerProducts"."ProductID" FROM "DBA"."CustomerProducts" INNER JOIN "GROUPO"."Customers" ON "GROUPO"."Customers"."ID" = "DBA"."CustomerProducts"."CustomerID" WHERE "GROUPO"."Customers"."Region" = 'Eastern';

c. Customers テーブルのダウンロードカーソルを更新して、東部地域の顧客情報だけをダウ ンロードするようにします。

Customers テーブル用の download_cursor イベントの既存の SQL スクリプトを次のクエ リに置き換えます。

SELECT "GROUPO"."Customers"."ID", "GROUPO"."Customers"."Surname", "GROUPO"."Customers"."GivenName", "GROUPO"."Customers"."City", "GROUPO"."Customers"."City", "GROUPO"."Customers"."State", "GROUPO"."Customers"."Country", "GROUPO"."Customers"."PostalCode", "GROUPO"."Customers"."Phone", "GROUPO"."Customers"."CompanyName", "GROUPO"."Customers"."Region" FROM "GROUPO"."Customers"."WHERE Region = 'Eastern';

d. SalesOrders ダウンロードカーソルを更新して、東部地域の顧客向けの注文情報だけをダウンロードするようにします。

SalesOrders テーブル用の download_cursor イベントの既存の SQL スクリプトを次のクエ リに置き換えます。

SELECT "GROUPO"."SalesOrders"."ID", "GROUPO"."SalesOrders"."CustomerID", "GROUPO"."SalesOrders"."OrderDate", "GROUPO"."SalesOrders"."FinancialCode", "GROUPO"."SalesOrders"."Region", "GROUPO"."SalesOrders"."SalesRepresentative" FROM "GROUPO"."SalesOrders"."Region" = 'Eastern' AND "GROUPO"."SalesOrders"."ID" IN (SELECT "DBA"."CustomerProducts"."SalesOrderID" FROM "DBA"."CustomerProducts"); 18. 同期モデルを保存します。

[ファイル] » [保存] をクリックします。

同期モデルが完成して、展開の準備が完了します。

結果

Mobile Link プロジェクトおよび同期モデルが作成されます。

次の手順

「レッスン4:同期モデルの展開」112ページに進みます。

参照

- ●「同期モデル」29ページ
- ●「統合データベースの設定」『Mobile Link サーバ管理』
- ●「Mobile Link サーバのシステムテーブル」『Mobile Link サーバ管理』
- ●「Mobile Link サーバシステムプロシージャ」『Mobile Link サーバ管理』
- ●「download_delete_cursor スクリプト」『Mobile Link サーバ管理』
- ●「パブリケーション」『Mobile Link クライアント管理』
- ●「同期モデルタスク」32ページ
- ●「ダウンロードタイプの変更」37ページ
- ●「テーブルとカラムのマッピング」33ページ

レッスン4:同期モデルの展開

同期モデル展開ウィザードを使用すると、統合データベースとリモートデータベースを展開できます。各データベースの展開は1つずつ行うこともできますが、両方一度に行うこともできます。同期モデル展開ウィザードでは、展開のオプションを順を追って設定できます。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル: SQL Anywhere 統合デー タベースでの Mobile Link の使用」

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」106ページを参照してください。

◆ タスク

- 1. Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、mlsqla_project、[同期モデル]、 sync_mlsqla の順に展開します。
- 2. [ファイル] » [展開] をクリックします。

同期モデル展開ウィザードが表示されます。生成されたファイルのデフォルトロケーション を選択し、[次へ]をクリックします。

- [クライアントネットワークオプション] ページのデフォルトをそのまま使用し、[次へ] をク リックします。
- 4. [Mobile Link のユーザとパスワード] ページで、次のタスクを実行します。
 - a. [どのような Mobile Link のユーザとパスワードを使用しますか] で、[これらの使用] を選択します。
 - b. [Mobile Link ユーザ] フィールドに、mlsqla_remote と入力します。
 - c. [Mobile Link パスワード] フィールドに、mlsqla pass と入力します。
 - d. [このユーザを統合データベースに登録します。]をオンにします。登録されたユーザは同 期することができます。
 - e. [次へ] をクリックします。
- 5. 同期プロファイル名を mlsqla remote syncprofile に変更し、[次へ] をクリックします。
- [データベースの同期を準備する方法を選択します]ページで、[統合データベースの同期を準備するために作成された SQL スクリプトの処理方法を指定してください] オプションの [統合データベースに対して実行します。] を選択します。
- [リモートデータベースの同期を準備するために作成された SQL スクリプトの処理方法を指定してください] オプションで [新しいリモートデータベースに対して実行します。] を選択します。 [次へ] をクリックします。
- 8. ウィザードによって作成された SQL スクリプトを確認するには、[表示] をクリックします。
- 9. [完了] をクリックします。
- 10. [閉じる] をクリックします。

結果

統合データベースを複数のリモートクライアントと同期するために設定し、1 つのリモートクラ イアントが正常に展開されました。他のリモートクライアントを展開する場合は、もう一度この ウィザードを実行し、新しい Mobile Link ユーザを作成して統合データベースと Mobile Link サー バの展開を終了します。統合データベースとリモートデータベースはすでに展開されているの で、実行する必要があるのは、他のリモート同期クライアントを展開することだけです。

次の手順

「レッスン5: Mobile Link サーバの起動」114 ページに進みます。

参照

- ●「同期モデルの展開」48ページ
- ●「リモートデータベースとしての SQL Anywhere データベースの使用」『Mobile Link クライア ント管理』
- ●「Mobile Link ユーザ」『Mobile Link クライアント管理』

レッスン5: Mobile Link サーバの起動

このレッスンでは、mlsrv16-c オプションを使って Mobile Link サーバを起動し、統合データベースに接続します。追加のオプションを使用すると、Mobile Link サーバの動作を設定できます。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル: SQL Anywhere 統合デー タベースでの Mobile Link の使用」

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」106ページを参照してください。

◆ Mobile Link サーバの起動

- 1. コマンドプロンプトで、c:¥mlsqla ディレクトリに移動します。
- 2. 次のコマンドを実行して、統合データベースに接続します。

mlsrv16 -c "DSN=SQL Anywhere 16 Demo" -o mlsrv.mls -v+ -dl -zf -zu+ -x tcpip

Mobile Link サーバメッセージウィンドウが表示されます。

このチュートリアルで使用している Mobile Link サーバの各オプションの説明を次に示しま す。オプション-o、-v、-dl は、デバッグとトラブルシューティングの情報を提供します。こ れらのロギングオプションは、開発環境での使用に適しています。パフォーマンス上の理由 から、一般的に-v+と-dl は運用環境では使用しません。 Mobile Link サーバオプションの完 全なリストについては、「Mobile Link サーバオプション」『Mobile Link サーバ管理』を参照し てください。

オプション	説明
-c	続いて接続文字列を指定します。
-0	メッセージログファイル mlsrv.mls を指定します。
-v+	ログを取る対象となる情報を指定します。- v+を使用して、最大冗長ロギングをオンに 設定します。
-dl	画面にすべてのログメッセージを表示しま す。
-zf	各同期の始めに Mobile Link サーバがスク リプトの変更をチェックします。
-zu+	自動的に新しいユーザを追加します。

オプション	説明
-X	Mobile Link クライアントの通信プロトコル とパラメータを設定します。

注意

-zf および-zu+ オプションは、デバッグと開発の目的でのみ使用してください。後のレッス ンで新しいスクリプトを統合データベースに追加するときにサーバを停止する必要をなくす るため、このチュートリアルでは-zf オプションが必要です。-zu+ オプションは、新しい Mobile Link ユーザを同期環境に自動的に追加します。

結果

Mobile Link サーバが起動し、統合データベースに接続されます。

次の手順

「レッスン6:同期」115ページに進みます。

レッスン6:同期

このレッスンでは、ulsync ユーティリティで同期を開始することによって、Mobile Link クライアントを Mobile Link サーバと同期します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル: SQL Anywhere 統合デー タベースでの Mobile Link の使用」

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」106ページを参照してください。

◆ タスク

1. *c*:*¥mlsqla¥mlsqla_project¥sync_mlsqla_deploy* ディレクトリに移動し、次のコマンドを実行して sync mlsqla remote データベースを同期します。

ulsync -c "DBF=sync_mlsqla_remote.udb" "Publications=sync_mlsqla_publication;MobiLinkUid=mlsqla_remote;MobiLinkPwd=mlsqla_pass;Scri ptVersion=sync_mlsqla_scriptversion;Stream=tcpip{port=2439}"

- DBF 実行していないデータベースを起動したときにロードして接続するデータベース ファイルを指定します。
- Publications 同期の実行時に使用するリモートデバイスのパブリケーション。(このパ ブリケーションは[同期モデル作成ウィザード] で作成されています。)
- MobiLinkUid Mobile Link サーバによる認証に使用するユーザ名。

● **MobiLinkPwd** Mobile Link サーバによる認証に使用するパスワード。

● ScriptVersion 同期の実行時に使用するリモートデバイスのスクリプトバージョン。 (このパブリケーションは[同期モデル作成ウィザード] で作成されています。)

● Stream ネットワークプロトコルを設定するオプションを設定します。

Mobile Link サーバのメッセージウィンドウに同期の進行状況が表示されます。このコマンドが正常に実行されると、ulsync アプリケーションによりリモートデータベースに統合データベースの情報のサブセットが格納されます。

同期が失敗した場合は、ulsync アプリケーションに渡した接続情報、および Mobile Link ユー ザ名とパスワードを確認します。それでも失敗する場合は、使用したパブリケーション名を 確認し、統合データベースと Mobile Link サーバが実行中であることを確認します。また、同 期ログ (サーバ、クライアントとも)の内容を確認することもできます。

注意

別のコンピュータにある ulsync アプリケーションを Mobile Link サーバから実行している場合、Mobile Link サーバのロケーションを指定する引数を渡す必要があります。

Mobile Link サーバを使用してリモートクライアントを正常に統合データベースに同期する と、リモートデータベースには1件の地域に関する情報が格納されます。SQL Anywhere 16 プラグインを使用すると、Sybase Central でデータベースにデータが移植されているかどうか を確認できます。

- 2. Sybase Central を開きます。
- 3. 次の手順でリモートデータベースに接続します。
 - a. 左ウィンドウ枠で [Ultra Light 16] を右クリックし、[接続] をクリックします。
 - b. [ユーザ ID] に DBA と入力し、[パスワード] に sql と入力します。
 - c. [データベースファイル] フィールドに、C:¥mlsqla¥mlsqla_project¥sync_mlsqla_deploy ¥sync_mlsqla_remote.udb と入力します。
 - d. [接続] をクリックします。
- 4. 左ウィンドウ枠で、[Ultra Light 16]、sync_mlsqla_remote、[テーブル]、[Customers] の順に 展開します。
- 5. 右ウィンドウ枠で、[データ] タブをクリックします。

Customers テーブルで、すべてのレコードは東部地域に関係する顧客のレコードです。この地域は、他の地域の顧客情報とは関係がありません。このため、地域を基準にしてローをフィルタ処理で除外するよう同期スクリプトを設定し、このデータベースのリモート ID を特定の地域識別子の値に設定します。この地域のデータベースの容量は小さくなり、同期に必要な時間も短くなります。リモートデータベースのサイズは最小限に抑えられているため、新しい顧客の入力やモバイルデバイスの変更の処理などの頻繁に行われる処理が迅速かつ効率的に実行されます。

結果

リモートデータベースと統合データベースの間で、データが同期されます。

次の手順

「クリーンアップ」117ページに進みます。

参照

●「Ultra Light 同期ユーティリティ (ulsync)」『Ultra Light データベース管理とリファレンス』

クリーンアップ

チュートリアルをコンピュータから削除します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル: SQL Anywhere 統合デー タベースでの Mobile Link の使用」

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」106ページを参照してください。

◆コンピュータからのチュートリアルの削除

1. 以下のアプリケーションのインスタンスをすべて閉じます。

•Sybase Central

•Interactive SQL

- 2. 統合データベースとリモートデータベースが保存されている C:¥mlsqla ディレクトリを削除 します。
- 次のコマンドを実行して、サンプルデータベースを消去し、新しいサンプルデータベースの コピーを元のオブジェクトおよびデータとともに作成します。

newdemo "%SQLANYSAMP16%¥demo.db"

プロンプトが表示されたら、既存のファイルをすべて消去することを選択します。

結果

チュートリアルがコンピュータから削除されます。

次の手順

なし。

チュートリアル: Oracle Database 10g での Mobile Link の使用

このチュートリアルでは、Mobile Link を使用して Oracle Database 10g を活用する方法について 説明します。ここでは、Oracle Database 10g と SQL Anywhere リモートデータベースの間の同期 を設定します。また、Ultra Light リモートデータベースを設定することもできます。

このチュートリアルは、販売チームに関するデータを活用することを目的としています。このシ ナリオでは、各販売担当者はリモート同期クライアントです。各販売担当者にはローカルの SQL Anywhere データベースが用意されています。このデータベースは Mobile Link を使用して 本社にある社内 Oracle データベースと同期されます。各販売担当者はラップトップまたはモバ イルデバイスを使用して社内データにアクセスし、リモートデータベースからデータを操作しま す。

このチュートリアルでは、Oracle Database 10g の基本インストールが行われていることを前提と しています。このインストールでは、orcl という名前のスターターデータベースが作成されま す。orcl データベースには OE (注文エントリ) と HR (人材) というサンプル スキーマがありま す。または、Oracle Database Configuration Assistant を使用して新しいデータベースを作成してサ ンプルスキーマをインストールしたり、SQL*Plus を介して空のデータベースにサンプルスキー マを手動でインストールすることもできます。サンプルスキーマ SQL ファイルは、個別の Example ダウンロードパッケージを介して Oracle から直接入手できます。これらのサンプルス キーマのインストールの詳細については、http://www.oracle.com/technetwork/testcontent/ connection11g-088156.html を参照してください。

このチュートリアルでは、SYSDBA 権限がある SYS ユーザとして Oracle に接続できることを前 提としています。これは Oracle システムビュー GV_\$TRANSACTION に対するパーミッション を付与するための必要条件です。SYS ユーザのパスワードは Oracle データベースのインストー ル時に設定されます。

必要なソフトウェア

•SQL Anywhere 16

●Oracle Database 10g Release 2 以降

概要

このチュートリアルでは、次の作業の方法について説明します。

- ●リモートスキーマの設計時に、リモートテーブルの同期方向などの重要な考慮事項を決定す る。
- ●統合データベースとリモートデータベースにユニークなプライマリキーを追加する。

●Mobile Link を Oracle Database 10g に接続する ODBC データソースを作成する。

●[同期モデル作成ウィザード]を使用して、統合データベースとリモートデータベースの間の同期を設定する。

●Sybase Central を使用して同期モデルをカスタマイズする。

●同期モデル展開ウィザードを使用して統合データベースとリモートデータベースを展開する。

●リモートクライアントを統合データベースと同期する。

パーミッションと権限

Oracle データベースで次のパーミッションを持つ必要があります。

●SYS.GV_\$TRANSACTION に対する SELECT

●SYS.GV \$SESSION に対する SELECT

●SYS.V_\$SESSION に対する SELECT

●SYS.GV_\$LOCK に対する SELECT

●SYS.DBMS_UTILITY に対する EXECUTE

●DBA_OBJECTS に対する SELECT

リモートデータベースで次のロールおよび権限を持つ必要があります。

●SYS_REPLICATION_ADMIN_ROLE システムロール

●SYS RUN REPLICATION ROLE システムロール

参照

- ●「Mobile Link 同期」1 ページ
- 「Sybase Central の Mobile Link プラグイン」 22 ページ

レッスン1:スキーマの設計

このチュートリアルでは、OE (注文エントリ) と HR (人材) というサンプルスキーマがインス トールされていることを前提としています。OE スキーマは統合データベースとして使用しま す。このスキーマには従業員、注文、顧客、製品に関する情報がまとめられています。このチュー トリアルでは、主に OE スキーマを使用します。

OE スキーマのテーブルのうち、このチュートリアルに関連するテーブルの概要を次に示します。

テーブル	説明
CUSTOMERS	レコードに情報が記録されている顧客。
INVENTORIES	各倉庫に保管されている各製品の数量。
ORDER_ITEMS	各注文の対象となっている製品のリスト。
ORDERS	特定の日付に販売担当者と顧客の間で成立し た販売の記録。

テーブル	説明
PRODUCT_DESCRIPTIONS	各製品に関する、各種言語での説明。
PRODUCT_INFORMATION	システム内の各製品の記録。

リモートスキーマの設計

販売担当者ごとに統合データベース全体をコピーしておくことは不要であり、非効率的です。リ モートスキーマは、1人の特定の販売担当者に関連する情報のみを格納するように設計します。 そのため、リモートスキーマは次のように設計します。

統合テーブル	リモートテーブル
CUSTOMERS	すべての行を抽出。
INVENTORIES	リモートでは使用しない。
ORDER_ITEMS	sales_rep_id を基準にフィルタ処理。
ORDERS	すべての行を抽出。
PRODUCT_DESCRIPTIONS	リモートでは使用しない。
PRODUCT_INFORMATION	すべての行を抽出。

各販売担当者はすべての顧客と製品の記録を保持し、すべての製品をどの顧客にも販売できるようにする必要があります。このチュートリアルでは、販売担当者は常に顧客と同じ言語を使用していることを前提としているため、PRODUCT_DESCRIPTIONSテーブルは必要ありません。各販売担当者には注文に関する情報が必要ですが、他の販売担当者に関する注文の情報は不要です。そのため、ローは販売担当者の識別子に基づいてフィルタされます。

次に、各テーブルの同期の方向を選択します。リモートデータベースで読み込む情報と、リモー トデータベースで作成、変更、または削除する情報について考慮する必要があります。この例で は、各販売担当者は製品と顧客のリストを必要としていますが、新しい製品の情報をシステムに 入力することはありません。製品と顧客の情報の入力は必ず本社の統合データベースから行う という制限が設けられています。一方で、販売担当者は新しい販売情報を常時記録できる必要が あります。このような理由から、各テーブルの同期については次のように決められています。

テーブル	同期
CUSTOMERS	リモートデータベースへのダウンロードの み。
ORDER_ITEMS	ダウンロードとアップロード。
ORDER	ダウンロードとアップロード。

テーブル	同期
PRODUCT_INFORMATION	リモートデータベースへのダウンロードの み。

「レッスン2:統合データベースの準備」121ページに進みます。

レッスン2:統合データベースの準備

OE データベースは Mobile Link で使用できるよう変更する必要があります。ユーザ定義型とし て作成されたカラムは削除されます。これらのユーザ定義型を SQL Anywhere が認識できる型 に変換することもできますが、このチュートリアルではこの操作は行いません。また、Mobile Link では OE のクレデンシャルを使用してトリガを作成する必要があるため、トリガを作成する ためのパーミッションを OE ユーザに付与する必要があります。

前提条件

このチュートリアルでは、OE (注文エントリ) サンプルデータベースがインストールされている ことを前提としています。このサンプルスキーマのインストールについては、http:// docs.oracle.com/cd/B14117 01/server.101/b10771.pdfの Oracle のマニュアルを参照してください。

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118 ページ

内容と備考

Oracle 11g 用のサンプルスキーマのインストールについては、http://docs.oracle.com/cd/B28359_01/ server.111/b28328.pdfの Oracle のマニュアルを参照してください。

◆ タスク

1. SYSDBA 権限を持つ SYS ユーザとして、Oracle SQL Plus アプリケーションを使用して接続します。コマンドプロンプトで次のコマンドを実行します。

sqlplus SYS/your password for sys as SYSDBA

2. ユーザ定義型として作成されたカラムを削除するには、次の文を実行します。

ALTER TABLE OE.CUSTOMERS DROP COLUMN CUST_ADDRESS; ALTER TABLE OE.CUSTOMERS DROP COLUMN PHONE_NUMBERS; ALTER TABLE OE.CUSTOMERS DROP COLUMN CUST_GEO_LOCATION; ALTER TABLE OE.PRODUCT_INFORMATION DROP COLUMN WARRANTY_PERIOD;

3. OE ユーザのロックを解除し、パスワードを sql に設定するには、次の文を実行します。

ALTER USER OE IDENTIFIED BY sql ACCOUNT UNLOCK;

4. OE ユーザがトリガを作成できるようにするには、次の文を実行します。

GRANT CREATE ANY TRIGGER TO OE;

5. orders_customer 外部キーを削除して、CUSTOMERS テーブルの customer_id を参照する新し い外部キーを作成するには、次のコマンドを実行します。

ALTER TABLE OE.ORDERS DROP CONSTRAINT ORDERS_CUSTOMER_ID_FK; ALTER TABLE OE.ORDERS ADD CONSTRAINT ORDERS_CUSTOMER_ID_FK FOREIGN KEY (CUSTOMER_ID) REFERENCES OE.CUSTOMERS (CUSTOMER_ID);

結果

ユーザ定義型として作成されたカラムが削除され、OE ユーザはトリガを作成できるようになり ます。

次の手順

「レッスン 3 : ユニークなキーの統合データベースへの追加」122 ページに進みます。

参照

- ●「Mobile Link 統合データベース」『Mobile Link サーバ管理』
- ●「Oracle 統合データベース」『Mobile Link サーバ管理』
- ●「ユニークなプライマリキー」『Mobile Link サーバ管理』

レッスン3:ユニークなキーの統合データベースへの追加

同期システムでは、テーブルのプライマリキーは、異なるデータベース内の同じローを識別する 唯一の方法であり、競合を検出する唯一の方法です。使用する各テーブルには、プライマリキー が必要です。プライマリキーが更新されることはありません。また、1つのデータベースに挿入 されたプライマリキーの値が別のデータベースに挿入されないようにする必要があります。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118ページ

内容と備考

ユニークなプライマリキーを生成する方法は複数あります。このチュートリアルでは、簡単に操作を行うため、複合プライマリキー方式を使用します。この方式では、統合データベースとリモートデータベースにまたがってユニークな複数のカラムを使用してプライマリキーを作成します。

◆タスク

1. コマンドプロンプトで次のコマンドを実行します。

sqlplus SYS/your password for sys as SYSDBA

 SALES_REP_ID に追加する値は HR.EMPLOYEES テーブルに存在する必要があります。 ORDERS_SALES_REP_FK 外部キーによりこのルールが強制的に適用されます。次の文を実 行して外部キーを削除します。

ALTER TABLE OE.ORDERS DROP CONSTRAINT ORDERS_SALES_REP_FK;

SALES_REP_ID カラムには NULL 値が含まれているため、このカラムをプライマリキーとして追加することはできません。このチュートリアルでは、NULL 値は1に置き換えます。次の文を実行します。

UPDATE OE.ORDERS SET SALES_REP_ID = 1 WHERE SALES_REP_ID IS NULL;

4. ORDER_ID カラムは ORDERS テーブルの現在のプライマリキーです。現在のプライマリ キーを削除するには、次の文を実行します。

ALTER TABLE OE.ORDERS DROP PRIMARY KEY CASCADE;

5. 複合プライマリキーは SALES_REP_ID カラムと ORDER_ID カラムにより構成されます。複 合プライマリキーを追加するには、次の文を実行します。

ALTER TABLE OE.ORDERS ADD CONSTRAINT salesrep_order_pk PRIMARY KEY (sales_rep_id, order_id);

結果

これらの文を実行した後、Mobile Link サーバでは統合データベースに接続し、任意の数のリモー トデータベースとの同期を設定できるようになります。

あとのレッスンでは、統合スキーマからリモートスキーマを作成するため、リモートスキーマは 統合スキーマと同じプライマリキーを持つことになります。

プライマリキーがすべてのデータベースに対してユニークになるようにカラムが選択されてい ます。ORDERS テーブルでは、プライマリキーは SALES_REP_ID カラムと ORDER_ID カラムで 構成されています。リモートデータベースの sales テーブルに挿入されるすべての値には、ユ ニークな注文番号が必要です (SALES_REP_ID の値は常に同じです)。これにより、リモートの 各 ORDERS テーブルで一意性が確保されます。統合データベースの ORDERS テーブルのプラ イマリキーは、複数の販売担当者がデータがアップロードした場合の競合を防止する役割があり ます。販売担当者ごとに SALES_REP_ID の値が異なるため、1 人の販売担当者が行うアップ ロードはいずれも他の販売担当者に対してユニークです。

次の手順

「レッスン4: Mobile Link の接続」124ページに進みます。

レッスン4: Mobile Link の接続

このレッスンでは、Mobile Link を統合データベースに接続する ODBC データソースを作成します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118 ページ

◆ タスク

1. ODBC データソースを作成します。

SQL Anywhere 16 に付属の SQL Anywhere 16 - Oracle ODBC ドライバを使用してください。 次の設定を使用してください。

ODBC タブのフィールド	值
[データソース名]	oracle_cons
[ユーザ ID]	OE
[パスワード]	sql
[TNS サービス名]	orcl
[プロシージャから結果が返される、または VARRAY パラメータを使用する]	選択
[配列サイズ]	60000

このチュートリアルでは、Oracle Database 10g の基本インストールが行われていることを前提 としています。このインストールでは、orcl という名前のスターターデータベースが作成さ れます。OE (注文エントリ) スキーマは自動的に orcl にインストールされます。OE スキー マを別のデータベースにインストールした場合、データベース名を TNS サービス名の値とし て使用します。

2. [テスト接続] をクリックして ODBC 接続をテストします。

結果

ODBC データソースが作成され、テストされます。

次の手順

「レッスン5: Mobile Link プロジェクトと同期モデルの作成」125ページに進みます。

参照

- http://www.sybase.com/detail?id=1011880
- ●「SQL Anywhere 16 Oracle ODBC ドライバ」『Mobile Link サーバ管理』

レッスン 5: Mobile Link プロジェクトと同期モデルの作成

このレッスンでは、新しい Mobile Link プロジェクトを作成して、統合データベースに接続しま す。プロジェクトを作成すると、同期モデルが自動的に作成されます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118ページ

◆タスク

- 1. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [Sybase Central] をクリッ クします。
- 2. [ツール] » [Mobile Link 16] » [新しいプロジェクト] をクリックします。

[プロジェクト作成ウィザード] が表示されます。

- (新しいプロジェクトの名前を指定してください。) フィールドに oracle_project と入力します。
- (新しいプロジェクトの保存場所を指定してください。) フィールドに C: ¥mlora と入力し、[次 へ] をクリックします。
- 5. [データベースの表示名] フィールドに oracle_cons と入力します。
- 6. [編集] をクリックします。
- 7. [汎用 ODBC データベースに接続] ページで次のタスクを実行します。
 - a. [ユーザ ID] フィールドに OE と入力します。
 - b. [パスワード] フィールドに sql アカウントのパスワードを入力します。
 - c. [ODBC データソース名] フィールドで、[参照] をクリックして oracle_cons を選択します。
 - d. [OK] をクリックし、[保存] をクリックします。
- 8. [パスワードを記憶] オプションを選択し、[次へ] をクリックします。
- 9. [統合スキーマの所有者] ウィンドウで、[選択した所有者のみを対象にデータベーススキーマ をロードする] を選択し、[OE ユーザ] をオンにします。

- 10. [新しいリモートデータベーススキーマ] ページの [リモートデータベースに含める統合デー タベースのテーブルとカラムを指定してください。] リストで、次のテーブルを選択し、[次 へ] をクリックします。
 - CUSTOMERS
 - ORDERS
 - ORDER_ITEMS
 - PRODUCT_INFORMATION
- 11. [リモートスキーマ名をプロジェクトに追加] オプションを選択します。
- 12. リモートスキーマ名に oracle remote schema と入力し、[次へ] をクリックします。
- 13. [SQL Anywhere] オプションを選択して、[完了] をクリックします。

Mobile Link が今回初めて統合データベースを使用する場合、Mobile Link システム設定をイン ストールするかどうかを確認するメッセージが表示されます。Mobile Link システム設定を インストールすると、Mobile Link システムテーブルと Mobile Link システムプロシージャが 追加されます。[はい] をクリックし、[OK] をクリックします。

結果

Mobile Link プロジェクトおよび同期モデルが作成されます。

次の手順

「レッスン6:同期モデルの変更」126ページに進みます。

レッスン6:同期モデルの変更

このレッスンでは、新規 Mobile Link プロジェクトの作成時に構築された統合データベースの同期モデルを変更します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118ページ

◆ タスク

- 1. oracle remote schema 同期モデルを右クリックして、[プロパティ]を選択します。
- 2. 最初のフィールドに sync_oracle と入力します。
- 3. 次のタスクを実行します。

- a. [パブリケーション名] フィールドに、sync oracle publication と入力します。
- b. [スクリプトバージョン] フィールドに sync_oracle_scriptversion と入力します。
 パブリケーションは、同期するデータを指定するリモートデータベース上のオブジェクトです。Mobile Link サーバのスクリプトにより、リモートデータベースからアップロードされたデータを統合データベースに適用する方法と、スクリプトバージョンによりスクリプトをグループ化する方法が定義されます。アプリケーションごとに異なるスクリプトバージョンを使用できるため、1 つの Mobile Link サーバを管理するだけで複数のアプリケーションを同期できます。
- c. [適用] をクリックしてから [OK] をクリックします。
- 4. 同期モデルのテーブルごとに、データの同期方向を設定します。

右ウィンドウ枠で [マッピング] タブをクリックし、[マッピング方向] カラムのローを次のように設定します。

- ORDERS と ORDER_ITEMS の各テーブルは、[双方向] (アップロードとダウンロードの 両方) に設定します。
- 残りのテーブルは、[**リモートにのみダウンロード**] に設定します。
- 5. すべての所有者の統合スキーマのロードのために時間がかかるというウィンドウが表示された場合、HR および OE ユーザのデータベーススキーマをロードすることを選択します。
- 6. リモートデータベースにダウンロードされたローを、リモート ID を基準として次のように フィルタします。
 - a. ORDERS テーブルを含むローを選択し、右ウィンドウ枠の下部にある [サブセットのダ ウンロード] タブをクリックします。
 - b. [サブセットのダウンロード] カラムを [カスタム] に変更します。
 - c. download_cursor スクリプトの WHERE 句に制限を追加することで、リモート ID を基準 としてローをフィルタします。これで、リモートデータベースがユニークに識別されま す。

[ダウンロードカーソルの WHERE 句で使用する SQL 式] フィールドに探索条件を入力 します。たとえば、次の SQL スクリプトは、ORDERS テーブルに使用できます。

OE.ORDERS.SALES_REP_ID = {ml s.remote_id}

ダウンロードカーソルスクリプトは、各テーブルからどのカラムとローをリモートデー タベースにダウンロードするかを指定します。探索条件の指定により、1人の営業担当 者(データベースのリモート ID が一致する営業担当者)に関する情報のみをダウンロー ドするようにできます。

- d. [ダウンロード削除サブセット] タブをクリックし、[ダウンロード削除サブセット] を [同 じ] から [なし] に変更します。
- 7. 同期モデルを保存します。

[ファイル] » [保存] をクリックします。

結果

同期モデルが完成して、展開の準備が完了します。

次の手順

「レッスン7:同期モデルの展開」128ページに進みます。

参照

- ●「統合データベースの設定」『Mobile Link サーバ管理』
- ●「Mobile Link サーバのシステムテーブル」『Mobile Link サーバ管理』
- ●「Mobile Link サーバシステムプロシージャ」『Mobile Link サーバ管理』
- ●「同期スクリプトの作成」『Mobile Link サーバ管理』
- ●「download_delete_cursor スクリプト」『Mobile Link サーバ管理』
- ●「競合の解決」『Mobile Link サーバ管理』
- ●「競合解決」『Mobile Link サーバ管理』
- ●「パブリケーション」『Mobile Link クライアント管理』
- ●「同期イベント」『Mobile Link サーバ管理』
- ●「同期モデルタスク」32ページ
- ●「同期の方法」『Mobile Link サーバ管理』
- ●「ダウンロードタイプの変更」37ページ
- ●「競合の検出と解決の変更」44ページ
- ●「テーブルとカラムのマッピング」33ページ

レッスン7:同期モデルの展開

同期モデル展開ウィザードを使用すると、統合データベースとリモートデータベースを展開できます。各データベースの展開は1つずつ行うこともできますが、両方一度に行うこともできます。同期モデル展開ウィザードでは、展開のオプションを順を追って設定できます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118ページ

◆タスク

- 1. Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、oracle_project、[同期モデル]、 sync_oracle の順に展開します。
- 2. [ファイル] » [展開] をクリックします。

同期モデル展開ウィザードが表示されます。

- 3. [ウィザードによって生成されたファイルを保管するフォルダを選択します。] フィールドの デフォルト設定を使用し、[次へ] をクリックします。
- 4. [クライアントネットワークオプション] ページのデフォルト設定をそのまま使用し、[次へ] をクリックします。
- 5. [どのような Mobile Link のユーザとパスワードを使用しますか]の下の [Mobile Link ユーザ とパスワード]で [これらの使用] を選択し、次のタスクを実行します。
 - a. [Mobile Link ユーザ] フィールドに、oracle_remote と入力します。
 - b. [Mobile Link パスワード] フィールドに、oracle pass と入力します。
 - c. [このユーザを統合データベースに登録します。]をオンにします。登録されたユーザに同 期が許可されるので、[次へ] をクリックし、もう一度 [次へ] をクリックします。
- [同期プロファイル] ページで、デフォルト同期プロファイル名を sync_oracle_publication_oracle_remote のまま使用します。[次へ] をクリックします。
- 7. [データベースの同期を準備する方法を選択します] ページで次のオプションを選択します。
 - a. [統合データベースの同期を準備するために作成された SQL スクリプトの処理方法を指 定してください] で [統合データベースに対して実行します。] を選択します。
 - b. [リモートデータベースの同期を準備するために作成された SQL スクリプトの処理方法 を指定してください] で [新しいリモートデータベースに対して実行します。] を選択します。
 - c. [次へ] をクリックします。
- 8. [完了] をクリックします。
- 9. [閉じる] をクリックします。

結果

統合データベースを複数のリモートクライアントと同期するために設定し、1 つのリモートクラ イアントが正常に展開されました。他のリモートクライアントを展開する場合は、もう一度この ウィザードを実行し、新しい Mobile Link ユーザを作成して統合データベースと Mobile Link サー バの展開を終了します。統合データベースと Mobile Link サーバはすでに展開されているので、 実行する必要があるのは、他のリモート同期クライアントを展開することだけです。

次の手順

「レッスン8:サーバとクライアントの起動」130ページに進みます。

参照

- ●「同期モデルの展開」48ページ
- ●「リモートデータベースとしての SQL Anywhere データベースの使用」『Mobile Link クライア ント管理』
- ●「Mobile Link ユーザ」『Mobile Link クライアント管理』

レッスン8:サーバとクライアントの起動

ここまでのレッスンで、ダウンロードカーソルスクリプトを変更して、1人の販売担当者に関する情報をダウンロードしています。このレッスンでは、リモート ID を販売担当者識別子に設定して販売担当者を指定し、Mobile Link 統合データベースとリモートデータベースを起動します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118ページ

内容と備考

デフォルトでは、Mobile Link は、アップロードとダウンロードに対してスナップショットアイ ソレーション/READ COMMITTED 独立性レベルを使用します。Mobile Link サーバがスナップ ショットアイソレーションを最大限有効に利用できるようにするには、Mobile Link サーバが使 用する Oracle アカウントは、Oracle システムビュー GV_\$TRANSACTION にアクセスできる必要 があります。アクセスできない場合には、警告が表示され、ローはダウンロードで失われること があります。

◆ タスク

1. SYSDBA 権限を持つ SYS ユーザとして、Oracle SQL Plus アプリケーションを使用して接続します。コマンドプロンプトで次のコマンドを実行します。

sqlplus SYS/your password for sys as SYSDBA

2. Oracle システムビュー GV_\$TRANSACTION へのアクセスを許可するには、次の文を実行します。

GRANT SELECT ON SYS.GV_\$TRANSACTION TO OE;

Oracle システムビュー V\$SESSION および GV_\$SESSION へのアクセスを許可するには、次の文を実行します。

GRANT SELECT ON SYS.V \$SESSION TO OE; GRANT SELECT ON SYS.GV_\$SESSION TO OE;

4. 他のシステムオブジェクトへのアクセスを許可するには、次の文を実行します。

GRANT SELECT ON SYS.GV_\$LOCK TO OE; GRANT EXECUTE ON SYS.DBMS_UTILITY TO OE; GRANT SELECT ON DBA_OBJECTS TO OE;

5. コマンドプロンプトで、同期モデルを作成したディレクトリに移動します。(このフォルダ は、**|同期モデル作成ウィザード|**の最初の手順で選択したルートディレクトリです。)

所定のディレクトリ名を使用している場合は、次のディレクトリに移動します。*mlorc ¥oracle_project¥sync_oracle_deploy¥* 6. Mobile Link サーバを起動するには、次のコマンドを実行します。

mlsrv.bat "DSN=oracle cons;UID=OE;PWD=sql"

- *mlsrv.bat* Mobile Link サーバを起動するために作成されたコマンドファイル。
- **DSN** ODBC データソース名。
- UID 統合データベースへの接続に使用するユーザ名。
- PWD 統合データベースへの接続に使用するパスワード。

このコマンドが正常に実行されると、Mobile Link サーバメッセージウィンドウに [Mobile Link サーバが起動しました。] というメッセージが表示されます。

Mobile Link サーバが起動しなかった場合は、統合データベースの接続情報を確認します。

 コマンドプロンプトで、同期モデル展開ウィザードによりリモートデータベースを作成した ディレクトリに移動します。

所定のディレクトリ名を使用している場合は、次のディレクトリに移動します。*mlora ¥oracle_project¥sync_oracle_deploy¥*

8. 次のコマンドを実行して、SQL Anywhere リモートデータベースを起動します。

dbsrv16 -n remote_eng sync_oracle_remote.db -n remote_db

● dbsrv16 SQL Anywhere データベースの起動に使用するデータベースサーバ。

- remote_eng データベースサーバ名。
- sync_oracle_remote.db remote_eng で起動するデータベースファイル。
- remote_db remote_eng にあるデータベースの名前。

結果

このコマンドが正常に実行されると、remote_engという名前の SQL Anywhere データベースサー バが起動し、remote db という名前のデータベースがロードされます。

次の手順

「レッスン9:リモート ID の設定」131ページに進みます。

参照

- ●「SQL Anywhere データベースサーバの構文」『SQL Anywhere サーバ データベース管理』
- ●「展開した同期モデル」52ページ
- ●「Mobile Link サーバ」『Mobile Link サーバ管理』
- ●「リモート ID」『Mobile Link クライアント管理』

レッスン 9:リモート ID の設定

リモートスキーマでは、各リモートデータベースは1人の販売担当者を表しています。作成した 同期スクリプトに含まれている論理により、Mobile Link サーバはリモートデータベースのリ モート ID に基づいてデータのサブセットをダウンロードします。リモートデバイスが最初に同 期するとき、選択した販売担当者に関するすべての情報がダウンロードされるため、最初の同期 の前に、データベースのリモート ID を有効な販売担当者識別子の値に設定する必要があります。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118ページ

◆ タスク

- 1. 有効な販売担当者識別子を選択します。
 - a. SYSDBA 権限を持つ SYS ユーザとして、Oracle SQL Plus アプリケーションを使用して接続します。コマンドプロンプトで次のコマンドを実行します。

sqlplus SYS/your-password-for-sys as SYSDBA

b. ORDERS テーブルで有効な販売担当者識別子のリストを表示するには、次の文を実行します。

SELECT COUNT(SALES_REP_ID), SALES_REP_ID FROM OE.ORDERS GROUP BY SALES_REP_ID;

この例では、リモートデータベースは SALES_REP_ID が 154 である販売担当者を表しています。

c. Oracle SQL Plus を終了するには、次のコマンドを実行します。

exit

2. データベースのリモート ID の値を 154 に設定するには、次のコマンドを実行します。

dbisql -c "SERVER=remote_eng;DBN=remote_db;UID=DBA;PWD=sql" "SET OPTION PUBLIC.ml remote id='154';"

- dbisql SQL Anywhere データベースに対して SQL コマンドを実行するためのアプリ ケーション。
- ENG データベースサーバ名として remote eng を指定します。
- DBN データベース名として remote db を指定します。
- UID リモートデータベースへの接続に使用するユーザ名。
- PWD リモートデータベースへの接続に使用するパスワード。
- SET OPTION PUBLIC.ml_remote_id='154' リモート ID を 154 に設定するための SQL 文。

結果

データベースのリモート ID が有効な販売担当者識別子の値に設定されます。

次の手順

「レッスン 10 : リモートクライアントの同期」133 ページに進みます。

レッスン 10: リモートクライアントの同期

このレッスンでは、リモートクライアントを初めて同期します。この作業は dbmlsync ユーティ リティで行います。dbmlsync はリモートデータベースに接続して Mobile Link サーバにより認 証されると、リモートデータベースのパブリケーションに基づいてリモートデータベースと統合 データベースの同期に必要なすべてのアップロードとダウンロードを実行します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118 ページ

◆タスク

- コマンドプロンプトで次のコマンドを実行します。
 - dbmlsync -c "SERVER=remote_eng;DBN=remote_db;UID=DBA;PWD=sql" -n sync_oracle_publication -u oracle remote -mp oracle_pass
 - dbmlsync 同期アプリケーション。
 - SERVER リモートデータベースサーバ名を指定。
 - DBN リモートデータベース名を指定。
 - UID リモートデータベースへの接続に使用するユーザ名を指定します。
 - PWD リモートデータベースへの接続に使用するパスワードを指定します。
 - sync_oracle_publication 同期の実行時に使用するリモートデバイスのパブリケーション。(このパブリケーションは[同期モデル作成ウィザード] で作成されています。)
 - oracle remote Mobile Link サーバによる認証に使用するユーザ名。
 - oracle_pass Mobile Link サーバによる認証に使用するパスワード。

注意

別のコンピュータにある dbmlsync アプリケーションを Mobile Link サーバから実行している 場合、Mobile Link サーバのロケーションを指定する引数を渡す必要があります。

結果

SQL Anywhere Mobile Link クライアントのメッセージウィンドウに同期の進行状況が表示され ます。このコマンドが正常に実行されると、dbmlsync アプリケーションによりリモートデータ ベースに統合データベースの情報のサブセットが格納されます。

同期が失敗した場合は、dbmlsync アプリケーションに渡す接続情報、および Mobile Link ユーザ 名とパスワードを確認します。それでも失敗する場合は、使用したパブリケーション名を確認 し、統合データベースと Mobile Link サーバが実行中であることを確認します。また、同期ログ (サーバ、クライアントとも)の内容を確認することもできます。

次の手順

「レッスン11:リモートデータベースのデータの表示」134ページに進みます。

参照

- ●「同期処理」13ページ
- ●「dbmlsync 構文」『Mobile Link クライアント管理』

レッスン11:リモートデータベースのデータの表示

Mobile Link サーバを使用してリモートクライアントを正常に統合データベースに同期すると、 リモートデータベースには1人の販売担当者に関する情報が格納されます。SQL Anywhere 16 プラグインを使用すると、Sybase Central でデータベースにデータが正しく移植されているかど うかを確認できます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118 ページ

◆ タスク

- 1. Sybase Central を起動します。
- 2. 次の手順でリモートデータベースに接続します。
 - a. 左ウィンドウ枠で [SQL Anywhere 16] を右クリックして [接続] をクリックします。
 - b. [認証] ドロップダウンリストで [データベース] をクリックし、[ユーザ ID] に DBA と入力し、[パスワード] に sql と入力します。
 - c. [アクション] ドロップダウンリストで、[このコンピュータで稼働しているデータベース に接続] をクリックします。[サーバ名] に remote_eng と入力し、[データベース名] に remote db と入力します。

- d. [接続] をクリックします。
- 3. Sybase Central の左ウィンドウ枠の [remote_db DBA] で、[テーブル] を展開し、[ORDERS テーブル] をクリックし、右ウィンドウ枠の [データ] タブをクリックします。

ORDERS テーブルでは、すべてのレコードが識別子 154 の販売担当者に関するものです。この販売担当者は、他の販売担当者の販売情報とは関係ありません。このため、リモート ID を 基準にしてローをフィルタ処理で除外するよう同期スクリプトを設定し、このデータベースのリモート ID を特定の販売担当者識別子の値に設定します。この販売担当者のデータベースの容量は小さくなり、同期に必要な時間も短くなります。リモートデータベースのサイズは最小限に抑えられているため、新しい販売記録の入力や過去の販売に対する払い戻し処理 などの頻繁に行われる処理が迅速かつ効率的に実行されます。

結果

リモートデータベースに、識別子154の販売担当員に関する情報のみが取り込まれます。

次の手順

「クリーンアップ」135ページに進みます。

クリーンアップ

注文エントリのデータベースを再生成して、チュートリアルのすべての教材をコンピュータから 削除します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」119ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。「チュートリアル: Oracle Database 10g での Mobile Link の使用」118ページ

◆タスク

- Oracle サンプルデータベースを再作成するには、http://www.oracle.com/technetwork/testcontent/ obeconnection-099425.html で説明する手順に従ってください。
- 2. Mobile Link プロジェクトを削除します。
 - a. Sybase Central を起動します。
 - b. 右ウィンドウ枠で、[Mobile Link 16] をダブルクリックします。
 - c. oracle_project が右ウィンドウ枠に表示されます。
 - d. oracle_project を右クリックして、[削除] をクリックします。
 - e. [プロジェクトの削除] ウィンドウで、[リストとコンピュータから削除] を選択し、[は い] をクリックします。

結果

注文エントリのデータベースが再生成され、チュートリアルのすべての教材がコンピュータから 削除されます。

次の手順

なし。

チュートリアル : Adaptive Server Enterprise 統合デー タベースと Mobile Link の使用

このチュートリアルでは、Mobile Link を使用して Adaptive Server Enterprise データベースを活用 する方法について説明します。ここでは、Adaptive Server Enterprise 統合データベースと SQL Anywhere リモートデータベースの間の同期を設定します。また、Ultra Light クライアントも使 用できます。

このチュートリアルでは、書店チェーンのデータを活用することを目的とします。このシナリオ に登場する各書店は、リモート同期環境です。各書店にはローカル SQL Anywhere データベース があり、本社の Adaptive Server Enterprise データベースと同期しています。各書店には、リモー トデータベースのデータにアクセスして操作できるコンピュータを複数設置することもできま す。

このチュートリアルでは、pubs2 サンプルスキーマが Adaptive Server Enterprise 15.7 サーバにイン ストールされていることを前提としています。pubs2 サンプルスキーマは Adaptive Server Enterprise 15.0 に付属しており、オプションとしてインストールされます。このチュートリアル では、統合データベースとして使用します。このサンプルに関する情報は、Adaptive Server Enterprise のマニュアルに記載されています。また、オンライン (http://infocenter.sybase.com/help/ topic/com.sybase.infocenter.dc31654.1570/html/sag1/X21150.htm) でも参照できます。

このチュートリアルでは、デフォルトの sa アカウントを使用します。Adaptive Server Enterprise のインストール時点では、sa アカウントのパスワードは NULL です。このチュートリアルでは、 NULL のパスワードが有効なパスワードに変更されていることを前提としています。Adaptive Server Enterprise での NULL のパスワード変更の詳細については、http://infocenter.sybase.com/help/ topic/com.sybase.infocenter.dc36272.1570/html/commands/CHDGIFDG.htm を参照してください。

必要なソフトウェア

•SQL Anywhere 16

• Adaptive Server Enterprise 15.7

概要

このチュートリアルでは、次の作業の方法について説明します。

●リモートスキーマの設計時に、リモートテーブルの同期方向などの重要な考慮事項を決定す る。

- ●統合データベースとリモートデータベースにユニークなプライマリキーを追加する。
- ●Mobile Link を Adaptive Server Enterprise データベースに接続する ODBC データソースを作成 する。
- ●[同期モデル作成ウィザード]を使用して、統合データベースとリモートデータベースの間の同期を設定する。

●Sybase Central を使用して同期設定をカスタマイズする。

●同期モデル展開ウィザードを使用して統合データベースとリモートデータベースを展開する。

●リモートクライアントを統合データベースと同期する。

パーミッションと権限

統合データベースで次のロールおよび権限を持つ必要があります。

●MASTER..SYSTRANSACTIONS と MASTER..SYSPROCESSES への SELECT パーミッション

SQL Anywhere リモートで次のロールおよび権限を持つ必要があります。

●SYS_REPLICATION_ADMIN_ROLE システムロール

●SYS_RUN_REPLICATION_ROLE システムロール

参照

●「Mobile Link 同期」1 ページ

レッスン1:スキーマの設計

pubs2 サンプルスキーマは統合データベーススキーマとして使用します。このスキーマには書店、タイトル、著者、出版社、販売に関する情報が格納されています。次の表は、Adaptive Server Enterprise データベースの各テーブルの説明です。

テーブル	説明
au_pix	著者の写真。
authors	システムに登録されている各タイトルの著者。
discounts	特定の書店で行われている各種の割引の記録。
sales	各販売レコードは、特定の書店での1件の販売を表しま す。
salesdetail	1件の販売で対象となった各タイトルに関する情報。

テーブル	説明
stores	各店舗レコードは、システムに登録されている1軒の書 店または支店を表しています。
titleauthor	どのタイトルがどの著者によって執筆されたかに関す る情報。
titles	システムに登録されているすべての本の記録。
blurbs, publishers, roysched	このチュートリアルでは必要ない情報。

リモートスキーマの設計

書店ごとに統合データベース全体をコピーしておくことは不要であり、非効率的です。リモート スキーマでは同じテーブル名を使用しますが、各書店に関する情報だけが格納されます。この設 定を実現するため、リモートスキーマは統合データベースのサブセットとして次のように設計さ れています。

統合テーブル	リモートテーブル
au_pix	すべての行を抽出。
authors	すべての行を抽出。
discounts	stor_id を基準にフィルタ処理。
sales	stor_id を基準にフィルタ処理。
salesdetail	stor_id を基準にフィルタ処理。
stores	stor_id を基準にフィルタ処理。
titleauthor	すべての行を抽出。
titles	すべての行を抽出。
blurbs	リモートでは使用しない。
publishers	リモートでは使用しない。
roysched	リモートでは使用しない。

各書店では、すべてのタイトルと著者の記録を保持し、顧客が書店の在庫を検索できるようにす る必要があります。一方で、出版社や印税に関する情報は書店では必要ないため、これらの情報 は各書店に対しては同期されません。各書店では販売と割引に関する情報が必要ですが、他の書 店の販売や割引に関する情報は必要ありません。そのため、ローは書店の識別子に基づいてフィ ルタされます。
注意

リモートデータベースで不要になるカラムがある場合は、テーブルからカラムのサブセットを取得することもできます。

次に、各テーブルの同期の方向を選択します。リモートデータベースで読み込む情報と、リモー トデータベースで作成、変更、または削除する情報について考慮する必要があります。この例で は、書店は著者とタイトルのリストにアクセスする必要がありますが、新しい著者名をシステム に入力することはありません。このため、著者とタイトルは必ず本社の統合データベースから入 力するという制限が適用されています。一方で、書店では新しい販売情報を常時記録できるよう にする必要があります。これらの要因から、各テーブルの同期の方向は次のようになります。

テーブル	同期
titleauthor	リモートデータベースへのダウンロードの み。
authors	リモートデータベースへのダウンロードの み。
au_pix	リモートデータベースへのダウンロードの み。
titles	リモートデータベースへのダウンロードの み。
stores	リモートデータベースへのダウンロードの み。
discounts	リモートデータベースへのダウンロードの み。
sales	ダウンロードとアップロード。
salesdetail	ダウンロードとアップロード。

「レッスン2:統合データベースの準備」139ページに進みます。

レッスン2:統合データベースの準備

このレッスンでは、Mobile Link 同期用の統合データベースのサイズを増やし、ユニークなプラ イマリキーを作成します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」137ページを参照してください。 このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

内容と備考

Mobile Link では、同期のためにシステムテーブルなどのオブジェクトを pubs2 データベースに 追加する必要があります。これらのオブジェクトを追加する場合は、pubs2 データベースのサイ ズを増やす必要があります。

◆タスク

1. Adaptive Server Enterprise の isql を使用して、pubs2 データベースに sa として接続します。コ マンドプロンプトで、次のコマンドをすべて1行に入力して実行します。

```
isql
-U sa
-P your-password-for-sa-account
-D pubs2
```

Adaptive Server Enterprise にリモートでアクセスしている場合は、-S オプションでサーバ名を 指定します。

2. データベースのサイズを増やすための所定のパーミッションを得るには、master データベー スにアクセスする必要があります。isql で次のコマンドを実行します。

```
use master
go
sp_dboption pubs2, "select into/bulkcopy/pllsort", true
go
```

Adaptive Server Enterprise では、データベースはディスクまたはディスクの一部に保存されます。pubs2 データベースのサイズを増やすには、次の文を実行します (pubs2 が格納されているディスクを指定する必要があります)。

ALTER DATABASE pubs2 ON disk-name = 33

結果

結果

次の手順

「レッスン3:ユニークなキーの統合データベースへの追加」140ページに進みます。

参照

●「Adaptive Server Enterprise 統合データベース」『Mobile Link サーバ管理』

レッスン3:ユニークなキーの統合データベースへの追加

このレッスンでは、ユニークなプライマリキーを統合データベースに追加します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

内容と備考

同期システムでは、テーブルのプライマリキーは、異なるデータベース内の同じローを識別する 唯一の方法であり、競合を検出する唯一の方法です。使用する各テーブルには、プライマリキー が必要です。プライマリキーが更新されることはありません。また、1つのデータベースに挿入 されたプライマリキーの値が別のデータベースに挿入されないようにする必要があります。

ユニークなプライマリキーを生成する方法は複数あります。このチュートリアルでは、簡単に操作を行うため、複合プライマリキー方式を使用します。この方式では、統合データベースとリ モートデータベースにまたがってユニークな複数のカラムを使用してプライマリキーを作成し ます。

◆タスク

1. Adaptive Server Enterprise の isql を使用して、pubs2 データベースに sa として接続します。コ マンドプロンプトで、次のコマンドをすべて1行に入力して実行します。

isql -U sa -P your-password-for-sa-account -D pubs2

Adaptive Server Enterprise にリモートでアクセスしている場合は、-S オプションでサーバ名を 指定します。

2. 次のローは、salesdetail テーブルに対して作成した複合プライマリキーを基準とするとユニー クではありません。操作を簡単にするために、次の文を実行してこのローを削除します。

DELETE FROM salesdetail WHERE stor_id = '5023' AND ord_num = 'NF-123-ADS-642-9G3' AND title_id = 'PC8888'

DELETE FROM salesdetail WHERE stor_id = '5023' AND ord_num = 'ZS-645-CAT-415-1B2' AND title_id = 'BU2075'

3. 次のインデックスは、前の手順でのプライマリキーの作成に干渉しています。このインデッ クスを削除するには、次の文を実行します。

DROP INDEX authors.auidind DROP INDEX titleauthor.taind DROP INDEX titles.titleidind DROP INDEX sales.salesind 4. 次の文を実行して、ユニークなプライマリキーを追加します。

ALTER TABLE au_pix ADD PRIMARY KEY (au_id) ALTER TABLE authors ADD PRIMARY KEY (au_id) ALTER TABLE titleauthor ADD PRIMARY KEY (au_id, title_id) ALTER TABLE titles ADD PRIMARY KEY (title_id) ALTER TABLE discounts ADD PRIMARY KEY (discountype) ALTER TABLE stores ADD PRIMARY KEY (stor_id) ALTER TABLE sales ADD PRIMARY KEY (stor_id, ord_num) ALTER TABLE salesdetail ADD PRIMARY KEY (stor_id, ord_num, title_id)

これらの文を実行した後、Mobile Link サーバでは統合データベースに接続し、任意の数のリモートデータベースとの同期を設定できるようになります。

注意

プライマリキーがない統合データベースとデータを同期することも可能です。ただし、他の テーブルでローを一意に識別するよう設計されたシャドウテーブルで機能する同期イベント を自分で作成する必要があります。

あとのレッスンでは、統合スキーマからリモートスキーマを作成するため、リモートスキー マは統合スキーマと同じプライマリキーを持つことになります。

プライマリキーがすべてのデータベースに対してユニークになるようにカラムが選択されて います。sales テーブルでは、プライマリキーは stor_id と ord_num カラムで構成されていま す。リモートデータベースの sales テーブルに挿入されるすべての値には、ユニークな注文番 号が必要です (stor_id の値は常に同じです)。これにより、リモートの各 sales テーブルで一意 性が確保されます。統合データベースの sales テーブルのプライマリキーは、複数の書店で データがアップロードされた場合の競合を防止する役割があります。書店ごとに stor_id の 値が異なるため、1 軒の書店からのアップロードはいずれも他の書店に対してユニークです。

salesdetail テーブルでは、プライマリキーは stor_id、ord_num、title_id の各カラムで構成され ています。1 件の注文に複数の本のタイトルが存在する場合もあります。リモートデータ ベースの sales テーブルでは、stor_id と ord_num については複数のローで同じ値になってもか まいませんが、title_id の値はローごとに異なる必要があります。この設定により、各リモー トデータベースの salesdetail テーブルで一意性が確保されます。sales テーブルと同様に、書 店ごとに stor_id の値が異なるため、1 軒の書店から統合データベースへのアップロードはい ずれも他の書店に対してユニークです。

結果

ユニークでないローは削除され、ユニークなプライマリキーが統合データベースに追加されま す。

次の手順

「レッスン4: Mobile Linkの接続」143ページに進みます。

参照

●「ユニークなプライマリキー」『Mobile Link サーバ管理』

レッスン4: Mobile Link の接続

このレッスンでは、Mobile Link を統合データベースに接続する ODBC データソースを作成します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

◆ タスク

1. ODBC データソースを作成します。

Adaptive Server Enterprise に付属の ODBC ドライバを使用する必要があります。このチュートリアルでは、次の設定を使用します。

[一般] タブのフィールド	値
データソース名	ase_cons
説明	
サーバ名 (ASE ホスト名)	localhost
サーバポート	5000
データベース名	pubs2
ログイン ID	sa
カーソルの使用	Off
[トランザクション] ダブのフィールト	
┃ ┃ サーバ初期化トランザクション	オフ

- 2. ODBC 接続をテストします。
 - a. [一般] タブで [テスト接続] をクリックします。 Adaptive Server Enterprise のログオン画面が表示されます。
 - b. sa アカウントのパスワードを入力し、[OK] をクリックします。 [ログインに成功しました] というメッセージが表示されます。

結果

ODBC データソースが作成され、テストされます。

次の手順

「レッスン5: Mobile Link プロジェクトと同期モデルの作成」144ページに進みます。

ODBC データソースを設定すると、Mobile Link 16 プラグインを使用して統合データベースに接続し、同期モデルを作成することができるようになります。

参照

http://www.sybase.com/detail?id=1011880

レッスン 5: Mobile Link プロジェクトと同期モデルの作成

このレッスンでは、新しい Mobile Link プロジェクトを作成して、統合データベースに接続しま す。新しいプロジェクトを作成すると、同期モデルが自動的に作成されます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

◆タスク

- 1. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [Sybase Central] をクリッ クします。
- 2. [ツール] » [Mobile Link 16] » [新しいプロジェクト] をクリックします。

[プロジェクト作成ウィザード] が表示されます。

- 3. [新しいプロジェクトの名前を指定してください。] フィールドに ase project と入力します。
- 4. [ロケーション] フィールドに C:¥mlase と入力し、[次へ] をクリックします。
- 5. [データベースの表示名] フィールドに ase cons と入力します。
- 6. [編集] をクリックします。
- 7. [汎用 ODBC データベースに接続] ページで次のタスクを実行します。
 - a. [ユーザ ID] フィールドに sa と入力します。
 - b. [パスワード] フィールドに sa アカウントのパスワードを入力します。

- c. [ODBC データソース名] フィールドで、[参照] をクリックして ase cons を選択します。
- d. [OK] をクリックし、[保存] をクリックします。
- 8. [パスワードを記憶] オプションを選択し、[次へ] をクリックします。
- 9. [新しいリモートデータベーススキーマ] ページの [リモートデータベースに含める統合デー タベースのテーブルとカラムを指定してください。] リストで、次のテーブルを選択します。
 - au_pix
 authors
 discounts
 sales
 salesdetail
 stores
 titleauthor
 - titles

[次へ] をクリックします。

- 10. [リモートスキーマ名をプロジェクトに追加] オプションを選択します。
- 11. リモートスキーマ名に ase remote schema と入力し、[次へ] をクリックします。
- 12. [SQL Anywhere] をオンにして、[完了] をクリックします。

Mobile Link が今回初めて統合データベースを使用する場合、Mobile Link システム設定をイン ストールするかどうかを確認するメッセージが表示されます。Mobile Link システム設定を インストールすると、Mobile Link システムテーブルと Mobile Link システムプロシージャが 追加されます。[はい] をクリックし、[OK] をクリックします。

結果

Mobile Link プロジェクトおよび同期モデルが作成されます。

次の手順

「レッスン6:同期モデルの変更」145ページに進みます。

レッスン 6:同期モデルの変更

このレッスンでは、同期モデルを変更します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

◆ タスク

- 1. ase remote schema 同期モデルを右クリックして、[プロパティ]を選択します。
- 2. 次のタスクを実行します。
 - a. 最初のフィールドに、sync_ase と入力します。
 - b. [パブリケーション名] フィールドに、sync_ase_publication と入力します。
 - c. [スクリプトバージョン] フィールドに sync_ase_scriptversion と入力します。
 - パブリケーションは、同期するデータを指定するリモートデータベース上のオブジェク トです。Mobile Link サーバのスクリプトにより、リモートデータベースからアップロー ドされたデータを統合データベースに適用する方法と、スクリプトバージョンによりス クリプトをグループ化する方法が定義されます。アプリケーションごとに異なるスクリ プトバージョンを使用できるため、1 つの Mobile Link サーバを管理するだけで複数のア プリケーションを同期できます。
 - d. [適用] をクリックしてから [OK] をクリックします。
- 3. 同期モデルのテーブルごとに、データの同期方向を設定します。

右ウィンドウ枠で [マッピング] タブをクリックし、[マッピング方向] カラムのローを次のように設定します。

- sales と salesdetail の各テーブルは、[双方向] (アップロードとダウンロードの両方) に設定 します。
- 残りのテーブルは、[**リモートにのみダウンロード**] に設定します。
- 4. リモートデータベースにダウンロードされたローを、リモート ID を基準として次のように フィルタします。
 - a. stores テーブルが含まれるローを選択し、[サブセットのダウンロード] タブをクリックします。
 - b. [サブセットのダウンロード] を [カスタム] に変更します。
 - c. download_cursor スクリプトの WHERE 句に制限を追加することで、リモート ID を基準 としてローをフィルタします。これで、リモートデータベースがユニークに識別されま す。

[ダウンロードカーソルの WHERE 句で使用する SQL 式] フィールドに探索条件を入力 します。たとえば、次の SQL スクリプトは、stores テーブルに使用できます。

"dbo"."stores"."stor_id" = {ml s.remote_id}

ダウンロードカーソルスクリプトは、各テーブルからどのカラムとローをリモートデー タベースにダウンロードするかを指定します。探索条件の指定により、1 つの書店 (デー タベースのリモート ID が一致する書店) に関する情報のみをダウンロードするようにで きます。

- d. [ダウンロード削除サブセット] タブをクリックし、[ダウンロード削除サブセット] を [同 じ] から [なし] に変更します。
- 5. sales、salesdetail、discountsの各テーブルを含むローについて、前の手順を繰り返します。

注意

SQL スクリプトで指定されたテーブルの名前は、編集するロー内のテーブル名に変更してください。

sales テーブルには、次の WHERE 句スクリプトを使用します。

"dbo"."sales"."stor_id" = {ml s.remote_id}

salesdetail テーブルには、次の WHERE 句スクリプトを使用します。

"dbo"."salesdetail"."stor_id" = {ml s.remote_id}

discounts テーブルには、次の WHERE 句スクリプトを使用します。

"dbo"."discounts"."stor_id" = {ml s.remote_id}

6. 同期モデルを保存します。

[ファイル]»[保存]をクリックします。

結果

同期モデルが完成して、展開の準備が完了します。

次の手順

「レッスン7:同期モデルの展開」147ページに進みます。

参照

- ●「統合データベースの設定」『Mobile Link サーバ管理』
- ●「Mobile Link サーバのシステムテーブル」『Mobile Link サーバ管理』
- ●「Mobile Link サーバシステムプロシージャ」『Mobile Link サーバ管理』
- ●「download_delete_cursor スクリプト」『Mobile Link サーバ管理』
- ●「競合の解決」『Mobile Link サーバ管理』
- ●「競合解決」『Mobile Link サーバ管理』
- ●「パブリケーション」『Mobile Link クライアント管理』
- ●「同期モデルタスク」32ページ
- ●「ダウンロードタイプの変更」37ページ
- ●「競合の検出と解決の変更」44ページ
- ●「テーブルとカラムのマッピング」33ページ

レッスン7:同期モデルの展開

同期モデル展開ウィザードを使用すると、統合データベースとリモートデータベースを展開できます。これらのデータベースの展開は1つずつ行うこともできますが、両方一度に行うこともできます。同期モデル展開ウィザードでは、展開のオプションを順を追って設定できます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

◆タスク

- 1. Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、ase_project、[同期モデル]、sync_ase の順に展開します。
- 2. [ファイル]»[展開]をクリックします。
- 3. [ウィザードによって生成されたファイルを保管するフォルダを選択します。] フィールドの デフォルト設定を使用し、[次へ] をクリックします。
- 4. [クライアントネットワークオプション] ページのデフォルト設定をそのまま使用し、[次へ] をクリックします。
- 5. [どのような Mobile Link のユーザとパスワードを使用しますか]の下の [Mobile Link ユーザ とパスワード] で [これらの使用] を選択し、次のタスクを実行します。

●[Mobile Link ユーザ] フィールドに、ase remote と入力します。

●[Mobile Link パスワード] フィールドに、ase pass と入力します。

●[このユーザを統合データベースに登録します。]をオンにします。登録されたユーザに同 期が許可されるので、[次へ] をクリックします。 [次へ] をクリックしてから、もう一度 [次へ] をクリックします。

- 6. [同期プロファイル] ページで、[同期プロファイル名] フィールドに sync_ase_profile と入力 し、[次へ] をクリックします。
- 7. [データベースの同期を準備する方法を選択します] ページで次のタスクを実行します。
 - a. [統合データベースの同期を準備するために作成された SQL スクリプトの処理方法を指 定してください] で [統合データベースに対して実行します。] を選択します。
 - b. [リモートデータベースの同期を準備するために作成された SQL スクリプトの処理方法 を指定してください] で [新しいリモートデータベースに対して実行します。] を選択しま す。[次へ] をクリックします。
- 8. [完了] をクリックします。
- 9. [閉じる] をクリックします。

結果

統合データベースを複数のリモートクライアントと同期するための設定がすべて完了し、1つの リモートクライアントが正常に展開されました。他のリモートクライアントを展開する場合は、 もう一度このウィザードを実行し、新しい Mobile Link ユーザを作成して統合データベースと Mobile Link サーバの展開を終了します。これらのものについては展開がすでに終了しているた め、あとは他のリモート同期クライアントを展開するだけです。

次の手順

「レッスン8:サーバとクライアントの起動」149ページに進みます。

参照

- ●「同期モデルの展開」48ページ
- ●「リモートデータベースとしての SQL Anywhere データベースの使用」『Mobile Link クライア ント管理』
- ●「Mobile Link ユーザ」『Mobile Link クライアント管理』

レッスン8:サーバとクライアントの起動

このレッスンでは、Mobile Link サーバとリモートデータベースを起動します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136ページ

内容と備考

ここまでのレッスンで、ダウンロードカーソルスクリプトを変更して、1 軒の書店に関する情報 をダウンロードしています。このレッスンでは、リモート ID を書店識別子に設定して、書店を 指定します。

◆タスク

1. コマンドプロンプトで、同期モデルを作成したフォルダに移動します。(このフォルダは、[同 期モデル作成ウィザード]の最初の手順で選択したルートディレクトリです。)

所定のディレクトリ名を使用している場合は、次のディレクトリに移動します。*mlase ¥ase_project¥sync_ase_deploy¥*.

2. Mobile Link サーバを起動するには、次のコマンドを実行します。

mlsrv.bat "DSN=ase_cons;UID=sa;PWD=sa;"

- *mlsrv.bat* Mobile Link サーバを起動するためのコマンドファイル。
- dsn ODBC データソース名。
- uid 統合データベースへの接続に使用するユーザ名 (Adaptive Server Enterprise の場合、 デフォルトは sa)。

● pwd sa として接続するためのパスワード。

このコマンドが正常に実行されると、Mobile Link サーバメッセージウィンドウに [Mobile Link サーバが起動しました。] というメッセージが表示されます。

Mobile Link サーバが起動しなかった場合は、統合データベースの接続情報を確認します。

3. コマンドプロンプトで、同期モデル展開ウィザードによりリモートデータベースを作成した ディレクトリに移動します。

所定のディレクトリ名を使用している場合は、次のディレクトリに移動します。*mlase ¥ase project¥sync ase deploy¥*.

4. SOL Anywhere リモートデータベースを起動するには、次のコマンドを入力します。

dbsrv16 -n remote_eng sync_ase_remote.db -n remote_db

- dbsrv16 SQL Anywhere データベースの起動に使用するデータベースサーバ。
- remote_eng データベースサーバ名。
- sync_ase_remote.db remote_eng で起動するデータベースファイル。
- **remote_db** remote eng にあるデータベースの名前。

結果

remote_eng という名前の SQL Anywhere データベースサーバが起動し、remote_db と呼ばれる データベースをロードします。

次の手順

「レッスン9:リモート ID の設定」150ページに進みます。

参照

- ●「SQL Anywhere データベースサーバの構文」『SQL Anywhere サーバ データベース管理』
- ●「展開した同期モデル」52ページ
- ●「Mobile Link サーバ」『Mobile Link サーバ管理』

レッスン 9:リモート ID の設定

リモートスキーマでは、各リモートデータベースは1軒の書店を表しています。作成した同期ス クリプトに含まれている論理により、Mobile Link サーバはリモートデータベースのリモート ID に基づいてデータのサブセットをダウンロードします。データベースのリモート ID は有効な書 店識別子の値に設定する必要があります。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

内容と備考

リモートデバイスが最初に同期する際に、書店(ここでは Thoreau Reading Discount Chain)に関す るすべての情報がダウンロードされるため、この手順は最初の同期の前に完了している必要があ ります。

◆タスク

- 1. 有効な書店識別子を選択します。
 - a. Adaptive Server Enterprise の isql を使用して、pubs2 データベースに sa として接続します。 コマンドプロンプトで、次のコマンドをすべて1行に入力して実行します。

```
isql
-U sa
-P your-password-for-sa-account
-D pubs2
```

Adaptive Server Enterprise にリモートでアクセスしている場合は、-S オプションでサーバ 名を指定します。

b. stores テーブルで有効な書店識別子のリストを表示するには、次の文を実行します。

SELECT * FROM stores

このチュートリアルでは、リモートデータベースは Thoreau Reading Discount Chain という名前の書店を表しています。書店識別子は 5023 です。

c. isqlを終了するには、次のコマンドを実行します。

exit

2. データベースのリモート ID を 5023 に設定するには、次のコマンドをすべて1行で入力して 実行します。

dbisql

-c "SERVER=remote_eng;DBN=remote_db;UID=DBA;PWD=sql" "SET OPTION PUBLIC.ml_remote_id='5023'"

- dbisql SQL Anywhere データベースに対して SQL コマンドを実行するためのアプリ ケーション。
- SERVER データベースサーバ名として remote_eng を指定します。
- DBN データベース名として remote db を指定します。
- UID リモートデータベースへの接続に使用するユーザ名を指定します。
- PWD リモートデータベースへの接続に使用するパスワードを指定します。
- SET OPTION PUBLIC.ml_remote_id='5023' リモート ID を 5023 に設定するための SQL コマンド。

結果

データベースのリモート ID が 5023 に設定されます。これは書店識別子の値です。

次の手順

「レッスン10:同期」152ページに進みます。

参照

●「リモート ID」『Mobile Link クライアント管理』

レッスン 10:同期

このレッスンでは、dbmlsync ユーティリティを使用して、リモートクライアントを初めて同期 します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

内容と備考

dbmlsyncは、リモートデータベースに接続し、リモートデータベースから同期情報をロードし、 トランザクションログをスキャンして、アップロードデータを生成します。次に、dbmlsyncは、 Mobile Link サーバに接続し、Mobile Link サーバにより認証されると、リモートデータベースの パブリケーションに基づいてリモートデータベースと統合データベースを同期するのに必要な すべてのアップロードとダウンロードを実行します。

◆ タスク

● コマンドプロンプトで、次のコマンドをすべて1行に入力して実行します。

dbmlsync -c "SERVER=remote_eng;DBN=remote_db;UID=DBA;PWD=sql;" -n sync_ase_publication -u ase_remote -mp ase_pass

- dbmlsync 同期アプリケーション。
- SERVER リモートデータベースサーバ名を指定します。
- DBN リモートデータベース名を指定します。
- UID リモートデータベースへの接続に使用するユーザ名を指定します。
- PWD リモートデータベースへの接続に使用するパスワードを指定します。
- sync_ase_publication 同期の実行に使用するリモートデバイスのパブリケーション名。 (このパブリケーションは [同期モデル作成ウィザード] で作成されています。)

● ase remote Mobile Link サーバによる認証に使用するユーザ名。

● ase_pass Mobile Link サーバによる認証に使用するパスワード。

注意

別のコンピュータにある dbmlsync アプリケーションを Mobile Link サーバから実行している 場合、Mobile Link サーバのロケーションを指定する引数も渡す必要があります。

結果

[SQL Anywhere Mobile Link クライアントのメッセージ] ウィンドウに同期の進行状況が表示されます。このコマンドが正常に実行されると、dbmlsync アプリケーションによりリモートデー タベースに統合データベースの情報のサブセットが格納されます。

同期が失敗した場合は、dbmlsync アプリケーションに渡す接続情報、および Mobile Link ユーザ 名とパスワードを確認します。それでも失敗する場合は、使用したパブリケーション名を確認 し、統合データベースと Mobile Link サーバが実行中であることを確認します。また、同期ログ (サーバ、クライアントとも)の内容を確認することもできます。

次の手順

「レッスン11:リモートデータベースのデータの表示」153ページに進みます。

参照

- ●「同期処理」13ページ
- ●「dbmlsync 構文」『Mobile Link クライアント管理』

レッスン11:リモートデータベースのデータの表示

Mobile Link サーバを使用してリモートクライアントを正常に統合データベースに同期すると、 リモートデータベースには1軒の書店に関する情報が格納されます。SQL Anywhere 16 プラグ インを使用すると、Sybase Central のリモートデータベースの内容を確認できます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

◆ タスク

- 1. Sybase Central を起動します。
- 2. 次の手順でリモートデータベースに接続します。
 - a. 左ウィンドウ枠で [SQL Anywhere 16] を右クリックして [接続] を選択します。

- b. [認証] ドロップダウンリストから [データベース] を選択し、次の手順を実行します。
 i. [ユーザ ID] フィールドに DBA と入力します。
 - ii. [パスワード] フィールドに sql と入力します。
- c. [アクション] ドロップダウンリストから、[このコンピュータで稼働しているデータベー スに接続] を選択します。
- d. [サーバ名] フィールドに remote_eng と入力し、[データベース名] フィールドに remote_db と入力します。
- e. [接続] をクリックします。
- 3. 統合データベースから作成されたテーブルが表示されていない場合は、次の手順を実行します。
 - a. remote db を右クリックして [所有者フィルタの設定] をクリックします。
 - b. [dbo] を選択して [OK] をクリックします。
 統合データベースから作成されたテーブルが左ウィンドウ枠に表示されます。dbo がこれらのテーブルに対して持っている所有権はリモートデータベースで保持されます。
- 4. 任意のリモートテーブルを選択して、右ウィンドウ枠の [データ] タブをクリックします。

sales、salesdetail、storeの各テーブルでは、すべてのレコードが識別子 5023の書店に関する ものです。この書店は、他の書店の販売情報とは関係ありません。このため、リモート ID を 基準にしてローをフィルタ処理で除外するよう同期スクリプトを設定し、このデータベース のリモート ID を特定の書店識別子の値に設定します。この書店のデータベースは容量が少 なく、同期に必要な時間も短くなります。リモートデータベースのサイズは最小限に抑えら れているため、新しい販売記録の入力や過去の販売に対する払い戻し処理などの頻繁に行わ れる処理が迅速かつ効率的に実行されます。

結果

リモートデータベース内のデータが表示されます。

次の手順

「クリーンアップ」154ページに進みます。

クリーンアップ

pubs2 データベースを再生成して、チュートリアルのすべての教材をコンピュータから削除します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:スキーマの設計」137ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「パーミッションと権限」セクションに一覧に なっているロールおよび権限を持っていることを前提としています。 「チュートリアル: Adaptive Server Enterprise 統合データベースと Mobile Link の使用」136 ページ

◆ タスク

1. pubs2 データベースを再生成します。

pubs2 データベースをインストールするスクリプトを実行するには、次のコマンドを実行します。

isql -U sa -P your-password-for-sa-account -i %SYBASE%¥%SYBASE_ASE%¥scripts¥instpbs2

Adaptive Server Enterprise にリモートでアクセスしている場合は、-S オプションでサーバ名を 指定します。また、instpbs2 ファイルをローカルのコンピュータにコピーする必要がありま す。-i オプションを更新して、instpbs2 ファイルの新しいロケーションを指定する必要があり ます。

- 2. 同期モデルを削除します。
 - a. Sybase Central を起動します。
 - b. 右ウィンドウ枠で [Mobile Link 16] をダブルクリックします。
 sync_ase モデルが表示されます。
 - c. sync_ase を右クリックして、[削除] を選択します。
- 3. dberase ユーティリティを使用して、リモートデータベースを消去します。

次のコマンドを実行します。

dberase sync_ase¥remote¥sync_ase_remote.db

結果

pubs2 データベースが再生成され、チュートリアルのすべての教材がコンピュータから削除されます。

次の手順

なし。

チュートリアル : カスタムユーザ認証用の Java と .NET の使用

Mobile Link 同期スクリプトは、SQL、Java、または .NET で作成できます。Java または .NET を 使用すると、同期処理の任意の時点にカスタムアクションを追加できます。 このチュートリアルでは、authenticate_user 接続イベントに対する Java または .NET メソッドを 追加します。authenticate_user イベントを使用すると、カスタム認証スキームを指定して、Mobile Link の組み込みクライアント認証を無効にできます。

必要なソフトウェア

•SQL Anywhere 16

●Java ソフトウェア開発キットまたは Microsoft .NET Framework

前提知識と経験

次の知識と経験が必要です。

●Java または .NET の知識

●Mobile Link イベントスクリプトの基本的な知識

権限

CustDB データベースで次の権限を持つ必要があります。

●SYS_REPLICATION_ADMIN_ROLE システムロール

●CREATE ANY TRIGGER システム権限

●CREATE ANY VIEW システム権限

●EXECUTE ANY PROCEDURE システム権限

概要

- このチュートリアルでは、次の作業の方法について説明します。
- ●Mobile Link サーバ API リファレンスを使用した、ソースファイルのコンパイル
- ●特定のテーブルレベルイベント用のクラスメソッドの指定
- ●-sl オプションを使用した、Mobile Link サーバ (mlsrv16) の実行
- ●サンプル Windows クライアントアプリケーションを使用した、同期のテスト

目的

次の項目について、知識と経験を得ることができます。

●Mobile Link カスタム認証

参照

- ●「ユーザ認証メカニズム」『Mobile Link クライアント管理』
- ●「外部サーバに対する認証」『Mobile Link クライアント管理』
- ●「.NET の同期スクリプト」『Mobile Link サーバ管理』
- ●「Java による同期スクリプトの作成」『Mobile Link サーバ管理』

レッスン1:カスタム認証用の Java クラスの作成 (サーバ側)

このレッスンでは、カスタム認証用の Java 論理を記述するクラスをコンパイルします。

前提条件

Java 同期論理を実行するには、Mobile Link サーバが *mlscript.jar* 内のクラスにアクセスできるこ とが必要です。*mlscript.jar* には、Java メソッドで利用する Mobile Link Java サーバ API クラスの レポジトリが入っています。Java クラスをコンパイルするときは、*mlscript.jar* を参照します。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル : カスタムユーザ認証用の Java と .NET の使用」155 ページ

内容と備考

顧客認証用の .NET クラスを作成するには、「レッスン1:カスタム認証用の .NET クラスの作成 (サーバ側)」158 ページを参照してください。

◆タスク

1. MobiLinkAuth というクラスを作成し、authenticateUser メソッドを作成します。

MobiLinkAuth クラスには、authenticate_user 同期イベントで使用する authenticateUser メソッ ドが含まれています。authenticate_user イベントは、ユーザパラメータとパスワードパラメー タを提供します。認証結果は、authentication status inout パラメータを使用して返します。

注意

authenticate_user 同期イベントの authenticateUser メソッドは、「レッスン2: authenticate_user イベント用の Java または.NET スクリプトの登録」160 ページで登録します。

アプリケーションサーバに次のコードを使用します。

import ianywhere.ml.script.*;

このコードは、簡単なカスタムユーザ認証の例を示します。128 で始まるユーザ名を使用し てクライアントが統合データベースにアクセスすると、認証に成功します。 2. コードを保存します。

このチュートリアルでは、*c*:¥*MLauth*をサーバサイドコンポーネントの作業フォルダとしま す。ファイルを *MobiLinkAuth.java* という名前でこのディレクトリに保存します。

- 3. クラスファイルをコンパイルします
 - a. Java ファイルが含まれるディレクトリに移動します。
 - b. MobiLinkAuth クラスをコンパイルし、Mobile Link サーバ Java API ライブラリを参照します。

次のコマンドを実行して、C:¥Program Files¥SQL Anywhere 16¥を SQL Anywhere 16ディレクトリに置き換えます。

javac MobiLinkAuth.java -classpath "C:¥Program Files¥SQL Anywhere 16¥java¥mlscript.jar"

結果

MobiLinkAuth.class ファイルが生成されます。

次の手順

「レッスン 2: authenticate_user イベント用の Java または .NET スクリプトの登録」160 ページに 進みます。

参照

- ●「authenticate user 接続イベント」『Mobile Link サーバ管理』
- ●「Java と .NET のユーザ認証」『Mobile Link クライアント管理』
- ●「Java 同期の例」『Mobile Link サーバ管理』
- ●「.NET 同期のサンプル」『Mobile Link サーバ管理』

レッスン1:カスタム認証用の.NET クラスの作成 (サーバ側)

このレッスンでは、カスタム認証用の .NET 論理を記述するクラスをコンパイルします。

前提条件

.NET 同期論理を実行するには、Mobile Link サーバが *iAnywhere.MobiLink.Script.dll* 内のクラスに アクセスできることが必要です。*iAnywhere.MobiLink.Script.dll* には、.NET メソッドで利用する Mobile Link .NET サーバ API クラスのレポジトリが入っています。.NET クラスをコンパイルす るときは、*iAnywhere.MobiLink.Script.dll* を参照します。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:カスタムユーザ認証用の Java と.NET の使用」155 ページ

内容と備考

顧客認証用の.NET クラスを作成するには、「レッスン1:カスタム認証用の Java クラスの作成 (サーバ側)」157 ページを参照してください。

◆タスク

1. MobiLinkAuth というクラスを作成し、authenticateUser メソッドを作成します。

MobiLinkAuth クラスには、authenticate_user 同期イベントで使用する authenticateUser メソッ ドが含まれています。authenticate_user イベントは、ユーザパラメータとパスワードパラメー タを提供します。認証結果は、authentication status inout パラメータを使用して返します。

注意

authenticate_user 同期イベントの authenticateUser メソッドは、「レッスン 2 : authenticate_user イベント用の Java または .NET スクリプトの登録」160 ページ で登録します。

```
アプリケーションサーバに次のコードを使用します。
```

```
using iAnywhere.MobiLink.Script;
```

```
public class MobiLinkAuth {
    public void authenticateUser(
        ref int authentication_status,
        string user,
        string pwd,
        string newPwd ) {
            if(user.StartsWith("128")) {
                // success: an auth status code of 1000
                authentication_status = 1000;
            } else {
                // fail: and authentication_status code of 4000
                authentication_status = 4000;
            }
        }
    }
}
```

このコードは、簡単なカスタムユーザ認証の例を示します。128 で始まるユーザ名を使用してクライアントが統合データベースにアクセスすると、認証に成功します。

2. コードを保存します。

このチュートリアルでは、c:¥MLauth をサーバサイドコンポーネントの作業フォルダとしま す。ファイルを MobiLinkAuth.cs という名前でこのディレクトリに保存します。

- 3. クラスファイルをコンパイルします
 - a. C#ファイルが含まれるディレクトリに移動します。
 - b. MobiLinkAuth クラスをコンパイルし、Mobile Link サーバ.NET API ライブラリを参照します。

```
次のコマンドを実行して、C:¥Program Files¥SQL Anywhere 16¥を SQL Anywhere 16ディレクトリに置き換えます。
```

csc /out:MobiLinkAuth.dll /target:library /reference:"C:¥Program Files¥SQL Anywhere 16¥Assembly¥v2¥iAnywhere.MobiLink.Script.dll" MobiLinkAuth.cs

結果

*MobiLinkAuth.dll*ファイルが生成されます。

次の手順

「レッスン 2: authenticate_user イベント用の Java または .NET スクリプトの登録」160 ページに 進みます。

参照

- ●「authenticate user 接続イベント」『Mobile Link サーバ管理』
- ●「Java と.NET のユーザ認証」『Mobile Link クライアント管理』
- ●「Java 同期の例」『Mobile Link サーバ管理』
- ●「.NET 同期のサンプル」『Mobile Link サーバ管理』

レッスン2: authenticate_user イベント用の Java また は.NET スクリプトの登録

SQL Anywhere には、同期できるように設定された SQL Anywhere サンプルデータベース (CustDB) が付属しています。たとえば、CustDB の ULCustomer テーブルは、さまざまなテーブ ルレベルスクリプトをサポートする同期テーブルです。このレッスンでは、authenticate_user 同 期イベント用の MobiLinkAuth authenticateUser メソッドを登録します。このスクリプトは、 Mobile Link サンプルデータベースである CustDB に追加します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:カスタム認証用の Java クラスの作成 (サーバ側)」157 ページを参照してくだ さい。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:カスタムユーザ認証用の Java と.NET の使用」155ページ

内容と備考

CustDB は、Ultra Light クライアントと SQL Anywhere クライアントの両方の統合データベース サーバとなるように設計されています。CustDB 統合データベースは、ODBC データソース SQL Anywhere 16 CustDB を使用します。

◆タスク

1. Interactive SQL からサンプルデータベースに接続します。

次のコマンドを実行します。

dbisql -c "DSN=SQL Anywhere 16 CustDB"

2. ml_add_java_connection_script または ml_add_dnet_connection_script ストアドプロシージャを 使用して、authenticate_user イベント用の authenticateUser メソッドを登録します。

Java の場合は、次の SQL 文を実行します。

CALL ml_server.ml_add_java_connection_script('custdb 16.0', 'authenticate_user', 'MobiLinkAuth.authenticateUser');

COMMIT;

.NET の場合は、次の SQL 文を実行します。

CALL ml_add_dnet_connection_script('custdb 16.0', 'authenticate_user', 'MobiLinkAuth.authenticateUser');

COMMIT;

結果

authenticate_user イベント用の authenticateUser メソッドが登録されます。

次の手順

「レッスン3: Mobile Link サーバの起動」161ページに進みます。

参照

- ●「スクリプトの追加と削除」『Mobile Link サーバ管理』
- ●「Java による同期スクリプトの作成」『Mobile Link サーバ管理』
- ●「.NET の同期スクリプト」『Mobile Link サーバ管理』
- ●「Java 同期の例」『Mobile Link サーバ管理』
- ●「.NET 同期のサンプル」『Mobile Link サーバ管理』
- ●「Java クラスのデバッグ」『Mobile Link サーバ管理』
- ●「.NET 同期論理のデバッグ」『Mobile Link サーバ管理』
- ●「同期スクリプトの作成」『Mobile Link サーバ管理』
- ●「同期イベント」『Mobile Link サーバ管理』
- ●「ml add java connection script システムプロシージャ」『Mobile Link サーバ管理』
- ●「ml add dnet connection script システムプロシージャ」『Mobile Link サーバ管理』

レッスン3: Mobile Link サーバの起動

このレッスンでは、-sl オプションを指定して Mobile Link サーバを実行し、コンパイル済みファ イルを検索するための一連のディレクトリを指定します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:カスタム認証用の Java クラスの作成 (サーバ側)」157 ページを参照してくだ さい。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:カスタムユーザ認証用の Java と.NET の使用」155 ページ

◆ タスク

● CustDB サンプルデータベースに接続して、mlsrv16 コマンドラインで Java クラスまた は.NET アセンブリをロードします。

c:¥MLauthは、ソースファイルがある実際のディレクトリに置き換えてください。

Java の場合は、次のコマンドを実行します。

mlsrv16 -c "DSN=SQL Anywhere 16 CustDB" -o serverOut.txt -v+ -sl java(-cp c:¥MLauth)

.NET の場合は、次のコマンドを実行します。

mlsrv16 -c "DSN=SQL Anywhere 16 CustDB" -o serverOut.txt -v+ -sl dnet(-MLAutoLoadPath=c: ¥MLauth)

結果

authenticate user synchronization イベントが発生すると、MobiLinkAuth メソッドが実行されます。

次の手順

「レッスン4:認証のテスト」162ページに進みます。

参照

- ●「Mobile Link サーバオプション」『Mobile Link サーバ管理』
- ●「-sl java mlsrv16 オプション」『Mobile Link サーバ管理』
- ●「-sl dnet mlsrv16 オプション」『Mobile Link サーバ管理』

レッスン4:認証のテスト

Ultra Light には、同期を開始できるサンプル Windows クライアントが用意されています。この レッスンでは、前のレッスンで起動した CustDB 統合データベースに対してアプリケーションを 実行します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:カスタム認証用の Java クラスの作成 (サーバ側)」157 ページを参照してくだ さい。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:カスタムユーザ認証用の Java と.NETの使用」155ページ

◆ タスク

1. サンプルアプリケーションを起動します。

[スタート] » [プログラム] » [SQL Anywhere 16] » [Ultra Light] » [Windows サンプルアプリ ケーション] をクリックします。 2. 無効な従業員 ID を入力して同期します。

このアプリケーションでは、従業員 ID は Mobile Link ユーザ名でもあります。128 で始まら ないユーザ名の場合、論理により同期が失敗します。Employee ID に値 50 を入力し、[OK] を クリックします。

結果

authenticate_user スクリプトからエラー 4000 が返されたことが、Mobile Link サーバのメッセージ ウィンドウに表示されます。

ユーザ ID またはパスワードが無効であることを示す SQLCODE -1497 の同期エラーが [Ultra Light CustDB デモ] ウィンドウに表示されます。 「ユーザ ID またはパスワードが無効です。」 『エラーメッセージ』を参照してください。

次の手順

「クリーンアップ」163ページに進みます。

参照

●「Mobile Link の CustDB サンプル」57 ページ

クリーンアップ

すべてのチュートリアルをコンピュータから削除し、Windows サンプルアプリケーションのデー タベースをリセットします。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:カスタム認証用の Java クラスの作成 (サーバ側)」157 ページを参照してくだ さい。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:カスタムユーザ認証用の Java と.NET の使用」155ページ

◆タスク

1. Java または.NET のソースファイルを削除します。

たとえば、c:¥mlauth ディレクトリを削除します。

警告 このディレクトリには、チュートリアル関連のファイルのみが含まれていることを確認して ください。

2. Interactive SQL と Ultra Light Windows クライアントアプリケーションを終了します。

各アプリケーションで、[ファイル]»[終了]をクリックします。

- 3. SQL Anywhere、Mobile Link、同期クライアントの各ウィンドウを閉じます。 それぞれのタスクバーを右クリックして、[閉じる] をクリックします。
- 4. Windows サンプルアプリケーションのデータベースをリセットします。

%SQLANYSAMP16%¥UltraLite¥CustDB ディレクトリから次のコマンドを実行します。

makedbs

結果

すべてのチュートリアルがコンピュータから削除され、Windows サンプルアプリケーションの データベースがリセットされます。

次の手順

なし。

チュートリアル:ダイレクトローハンドリングの使用

ダイレクトローハンドリングを使用すると、サポートされている統合データベース以外にも、リ モートデータと中央のデータソース、アプリケーション、または Web サービスとの通信ができ ます。

このチュートリアルでは、Java 用または.NET 用の Mobile Link サーバ API を使用して簡単なダ イレクトローハンドリングを行う方法を習得します。また、クライアントの RemoteOrders テー ブルを統合データベースと同期し、OrderComments テーブル用に特別なダイレクトローハンドリ ング処理を追加する方法も習得します。

必要なソフトウェア

SQL Anywhere 16

●Java ソフトウェア開発キットまたは Microsoft .NET Framework

前提知識と経験

次の知識と経験が必要です。

●Java または .NET の知識

●Mobile Link イベントスクリプトの基本的な知識

権限

統合データベースで次のロールおよび権限を持つ必要があります。

●SYS AUTH RESOURCE ROLE 互換ロール

●MONITOR システム権限

リモートデータベースで次のロールおよび権限を持つ必要があります。

●SYS REPLICATION ADMIN ROLE システムロール

●SYS RUN REPLICATION ROLE システムロール

概要

このチュートリアルでは、次の作業の方法について説明します。

- ●Java 用または.NET 用の Mobile Link サーバ API の使用
- ●Mobile Link ダイレクトローハンドリング用のメソッドの作成

参照

- ●「Mobile Link 同期」1ページ
- ●「同期の方法」『Mobile Link サーバ管理』
- ●「ダイレクトローハンドリング」『Mobile Link サーバ管理』
- http://www.sybase.com/detail?id=1058600#319 (このページを表示するには、Sybase.com アカウ ントが必要です。)
- http://sqlanywhere-forum.sybase.com

レッスン1:テキストファイルデータソースの設定

このレッスンでは、注文情報を保存する新しいテキストファイルを作成します。

前提条件

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:ダイレクトローハンドリン グの使用」164ページ

◆ タスク

- 1. 新しい空のテキストファイルを作成します。
- 2. comment_id、order_id、order_comment を表す、次のタブで区切った値をファイルに追加します。
 - 786 34 OK, ship promotional material.
 - 787 35 Yes, the product is going out of production.
 - 788 36 No, your commission cannot be increased...
- 3. そのファイルを作業ディレクトリに保存します。

このチュートリアルでは、*c*:*¥MLdirect*をサーバサイドコンポーネントの作業ディレクトリとします。ファイルを orderResponses.txt という名前でこのディレクトリに保存します。

結果

テキストファイルが作成されます。

次の手順

「レッスン2: Mobile Link 統合データベースの設定」166ページに進みます。

レッスン2: Mobile Link 統合データベースの設定

このレッスンでは、データベースを作成し、ODBC データソースを定義します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:テキストファイルデータソースの設定」165ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:ダイレクトローハンドリン グの使用」164ページ

内容と備考

Mobile Link 統合データベースはデータの中央レポジトリであり、同期処理の管理に使用する Mobile Link のシステムテーブルとストアドプロシージャが含まれます。ダイレクトローハンド リングでは、統合データベース以外のデータソースと同期しますが、Mobile Link サーバが使用 する情報を保持するために統合データベースも必要です。

注意

Mobile Link 統合データベースを Mobile Link システムオブジェクトと ODBC データソースを使用して設定済みの場合は、このレッスンは省略できます。

◆ タスク

- 1. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [Sybase Central] をクリック します。
- 2. [ツール] » [SQL Anywhere 16] » [データベースの作成] をクリックします。
- 3. [次へ] をクリックします。
- 4. デフォルト値 [このコンピュータにデータベースを作成] を受け入れて、[次へ] をクリックします。
- 5. [メインデータベースファイルを保存] フィールドに、データベースのファイル名およびパス を入力します。たとえば、c:¥MLdirect¥MLconsolidated.db と入力します。

- データベース作成ウィザードの残りの指示に従い、デフォルト値をそのまま使用します。
 [データベースへの接続] ページで、[最終切断後にデータベースを停止] オプションをオフにします。
- 7. [完了] をクリックします。

ML consolidated データベースが Sybase Central に表示されます。

- 8. ウィンドウが自動的に閉じない場合は、[データベースの作成] ウィンドウの [閉じる] をク リックします。
- 9. SQL Anywhere 16 ドライバを使用して、ML consolidated データベース用の ODBC データソー スを定義します。

Sybase Central で、[ツール] » [SQL Anywhere 16] » [ODBC アドミニストレータを開く] をク リックします。

- 10. [ユーザ DSN] タブをクリックし、[追加] をクリックします。
- [データ ソースの新規作成] ウィンドウで、[SQL Anywhere 16] をクリックし、[完了] をクリックします。
- 12. [SQL Anywhere の ODBC 設定] ウィンドウで、次の操作を行います。
 - a. [ODBC] タブをクリックします。
 - b. [データソース名] フィールドに mldirect_db と入力します。
 - c. [ログイン] タブをクリックします。
 - d. [ユーザ ID] フィールドに DBA と入力します。
 - e. [パスワード] フィールドに sql と入力します。
 - f. [サーバ名] フィールドに、ML consolidated と入力します。
 - g. [OK] をクリックします。
- 13. ODBC データソースアドミニストレータを閉じます。

[ODBC データソースアドミニストレータ] ウィンドウで [OK] をクリックします。

結果

統合データベースと ODBC データソースが作成されます。

次の手順

「レッスン3: Mobile Link 統合データベースでのテーブルの作成」168ページに進みます。

参照

●「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』

●「Mobile Link 統合データベース」『Mobile Link サーバ管理』

レッスン 3: Mobile Link 統合データベースでのテーブルの作成

このレッスンでは、Mobile Link 統合データベースに RemoteOrders テーブルを作成します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:テキストファイルデータソースの設定」165ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。 「チュートリアル:ダイレクトローハンドリン グの使用」164ページ

内容と備考

作成する RemoteOrders テーブルには次のカラムが含まれます。

order id 注文のユニークな識別子です。

product id 製品のユニークな識別子です。

quantity 品目の販売数です。

order_status 注文のステータスです。

last_modified ローが最後に変更された日です。このカラムはタイムスタンプベースのダウン ロードに使用します。このダウンロード方法は、効率的な同期のためにローをフィルタする一般 的な方法です。

◆ RemoteOrders テーブルの作成

1. Interactive SQL からデータベースに接続します。

Interactive SQL は、Sybase Central またはコマンドプロンプトから起動できます。

●Sybase Central から Interactive SQL を起動するには、MLconsolidated - DBA データベースを 右クリックし、[Interactive SQL を開く] をクリックします。

●コマンドプロンプトで Interactive SQL を起動するには、次のコマンドを実行します。

dbisql -c "DSN=mldirect db"

2. Interactive SQL で次の SQL 文を実行し、RemoteOrders テーブルを作成します。

```
CREATE TABLE RemoteOrders (
order_id INTEGER NOT NULL,
product_id INTEGER NOT NULL,
quantity INTEGER,
order_status VARCHAR(10) DEFAULT 'new',
last_modified TIMESTAMP DEFAULT CURRENT TIMESTAMP,
PRIMARY KEY(order_id)
);
```

Interactive SQL によって、統合データベースに RemoteOrders テーブルが作成されます。

3. Interactive SQL で次の文を実行して Mobile Link のシステムテーブルとストアドプロシージャを作成します。

*C:Program Files¥SQL Anywhere 16¥*は、SQL Anywhere 16 インストール環境のロケーション に置き換えてください。

READ "C: ¥Program Files ¥SQL Anywhere 16 ¥MobiLink ¥setup ¥syncsa.sql";

Interactive SQL によって *syncsa.sql* が統合データベースに適用されます。*syncsa.sql* を実行すると、前に **ml**_ が付いた一連のシステムテーブルとストアドプロシージャが作成されます。これらのテーブルとストアドプロシージャは、同期処理中に Mobile Link サーバによって使用されます。

結果

RemoteOrders テーブルが作成され、Mobile Link のシステムテーブルとストアドプロシージャが 統合データベースに追加されます。

次の手順

「レッスン4:同期スクリプトの追加」169ページに進みます。

参照

●「CREATE TABLE 文」『SQL Anywhere サーバ SQL リファレンス』

レッスン4:同期スクリプトの追加

ダイレクトローハンドリングを使用し、ストアドプロシージャを使用して Mobile Link 統合デー タベースに同期スクリプト情報を追加します。このレッスンでは、handle_UploadData、 handle_DownloadData、end_download、download_cursor、download_delete_cursorの各イベントに 対応するメソッド名を登録します。独自の Java または .NET クラスをレッスンの後半で作成し ます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:テキストファイルデータソースの設定」165ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。 「チュートリアル:ダイレクトローハンドリン グの使用」164ページ

内容と備考

SQL ローハンドリングを使用すると、リモートデータを、Mobile Link 統合データベース内のテーブルと同期できます。SQL ベースのスクリプトでは、次の情報を定義します。

●Mobile Link クライアントからアップロードするデータを統合データベースに適用する方法。

●統合データベースからダウンロードするデータ。

次の SQL ベースのアップロードイベントとダウンロードイベントが作成されます。

- upload_insert このイベントは、クライアントデータベースに挿入された新しい注文を統合 データベースに適用する方法を定義します。
- download_cursor このイベントは、リモートクライアントにダウンロードする注文を定義 します。
- download_delete_cursor このイベントは、アップロード専用の同期スクリプトを使用する 場合に必要です。このチュートリアルでは、このイベントを無視するように Mobile Link サー バを設定します。

◆ タスク

1. まだ接続していない場合は、次のコマンドを実行して、Interactive SQL から統合データベー スに接続します。

dbisql -c "DSN=mldirect_db"

 ml_add_table_script ストアドプロシージャを使用して、upload_insert、download_cursor、 download_delete_cursorの各イベント用の SQL ベースのテーブルスクリプトを追加します。

Interactive SQL で次の SQL 文を実行します。upload_insert のスクリプトでは、アップロード された order_id、product_id、quantity、order_status を Mobile Link 統合データベースに挿入し ます。download_cursor のスクリプトでは、タイムスタンプベースのフィルタを使用して、更 新されたローをリモートクライアントにダウンロードします。

CALL ml_add_table_script('default', 'RemoteOrders', 'upload_insert', 'INSERT INTO RemoteOrders(order_id, product_id, quantity, order_status) VALUES({ml r.order_id}, {ml r.product_id}, {ml r.quantity}, {ml r.order_status})'); CALL ml_add_table_script('default', 'RemoteOrders', 'download_cursor', 'SELECT order_id, product_id, quantity, order_status FROM RemoteOrders WHERE last_modified >= {ml s.last_table_download}');

CALL ml_add_table_script('default', 'RemoteOrders', 'download_delete_cursor', '--{ml_ignore}');

COMMIT;

3. end_download イベント用の Java または .NET メソッドを登録します。

Mobile Link サーバが end_download 接続イベントを実行するときに、このメソッドを使用し てメモリリソースを解放します。

Java の場合は、Interactive SQL で次の文を実行します。

CALL ml_add_java_connection_script('default', 'end_synchronization', 'MobiLinkOrders.EndSync'); .NET の場合は、Interactive SQL で次の文を実行します。

Interactive SQL によって、ユーザ定義の EndDownload メソッドが end_download イベント用に 登録されます。

4. handle_UploadData と handle_DownloadData の各イベント用の Java または .NET メソッドを登録します。

Java の場合は、Interactive SQL で次の文を実行します。

.NET の場合は、Interactive SQL で次の文を実行します。

Interactive SQL によって、ユーザ定義の GetUpload と SetDownload の各メソッドが、それぞれ handle_UploadData と handle_DownloadData の各イベント用に登録されます。これらのメ ソッドは次のレッスンで作成します。

5. download_cursor と download_delete_cursor の各イベントを登録します。

Interactive SQL で次の文を実行します。

CALL ml_add_table_script('default', 'OrderComments', 'download_cursor', '--{ml_ignore}');

CALL ml_add_table_script('default', 'OrderComments', 'download_delete_cursor', '--{ml_ignore}');

同期は双方向であり、アップロード専用ではないため、スクリプトを使用するときには、 OrderComments テーブル用に download_cursor と download_delete_cursor の各イベントを登録 してください。 「必要なスクリプト」『Mobile Link サーバ管理』を参照してください。

6. これまでの変更内容をコミットします。

Interactive SQL で次の文を実行します。

COMMIT;

7. Interactive SQL を閉じます。

結果

handle_UploadData、handle_DownloadData、end_download、download_cursor、および download delete cursor イベントに対応するメソッド名が登録されます。

次の手順

「レッスン5: Mobile Link のダイレクトローハンドリングのための Java または .NET クラスの作成」172ページに進みます。

参照

- ●「Mobile Link イベントの概要」『Mobile Link サーバ管理』
- ●「スクリプトの追加と削除」『Mobile Link サーバ管理』
- ●「ローをアップロードするスクリプト」『Mobile Link サーバ管理』
- ●「ローをダウンロードするスクリプト」『Mobile Link サーバ管理』
- ●「upload insert テーブルイベント」『Mobile Link サーバ管理』
- ●「upload update テーブルイベント」『Mobile Link サーバ管理』
- 「upload delete テーブルイベント」 『Mobile Link サーバ管理』
- ●「download cursor テーブルイベント」『Mobile Link サーバ管理』
- ●「download delete cursor テーブルイベント」『Mobile Link サーバ管理』
- ●「ダイレクトローハンドリング」『Mobile Link サーバ管理』
- ●「ダイレクトアップロード」『Mobile Link サーバ管理』
- ●「ダイレクトダウンロード」『Mobile Link サーバ管理』
- ●「タイムスタンプベースのダウンロードの実装」『Mobile Link サーバ管理』
- ●「リモートデータベース間でローを分割する」『Mobile Link サーバ管理』

レッスン 5: Mobile Link のダイレクトローハンドリングのための Java または .NET クラスの作成

このレッスンでは、ダイレクトローハンドリングを使用して、クライアントデータベース内の OrderComments テーブルのローを処理します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:テキストファイルデータソースの設定」165ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。 「チュートリアル:ダイレクトローハンドリン グの使用」164ページ

内容と備考

ダイレクトローハンドリング用に次のメソッドを追加します。

- **GetUpload** このメソッドは handle_UploadData イベントに使用します。GetUpload では、 アップロードされたコメントを orderComments.txt というファイルに書き込みます。
- SetDownload このメソッドは handle_DownloadData イベントに使用します。SetDownload では、*orderResponses.txt* ファイルを使用してリモートクライアントに応答をダウンロードします。
- EndDownload このメソッドを end_download イベントに使用します。EndDownload では、 メモリリソースが解放されます。

次の手順では、処理用メソッドを含む Java または .NET のクラスを作成する方法を示します。 完全なリストについては、「MobiLinkOrders コードの全リスト (Java)」179 ページまたは 「MobiLinkOrders コードの全リスト (.NET)」181 ページを参照してください。

◆ タスク

1. Java または .NET で、MobiLinkOrders というクラスを作成します。

Java の場合は、次のコードを入力します。

import ianywhere.ml.script.*; import java.io.*; import java.sql.*;

public class MobiLinkOrders {

.NET の場合は、次のコードを入力します。

using iAnywhere.MobiLink.Script; using System.IO; using System.Data; using System.Text;

public class MobiLinkOrders {

2. クラスレベルの DBConnectionContext インスタンスを宣言します。

Java の場合は、次のコードを入力します。

// Class level DBConnectionContext
DBConnectionContext _cc;

.NET の場合は、次のコードを入力します。

// Class level DBConnectionContext
private DBConnectionContext _cc = null;

Mobile Link サーバによって DBConnectionContext のインスタンスがクラスコンストラクタに 渡されます。DBConnectionContext には、Mobile Link 統合データベースとの現在の接続に関 する情報がカプセル化されます。

3. ファイル入出力に使用するオブジェクトを宣言します。

Java の場合は、java.io.FileWriter と java.io.BufferedReader を次のように宣言します。

// Java objects for file i/o FileWriter my_writer; BufferedReader my_reader;

.NET の場合は、StreamWriter と StreamReader を次のように宣言します。

// Instances for file I/O
private static StreamWriter my_writer = null;
private static StreamReader my_reader = null;

4. クラスコンストラクタを作成します。

```
クラスコンストラクタが、クラスレベルの DBConnectionContext インスタンスを設定します。
```

```
Java の場合は、次のコードを入力します。
public MobiLinkOrders( DBConnectionContext cc )
throws IOException, FileNotFoundException
```

```
{
    // Declare a class-level DBConnectionContext
    _cc = cc;
}
```

```
.NET の場合は、次のコードを入力します。
```

```
public MobiLinkOrders(DBConnectionContext cc) {
    _cc = cc;
}
```

5. GetUpload メソッドを作成します。

GetUpload メソッドでは、OrderComments テーブルを表す UploadedTableData クラスインスタンスを取得します。OrderComments テーブルには、遠隔地の営業部員による特別なコメントが含まれます。このテーブルはレッスンの後半で作成します。

UploadedTableData の getInserts メソッドでは、注文に対する新しいコメントの結果セットを 返します。writeOrderComment メソッドでは、結果セット内の各ローをテキストファイルに書 き出します。

```
Java の場合は、次のコードを入力します。
```

```
public void writeOrderComment( int commentID, int orderID, String comments )
    throws IOException
  {
    if (my writer == null)
       // A FileWriter for writing order comments
       my writer = new FileWriter( "C:¥¥MLdirect¥¥orderComments.txt",true);
    // Write out the order comments to remoteOrderComments.txt
    my_writer.write(_commentID + "¥t" + _orderID + "¥t" + _comments);
    my_writer.write("¥n");
    my_writer.flush();
  }
  // Method for the handle UploadData synchronization event
  public void GetUpload( UploadData ut )
    throws SQLException, IOException
  ł
    // Get an UploadedTableData for OrderComments
```
```
UploadedTableData orderCommentsTbl = ut.getUploadedTableByName("OrderComments");
          // Get inserts uploaded by the MobiLink client
          ResultSet insertResultSet = orderCommentsTbl.getInserts();
          while (insertResultSet.next())
            // Get order comments
            int commentID = insertResultSet.getInt("comment id");
            int orderID = insertResultSet.getInt("order id");
             String specialComments = insertResultSet.getString("order comment");
            if ( specialComments != null) {
               writeOrderComment( commentID, orderID, specialComments);
            }
          insertResultSet.close();
        }
   .NET の場合は、次のコードを入力します。
        public void WriteOrderComment(int comment id,
          int order id.
          string comments)
          if (my writer == null)
            my writer = new StreamWriter("c:¥¥MLdirect¥¥orderComments.txt");
          my_writer.WriteLine("{0}+t{1}+t{2}", comment_id, order_id, comments);
          my writer.Flush();
        }
      // Method for the handle UploadData synchronization event.
        public void GetUpload(UploadData ut)
          // Get UploadedTableData for remote table called OrderComments
          UploadedTableData order comments table data =
            ut.GetUploadedTableByName("OrderComments");
          // Get inserts uploaded by the MobiLink client
          using( IDataReader new_comment_reader = order_comments_table_data.GetInserts() ) {
                 while (new_comment_reader.Read()) {
                        // Columns are
                        // 0 - "order comment"
                        // 1 - "comment_id"
                        // 2 - "order id"
                        // You can look up these values using the DataTable returned by:
                        // order comments table data.GetSchemaTable().
                        // In this example, you just use the known column order to
                        // determine the column indexes; alternatively, you could use
                        // the column names
                        // Only process this insert if the order comment is not null
                        if (!new comment reader.lsDBNull(2)) {
                           int comment id = new comment reader.GetInt32(0);
                           int order id = new comment reader.GetInt32(1);
                           string comments = new_comment_reader.GetString(2);
                           WriteOrderComment(comment id, order id, comments);
                        }
                 }
               }
        }
6. SetDownload メソッドを作成します。
```

a. OrderComments テーブルを表すクラスインスタンスを取得します。

DBConnectionContextの getDownloadData メソッドを使用して DownloadData のインスタ ンスを取得します。DownloadData の getDownloadTableByName メソッドを使用して、 OrderComments テーブルの DownloadTableData インスタンスを返します。

```
Java の場合は、次のコードを入力します。
```

```
public void SetDownload()
    throws SQLException, IOException
{
    DownloadData download_d = _cc.getDownloadData();
    DownloadTableData download_td =
    download_d.getDownloadTableByName( "OrderComments" );
.NET の場合は、次のコードを入力します。
    private const string read_file_path =
        "c:¥¥MLdirect¥¥orderResponses.txt";
    // Method for the handle_DownloadData synchronization event
    public void SetDownload() {
        if ((my_reader == null) && !File.Exists(read_file_path)) {
            System.Console.Out.Write("There is no file to read.");
            return;
            pownloadTableData comments_for_download =
            _cc.GetDownloadData().GetDownloadTableByName("OrderComments");
```

注意

OrderComments テーブルは、「レッスン7: Mobile Link クライアントデータベースの設定」185ページでリモートデータベースに作成します。

b. 準備文または IDbCommand を取得します。これを使用すると、ダウンロードに操作を追加、挿入、または更新できます。

Java の場合は、DownloadTableData の getUpsertPreparedStatement メソッドを使用して java.sql.PreparedStatement のインスタンスを次のように返します。

PreparedStatement update_ps = download_td.getUpsertPreparedStatement();

.NET の場合は、DownloadTableData の GetUpsertCommand メソッドを次のように使用します。

// Add upserts to the set of operation that are going to be // applied at the remote database IDbCommand comments_upsert = comments for download.GetUpsertCommand();

c. 各ローのダウンロードデータを設定します。

このコードは、orderResponses.txt を参照して、Mobile Link ダウンロードにデータを追加 しています。

Java の場合は、次のコードを入力します。

try {
 // A BufferedReader for reading in responses
 if (my reader == null)

```
my_reader = new BufferedReader(new FileReader("C:¥¥MLdirect¥¥orderResponses.txt"));
```

```
// Get the next line from orderResponses
       String commentLine;
       commentLine = my_reader.readLine();
      // Send comment responses down to clients
       while (commentLine != null) {
       // Get the next line from orderResponses.txt
       String[] response details = commentLine.split("¥t");
       if (response details.length != 3) {
         System.err.println("Error reading from orderResponses.txt");
         System.err.println("Error setting direct row handling download");
         return:
       int comment id = Integer.parseInt(response details[0]);
       int order id = Integer.parseInt(response details[1]);
       String updated comment = response details[2];
      // Set an order comment response in the MobiLink download
       update ps.setInt(1, comment id);
       update ps.setInt(2, order id);
       update_ps.setString(3, updated_comment);
       update ps.executeUpdate();
      // Get next line
       commentLine = my reader.readLine();
    }
.NET の場合は、次のコードを入力します。
       if (my reader == null) {
         my_reader = new StreamReader(read_file_path);
       string comment line;
       while ((comment_line = my_reader.ReadLine()) != null) {
         // Three values are on each line separated by '¥t'
         string[] response details = comment_line.Split('¥t');
         if (response details.Length != 3) {
           throw (new SynchronizationException(
              "Error reading from orderResponses.txt"));
         int comment id = System.Int32.Parse(response details[0]);
         int order_id = System.Int32.Parse(response_details[1]);
         string comments = response_details[2];
         // Parameters of the correct number and type have
         // already been added so you just need to set the
         // values of the IDataParameter
         ((IDataParameter)(comments upsert.Parameters[0])).Value =
            comment id;
         ((IDataParameter)(comments upsert.Parameters[1])).Value =
           order id;
         ((IDataParameter)(comments upsert.Parameters[2])).Value =
           comments;
         // Add the upsert operation
         comments_upsert.ExecuteNonQuery();
      }
    }
ダウンロードに挿入操作または更新操作を追加する準備文を終了します。
```

d.

Java の場合は、次のコードを入力します。 finally { update_ps.close(); } }

.NET の場合、IDbCommand を閉じる必要はありません。オブジェクトは、ダウンロードの終わりに自動的に破棄されます。

7. EndDownload メソッドを作成します。

このメソッドでは、end download 接続イベントを処理し、またリソースを解放できます。

```
Java の場合は、次のコードを入力します。
```

```
public void EndSync()
  throws IOException
{
    // Close i/o resources
    if (my_reader != null) {
        my_reader.close();
        my_reader = null;
    }
    if (my_writer != null) {
        my_writer.close();
        my_writer = null;
    }
}
```

.NET の場合は、次のコードを入力します。

public void EndSync()
{
 if (my_writer != null) {
 my_writer.Close();
 my_writer = null;
 }
 if (my_reader != null) {

```
my_reader.Close();
my_reader = null;
}
```

8. コードを保存します。

Java の場合は、コードを MobiLinkOrders.java という名前で作業ディレクトリ c:¥MLdirect に 保存します。

.NET の場合は、コードを *MobiLinkOrders.cs* という名前で作業ディレクトリ *c:*¥*MLdirect* に保存します。

 コードを検証する場合は、「MobiLinkOrders コードの全リスト (Java)」179 ページまたは 「MobiLinkOrders コードの全リスト (.NET)」181 ページを参照してください。

10. クラスファイルをコンパイルします

- a. Java または.NET のソースファイルが含まれるディレクトリに移動します。
- b. MobiLinkOrders をコンパイルし、Java または .NET 用の Mobile Link サーバ API ライブラ リを参照します。

Java の場合、%SQLANY16%¥java にある mlscript.jar を参照する必要があります。

Java の場合は、次のコマンドを実行して、*C:*¥*Program Files*¥*SQL Anywhere 16*¥を SQL Anywhere 16 ディレクトリに置き換えます。

javac -classpath "C:¥Program Files¥SQL Anywhere 16¥java¥mlscript.jar" MobiLinkOrders.java

.NET の場合は、次のコマンドを実行して、*C:¥Program Files¥SQL Anywhere 16¥*を SQL Anywhere 16 ディレクトリに置き換えます。

csc /out:MobiLinkServerCode.dll /target:library /reference:"C:¥Program Files¥SQL Anywhere 16¥Assembly¥v2¥iAnywhere.MobiLink.Script.dll" MobiLinkOrders.cs

注意

この例では、プライマリキーの値がユニークとはかぎりません。 「ユニークなプライマリ キー」『Mobile Link サーバ管理』を参照してください。

結果

Mobile Link のダイレクトローハンドリングのための Java クラスまたは .NET クラスが作成されます。

次の手順

「レッスン6: Mobile Link サーバの起動」183ページに進みます。

参照

- ●「ダイレクトローハンドリング」『Mobile Link サーバ管理』
- ●「Java による同期スクリプトの作成」『Mobile Link サーバ管理』
- ●「.NET の同期スクリプト」『Mobile Link サーバ管理』

MobiLinkOrders コードの全リスト (Java)

Java でダイレクトローハンドリングを行う場合の MobiLinkOrders の全リストは次のとおりで す。 手順を追った説明については、「レッスン 5: Mobile Link のダイレクトローハンドリングの ための Java または .NET クラスの作成」172 ページを参照してください。

import ianywhere.ml.script.*; import java.io.*; import java.sql.*;

public class MobiLinkOrders {

// Class level DBConnectionContext
DBConnectionContext _cc;

// Java objects for file i/o
FileWriter my_writer;
BufferedReader my reader;

```
public MobiLinkOrders( DBConnectionContext cc )
  throws IOException, FileNotFoundException
  // Declare a class-level DBConnectionContext
   cc = cc;
public void writeOrderComment( int commentID, int orderID, String comments )
  throws IOException
  if (my writer == null)
    // A FileWriter for writing order comments
    my writer = new FileWriter( "C:¥¥MLdirect¥¥orderResponses.txt",true);
  // Write out the order comments to remoteOrderComments.txt
  my_writer.write(_commentID + "¥t" + _orderID + "¥t" + _comments);
my_writer.write( "¥n" );
  my writer.flush();
}
// Method for the handle UploadData synchronization event
public void GetUpload( UploadData ut )
  throws SQLException, IOException
  // Get an UploadedTableData for OrderComments
  UploadedTableData orderCommentsTbl = ut.getUploadedTableByName("OrderComments");
  // Get inserts uploaded by the MobiLink client
  ResultSet insertResultSet = orderCommentsTbl.getInserts();
  while (insertResultSet.next())
    // Get order comments
    int commentID = insertResultSet.getInt("comment id");
    int orderID = insertResultSet.getInt("order id")
    String specialComments = insertResultSet.getString("order comment");
    if ( specialComments != null) {
       writeOrderComment( commentID, orderID, specialComments);
  insertResultSet.close();
public void SetDownload()
  throws SQLException, IOException
ł
  DownloadData download d = cc.getDownloadData();
  DownloadTableData download td = download d.getDownloadTableByName( "OrderComments" );
  PreparedStatement update ps = download td.getUpsertPreparedStatement();
try {
  // A BufferedReader for reading in responses
  if (my reader == null)
  my reader = new BufferedReader(new FileReader("C:¥¥MLdirect¥¥orderResponses.txt"));
  // Get the next line from orderResponses
  String commentLine;
  commentLine = my reader.readLine();
  // Send comment responses down to clients
  while (commentLine != null) {
  // Get the next line from orderResponses.txt
```

```
String[] response details = commentLine.split("¥t");
  if (response details.length != 3) {
     System.err.println("Error reading from orderResponses.txt");
     System.err.println("Error setting direct row handling download");
     return;
  int comment id = Integer.parseInt(response details[0]);
  int order id = Integer parseInt(response details[1]);
  String updated comment = response details[2];
  // Set an order comment response in the MobiLink download
  update ps.setInt(1, comment id);
  update_ps.setInt(2, order id);
  update ps.setString(3, updated comment);
  update ps.executeUpdate();
  // Get next line
  commentLine = my reader.readLine();
} finally {
  update ps.close();
public void EndDownload()
  throws IOException
   // Close i/o resources
  if (my reader != null) {
     my reader.close();
     my reader = null;
  if (my_writer != null) {
    my_writer.close();
     my writer = null;
  }
}
```

MobiLinkOrders コードの全リスト (.NET)

.NET でダイレクトローハンドリングを行う場合の MobiLinkOrders の全リストは次のとおりで す。 手順を追った説明については、「レッスン 5: Mobile Link のダイレクトローハンドリングの ための Java または .NET クラスの作成」172 ページを参照してください。

```
using iAnywhere.MobiLink.Script;
using System.IO;
using System.Data;
using System.Text;
```

}

```
public class MobiLinkOrders {
    // Class level DBConnectionContext
    private DBConnectionContext _cc = null;
```

```
// Instances for file I/O
private static StreamWriter my_writer = null;
private static StreamReader my_reader = null;
```

public MobiLinkOrders(DBConnectionContext cc) {

```
_cc = cc;
}
public void WriteOrderComment(int comment id,
  int order id,
  string comments)
ł
  if (my_writer == null) {
     my writer = new StreamWriter("c:¥¥MLdirect¥¥orderComments.txt");
  my writer.WriteLine("{0}¥t{1}¥t{2}", comment id, order id, comments);
  my_writer.Flush();
// Method for the handle UploadData synchronization event.
public void GetUpload(UploadData ut)
  // Get UploadedTableData for remote table called OrderComments
  UploadedTableData order_comments_table_data =
ut.GetUploadedTableByName("OrderComments");
  // Get inserts uploaded by the MobiLink client
  IDataReader new comment reader =
     order comments table data.GetInserts();
  while (new comment reader.Read()) {
     // Columns are
     // 0 - "order_comment"
     // 1 - "comment id"
     // 2 - "order_id"
     // You can look up these values using the DataTable returned by:
     // order comments table data.GetSchemaTable().
     // In this example, you just use the known column order to
     // determine the column indexes
     // Only process this insert if the order comment is not null
     if (!new comment reader.IsDBNull(2)) {
       int comment id = new comment reader.GetInt32(0);
       int order id = new comment reader.GetInt32(1);
       string comments = new_comment_reader.GetString(2);
       WriteOrderComment(comment id, order id, comments);
     }
  // Always close the reader when you are done with it!
  new comment reader.Close();
}
private const string read file path =
   "c:¥¥MLdirect¥¥orderResponses.txt";
// Method for the handle_DownloadData synchronization event
public void SetDownload() {
    if ((my_reader == null) && !File.Exists(read_file_path)) {
     System.Console.Out.Write("There is no file to read.");
     return;
  DownloadTableData comments for download =
     cc.GetDownloadData().GetDownloadTableByName("OrderComments");
  // Add upserts to the set of operation that are going to be
  // applied at the remote database
  IDbCommand comments upsert =
     comments for download.GetUpsertCommand();
```

```
if (my reader == null)
       my reader = new StreamReader(read file path);
     string comment line;
     while ((comment line = my reader.ReadLine()) != null) {
       // Three values are on each line separated by '¥t'
       string[] response details = comment line.Split('¥t');
       if (response details.Length != 3) {
          throw (new SynchronizationException(
            "Error reading from orderResponses.txt"));
       int comment_id = System.Int32.Parse(response details[0]);
       int order id = System.Int32.Parse(response details[1]);
       string comments = response details[2];
       // Parameters of the correct number and type have
       // already been added so you just need to set the
       // values of the IDataParameter
       ((IDataParameter)(comments_upsert.Parameters[0])).Value =
          comment id;
       ((IDataParameter)(comments upsert.Parameters[1])).Value =
          order id;
       ((IDataParameter)(comments upsert.Parameters[2])).Value =
          comments;
       // Add the upsert operation
       comments upsert.ExecuteNonQuery();
  }
  public void EndDownload()
     if (my_writer != null) {
       my_writer.Close();
       my writer = null;
     if (my reader != null) {
       my reader.Close();
       my reader = null;
    }
  }
}
```

レッスン 6: Mobile Link サーバの起動

このレッスンでは、Mobile Link サーバを起動します。-c オプションを使って Mobile Link サー バ (mlsrv16) を起動し、統合データベースに接続します。-sl java オプションまたは -sl dnet オプ ションを使用して、Java クラスまたは .NET クラスをそれぞれロードします。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:テキストファイルデータソースの設定」165ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:ダイレクトローハンドリン グの使用」164ページ

◆ タスク

● 統合データベースに接続し、mlsrv16のコマンドラインでクラスをロードします。

c:¥MLdirect は、ソースファイルがある実際のディレクトリに置き換えてください。

Java の場合は、次のコマンドを実行します。

mlsrv16 -c "DSN=mldirect_db" -o serverOut.txt -v+ -dl -zu+ -x tcpip -sl java (-cp c:¥MLdirect)

.NET の場合は、次のコマンドを実行します。

mlsrv16 -c "DSN=mldirect_db" -o serverOut.txt -v+ -dl -zu+ -x tcpip -sl dnet (-MLAutoLoadPath=c: ¥MLdirect)

Mobile Link サーバメッセージウィンドウが表示されます。

このチュートリアルで使用している Mobile Link サーバの各オプションの説明を次に示しま す。オプション -o、-v、-dl は、デバッグとトラブルシューティングの情報を提供します。こ れらのロギングオプションは、開発環境での使用に適しています。パフォーマンス上の理由 から、一般的に -v+ と -dl は運用環境では使用しません。

オプション	説明
-c	続いて接続文字列を指定します。
-0	メッセージログファイル serverOut.txt を指定します。
-v+	-v オプションは、ログを取る情報を指定します。-v+を使用して、最大冗長ロギングをオンに設定します。
-dl	画面にすべてのログメッセージを表示しま す。
-zu+	自動的に新しいユーザを追加します。
-x	Mobile Link クライアントの通信プロトコル とパラメータを設定します。
-sl java	クラスファイルを検索する一連のディレク トリを指定し、またサーバ起動時に Java VM をロードします。
-sl dnet	.NET アセンブリのロケーションを指定し、 またサーバ起動時に CLR をロードします。

結果

統合データベースに接続する Mobile Link サーバが起動し、作成したクラスがロードされます。

次の手順

「レッスン7: Mobile Link クライアントデータベースの設定」185ページに進みます。

参照

- ●「Mobile Link サーバオプション」『Mobile Link サーバ管理』
- ●「-sl java mlsrv16 オプション」『Mobile Link サーバ管理』
- ●「-sl dnet mlsrv16 オプション」『Mobile Link サーバ管理』

レッスン7: Mobile Link クライアントデータベースの設定

このレッスンでは、SQL Anywhere データベースを統合データベースと Mobile Link クライアント に使用します。また、このチュートリアルの目的上、Mobile Link クライアント、統合データベー ス、および Mobile Link サーバはすべて同じコンピュータに置きます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:テキストファイルデータソースの設定」165ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:ダイレクトローハンドリン グの使用」164ページ

内容と備考

Mobile Link クライアントデータベースを設定するには、RemoteOrders と OrderComments の各 テーブルを作成します。RemoteOrders テーブルは、統合データベースの RemoteOrders テーブル に対応します。Mobile Link サーバでは、SQL ベースのスクリプトを使用してリモート注文が同 期されます。OrderComments テーブルは、クライアントデータベースだけで使用されます。 Mobile Link サーバでは、特別なイベントを使用して OrderComments テーブルが処理されます。

テーブルの作成後に、クライアントデータベースで同期ユーザ、パブリケーション、サブスクリ プションを作成します。パブリケーションは、リモートデータベース上の同期対象となるテーブ ルとカラムを識別します。これらのテーブルとカラムを「アーティクル」と呼びます。同期サブ スクリプションは、パブリケーションに対する Mobile Link ユーザのサブスクリプションです。

◆ タスク

1. dbinit コマンドラインユーティリティを使用して、Mobile Link クライアントデータベースを 作成します。

次のコマンドを実行します。

dbinit -i -k -dba DBA,sql remote1

-i オプションと -k オプションは、それぞれ jConnect のサポートと Watcom SQL の互換ビュー を省略します。 2. dbsrv16 コマンドラインユーティリティを使用して、Mobile Link クライアントデータベース を起動します。

次のコマンドを実行します。

dbsrv16 remote1

3. 次のコマンドを実行して、Interactive SQL から Mobile Link クライアントデータベースに接続 します。

```
dbisql -c "SERVER=remote1;UID=DBA;PWD=sql"
```

4. Interactive SQL で次の SQL 文を実行して RemoteOrders テーブルを作成します。

```
CREATE TABLE RemoteOrders (
order_id INTEGER NOT NULL,
product_id INTEGER NOT NULL,
quantity INTEGER,
order_status VARCHAR(10) DEFAULT 'new',
PRIMARY KEY(order_id)
);
```

5. Interactive SQL で次の文を実行して OrderComments テーブルを作成します。

```
CREATE TABLE OrderComments (
comment_id INTEGER NOT NULL,
order_id INTEGER NOT NULL,
order_comment VARCHAR(255),
PRIMARY KEY(comment_id),
FOREIGN KEY(order_id) REFERENCES RemoteOrders(order_id)
);
```

6. Interactive SQL で次の文を実行して、Mobile Link 同期ユーザ、パブリケーション、サブスク リプションを作成します。

```
CREATE SYNCHRONIZATION USER ml_sales1;
CREATE PUBLICATION order_publ (TABLE RemoteOrders, TABLE OrderComments);
CREATE SYNCHRONIZATION SUBSCRIPTION TO order_publ FOR ml_sales1
TYPE TCPIP ADDRESS 'host=localhost';
```

注意

Mobile Link サーバに接続する方法は、CREATE SYNCHRONIZATION SUBSCRIPTION 文の TYPE 句と ADDRESS 句を使用して指定します。

パブリケーションを使用して、同期するデータを指定できます。この例では、RemoteOrders と OrderComments の各テーブル全体を指定します。

結果

リモート SQL Anywhere データベースが作成されます。

次の手順

「レッスン8:同期」187ページに進みます。

参照

- ●「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』
- ●「Mobile Link クライアント」『Mobile Link クライアント管理』
- 「CREATE SYNCHRONIZATION USER 文 [Mobile Link]」 『SQL Anywhere サーバ SQL リファ レンス』
- 「CREATE PUBLICATION 文 [Mobile Link] [SQL Remote]」 『SQL Anywhere サーバ SQL リファ レンス』
- ●「CREATE SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』

レッスン8:同期

dbmlsync ユーティリティを使用して、SQL Anywhere リモートデータベースの Mobile Link 同期 を開始します。dbmlsync を起動する前に、注文データとコメントをリモートデータベースに追 加します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:テキストファイルデータソースの設定」165ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:ダイレクトローハンドリン グの使用」164ページ

◆ タスク

1. まだ接続していない場合は、次のコマンドを実行して、Interactive SQL から Mobile Link クラ イアントデータベースに接続します。

dbisql -c "SERVER=remote1;UID=DBA;PWD=sql"

2. 次の文を実行して、クライアントデータベース内の RemoteOrders テーブルに注文を追加しま す。

INSERT INTO RemoteOrders (order_id, product_id, quantity, order_status) VALUES (1,12312,10,'new');

3. 次の文を Interactive SQL で実行して、クライアントデータベース内の OrderComments テーブ ルにコメントを追加します。

INSERT INTO OrderComments (comment_id, order_id, order_comment) VALUES (1,1,'send promotional material with the order');

4. Interactive SQL で次の文を実行して、変更をコミットします。

COMMIT;

5. 次のコマンドを実行します。

dbmlsync -c "SERVER=remote1;UID=DBA;PWD=sql" -o rem1.txt -v+

次のテーブルには、このレッスンで使用する各 dbmlsync オプションの説明が含まれています。

オプション	説明
-c	接続文字列を指定します。
-0	メッセージログファイル rem1.txt を指定します。
-v+	-v オプションは、ログを取る情報を指定します。-v+を使用して、最大冗 長ロギングをオンに設定します。

Mobile Link 同期クライアントの起動が完了すると、同期が成功したことを示す出力画面が表示されます。SQL ベースの同期によって、クライアントの RemoteOrders テーブル内のローが、統合データベース内の RemoteOrders テーブルに転送されます。

Java または .NET の処理によってコメントが orderComments.txt に挿入されました。

- 6. SQL Anywhere Mobile Link クライアントウィンドウをすべて閉じます。
- 7. 応答を orderResponses.txt に挿入してリモートデータベースにダウンロードします。この操作はサーバ側で行います。

次のテキストを order Responses.txt に追加します。エントリはタブ文字で区切ります。行末 で、[Enter] キーを押します。

- 1 1 Promotional material shipped
- 8. dbmlsync クライアントユーティリティを使用して同期を実行します。

この操作はクライアント側で行います。

次のコマンドを実行します。

dbmlsync -c "SERVER=remote1;UID=DBA;PWD=sql" -o rem1.txt -v+

Mobile Link クライアントユーティリティが表示されます。

注意

ダイレクトローハンドリングを使用してダウンロードされたローは、mlsrv16-v+オプション によっては出力されず、dbmlsync-v+オプションによってリモートログに出力されます。

Interactive SQL で、OrderComments テーブルを選択して、ローがダウンロードされたことを確認します。

次の SQL 文を実行します。

SELECT * FROM OrderComments;

結果

SQL Anywhere リモートデータベースが更新され、統合データベースと同期されます。

次の手順

「クリーンアップ」189ページに進みます。

参照

- ●「SQL Anywhere クライアント」『Mobile Link クライアント管理』
- ●「Mobile Link SQL Anywhere クライアントユーティリティ (dbmlsync)」『Mobile Link クライア ント管理』

クリーンアップ

チュートリアルをコンピュータから削除します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:テキストファイルデータソースの設定」165ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:ダイレクトローハンドリン グの使用」164ページ

◆タスク

- 1. Interactive SQL のすべてのインスタンスを閉じます。
- 2. SQL Anywhere、Mobile Link、同期クライアントの各ウィンドウを閉じます。
- 3. 次の手順で、チュートリアルに関連するすべての ODBC データソースを削除します。
 - a. ODBC アドミニストレータを起動します。
 次のコマンドを実行します。

odbcad32

- b. mldirect_db データソースを削除します。
- 4. 統合データベースとリモートデータベースを削除します。
 - a. 統合データベースとリモートデータベースが保存されているディレクトリに移動しま す。
 - b. MLconsolidated.db、MLconsolidated.log、remote1.db、remote1.log を削除します。

結果

チュートリアルがコンピュータから削除されます。

次の手順

なし。

チュートリアル: Microsoft Excel との同期

ダイレクトローハンドリングを使用して、リモートデータと中央のデータソース、アプリケーション、または Web サービスとの通信ができます。

このチュートリアルでは、ダイレクトローハンドリングを使用して、Microsoft Excel のスプレッドシートにあるデータを Mobile Link クライアントと同期する基本的な手順について説明します。Java 実装を例として使用し、Mobile Link ダイレクトローハンドリングを実装して、サポートされている統合データソース以外のデータソースを使用できるようにする方法について説明します。

必要なソフトウェア

•SQL Anywhere 16

●Java ソフトウェア開発キット

●Microsoft Office Excel 2007 以降

前提知識と経験

次の知識と経験が必要です。

●Java の知識

●Microsoft Excel の知識

●Mobile Link イベントスクリプトの基本的な知識

権限

統合データベースで次のロールおよび権限を持つ必要があります。

●SYS AUTH RESOURCE ROLE 互換ロール

●MONITOR システム権限

リモートデータベースで次のロールおよび権限を持つ必要があります。

●SYS REPLICATION ADMIN ROLE システムロール

●SYS RUN REPLICATION ROLE システムロール

概要

このチュートリアルでは、次の作業の方法について説明します。

●Java 用 Mobile Link サーバ API の使用

●Mobile Link ダイレクトローハンドリング用のメソッドの作成

●Java を使用した Microsoft Excel ワークシートにあるデータへのアクセス

参照

- ●「Mobile Link 同期」1 ページ
- ●「同期の方法」『Mobile Link サーバ管理』
- ●「ダイレクトローハンドリング」『Mobile Link サーバ管理』
- http://www.sybase.com/detail?id=1058600#319(このページを表示するには、Sybase.com アカウ ントが必要です。)
- http://sqlanywhere-forum.sybase.com

レッスン1: Excel ワークシートの設定

このレッスンでは、Excel ワークシートを作成し、Microsoft Excel ドライバを使用して ODBC デー タソースを定義します。Excel ワークシートには製品情報を格納します。

前提条件

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Microsoft Excel との同期」 190ページ

Microsoft Excel ドライバは 32 ビットドライバであるため、このチュートリアルでは、32 ビット バージョンの ODBC データソースアドミニストレータが必要です。

◆ タスク

- 1. サーバサイドコンポーネント用に c:¥MLobjexcel という名前の作業ディレクトリを作成します。
- 2. Microsoft Excel を開き、新しいワークブックを作成します。
- 3. デフォルトのワークシートで、A、B、Cの各カラムヘッダで次の内容を追加します。

comment_id	order_id	order_comment
2	1	出荷する宣伝用資料
3	1	必要な資料の詳細

- 4. デフォルトのワークシート名 Sheet1 を order_sheet に変更します。
 - a. [Sheet1] タブをダブルクリックします。
 - b. order_sheet と入力します。
- 5. Excel ワークブックを保存します。

ワークブックを c:¥MLobjexcel に order_central.xlsx という名前で保存します。

6. Microsoft Excel ドライバを使用して ODBC データソースを作成します。

- a. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [ODBC データソースア ドミニストレータ] をクリックします。
- b. [ユーザ DSN] タブをクリックします。
- c. [追加] をクリックします。
- d. [Microsoft Excel Driver (*.xls、*.xlsx、*.xlsm、*.xlsb)] をクリックします。
- e. [完了] をクリックします。
- f. [データ ソース名] フィールドに excel_datasource と入力します。
- g. [ブックの選択] をクリックし、*c:¥MLobjexcel¥order_central.xlsx* (ワークシートが含まれる ファイル)を選択します。
- h. [読み込み専用] オプションをクリアします。
- i. 開いているすべての [ODBC データソースアドミニストレータ] ウィンドウで [OK] をク リックします。

結果

Excel ワークシートと ODBC データソースが作成されます。

次の手順

「レッスン2: Mobile Link 統合データベースの設定」192ページに進みます。

レッスン2: Mobile Link 統合データベースの設定

このレッスンでは、データベースを作成し、ODBC データソースを定義します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Excel ワークシートの設定」191ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Microsoft Excel との同期」 190ページ

内容と備考

Mobile Link 統合データベースはデータの中央レポジトリであり、同期処理の管理に使用する Mobile Link のシステムテーブルとストアドプロシージャが含まれます。ダイレクトローハンド リングでは、統合データベース以外のデータソースと同期しますが、Mobile Link サーバが使用 する情報を保持するために統合データベースも必要です。

注意

Mobile Link 統合データベースを Mobile Link システムオブジェクトと DSN を使用して設定済みの場合は、このレッスンは省略できます。

◆タスク

- 1. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [Sybase Central] をクリック します。
- 2. [ツール] » [SQL Anywhere 16] » [データベースの作成] をクリックします。
- 3. [次へ] をクリックします。
- [このコンピュータにデータベースを作成] をデフォルトのままにし、[次へ] をクリックします。
- 5. [メインデータベースファイルを保存] フィールドに、データベースのファイル名およびパス を入力します。たとえば、c:¥MLobjexcel¥MLconsolidated.db と入力します。
- 6. データベース作成ウィザードの残りの指示に従い、デフォルト値をそのまま使用します。 DBA ユーザのユーザ ID とパスワードを指定するように求められたら、それぞれ DBA と sql を入力します。

[データベースへの接続] ページで、[最終切断後にデータベースを停止] オプションをオフに します。

7. [完了] をクリックします。

ML consolidated データベースが Sybase Central に表示されます。

- Sybase Central で、[ツール] » [SQL Anywhere 16] » [ODBC アドミニストレータを開く] をク リックします。
- 9. [ユーザ DSN] タブをクリックしてから、[追加] をクリックします。
- 10. [データソースの新規作成] ウィンドウで、[SQL Anywhere 16] をクリックし、[完了] をクリッ クします。
- 11. [SQL Anywhere の ODBC 設定] ウィンドウで、次の操作を行います。
 - a. [ODBC] タブをクリックします。
 - b. [データソース名] フィールドに mlexcel_db と入力します。
 - c. [ログイン] タブをクリックします。
 - d. [ユーザ ID] フィールドに DBA と入力します。
 - e. [パスワード] フィールドに sql と入力します。
 - f. [アクション] ドロップダウンリストで、[このコンピュータで稼働しているデータベース に接続] をクリックします。
 - g. [サーバ名] フィールドに、ML consolidated と入力します。
 - h. [OK] をクリックします。
- 12. ODBC データソースアドミニストレータを閉じます。

[ODBC データソースアドミニストレータ] ウィンドウで [OK] をクリックします。

結果

統合データベースと統合データベース用の ODBC データソースが作成されます。

次の手順

「レッスン3: Mobile Link 統合データベースでのテーブルの作成」194ページに進みます。

参照

- ●「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』
- ●「CREATE TABLE 文」『SQL Anywhere サーバ SQL リファレンス』
- ●「Mobile Link 統合データベース」『Mobile Link サーバ管理』

レッスン 3: Mobile Link 統合データベースでのテーブルの作成

このレッスンでは、Mobile Link 統合データベースに RemoteOrders テーブルを作成します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1: Excel ワークシートの設定」191ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Microsoft Excel との同期」 190ページ

内容と備考

このレッスンで作成する RemoteOrders テーブルには、次のカラムが含まれます。

order_id 注文のユニークな識別子です。

product_id 製品のユニークな識別子です。

quantity 品目の販売数です。

order_status 注文のステータスです。

last_modified ローが最後に変更された日です。このカラムはタイムスタンプベースのダウン ロードに使用します。このダウンロード方法は、効率的な同期のためにローをフィルタする一般 的な方法です。

◆タスク

1. Interactive SQL からデータベースに接続します。

Interactive SQL は、Sybase Central またはコマンドプロンプトから起動できます。

●Sybase Central から Interactive SQL を起動するには、MLconsolidated - DBA データベースを 右クリックし、[Interactive SQL を開く] をクリックします。 ●コマンドプロンプトで Interactive SQL を起動するには、次のコマンドを実行します。

```
dbisql -c "DSN=mlexcel db"
```

2. Interactive SQL で次の SQL 文を実行し、RemoteOrders テーブルを作成します。

```
CREATE TABLE RemoteOrders (
order_id INTEGER NOT NULL,
product_id INTEGER NOT NULL,
quantity INTEGER,
order_status VARCHAR(10) DEFAULT 'new',
last_modified TIMESTAMP DEFAULT CURRENT TIMESTAMP,
PRIMARY KEY(order_id)
);
```

Interactive SQL によって、統合データベースに RemoteOrders テーブルが作成されます。

3. Interactive SQL で次の SQL 文を実行し、Mobile Link のシステムテーブルとストアドプロシー ジャを作成します。

*C:¥Program Files¥SQL Anywhere 16¥*は、SQL Anywhere 16 インストール環境のロケーション に置き換えてください。

READ "C: ¥Program Files ¥SQL Anywhere 16¥MobiLink ¥setup ¥syncsa.sql";

Interactive SQL によって *syncsa.sql* が統合データベースに適用されます。*syncsa.sql* を実行すると、前に **ml**_ が付いた一連のシステムテーブルとストアドプロシージャが作成されます。これらのテーブルとストアドプロシージャは、同期処理中に Mobile Link サーバによって使用されます。

結果

統合データベースに RemoteOrders テーブルが作成されます。

次の手順

「レッスン4:同期スクリプトの追加」195ページに進みます。

レッスン4:同期スクリプトの追加

このレッスンでは、SQL ローハンドリングとダイレクトローハンドリング用のスクリプトを統 合データベースに追加します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1: Excel ワークシートの設定」191ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Microsoft Excel との同期」 190ページ

内容と備考

SQL ローハンドリングを使用すると、リモートデータを、Mobile Link 統合データベース内のテー ブルと同期できます。SQL ベースのスクリプトでは、次の情報を定義します。

●Mobile Link クライアントからアップロードするデータを統合データベースに適用する方法。

●統合データベースからダウンロードするデータ。

このレッスンでは、次の SQL ベースのアップロードイベントとダウンロードイベント用の同期 スクリプトを作成します。

- upload_insert このイベントは、クライアントデータベースに挿入された新しい注文を統合 データベースに適用する方法を定義します。
- download_cursor このイベントは、リモートクライアントにダウンロードする注文を定義 します。
- download_delete_cursor このイベントは、アップロード専用の同期スクリプトを使用する 場合に必要です。このチュートリアルでは、このイベントを無視するように Mobile Link サー バを設定します。

ダイレクトローハンドリングを使用して特別な処理を SQL ベースの同期システムに追加しま す。この手順では、handle_UploadData、handle_DownloadData、download_cursor、 download_delete_cursor の各イベントに対応するメソッド名を登録します。独自の Java クラスを レッスンの後半で作成します。

◆タスク

1. 統合データベースに接続していない場合は、Interactive SQL から接続します。

次のコマンドを実行します。

dbisql -c "DSN=mlexcel_db"

2. ml_add_table_script ストアドプロシージャを使用して、upload_insert、download_cursor、 download delete cursorの各イベント用の SQL ベースのテーブルスクリプトを追加します。

Interactive SQL で次の SQL 文を実行します。upload_insert のスクリプトでは、アップロード された order_id、product_id、quantity、order_status を Mobile Link 統合データベースに挿入し ます。download_cursor のスクリプトでは、タイムスタンプベースのフィルタを使用して、更 新されたローをリモートクライアントにダウンロードします。

CALL ml_add_table_script('default', 'RemoteOrders', 'upload_insert', 'INSERT INTO RemoteOrders(order_id, product_id, quantity, order_status) VALUES({ml r.order_id}, {ml r.product_id}, {ml r.quantity}, {ml r.order_status})'); CALL ml_add_table_script('default', 'RemoteOrders', 'download_cursor', 'SELECT order id, product id, quantity, order status

FROM RemoteOrders WHERE last modified >= {ml s.last table download}');

CALL ml_add_table_script('default', 'RemoteOrders', 'download_delete_cursor', '--{ml_ignore}');

COMMIT

3. handle UploadData と handle DownloadData の各イベント用の Java メソッドを登録します。

Interactive SQL で次の SQL 文を実行します。

Interactive SQL によって、GetUpload と SetDownload の各メソッドが、handle_UploadData と handle_DownloadData の各イベント用にそれぞれ登録されます。これらのメソッドは次の レッスンで作成します。

4. download cursor と download delete cursor の各イベントを登録します。

Interactive SQL で次の SQL スクリプトを実行します。

CALL ml_add_table_script('default', 'OrderComments', 'download_cursor', '--{ml_ignore}');

CALL ml_add_table_script('default', 'OrderComments', 'download_delete_cursor', '--{ml_ignore}');

同期は双方向であり、アップロード専用ではないため、スクリプトを使用するときには、 OrderComments テーブル用に download_cursor と download_delete_cursor の各イベントを登録 してください。 「必要なスクリプト」『Mobile Link サーバ管理』を参照してください。

5. これまでの変更内容をコミットします。

Interactive SQL で次の SQL 文を実行します。

COMMIT;

結果

upload_insert、download_cursor、download_delete_cursor の各イベントがデータベースに追加され ます。handle_UploadData、handle_DownloadData、download_cursor、および download_delete_cursor イベントに対応するメソッド名が登録されます。

次の手順

「レッスン 5: Mobile Link のダイレクトローハンドリングのための Java クラスの作成」 198 ページに進みます。

参照

- ●「Mobile Link イベントの概要」『Mobile Link サーバ管理』
- ●「スクリプトの追加と削除」『Mobile Link サーバ管理』
- ●「ローをアップロードするスクリプト」『Mobile Link サーバ管理』
- ●「ローをダウンロードするスクリプト」『Mobile Link サーバ管理』
- ●「upload_insert テーブルイベント」『Mobile Link サーバ管理』
- ●「upload_update テーブルイベント」『Mobile Link サーバ管理』
- 「upload_delete テーブルイベント」 『Mobile Link サーバ管理』
- ●「download_cursor テーブルイベント」『Mobile Link サーバ管理』
- ●「download_delete_cursor テーブルイベント」『Mobile Link サーバ管理』
- ●「ダイレクトローハンドリング」『Mobile Link サーバ管理』
- ●「ダイレクトアップロード」『Mobile Link サーバ管理』
- ●「ダイレクトダウンロード」『Mobile Link サーバ管理』
- ●「タイムスタンプベースのダウンロードの実装」『Mobile Link サーバ管理』
- ●「リモートデータベース間でローを分割する」『Mobile Link サーバ管理』

レッスン 5: Mobile Link のダイレクトローハンドリングのための Java クラスの作成

このレッスンでは、ダイレクトローハンドリングを使用して、クライアントデータベース内の OrderComments テーブルのローを処理します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Excel ワークシートの設定」191ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Microsoft Excel との同期」 190ページ

内容と備考

このレッスンでは、ダイレクトローハンドリング用に次のメソッドを追加します。

- **GetUpload** このメソッドは handle_UploadData イベントに使用します。GetUpload では、 アップロードされたコメントを order_central.xlsx という Excel ワークシートに書き込みま す。
- SetDownload このメソッドは handle_DownloadData イベントに使用します。SetDownload は、Excel ワークシート order_central.xlsx に格納されたデータを取り出し、リモートクライア ントに送信します。

次の手順では、処理用メソッドを含む Java クラスを作成する方法を示します。 完全なリストについては、「MobiLinkOrders コードの全リスト (Java)」202 ページを参照してください。

◆ タスク

1. MobiLinkOrders という新しいクラスの作成を開始します。

```
次のコードを作成します。
```

import ianywhere.ml.script.*;
import java.io.*;
import java.sql.*;

public class MobiLinkOrders {

2. クラスレベルの DBConnectionContext インスタンスを宣言します。

次のコードを追加します。

// Class level DBConnectionContext
DBConnectionContext _cc;

Mobile Link サーバによって DBConnectionContext のインスタンスがクラスコンストラクタに 渡されます。DBConnectionContext には、Mobile Link 統合データベースとの現在の接続に関 する情報がカプセル化されます。

3. クラスコンストラクタを作成します。

クラスコンストラクタが、クラスレベルの DBConnectionContext インスタンスを設定します。

次のコードを追加します。

```
public MobiLinkOrders( DBConnectionContext cc )
  throws IOException, FileNotFoundException {
    // Declare a class-level DBConnectionContext
    _cc = cc;
}
```

4. GetUpload メソッドを作成します。

GetUpload メソッドでは、OrderComments テーブルを表す UploadedTableData クラスインスタンスを取得します。OrderComments テーブルには、遠隔地の営業部員による特別なコメントが含まれます。このテーブルはレッスンの後半で作成します。

UploadedTableData の getInserts メソッドでは、注文に対する新しいコメントの結果セットを 返します。

次のコードを追加します。

```
// Method for the handle_UploadData synchronization event
public void GetUpload( UploadData ut )
    throws SQLException, IOException {
        // Get an UploadedTableData for OrderComments
        UploadedTableData orderCommentsTbl = ut.getUploadedTableByName("OrderComments");
        // Get inserts uploaded by the MobiLink client
        ResultSet insertResultSet = orderCommentsTbl.getInserts();
        try {
            // Connect to the excel worksheet through ODBC
```

```
Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
  Connection con = DriverManager.getConnection( "jdbc:odbc:excel datasource" );
  while( insertResultSet.next() ) {
     // Get order comments
     int commentID = insertResultSet.getInt("comment id");
     int orderID = insertResultSet.getInt("order id");
     String specialComments = insertResultSet.getString("order comment");
     // Execute an insert statement to add the order comment to the worksheet
     PreparedStatement st = con.prepareStatement("INSERT INTO [order sheet$]"
       + "(order id, comment id, order comment) VALUES (?,?,?)");
     st.setString(1, Integer.toString(_orderID));
st.setString(2, Integer.toString(_commentID));
     st.setString(3, specialComments);
     st.executeUpdate();
     st.close();
  con.close();
} catch(Exception ex) {
  System.err.print("Exception: ");
  System.err.println(ex.getMessage());
} finally {
insertResultSet.close();
```

- 5. SetDownload メソッドを作成します。
 - a. OrderComments テーブルを表すクラスインスタンスを取得します。

DBConnectionContext の getDownloadData メソッドを使用して DownloadData のインスタ ンスを取得します。DownloadData の getDownloadTableByName メソッドを使用して、 OrderComments テーブルの DownloadTableData インスタンスを返します。

次のコードを追加します。

```
public void SetDownload() throws SQLException, IOException {
    DownloadData download_d = _cc.getDownloadData();
    DownloadTableData download_td =
    download_d.getDownloadTableByName( "OrderComments" );
```

注意

```
このテーブルは、「レッスン7: Mobile Link クライアントデータベースの設定」205ページ
でリモートデータベースに作成します。
```

b. 準備文または IDbCommand を取得します。これを使用すると、ダウンロードに挿入操作 や更新操作を追加できます。

DownloadTableDataのgetUpsertPreparedStatementメソッドを使用してjava.sql.PreparedStatementのインスタンスを返します。

次のコードを追加します。

// Prepared statement to compile upserts (inserts or updates).
PreparedStatement download_upserts = download_td.getUpsertPreparedStatement();

c. 各ローのダウンロードデータを設定します。

次のコードでは、order_central.xlsx ワークシートを参照して、Mobile Link ダウンロード にデータを追加しています。

次のコードを追加します。 try { // Connect to the excel worksheet through ODBC // Connect to the excel worksheet through ODBC Class.forName("sun.jdbc.odbc.JdbcOdbcDriver"); Connection con = DriverManager.getConnection("jdbc:odbc:excel datasource"); // Retrieve all the rows in the worksheet Statement st = con.createStatement(); ResultSet Excel rs = st.executeQuery("select * from [order sheet\$]"); while (Excel_rs.next()) { // Retrieve the row data int Excel_comment_id = Excel_rs.getInt(1); int Excel_order_id = Excel_rs.getInt(2); String Excel comment = Excel rs.getString(3); // Add the Excel data to the MobiLink download. download_upserts.setInt(1, Excel_comment_id); download_upserts.setInt(2, Excel_order_id); download upserts.setString(3, Excel comment); download upserts.executeUpdate(); } // Close the excel result set, statement, and connection. Excel rs.close(); st.close(); con.close(); } catch (Exception ex) { System.err.print("Exception: "); System.err.println(ex.getMessage()); d. ダウンロードに挿入操作または更新操作を追加する準備文を終了し、メソッドとクラス を終了します。 次のコードを追加します。 finally { download upserts.close(); }

6. Java コードを *MobiLinkOrders.java* という名前で作業ディレクトリ *c*:¥MLobjexcel に保存します。

MobiLinkOrders.java のコードを検証する場合は、「MobiLinkOrders コードの全リスト (Java)」 202 ページを参照してください。

- 7. クラスファイルをコンパイルします
 - a. Java のソースファイルが含まれるディレクトリに移動します。
 - b. Java 用の Mobile Link サーバ API ライブラリを参照する MobiLinkOrders をコンパイルします。

%SQLANY16%¥Java にある mlscript.jar を参照する必要があります。

次のコマンドを実行して、C:¥Program Files¥SQL Anywhere 16¥を SQL Anywhere 16ディレクトリに置き換えます。

javac -classpath "C:¥Program Files¥SQL Anywhere 16¥java¥mlscript.jar" MobiLinkOrders.java

結果

クライアントデータベース内の OrderComments テーブルのローが更新されます。

次の手順

「レッスン6: Mobile Link サーバの起動」204ページに進みます。

参照

- ●「ダイレクトローハンドリング」『Mobile Link サーバ管理』
- ●「Java による同期スクリプトの作成」『Mobile Link サーバ管理』

MobiLinkOrders コードの全リスト (Java)

```
このチュートリアルで使用している Java の MobiLinkOrders クラスの全コードを次に示します。
手順を追った説明については、「レッスン 5: Mobile Link のダイレクトローハンドリングのため
の Java クラスの作成」198 ページを参照してください。
```

```
import ianywhere.ml.script.*;
  import java.io.*;
  import java.sql.*;
  public class MobiLinkOrders {
  // Class level DBConnectionContext
  DBConnectionContext cc;
  public MobiLinkOrders( DBConnectionContext cc )
    throws IOException, FileNotFoundException {
    // Declare a class-level DBConnectionContext
    _cc = cc;
  // Method for the handle UploadData synchronization event
  public void GetUpload( UploadData ut )
    throws SQLException, IOException {
    // Get an UploadedTableData for OrderComments
    UploadedTableData orderCommentsTbl = ut.getUploadedTableByName("OrderComments");
    // Get inserts uploaded by the MobiLink client
    ResultSet insertResultSet = orderCommentsTbl.getInserts();
    try {
       // Connect to the excel worksheet through ODBC
       Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
       Connection con = DriverManager.getConnection( "jdbc:odbc:excel datasource" );
       while( insertResultSet.next() ) {
         // Get order comments
         int commentID = insertResultSet.getInt("comment id");
         int orderID = insertResultSet.getInt("order id");
         String specialComments = insertResultSet.getString("order comment");
         // Execute an insert statement to add the order comment to the worksheet
         PreparedStatement st = con.prepareStatement("INSERT INTO [order sheet$]"
```

```
+ "(order id, comment id, order comment) VALUES (?,?,?)" );
           st.setString(1, Integer.toString(_orderID));
st.setString(2, Integer.toString(_commentID));
st.setString(3, _specialComments);
           st.executeUpdate();
           st.close();
        con.close();
     } catch(Exception ex) {
        System.err.print("Exception: ");
        System.err.println(ex.getMessage());
     } finally {
     insertResultSet.close();
  public void SetDownload() throws SQLException, IOException {
     DownloadData download_d = _cc.getDownloadData();
DownloadTableData download_td = download_d.getDownloadTableByName( "OrderComments" );
     // Prepared statement to compile upserts (inserts or updates).
     PreparedStatement download upserts = download td.getUpsertPreparedStatement();
     try {
// Connect to the excel worksheet through ODBC
// Connect to the excel worksheet through ODBC
        Class.forName("sun.jdbc.odbc.JdbcOdbcDriver");
        Connection con = DriverManager.getConnection( "jdbc:odbc:excel datasource" );
        // Retrieve all the rows in the worksheet
        Statement st = con.createStatement();
        ResultSet Excel rs = st.executeQuery( "select * from [order sheet$]" );
        while (Excel rs.next()) {
           // Retrieve the row data
           int Excel comment id = Excel rs.getInt(1);
           int Excel order id = Excel rs.getInt(2);
           String Excel comment = Excel rs.getString(3);
           // Add the Excel data to the MobiLink download.
           download_upserts.setInt( 1, Excel_comment_id );
download_upserts.setInt( 2, Excel_order_id );
           download upserts.setString( 3, Excel comment );
           download_upserts.executeUpdate();
        }
        // Close the excel result set, statement, and connection.
        Excel rs.close();
        st.close();
        con.close();
     } catch (Exception ex) {
        System.err.print("Exception: ");
        System.err.println(ex.getMessage());
     } finally {
     download upserts.close();
}
```

レッスン 6: Mobile Link サーバの起動

このレッスンでは、-c オプションを使用して Mobile Link サーバ (mlsrv16) を起動して統合データ ベースに接続し、-sl java オプションを使用して Java クラスをロードします。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Excel ワークシートの設定」191ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Microsoft Excel との同期」 190ページ

◆ タスク

● 統合データベースに接続し、mlsrv16のコマンドラインでクラスをロードします。

次のコマンドを実行します。*c:¥MLobjexcel*は、Java ソースファイルがある実際のディレクト リに置き換えてください。

mlsrv16 -c "DSN=mlexcel_db" -o serverOut.txt -v+ -dl -zu+ -x tcpip -sl java (-cp c:¥MLobjexcel)

Mobile Link サーバメッセージウィンドウが表示されます。

このチュートリアルで使用している Mobile Link サーバの各オプションの説明を次に示しま す。オプション-o、-v、-dl は、デバッグとトラブルシューティングの情報を提供します。こ れらのロギングオプションは、開発環境での使用に適しています。パフォーマンス上の理由 から、一般的に-v+と-dl は運用環境では使用しません。

オプション	説明
-c	続いて接続文字列を指定します。
-0	メッセージログファイル serverOut.txt を指 定します。
-v+	-v オプションは、ログを取る情報を指定します。-v+を使用して、最大冗長ロギングをオンに設定します。
-dl	画面にすべてのログメッセージを表示しま す。
-zu+	自動的に新しいユーザを追加します。
-X	Mobile Link クライアントの通信プロトコル とパラメータを設定します。

オプション	説明
-sl java	クラスファイルを検索する一連のディレク トリを指定し、またサーバ起動時に Java VM をロードします。

結果

Mobile Link サーバが起動し、ダイレクトローハンドリングの準備が整います。

次の手順

「レッスン7: Mobile Link クライアントデータベースの設定」205ページに進みます。

参照

●「Mobile Link サーバオプション」『Mobile Link サーバ管理』

●「-sl java mlsrv16 オプション」『Mobile Link サーバ管理』

レッスン7: Mobile Link クライアントデータベースの設定

このレッスンでは、SQL Anywhere データベースを統合データベースと Mobile Link クライアント に使用します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1: Excel ワークシートの設定」191ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Microsoft Excel との同期」 190ページ

内容と備考

また、このチュートリアルの目的上、Mobile Link クライアント、統合データベース、および Mobile Link サーバはすべて同じコンピュータに置きます。

Mobile Link クライアントデータベースを設定するには、RemoteOrders と OrderComments の各 テーブルを作成します。RemoteOrders テーブルは、統合データベースの RemoteOrders テーブル に対応します。Mobile Link サーバでは、SQL ベースのスクリプトを使用してリモート注文が同 期されます。OrderComments テーブルは、クライアントデータベースだけで使用されます。 Mobile Link サーバでは、特別なイベントを使用して OrderComments テーブルが処理されます。

◆ タスク

1. dbinit コマンドラインユーティリティを使用して、Mobile Link クライアントデータベースを 作成します。 次のコマンドを実行します。

dbinit -i -k -dba DBA,sql remote1

-i オプションと -k オプションは、それぞれ jConnect のサポートと Watcom SQL の互換ビュー を省略します。

2. dbsrv16 コマンドラインユーティリティを使用して、Mobile Link クライアントデータベース を起動します。

```
次のコマンドを実行します。
```

dbsrv16 remote1

3. Interactive SQL を使用して Mobile Link クライアントデータベースに接続します。

```
次のコマンドを実行します。
```

dbisql -c "SERVER=remote1;UID=DBA;PWD=sql"

4. RemoteOrders テーブルを作成します。

Interactive SQL で次の SQL 文を実行します。

```
CREATE TABLE RemoteOrders (
order_id INTEGER NOT NULL,
product_id INTEGER NOT NULL,
quantity INTEGER,
order_status VARCHAR(10) DEFAULT 'new',
PRIMARY KEY(order_id)
);
```

5. OrderComments テーブルを作成します。

Interactive SQL で次の SQL 文を実行します。

```
CREATE TABLE OrderComments (
comment_id INTEGER NOT NULL,
order_id INTEGER NOT NULL,
order_comment VARCHAR(255),
PRIMARY KEY(comment_id),
FOREIGN KEY(order_id) REFERENCES RemoteOrders(order_id)
);
```

6. Mobile Link 同期ユーザ、パブリケーション、サブスクリプションを作成します。

Interactive SQL で次の SQL 文を実行します。

```
CREATE SYNCHRONIZATION USER ml_sales1;
CREATE PUBLICATION order_publ (TABLE RemoteOrders, TABLE OrderComments);
CREATE SYNCHRONIZATION SUBSCRIPTION TO order_publ FOR ml_sales1
TYPE TCPIP ADDRESS 'host=localhost';
```

注意

Mobile Link サーバに接続する方法は、CREATE SYNCHRONIZATION SUBSCRIPTION 文の TYPE 句と ADDRESS 句を使用して指定します。 パブリケーションを使用して、同期するデータを指定できます。この例では、RemoteOrders と OrderComments のテーブルをすべて指定します。

結果

SQL Anywhere クライアントデータベースが作成され、同期の準備が行われます。

次の手順

「レッスン8:同期」207ページに進みます。

参照

- ●「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』
- ●「Mobile Link クライアント」『Mobile Link クライアント管理』
- 「CREATE SYNCHRONIZATION USER 文 [Mobile Link]」 『SQL Anywhere サーバ SQL リファ レンス』
- 「CREATE PUBLICATION 文 [Mobile Link] [SQL Remote]」 『SQL Anywhere サーバ SQL リファ レンス』
- ●「CREATE SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』

レッスン8:同期

dbmlsync ユーティリティを使用して、SQL Anywhere リモートデータベースの Mobile Link 同期 を開始します。dbmlsync を起動する前に、注文データとコメントをリモートデータベースに追 加します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1: Excel ワークシートの設定」191ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Microsoft Excel との同期」 190ページ

◆ タスク

1. Mobile Link クライアントデータベースに接続していない場合は、Interactive SQL から接続します。

次のコマンドを実行します。

dbisql -c "SERVER=remote1;UID=DBA;PWD=sql"

2. クライアントデータベース内の RemoteOrders テーブルに注文を追加します。

Interactive SQL で次の SQL 文を実行します。

INSERT INTO RemoteOrders (order_id, product_id, quantity, order_status) VALUES (1,12312,10,'new');

3. クライアントデータベース内の OrderComments テーブルにコメントを追加します。

Interactive SQL で次の SQL 文を実行します。

INSERT INTO OrderComments (comment_id, order_id, order_comment) VALUES (1,1,'send promotional material with the order');

4. これまでの変更内容をコミットします。

Interactive SQL で次の SQL 文を実行します。

COMMIT;

5. コマンドプロンプトで次のコマンドを実行します。

dbmlsync -c "SERVER=remote1;UID=DBA;PWD=sql" -o rem1.txt -v+

次の表は、使用されている各 dbmlsync オプションを説明しています。

オプション	説明
-c	接続文字列を指定します。
-0	メッセージログファイル rem1.txt を指定します。
-v+	-v オプションは、ログを取る情報を指定します。-v+を使用して、最大冗 長ロギングをオンに設定します。

Mobile Link 同期クライアントの起動が完了すると、同期が成功したことを示す出力画面が表示されます。SQL ベースの同期によって、クライアントの RemoteOrders テーブル内のローが、統合データベース内の RemoteOrders テーブルに転送されました。

Java の処理によって、コメントが order_central.xlsx ワークシートに挿入されました。 order central.xlsx ワークシートに格納された情報がクライアントにダウンロードされます。

Interactive SQL で、OrderComments テーブルを選択して、ローがダウンロードされたことを確認します。

Interactive SQL で次の SQL 文を実行します。

SELECT * FROM OrderComments;

注意

ダイレクトローハンドリングを使用してダウンロードされたローは、mlsrv16-v+オプション によっては出力されず、dbmlsync-v+オプションによってリモートログに出力されます。

結果

リモートデータベースの注文データとコメントが更新され、リモートデータベースと統合データ ベースが同期されます。

次の手順

「クリーンアップ」209ページに進みます。

参照

- ●「SQL Anywhere クライアント」『Mobile Link クライアント管理』
- 「Mobile Link SQL Anywhere クライアントユーティリティ (dbmlsync)」『Mobile Link クライア ント管理』

クリーンアップ

チュートリアルをコンピュータから削除します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1: Excel ワークシートの設定」191ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Microsoft Excel との同期」 190ページ

◆ タスク

1. 以下のアプリケーションのインスタンスをすべて閉じます。

●Interactive SQL

Microsoft Excel

- 2. Excel ワークブック order_central.xlsx を削除します。
- 3. SQL Anywhere、Mobile Link、同期クライアントの各ウィンドウを閉じます。
- 4. 次の手順で、チュートリアルに関連するすべての ODBC データソースを削除します。
 - a. ODBC アドミニストレータを起動します。 次のコマンドを実行します。

odbcad32

- b. excel_datasource と mlexcel_db の各データソースを削除します。
- 5. 統合データベースとリモートデータベースを削除します。
 - a. 統合データベースとリモートデータベースが保存されているディレクトリに移動しま す。
 - b. *MLconsolidated.db、MLconsolidated.log、remote1.db、remote1.log*を削除します。

結果

チュートリアルがコンピュータから削除されます。

次の手順

なし。

チュートリアル:XML との同期

このチュートリアルでは、XMLファイルとリモートクライアントの間でデータを同期する方法 を示します。

ダイレクトローハンドリングを使用して、リモートデータと中央のデータソース、アプリケーション、または Web サービスとの通信ができます。

このチュートリアルでは、Mobile Link ダイレクトローハンドリングを実装して、サポートされ ている統合データソース以外のデータソースを使用できるようにします。このチュートリアル では、例として Java 実装を使用します。

必要なソフトウェア

• SQL Anywhere 16

●Java ソフトウェア開発キット

●XML DOM ライブラリ

前提知識と経験

次の知識と経験が必要です。

●Java の知識

●XML の知識

●XML DOM の知識

●Mobile Link イベントスクリプトの基本的な知識

権限

統合データベースで次のロールおよび権限を持つ必要があります。

- ●SYS AUTH RESOURCE ROLE 互換ロール
- ●MONITOR システム権限

リモートデータベースで次のロールおよび権限を持つ必要があります。

●SYS REPLICATION ADMIN ROLE システムロール
●SYS_RUN_REPLICATION_ROLE システムロール

概要

このチュートリアルでは、次の作業の方法について説明します。

●Java 用 Mobile Link サーバ Java API の使用

●Mobile Link ダイレクトローハンドリング用のメソッドの作成

参照

- ●「Mobile Link 同期」1 ページ
- ●「同期の方法」『Mobile Link サーバ管理』
- ●「ダイレクトローハンドリング」『Mobile Link サーバ管理』
- http://www.sybase.com/detail?id=1058600#319(このページを表示するには、Sybase.com アカウ ントが必要です。)
- http://sqlanywhere-forum.sybase.com

レッスン1:XML データソースの設定

このレッスンでは、注文情報を保存する XML ファイルを作成します。

前提条件

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: XML との同期」210ページ

◆タスク

1. 次の内容の XML ファイルを作成します。

<?xml version="1.0" encoding="UTF-8"?> <orders></orders>

2. XML ファイルを保存します。

このチュートリアルでは、*c:¥MLobjxml*をサーバサイドコンポーネントの作業ディレクトリ とします。XML ファイルを order_comments.xml という名前でこのディレクトリに保存しま す。

結果

XML ファイルが作成されます。

次の手順

「レッスン2: Mobile Link 統合データベースの設定」212ページに進みます。

レッスン2: Mobile Link 統合データベースの設定

このレッスンでは、データベースを作成し、ODBC データソースを定義します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:XMLデータソースの設定」211ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:XML との同期」210ページ

内容と備考

Mobile Link 統合データベースはデータの中央レポジトリであり、同期処理の管理に使用する Mobile Link のシステムテーブルとストアドプロシージャが含まれます。ダイレクトローハンド リングでは、統合データベース以外のデータソースと同期しますが、Mobile Link サーバが使用 する情報を保持するために統合データベースも必要です。

注意

Mobile Link 統合データベースを Mobile Link システムオブジェクトと DSN を使用して設定済みの場合は、このレッスンは省略できます。

◆ タスク

1. Sybase Central を起動します。

[スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [Sybase Central] をクリックします。

- 2. [ツール] » [SQL Anywhere 16] » [データベースの作成] をクリックします。
- 3. [次へ] をクリックします。
- (このコンピュータにデータベースを作成)をデフォルトのままにし、[次へ]をクリックします。
- 5. [メインデータベースファイルを保存] フィールドに、データベースのファイル名およびパス を入力します。たとえば、c:¥MLobjxml¥MLconsolidated.db と入力します。[次へ] をクリック します。
- 6. データベース作成ウィザードの残りの指示に従い、デフォルト値をそのまま使用します。 DBA ユーザのユーザ ID とパスワードを指定するように求められたら、それぞれ DBA と sql を入力します。

[データベースへの接続] ページで、[最終切断後にデータベースを停止] オプションをオフに します。

7. [完了] をクリックします。

ML consolidated データベースが Sybase Central に表示されます。

- 8. ウィンドウが自動的に閉じない場合は、[データベースの作成] ウィンドウの [閉じる] をク リックします。
- 9. [ツール] » [SQL Anywhere 16] » [ODBC アドミニストレータを開く] をクリックします。
- 10. [ユーザ DSN] タブをクリックし、[追加] をクリックします。
- 11. [データソースの新規作成] ウィンドウで、[SQL Anywhere 16] をクリックし、[完了] をクリッ クします。
- 12. [SQL Anywhere の ODBC 設定] ウィンドウで、次の操作を行います。
 - a. [ODBC] タブをクリックします。
 - b. [データソース名] フィールドに mlxml db と入力します。
 - c. [ログイン] タブをクリックします。
 - d. [ユーザ ID] フィールドに DBA と入力します。
 - e. [パスワード] フィールドに sql と入力します。
 - f. [サーバ名] フィールドに、ML consolidated と入力します。
 - g. [OK] をクリックします。
- 13. ODBC データソースアドミニストレータを閉じます。

[ODBC データソースアドミニストレータ] ウィンドウで [OK] をクリックします。

結果

データベースが作成され、ODBC データソースが定義されます。

次の手順

「レッスン3: Mobile Link 統合データベースでのテーブルの作成」213ページに進みます。

参照

- ●「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』
- ●「Mobile Link 統合データベース」『Mobile Link サーバ管理』

レッスン 3: Mobile Link 統合データベースでのテーブルの作成

このレッスンでは、Mobile Link 統合データベースに RemoteOrders テーブルを作成します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:XMLデータソースの設定」211ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: XML との同期」210ページ

内容と備考

RemoteOrders テーブルには次のカラムが含まれます。

order_id 注文のユニークな識別子です。

product_id 製品のユニークな識別子です。

quantity 品目の販売数です。

order_status 注文のステータスです。

last_modified ローが最後に変更された日です。このカラムはタイムスタンプベースのダウン ロードに使用します。このダウンロード方法は、効率的な同期のためにローをフィルタする一般 的な方法です。

◆ タスク

1. Interactive SQL を使用してデータベースに接続します。

Interactive SQL は、Sybase Central またはコマンドプロンプトから起動できます。

●Sybase Central から Interactive SQL を起動するには、MLconsolidated - DBA データベースを 右クリックし、[Interactive SQL を開く] をクリックします。

●コマンドプロンプトで Interactive SQL を起動するには、次のコマンドを実行します。

dbisql -c "DSN=mlxml_db"

2. Interactive SQL で次の SQL 文を実行し、RemoteOrders テーブルを作成します。

```
CREATE TABLE RemoteOrders (
order_id INTEGER NOT NULL,
product_id INTEGER NOT NULL,
quantity INTEGER,
order_status VARCHAR(10) DEFAULT 'new',
last_modified TIMESTAMP DEFAULT CURRENT TIMESTAMP,
PRIMARY KEY(order_id)
);
```

Interactive SQL によって、統合データベースに RemoteOrders テーブルが作成されます。

3. Interactive SQL で次の文を実行して Mobile Link のシステムテーブルとストアドプロシージャを作成します。

*C:¥Program Files¥SQL Anywhere 16¥*は、SQL Anywhere 16 インストール環境のロケーション に置き換えてください。

READ "C: ¥Program Files ¥SQL Anywhere 16 ¥MobiLink ¥setup ¥syncsa.sql";

Interactive SQL によって *syncsa.sql* が統合データベースに適用されます。*syncsa.sql* を実行すると、前に**ml** が付いた一連のシステムテーブルとストアドプロシージャが作成されます。

これらのテーブルとストアドプロシージャは、同期処理中に Mobile Link サーバによって使用 されます。

結果

RemoteOrders テーブルが作成され、Mobile Link のシステムテーブルとストアドプロシージャが インストールされます

次の手順

「レッスン4:同期スクリプトの追加」215 ページに進みます。

参照

●「CREATE TABLE 文」『SQL Anywhere サーバ SQL リファレンス』

レッスン4:同期スクリプトの追加

このレッスンでは、SQL ローハンドリングとダイレクトローハンドリング用のスクリプトを統 合データベースに追加します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:XMLデータソースの設定」211ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: XML との同期」210ページ

内容と備考

SQL ローハンドリングを使用すると、リモートデータを、Mobile Link 統合データベース内のテー ブルと同期できます。SQL ベースのスクリプトでは、次の情報を定義します。

●Mobile Link クライアントからアップロードするデータを統合データベースに適用する方法。

●統合データベースからダウンロードするデータ。

このレッスンでは、次の SQL ベースのアップロードイベントとダウンロードイベント用の同期 スクリプトを作成します。

- upload_insert このイベントは、クライアントデータベースに挿入された新しい注文を統合 データベースに適用する方法を定義します。
- download_cursor このイベントは、リモートクライアントにダウンロードする注文を定義 します。
- download_delete_cursor このイベントは、アップロード専用の同期スクリプトを使用する 場合に必要です。このチュートリアルでは、このイベントを無視するように Mobile Link サー バを設定します。

ダイレクトローハンドリングを使用して特別な処理を SQL ベースの同期システムに追加しま す。このレッスンでは、handle_UploadData、download_cursor、download_delete_cursor の各イベン トに対応するメソッド名を登録します。 独自の Java クラスを「レッスン5: Mobile Link のダイ レクトローハンドリングのための Java クラスの作成」217 ページで作成します。

◆ タスク

1. 統合データベースに接続していない場合は、Interactive SQL で接続します。

次のコマンドを実行します。

dbisql -c "DSN=mlxml_db"

2. ml_add_table_script ストアドプロシージャを使用して、upload_insert、download_cursor、 download delete cursorの各イベント用の SQL ベースのテーブルスクリプトを追加します。

Interactive SQL で次の SQL 文を実行します。upload_insert のスクリプトでは、アップロード された order_id、product_id、quantity、order_status を Mobile Link 統合データベースに挿入し ます。download_cursor のスクリプトでは、タイムスタンプベースのフィルタを使用して、更 新されたローをリモートクライアントにダウンロードします。

CALL ml_add_table_script('default', 'RemoteOrders', 'upload_insert', 'INSERT INTO RemoteOrders(order_id, product_id, quantity, order_status) VALUES({ml r.order_id}, {ml r.product_id}, {ml r.quantity}, {ml r.order_status})');

CALL ml_add_table_script('default', 'RemoteOrders', 'download_cursor', 'SELECT order_id, product_id, quantity, order_status FROM RemoteOrders WHERE last_modified >= {ml s.last_table_download}');

CALL ml_add_table_script('default', 'RemoteOrders', 'download_delete_cursor', '--{ml_ignore}');

COMMIT;

3. handle_UploadData イベント用に Java メソッドを登録します。

Interactive SQL で次の SQL 文を実行します。

Interactive SQL によって、handle_UploadData イベント用の GetUpload メソッドが登録されま す。次のレッスンでは、挿入されたデータを Mobile Link クライアントデータベース内の OrderComments テーブルから取得する GetUpload メソッドを作成します。

4. download_cursor と download_delete_cursor の各イベントを登録します。

Interactive SQL で次の SQL 文を実行します。

CALL ml_add_table_script('default', 'OrderComments', 'download_cursor', '--{ml_ignore}');

CALL ml_add_table_script('default', 'OrderComments', 'download_delete_cursor', '--{ml_ignore}'); 同期は双方向であり、アップロード専用ではないため、スクリプトを使用するときには、 OrderComments テーブル用に download_cursor と download_delete_cursor の各イベントを登録 してください。 「必要なスクリプト」『Mobile Link サーバ管理』を参照してください。

5. これまでの変更内容をコミットします。

Interactive SQL で次の SQL 文を実行します。

COMMIT;

6. Interactive SQL を閉じます。

結果

handle_UploadData、handle_DownloadData、end_download、download_cursor、および download_delete_cursor イベントに対応するメソッド名が登録されます。

次の手順

「レッスン 5: Mobile Link のダイレクトローハンドリングのための Java クラスの作成」 217ページに進みます。

参照

「Mobile Link イベントの概要」『Mobile Link サーバ管理』
「スクリプトの追加と削除」『Mobile Link サーバ管理』
「ローをアップロードするスクリプト」『Mobile Link サーバ管理』
「ローをダウンロードするスクリプト」『Mobile Link サーバ管理』
「upload_insert テーブルイベント」『Mobile Link サーバ管理』
「upload_update テーブルイベント」『Mobile Link サーバ管理』
「upload_delete テーブルイベント」『Mobile Link サーバ管理』
「download_cursor テーブルイベント」『Mobile Link サーバ管理』
「download_delete_cursor テーブルイベント」『Mobile Link サーバ管理』
「ダイレクトローハンドリング」『Mobile Link サーバ管理』
「ダイレクトアップロード」『Mobile Link サーバ管理』
「ダイレクトダウンロード」『Mobile Link サーバ管理』
「タイムスタンプベースのダウンロードの実装」『Mobile Link サーバ管理』

●「リモートデータベース間でローを分割する」『Mobile Link サーバ管理』

レッスン 5: Mobile Link のダイレクトローハンドリングのための Java クラスの作成

このレッスンでは、ダイレクトローハンドリングを使用して、クライアントデータベース内の OrderComments テーブルのローを処理します。handle_UploadData イベント用のダイレクトロー ハンドリングの GetUpload メソッドを追加します。GetUpload では、アップロードされたコメン トを XML ファイルに書き込みます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:XMLデータソースの設定」211ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: XML との同期」210ページ

内容と備考

次の手順では、処理用メソッドを含む Java クラスを作成する方法を示します。 完全なリストについては、「MobiLinkOrders の Java コードのリスト」222 ページを参照してください。

◆ タスク

1. MobiLinkOrders というクラスを作成します。

次のコードを作成します。

import ianywhere.ml.script.*; import java.io.*; import java.sql.*;

import javax.xml.parsers.DocumentBuilder; import javax.xml.parsers.DocumentBuilderFactory; import javax.xml.parsers.ParserConfigurationException;

import org.xml.sax.SAXException;

import org.w3c.dom.Document; import org.w3c.dom.Element; import org.w3c.dom.Node; import org.w3c.dom.NodeList;

// For write operation
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.TransformerConfigurationException;
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;

public class MobiLinkOrders {

2. クラスレベルの DBConnectionContext インスタンスおよび Document インスタンスを宣言します。Document クラスは、XML 文書をオブジェクトとして表します。

次のコードを作成します。

// Class level DBConnectionContext
DBConnectionContext _cc;
Document doc;

Mobile Link サーバによって DBConnectionContext のインスタンスがクラスコンストラクタに 渡されます。DBConnectionContext には、Mobile Link 統合データベースとの現在の接続に関 する情報がカプセル化されます。 3. クラスコンストラクタを作成します。

クラスコンストラクタが、クラスレベルの DBConnectionContext インスタンスを設定します。

```
次のコードを作成します。
```

public MobiLinkOrders(DBConnectionContext cc) throws IOException, FileNotFoundException {
 // Declare a class-level DBConnectionContext
 _cc = cc;
}

4. GetUpload メソッドを作成します。

GetUpload メソッドでは、OrderComments テーブルを表す UploadedTableData クラスインスタンスを取得します。OrderComments テーブルには、遠隔地の営業部員による特別なコメントが含まれます。このテーブルはレッスンの後半で作成します。

UploadedTableData の getInserts メソッドでは、注文に対する新しいコメントの結果セットを 返します。

a. メソッドの宣言を作成します。

次のコードを作成します。

// Method for the handle_UploadData synchronization event
public void GetUpload(UploadData ut) throws SQLException, IOException {

b. アップロード済み挿入を Mobile Link クライアントから取得するコードを作成します。

次のコードを作成します。

// Get an UploadedTableData for the remote table UploadedTableData remoteOrdersTable = ut.getUploadedTableByName("OrderComments");

// Get inserts uploaded by the MobiLink client
// as a java.sql.ResultSet
ResultSet insertResultSet = remoteOrdersTable.getInserts();

c. 既存の XML ファイル order_comments.xml を読み込むコードを作成します。

次のコードを作成します。

try {
readDom("order_comments.xml");

d. すべてのアップロード済み挿入を XML ファイルに追加するコードを作成します。

```
次のコードを作成します。
```

```
// Write out each insert in the XML file
while( insertResultSet.next() ) {
    buildXML(insertResultSet);
}
```

e. XML ファイルに出力するコードを作成します。 次のコードを作成します。

```
writeXML();
}
```

f. ResultSet を閉じるコードを作成します。 次のコードを作成します。

```
finally {
    // Close the result set of uploaded inserts
    insertResultSet.close();
}
```

5. buildXML メソッドを作成します。

```
次のコードを作成します。
```

```
private void buildXML( ResultSet rs ) throws SQLException {
            int order_id = rs.getInt(1);
            int comment_id = rs.getInt(2);
            String order_comment = rs.getString(3);
            // Create the comment object to be added to the XML file
            Element comment = _doc.createElement("comment");
comment.setAttribute("id", Integer.toString(comment_id));
            comment.appendChild(_doc.createTextNode(order_comment));
            // Get the root element (orders)
            Element root = _doc.getDocumentElement();
            // Get each individual order
            NodeList rootChildren = root.getChildNodes();
            for(int i = 0; i < rootChildren.getLength(); i++) {</pre>
              // If the order exists, add the comment to the order
              Node n = rootChildren.item(i);
              if(n.getNodeType() == Node.ELEMENT_NODE) {
                 Element e = (Element) n;
                 int idIntVal = integer.parseInt(e.getAttribute("id"));
                 if(idIntVal == order id) {
                    e.appendChild(comment);
                    // The comment has been added to the file, so exit
                    // the function.
                    return;
                 }
              }
            }
            // If the order did not exist already, create it
            Element order = _doc.createElement("order");
order.setAttribute("id", Integer.toString(order_id));
            // Add the comment to the new order
            order.appendChild(comment);
            root.appendChild(order);
         }
6. writeXML メソッドを作成します。
    次のコードを作成します。
         private void writeXML() {
```

```
TransformerFactory tFactory = TransformerFactory.newInstance();
              Transformer transformer = tFactory.newTransformer();
              // The XML source is doc
              DOMSource source = new DOMSource( doc);
              // Write the xml data to order comments.xml
              StreamResult result = new StreamResult(new File("order comments.xml"));
              transformer.transform(source, result);
           } catch (TransformerConfigurationException tce) {
              // Error generated by the parser
System.out.println ("¥n** Transformer Factory error");
System.out.println(" " + tce.getMessage() );
              // Use the contained exception, if any
              Throwable x = tce;
              if (tce.getException() != null) x = tce.getException();
              x.printStackTrace();
           } catch (TransformerException te) {
              // Error generated by the parser
              System.out.println ("¥n** Transformation error");
System.out.println(" " + te.getMessage() );
              // Use the contained exception, if any
              Throwable x = te;
              if (te.getException() != null) x = te.getException();
              x.printStackTrace();
           }
         }
7. readDom メソッドを作成します。
    次のコードを作成します。
         private void readDom(String filename) {
           DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
           try {
              //parse the Document data into _doc
              DocumentBuilder builder = factory.newDocumentBuilder();
              _doc = builder.parse( new File(filename) );
           } catch (SAXException sxe) {
              // Error generated during parsing)
              Exception x = sxe;
              if (sxe.getException() != null) x = sxe.getException();
              x.printStackTrace();
           } catch (ParserConfigurationException pce) {
              // Parser with specified options can't be built
              pce.printStackTrace();
           } catch (IOException ioe) {
              // I/O error
              ioe.printStackTrace();
        }
      }
```

8. Java コードを *MobiLinkOrders.java* という名前で作業ディレクトリ *c:*¥*MLobjxml* に保存しま す。 *MobiLinkOrders.java* のコードを検証する場合は、「MobiLinkOrders の Java コードのリスト」 222 ページを参照してください。

- 9. クラスファイルをコンパイルします
 - a. Java のソースファイルが含まれるディレクトリに移動します。
 - b. Java 用の Mobile Link サーバ API ライブラリを参照する MobiLinkOrders をコンパイルします。

%*SQLANY16%*¥*Java* にある *mlscript.jar* を参照し、XML DOM ライブラリが正しくインストールされていることを確認する必要があります。

次のコマンドを実行して、C:¥Program Files¥SQL Anywhere 16¥を SQL Anywhere 16ディ レクトリに置き換えます。

javac -classpath "C:¥Program Files¥SQL Anywhere 16¥java¥mlscript.jar" MobiLinkOrders.java

結果

Mobile Link のダイレクトローハンドリングのための Java クラスが作成されます。

次の手順

「レッスン6: Mobile Link サーバの起動」225ページに進みます。

参照

- ●「ダイレクトローハンドリング」『Mobile Link サーバ管理』
- ●「Java による同期スクリプトの作成」『Mobile Link サーバ管理』

MobiLinkOrders の Java コードのリスト

このチュートリアルで使用している Java の MobiLinkOrders クラスの全コードを次に示します。 手順を追った説明については、「レッスン5: Mobile Link のダイレクトローハンドリングのため の Java クラスの作成」217ページを参照してください。

```
import ianywhere.ml.script.*;
import java.io.*;
import java.sql.*;
```

import javax.xml.parsers.DocumentBuilder; import javax.xml.parsers.DocumentBuilderFactory; import javax.xml.parsers.ParserConfigurationException;

import org.xml.sax.SAXException;

import org.w3c.dom.Document; import org.w3c.dom.Element; import org.w3c.dom.Node; import org.w3c.dom.NodeList;

// For write operation
import javax.xml.transform.Transformer;
import javax.xml.transform.TransformerException;
import javax.xml.transform.TransformerFactory;
import javax.xml.transform.TransformerConfigurationException;

```
import javax.xml.transform.dom.DOMSource;
import javax.xml.transform.stream.StreamResult;
public class MobiLinkOrders {
  // Class level DBConnectionContext
  DBConnectionContext cc;
  Document doc;
  public MobiLinkOrders( DBConnectionContext cc ) throws IOException, FileNotFoundException {
    // Declare a class-level DBConnectionContext
     _cc = cc;
  }
  // Method for the handle UploadData synchronization event
  public void GetUpload( UploadData ut ) throws SQLException, IOException {
    // Get an UploadedTableData for the remote table
    UploadedTableData remoteOrdersTable = ut.getUploadedTableByName("OrderComments");
    // Get inserts uploaded by the MobiLink client
    // as a java.sql.ResultSet
    ResultSet insertResultSet = remoteOrdersTable.getInserts();
  try {
    readDom("order comments.xml");
    // Write out each insert in the XML file
    while( insertResultSet.next() ) {
    buildXML(insertResultSet);
    writeXML();
  } finally {
    // Close the result set of uploaded inserts
    insertResultSet.close();
  private void buildXML( ResultSet rs ) throws SQLException {
    int order_id = rs.getInt(1);
    int comment id = rs.getInt(2);
     String order comment = rs.getString(3);
    // Create the comment object to be added to the XML file
    Element comment = doc.createElement("comment");
    comment.setAttribute("id", Integer.toString(comment_id));
    comment.appendChild(_doc.createTextNode(order_comment));
     // Get the root element (orders)
     Element root = doc.getDocumentElement();
    // Get each individual order
    NodeList rootChildren = root.getChildNodes();
    for(int i = 0; i < rootChildren.getLength(); i++) {</pre>
       If the order exists, add the comment to the order
       Node n = rootChildren.item(i);
       if(n.getNodeType() == Node.ELEMENT NODE) {
         Element e = (Element) n;
         int idIntVal = Integer.parseInt(e.getAttribute("id"));
         if(idIntVal == order id) {
            e.appendChild(comment);
            // The comment has been added to the file, so exit
```

```
// the function
           return;
        }
     }
  }
  // If the order did not exist already, create it
  Element order = _doc.createElement("order");
order.setAttribute("id", Integer.toString(order_id));
  // Add the comment to the new order
  order.appendChild(comment);
  root.appendChild(order);
}
private void writeXML() {
  try {
// Use a Transformer for output
     TransformerFactory tFactory = TransformerFactory.newInstance();
     Transformer transformer = tFactory.newTransformer();
     // The XML source is doc
     DOMSource source = new DOMSource( doc);
     // Write the xml data to order comments.xml
     StreamResult result = new StreamResult(new File("order comments.xml"));
     transformer.transform(source, result);
  } catch (TransformerConfigurationException tce) {
     // Error generated by the parser
System.out.println ("¥n** Transformer Factory error");
System.out.println(" " + tce.getMessage() );
     // Use the contained exception, if any
     Throwable x = tce;
     if (tce.getException() != null) x = tce.getException();
     x.printStackTrace();
  } catch (TransformerException te) {
     // Error generated by the parser
System.out.println ("¥n** Transformation error");
System.out.println(" " + te.getMessage() );
     // Use the contained exception, if any
     Throwable x = te;
     if (te.getException() != null) x = te.getException();
     x.printStackTrace();
  }
}
private void readDom(String filename) {
  DocumentBuilderFactory factory = DocumentBuilderFactory.newInstance();
  try
     //parse the Document data into doc
     DocumentBuilder builder = factory.newDocumentBuilder();
     doc = builder.parse( new File(filename) );
  } catch (SAXException sxe) {
     // Error generated during parsing)
     Exception x = sxe;
     if (sxe.getException() != null) x = sxe.getException();
     x.printStackTrace();
  } catch (ParserConfigurationException pce) {
     // Parser with specified options can't be built
     pce.printStackTrace();
```

```
} catch (IOException ioe) {
    // I/O error
    ioe.printStackTrace();
    }
}
```

レッスン 6: Mobile Link サーバの起動

このレッスンでは、Mobile Link サーバを起動します。-c オプションを使用して Mobile Link サーバ (mlsrv16) を起動して統合データベースに接続し、-sl java オプションを使用して Java クラスを ロードします。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:XMLデータソースの設定」211ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: XML との同期」210ページ

◆タスク

● 統合データベースに接続し、mlsrv16のコマンドラインでクラスをロードします。

*c:¥MLobjxml*をソースファイルがある実際のディレクトリに置き換えて、次のコマンドを実行します。

mlsrv16 -c "DSN=mlxml_db" -o serverOut.txt -v+ -dl -zu+ -x tcpip -sl java (-cp c:¥MLobjxml)

Mobile Link サーバメッセージウィンドウが表示されます。

このチュートリアルで使用している Mobile Link サーバの各オプションの説明を次に示しま す。オプション-o、-v、-dl は、デバッグとトラブルシューティングの情報を提供します。こ れらのロギングオプションは、開発環境での使用に適しています。パフォーマンス上の理由 から、一般的に-v+と-dl は運用環境では使用しません。

オプション	説明
-c	続いて接続文字列を指定します。
-0	メッセージログファイル serverOut.txt を指 定します。
-v+	-v オプションは、ログを取る情報を指定します。-v+を使用して、最大冗長ロギングをオンに設定します。

オプション	説明
-dl	画面にすべてのログメッセージを表示しま す。
-zu+	自動的に新しいユーザを追加します。
-x	Mobile Link クライアントの通信プロトコル とパラメータを設定します。
-sl java	クラスファイルを検索する一連のディレク トリを指定し、またサーバ起動時に Java VM をロードします。

結果

Mobile Link サーバが起動します。

次の手順

「レッスン7: Mobile Link クライアントデータベースの設定」226ページに進みます。

参照

- ●「Mobile Link サーバオプション」『Mobile Link サーバ管理』
- ●「-sl java mlsrv16 オプション」『Mobile Link サーバ管理』

レッスン7: Mobile Link クライアントデータベースの設定

このレッスンでは、SQL Anywhere データベースを統合データベースと Mobile Link クライアント に使用します。また、このチュートリアルの目的上、Mobile Link クライアント、統合データベー ス、および Mobile Link サーバはすべて同じコンピュータに置きます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:XMLデータソースの設定」211ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。 「チュートリアル: XML との同期」210ページ

内容と備考

Mobile Link クライアントデータベースを設定するには、RemoteOrders と OrderComments の各 テーブルを作成します。RemoteOrders テーブルは、統合データベースの RemoteOrders テーブル に対応します。Mobile Link サーバでは、SQL ベースのスクリプトを使用してリモート注文が同 期されます。OrderComments テーブルは、クライアントデータベースだけで使用されます。 Mobile Link サーバでは、特別なイベントを使用して OrderComments テーブルが処理されます。

◆タスク

1. dbinit コマンドラインユーティリティを使用して、Mobile Link クライアントデータベースを 作成します。

c:¥MLobjxmlに移動して、次のコマンドを実行します。

dbinit -i -k -dba DBA,sql remote1

-i オプションと -k オプションは、それぞれ jConnect のサポートと Watcom SQL の互換ビュー を省略します。

2. dbsrv16 コマンドラインユーティリティを使用して、Mobile Link クライアントデータベース を起動します。

次のコマンドを実行します。

dbsrv16 remote1

3. Interactive SQL を使用して Mobile Link クライアントデータベースに接続します。

次のコマンドを実行します。

dbisql -c "SERVER=remote1;UID=DBA;PWD=sql"

4. RemoteOrders テーブルを作成します。

Interactive SQL で次の SQL 文を実行します。

```
CREATE TABLE RemoteOrders (
order_id INTEGER NOT NULL,
product_id INTEGER NOT NULL,
quantity INTEGER,
order_status VARCHAR(10) DEFAULT 'new',
PRIMARY KEY(order_id)
);
```

5. OrderComments テーブルを作成します。

Interactive SQL で次の SQL 文を実行します。

CREATE TABLE OrderComments (comment_id INTEGER NOT NULL, order_id INTEGER NOT NULL, order_comment VARCHAR(255), PRIMARY KEY(comment_id), FOREIGN KEY(order_id) REFERENCES RemoteOrders(order_id));

6. Mobile Link 同期ユーザ、パブリケーション、サブスクリプションを作成します。

Interactive SQL で次の SQL 文を実行します。

CREATE SYNCHRONIZATION USER ml_sales1; CREATE PUBLICATION order_publ (TABLE RemoteOrders, TABLE OrderComments); CREATE SYNCHRONIZATION SUBSCRIPTION TO order_publ FOR ml_sales1 TYPE TCPIP ADDRESS 'host=localhost';

注意

Mobile Link サーバに接続する方法は、CREATE SYNCHRONIZATION SUBSCRIPTION 文の TYPE 句と ADDRESS 句を使用して指定します。

パブリケーションを使用して、同期するデータを指定できます。この例では、RemoteOrders と OrderComments のテーブルをすべて指定します。

結果

リモートデータベースに作成され、同期用に設定されます。

次の手順

「レッスン8:同期」228ページに進みます。

参照

- ●「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』
- ●「Mobile Link クライアント」『Mobile Link クライアント管理』
- 「CREATE SYNCHRONIZATION USER 文 [Mobile Link]」 『SQL Anywhere サーバ SQL リファ レンス』
- 「CREATE PUBLICATION 文 [Mobile Link] [SQL Remote]」 『SQL Anywhere サーバ SQL リファ レンス』
- ●「CREATE SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』

レッスン8:同期

このレッスンでは、dbmlsync ユーティリティを使用して Mobile Link 同期を開始します。 dbmlsync を起動する前に、注文データとコメントをリモートデータベースに追加します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:XMLデータソースの設定」211ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル:XML との同期」210ページ

◆タスク

1. Mobile Link クライアントデータベースに接続していない場合は、Interactive SQL から接続します。

次のコマンドを実行します。

dbisql -c "SERVER=remote1;UID=DBA;PWD=sql"

2. クライアントデータベース内の RemoteOrders テーブルに注文を追加します。

Interactive SQL で次の SQL 文を実行します。

INSERT INTO RemoteOrders (order_id, product_id, quantity, order_status) VALUES (1,12312,10,'new');

3. クライアントデータベース内の OrderComments テーブルにコメントを追加します。

Interactive SQL で次の SQL 文を実行します。

INSERT INTO OrderComments (comment_id, order_id, order_comment) VALUES (1,1,'send promotional material with the order');

4. これまでの変更内容をコミットします。

Interactive SQL で次の SQL 文を実行します。

COMMIT;

5. コマンドプロンプトで次のコマンドを実行します。

dbmlsync -c "SERVER=remote1;UID=DBA;PWD=sql" -o rem1.txt -v+

次のテーブルには、このレッスンで使用する各 dbmlsync オプションの説明が含まれています。

オプション	説明
-c	接続文字列を指定します。
-0	メッセージログファイル rem1.txt を指定します。
-v+	-v オプションは、ログを取る情報を指定します。-v+を使用して、最大冗 長ロギングをオンに設定します。

Mobile Link 同期クライアントの起動が完了すると、同期が成功したことを示す出力画面が表示されます。

6. SQL ベースの同期によって、クライアントの RemoteOrders テーブル内のローが、統合データ ベース内の RemoteOrders テーブルに転送されました。

次の手順を実行して、クライアントの RemoteOrders テーブルに追加された情報が、統合デー タベース内の RemoteOrders テーブルに転送されたことを確認します。

a. コマンドプロンプトで Interactive SQL を起動するには、次のコマンドを実行します。

dbisql -c "DSN=mlxml_db"

b. Interactive SQL で次の SQL 文を実行します。

SELECT * FROM RemoteOrders;

7. Java の処理によってコメントが XML ファイルに挿入されました。

c:¥MLobjxml に移動し、テキストエディタで *order_comments.xml* を開いて、コメントが挿入されていることを確認します。

結果

リモートデータベースと統合データベースが同期されます。

次の手順

「クリーンアップ」230ページに進みます。

参照

- ●「SQL Anywhere クライアント」『Mobile Link クライアント管理』
- ●「Mobile Link SQL Anywhere クライアントユーティリティ (dbmlsync)」『Mobile Link クライア ント管理』

クリーンアップ

チュートリアルをコンピュータから削除します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:XMLデータソースの設定」211ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: XML との同期」210ページ

◆タスク

- 1. Interactive SQL のすべてのインスタンスを閉じます。
- 2. SQL Anywhere、Mobile Link、同期クライアントの各ウィンドウを閉じます。
- 3. 次の手順で、チュートリアルに関連するすべての ODBC データソースを削除します。
 - a. ODBC アドミニストレータを起動します。

次のコマンドを実行します。

odbcad32

- b. **mlxml_db** データソースを削除します。
- 4. 統合データベースとリモートデータベースを削除します。
 - a. 統合データベースとリモートデータベースが保存されているディレクトリに移動しま す。
 - b. MLconsolidated.db、MLconsolidated.log、remote1.db、remote1.logを削除します。

結果

チュートリアルがコンピュータから削除されます。

次の手順

なし。

チュートリアル:リモートデータベースの集中管理の使 用

このチュートリアルでは、リモートデータベースの集中管理の設定プロセスについての説明と、 いくつかの一般操作の実行方法が、順を追って示されます。

このチュートリアルに従って、集中管理を最初から設定したり、既存の同期システムに集中管理 を追加したりできます。チュートリアルでは、手順全体を通して、既存の同期システムに集中管 理を追加する場合に異なる処理が必要となる箇所では、そのことが示されます。

リモートデータベースの集中管理に関するいくつかの紹介ビデオやチュートリアル用ビデオが、 オンラインで提供されています。詳細については、http://www.sybase.jp/detail?id=1081142 を参照 してください。

注意

ビデオチュートリアルは、バージョン 12.0.0 の SQL Anywhere に基づいています。図やプロシージャのいくつかは SQL Anywhere 16.0 とは異なります。

必要なソフトウェア

このチュートリアルでは、チュートリアルを実行するローカルコンピュータに SQL Anywhere (Mobile Link と Sybase Central を含む) が完全にインストールされていることを前提としています。

Mobile Link エージェントの展開の詳細については、「SQL Anywhere Mobile Link クライアントの 配備」『Mobile Link サーバ管理』と「Ultra Light Mobile Link クライアントの配備」『Mobile Link サーバ管理』を参照してください。

権限

統合データベースで次のロールおよび権限を持つ必要があります。

●SYS_AUTH_RESOURCE_ROLE 互換ロール

●MONITOR システム権限

リモートデータベースで次のロールおよび権限を持つ必要があります。

●SYS REPLICATION ADMIN ROLE システムロール

●SYS RUN REPLICATION ROLE システムロール

概要

このチュートリアルでは、次の作業の方法について説明します。

- ●統合データベースと Mobile Link プロジェクトを作成します。
- ●Mobile Link サーバを起動し、Mobile Link ユーザとエージェントを定義し、リモートデバイス でエージェントを設定します。
- ●同期モデルを作成し、展開します。
- ●リモートタスクを操作します。

参照

●「リモートデータベースの集中管理」『Mobile Link サーバ管理』

レッスン1:統合データベースの作成

このレッスンでは、統合データベースを設定します。既存の同期システムがある場合は、「レッ スン2: Mobile Link プロジェクトの作成」233ページに進みます。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

◆タスク

 次のコマンドを実行して、このチュートリアルで使用するディレクトリを作成します。通常、 統合ディレクトリには、中央のサーバに存在するすべてのデータベースと他のファイルが含 まれます。

md c:¥cadmin_demo md c:¥cadmin_demo¥consolidated

2. SQL Anywhere 統合データベースと、それに接続する ODBC データソースを作成します。

cd c:¥cadmin_demo¥consolidated dbinit -dba DBA,sql consol.db start dbsrv16 consol.db dbdsn -w cadmin_tutorial_consol consol -y -c "UID=DBA;PWD=sql;DBF=consol.db;SERVER=consol" cd ..

3. Interactive SQL でデータベースに接続します。次のコマンドを実行します。

dbisql -c "DSN=cadmin_tutorial_consol"

4. Interactive SQL で次の文を実行して、*syncsa.sql* 設定スクリプトを使用して Mobile Link のシス テムテーブルとストアドプロシージャを作成します。*C:¥Program Files¥SQL Anywhere 16¥* を、SQL Anywhere 16 のインストール環境のロケーションに置き換えてください。

READ "C: ¥Program Files ¥SQL Anywhere 16 ¥MobiLink ¥setup ¥syncsa.sql";

5. Interactive SQL を閉じます。SQL 文を保存する必要はありません。

結果

SQL Anywhere データベースが作成され、これに接続されます。

次の手順

「レッスン2: Mobile Link プロジェクトの作成」233ページに進みます。

レッスン2: Mobile Link プロジェクトの作成

集中管理を行う場合は、Mobile Link プロジェクトを作成してください。プロジェクトは、集中 管理用に定義するさまざまなオブジェクトのコンテナとしての役割を果たします。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

◆タスク

- 1. Sybase Central を起動するには、[スタート]»[プログラム]»[SQL Anywhere 16]»[管理ツー ル]»[Sybase Central] をクリックします。
- 2. [ツール] » [Mobile Link 16] » [新しいプロジェクト] をクリックします。

[プロジェクト作成ウィザード] が表示されます。

- 3. [ようこそ] ページで、プロジェクト名を Central Admin Tutorial に変更し、プロジェクトファ イルのロケーションとして C:/cadmin を入力します。[次へ] をクリックします。
- 4. [統合データベースを指定] ページで、[データベースの表示名] に Tutorial を入力します。
- 5. [接続文字列] に次の値を入力します。

UID=DBA;PWD=sql;DSN=cadmin_tutorial_consol

6. [パスワードを記憶] を選択し、[次へ] をクリックします。

データベースにテーブルが存在しないために同期モデルが作成されないことを示す警告を受信する場合があります。[OK]をクリックします。

- 7. [リモートスキーマ名をプロジェクトに追加] を選択し、スキーマ名として Tutorial Application v1.0 と入力します。[次へ] をクリックします。
- 8. データベースタイプとして [SQL Anywhere] を選択し、[完了] をクリックします。

Mobile Link が今回初めて統合データベースを使用する場合、Mobile Link システム設定をイン ストールするかどうかを確認するメッセージが表示されます。Mobile Link システム設定を インストールすると、Mobile Link システムテーブルと Mobile Link システムプロシージャが 追加されます。[はい] をクリックし、[OK] をクリックします。

結果

Mobile Link プロジェクトが作成されます。

次の手順

「レッスン3: Mobile Link サーバの起動」234ページに進みます。

レッスン3: Mobile Link サーバの起動

このレッスンでは、Mobile Link サーバを起動します。Mobile Link サーバは、各リモートデバイ スの統合データベースとエージェントデータベースの間において、リモートデータベースのデー タの同期と、タスクおよびタスク結果の同期の両方を実行するために必要となります。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

既存の同期システムがある場合は、サーバがすでに稼働しているため、この項を省略してもかまいません。ただし、サーバコマンドラインで-ftrオプションと-ftruオプションが指定されていることを確認してください。これらのオプションは、リモートデバイスにファイルをダウンロードしたり、リモートデバイスからファイルをアップロードしたりする場合に必要です。

◆ タスク

● コマンドプロンプトで次のコマンドを実行します。

md c:¥cadmin_demo¥consolidated¥upload md c:¥cadmin_demo¥consolidated¥download cd c:¥cadmin_demo¥consolidated start mlsrv16.exe -c "DSN=cadmin_tutorial_consol;UID=DBA;PWD=sql" -ftr download -ftru upload -x tcpip(port=2439) -v+ -ot mlsrv.txt cd ..

次に、使用するオプションの概要を示します。

● -c 統合データベースに接続するために Mobile Link で使用される接続パラメータを指定 します。

- -ftr Mobile Link でダウンロードするファイルを検索するディレクトリを指定します。
- -ftru Mobile Link でアップロードされたファイルを保存するディレクトリを指定します。
- -x 同期クライアントを Mobile Link サーバに接続する方法を定義する通信パラメータを 指定します。
- -v+ 最大冗長を指定します。この設定はデバッグに役立ちますが、運用環境ではパフォーマンスが低下する場合があります。
- -ot Mobile Link 出力メッセージが記録されるファイルを指定します。

結果

Mobile Link サーバが起動され、リモートデバイスからアップロードまたはリモートデバイスに ダウンロードされるファイルが入るアップロードディレクトリおよびダウンロードディレクト リが作成されます。

次の手順

「レッスン4: Mobile Link ユーザの定義」235ページに進みます。

レッスン4: Mobile Link ユーザの定義

このレッスンでは、エージェントが使用する Mobile Link ユーザを定義します。既存の同期シス テムがあり、既存のいずれかの Mobile Link ユーザを Mobile Link エージェントが使用して同期す る場合は、この項を省略してもかまいません。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

エージェントがエージェントデータベースを同期する場合、Mobile Link サーバに対してエー ジェント自体を認証する必要があります。Mobile Link ユーザとオプションのパスワードを使用 することによって認証します。通常、リモートデータベースの同期に使用する Mobile Link ユー ザとパスワードは、エージェントデータベースの同期にエージェントが使用するものと同じで す。

◆タスク

1. Sybase Central で、[ビュー]»[フォルダ] をクリックします。

- 2. [Mobile Link 16] で、[Central Admin Tutorial]、[統合データベース]、Tutorial の順に展開し ます。
- 3. [ユーザ] を右クリックし、[新規] »[ユーザ] をクリックします。

ユーザ作成ウィザードが表示されます。

- 4. [ようこそ] ページで、新規ユーザの名前に JOHN と入力し、[次へ] をクリックします。
- 5. [認証] ページで、[このユーザは、標準 Mobile Link 認証を使用する場合、接続のためにパス ワードが必要です。] をオンにし、[パスワード] と [パスワードの確認] の両方のフィールド に sql と入力します。[完了] をクリックします。

同期しようとするエージェントを認証しない場合は、この手順を省略し、-zu+オプションを Mobile Link サーバコマンドラインに追加します。-zu+を指定すると、最初に同期を行おうと するときに、各 Mobile Link ユーザが登録されます。「-zu mlsrv16 オプション」『Mobile Link サーバ管理』を参照してください。

結果

Mobile Link ユーザが作成されます。

次の手順

「レッスン5:エージェントの定義」236ページに進みます。

レッスン5:エージェントの定義

このレッスンでは、エージェントを定義します。このエージェントは、リモートデバイスで実行 している Mobile Link エージェントのインスタンスを表します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

管理するリモートデバイスごとに別のエージェントを作成してください。

◆タスク

1. [Mobile Link 16] で、[Central Admin Tutorial]、[統合データベース]、Tutorial の順に展開し ます。 2. [エージェント] を右クリックし、[新規] » [エージェント] をクリックします。

Mobile Link エージェント作成ウィザードが表示されます。

- 3. [ようこそ]ページで、[エージェントを1つ設定する]を選択し、[次へ]をクリックします。
- [エージェント ID] ページで、[エージェント ID] に AID_JOHN と入力します。エージェント ID には任意の値を指定できますが、各エージェントにはユニークな ID が必要です。エージェント ID は、プレフィクス AID_で始まり、その後にエージェントが使用する Mobile Link ユーザ名が続くのが一般的です。必要に応じて、[説明] フィールドにエージェントの説明を入力することもできます。[次へ] をクリックします。
- [リモートデータベース]ページでは、このエージェントが管理するリモートデータベースを 定義できます。この処理ではデータベースは作成されません。実際の作成は後で行います。
 [リモートスキーマ名]で [Tutorial Application v1.0] を選択します。これは、前のレッスンで ドロップダウンリストから定義した名前です。
- 6. [データベース接続文字列] フィールドに次の接続文字列を入力します。

start=dbsrv16;SERVER=tutorial_v1;DBF={db_location}¥tutorial_v1.db;UID=DBA;PWD=sql

この文字列値はマクロ {db_location} を使用しています。このマクロは、アプリケーション データベースが保存される時点で、リモートデバイスでディレクトリに置き換えられます。 [次へ] をクリックします。

- [エージェント設定] ページで、30 と入力し、[同期間隔] に [秒] を選択します。同期間隔は、 エージェントがエージェントデータベースを同期する頻度を制御します。エージェントデー タベースの同期は、エージェントが実行する新しいタスクを受け取り、実行済みのタスクの 結果をアップロードする方法となります。
- [エージェント設定] ページで、10 と入力し、[同期ポーリング間隔] に [秒] を選択します。管理ポーリング間隔によって、サーバからの同期要求や他のアクションの実行要求をエージェントがチェックする頻度が決まります。

注意

同期間隔または管理ポーリング間隔に選択した値が小さいと、応答性に非常に優れたエー ジェントとなり、デモやトラブルシューティングに効果的です。ただし、小さい値を運用環 境でグローバルに使用すると、サーバの負荷が大きくなり、パフォーマンスが低下します。

9. [完了] をクリックします。

結果

エージェントが作成および設定されます。

次の手順

「レッスン6:リモートデバイスでのエージェントの設定」238ページに進みます。

レッスン6:リモートデバイスでのエージェントの設定

このレッスンでは、Mobile Link エージェントを実行します。Mobile Link エージェントは、集中 管理の対象となる、それぞれのリモートデバイス上で実行されている必要があります。この チュートリアルでは、Mobile Link サーバと同じコンピュータでエージェントを実行します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

◆ タスク

1. 通常はリモートデバイス上にあるファイルを含むディレクトリを作成します。

md c:¥cadmin_demo¥remote cd c:¥cadmin_demo¥remote

2. Mobile Link エージェントを設定モードで次のように実行します。

mlagent -cr -db . -x tcpip{host=localhost;port=2439} -a AID_JOHN -u JOHN -p sql

この手順では、エージェントデータベースを作成し、一部の設定情報をそこに保存します。 指定されたオプションがデータベースに保存されると、エージェントが停止します。次に、 使用するオプションの概要を示します。

- -cr エージェントを設定モードで実行し、設定モードの前の実行で保存された設定内容 を破棄することを指定します。
- -db エージェントがアプリケーションデータベースを作成する場所を指定します。この場所は、{db location} マクロの値になります。
- -x エージェントデータベースを同期する(新しいタスクを受け取り、実行済みのタスクの結果をアップロードする)ために、エージェントを Mobile Link サーバに接続する方法を指定します。集中管理を既存の同期システムに追加している場合は、このオプションで指定した値を、Mobile Link サーバへの接続に適した文字列に変更する必要があります。
- -a このエージェントのエージェント ID を指定します。Sybase Central を使用して統合 データベースで前に作成したエージェント ID と同じ値が指定されています。
- -u エージェントデータベースの同期時にエージェントが使用する Mobile Link ユーザ を指定します。この値は、主にエージェントを認証するために Mobile Link サーバで使用 されます。
- -p -u オプションで指定された Mobile Link ユーザのパスワードを指定します。
- 3. リモートデバイスで Mobile Link エージェントを実行します。このチュートリアルでは、エージェントの実行を次のように明示的に開始します。

start mlagent -v9 -ot agent.txt

次に、エージェントを実行するためにこのレッスンで使用するオプションの概要を示します。

- -v9 最大冗長を使用します。このロギングオプションは、開発環境での使用に適しています。パフォーマンス上の理由から、一般的に-v9 は運用環境では使用しません。
- -ot エージェントログの出力ファイルを指定します。
- Chrö、Mobile Link エージェントが稼働し、正常に同期されます。確認するには、Sybase Central に戻ります。[フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] » [統 合データベース] » Tutorial » [エージェント] の順に展開します。AID_JOHN を選択し、右 ウィンドウ枠で [イベント] タブを確認します。エージェントの最初の同期を示すエントリ が表示されています。

注意

エージェント設定に関する運用時の考慮事項 集中管理を運用環境で使用する場合は、次の考 慮事項に注意してください。

- ○-u オプションと -p オプションに指定した値を、同期システムに適した Mobile Link ユーザ とパスワードの組み合わせに変更する必要があります。
- ○-on オプションを使用すると、エージェントが生成するログファイルのサイズを制限できま す。
- ○Mobile Link エージェントがリモートデバイスで稼働している場合、リモートデバイスには リモート管理のみが可能です。エージェントが常に稼働していることを確認する措置を 取ってください。このことを行うには、たとえば、エージェントをサービスとして実行し たり、エージェントをレジストリで Run 起動グループに追加したりします。

結果

Mobile Link エージェントが実行され、同期されます。

次の手順

「レッスン7:同期モデルの作成」239ページに進みます。

レッスン7:同期モデルの作成

このレッスンでは、同期モデルを作成します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

集中管理を既存の同期システムに追加している場合は、「レッスン8:同期モデルの展開」 241ページに進みます。

◆タスク

- 統合データベースでリモートデータベース用にテーブルを定義します。Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial]、[統合データベース] の順 に展開します。Tutorial - DBA を右クリックして、[Interactive SQL を開く] をクリックしま す。
- 2. [SQL 文] ウィンドウ枠で、次のコマンドを入力します。

CREATE TABLE customer(cust_id INTEGER PRIMARY KEY, f_name VARCHAR(100), I_name VARCHAR(100))

- 3. [F5] キーを押して SQL を実行します。Interactive SQL を閉じます。SQL 文を保存する必要 はありません。
- 4. Sybase Central の [フォルダ] ビューで、[Central Admin Tutorial] » [新規] » [同期モデル] を右 クリックします。
- 5. [ようこそ] ページで、新しい同期モデルの名前に tutorial1 と入力します。[次へ] をクリック します。
- [プライマリキー要件]ページで、スキーマが同期要件を満たしていることを確認するために 3つのチェックボックスをすべてオンにします。[次へ]をクリックします。
- 7. [統合データベーススキーマ] ページで Tutorial データベースを選択し、[次へ] をクリックします。
- 8. [リモートデータベーススキーマ] ページで [いいえ、新しいリモートデータベーススキーマを 作成します] を選択し、[次へ] をクリックします。
- 9. [新しいリモートデータベーススキーマ] ページで、customer テーブルが選択されていること を確認し、[次へ] をクリックします。[完了] をクリックします。

結果

これで、customerという単一のテーブルを含む同期モデルが作成されました。このテーブルは、 リモートデータベースと統合データベース間で同期できます。次の手順では、このモデルを展開 して統合データベースに同期オブジェクトを作成し、リモートデータベースを作成する SQL を 生成します。

次の手順

「レッスン8:同期モデルの展開」241ページに進みます。

レッスン8:同期モデルの展開

このレッスンでは、前のレッスンで作成した同期モデルを展開します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

◆タスク

 Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] » [同期モ デル] を展開します。tutorial1 を右クリックして、[展開] をクリックします。

同期モデル展開ウィザードが表示されます。

- 2. [ようこそ] ページで、ウィザードによって生成されたファイルを保管するデフォルトロケー ションをそのまま使用し、[次へ] をクリックします。
- [クライアントネットワークオプション] ページで、以下のオプションを選択し、[次へ] をク リックします。
 - プロトコル TCP/IP
 - ホスト localhost
 - ポート 2439
- [Mobile Link ユーザとパスワード] ページで、[リモートタスクに適したマクロの値の使用] を 選択します。

{ml_username} および {ml_password} のマクロ値は、生成された SQL ファイルで使用され、 リモートデバイスで SQL が実行されるときに Mobile Link エージェントが使用する Mobile Link ユーザとパスワードに置き換えられます。同期プロファイルは tutorial1_{ml_username} という名前で自動的に作成されます。このとき、 {ml_username} マクロは Mobile Link ユーザ 名に置き換えられ、ここでは JOHN に置き換えられます。

- 5. [同期プロファイル] ページで、[同期プロファイル名] フィールドに tutorial1_JOHN と入力します。
- [データベースの同期を準備する方法を選択します]ページが表示されるまで [次へ] をクリックし、以下のタスクを実行します。
 - ●[統合データベースの同期を準備するために作成された SQL スクリプトの処理方法を指定 してください] で [統合データベースに対して実行します。] を選択します。
 - ●[リモートデータベースの同期を準備するために作成された SQL スクリプトの処理方法を 指定してください] で [実行しません] を選択します。

●[次へ]をクリックしてから、[完了]をクリックします。

同期モデルから移動する場合、変更内容を保存するよう求められます。[はい]をクリックします。

結果

これで、同期モデルの作成と展開が完了しました。モデルを展開すると、スクリプトが統合デー タベースに追加され、リモートデータベースを同期できるようになります。また、SQLファイ ルも c:¥cadmin_demo¥Central Admin Tutorial¥tutorial1_deploy¥ディレクトリに生成してあります。 これはリモートデータベースを作成する場合に使用できます。これらのファイルをこの時点で 確認してください。

次の手順

「レッスン9:リモートタスクの作成」242ページに進みます。

レッスン9:リモートタスクの作成

このレッスンでは、リモートタスクを作成して、「Hello World」というメッセージをリモートデ バイスに表示します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

集中管理におけるほとんどのアクションには、リモートタスクが関係しています。リモートタス クは、管理者が作成するコマンドのコレクションです。リモートタスクは、1つまたは複数の エージェントに割り当てることができます。エージェントに割り当てられたリモートタスクは、 エージェントが次にエージェントデータベースを同期するときにエージェントにダウンロード されます。エージェントは、タスクを適切な時間に実行し、実行に関する情報をアップロードし ます。

◆タスク

1. 新しいリモートタスクを作成します。Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] を展開し、[リモートタスク] を右クリックして [新規]»[リモー トタスク] をクリックします。

リモートタスク作成ウィザードが表示されます。

- [ようこそ] ページで、[名前] フィールドに Hello World と入力します。[次へ] をクリックします。
- [トリガのメカニズム] ページで、[エージェントで受信されたとき] をオンにし、[完了] をク リックしてウィザードを完了します。
- 4. [フォルダ] ビューで、新しく作成した [Hello World] タスクをクリックします。右ウィンドウ 枠に、コマンドをタスクに追加できる [コマンド] タブが表示されます。
- 5. [コマンド] タブで、[コマンドタイプ] ドロップダウンリストから [プロンプト] を選択します。 [メッセージ] フィールドに Hello World と入力します。
- 2つ目のコマンドをタスクに追加するには、新しいコマンドが表示されるまで [Tab] キーを押 すか、[コマンドを追加] ボタンをクリックします。2つ目のコマンドのコマンドタイプを [プ ロンプト] に設定し、[メッセージ] フィールドに Hello Again と入力します。

ここで作成した Hello World タスクは、デザイン時のタスクです。このタスクは、ローカルコ ンピュータでプロジェクトに保存されます。タスクをエージェントに割り当てる前に、タス クを展開して統合データベースにコピーする必要があります。

結果

リモートタスクが作成され、展開の準備が完了します。

次の手順

「レッスン10:リモートタスクの展開」243ページに進みます。

レッスン 10: リモートタスクの展開

このレッスンでは、リモートタスクを展開して、エージェントに割り当てできるようにします。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

◆タスク

 Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial]、[リモー トタスク] の順に展開し、[Hello World] タスクを右クリックして [展開] をクリックします。

リモートタスク展開ウィザードが表示されます。

2. [タスクの名前と説明] ページでデフォルトをそのまま使用し、[次へ] をクリックします。

このようにすると、展開済みタスクが、デザイン時のタスクと同じ名前になります。同じデ ザイン時のタスクを2回目に展開しないかぎり、通常はこのようにします。2回目に展開す る場合は、展開済みタスクの名前を変更する必要があります。

- [受信者] ページでは、展開済みタスクを既存のエージェントに割り当てることができます。 この操作を別の手順で行うこともできます。[受信者] ドロップダウンから [特定のエージェント] を選択します。[エージェント] リストで [AID_JOHN] を選択し、[次へ] をクリックします。
- 4. [配信オプション] ページで、[エージェントの次回の同期時] をオンにし、[次へ] をクリック します。
- 5. [結果とステータスのレポート] ページで、どちらの質問に対しても [すぐに結果とステータスを送信] をオンにします。これにより、タスクを実行したときに通知をタイムリーに受け取ることができます。 ルーチンタスクや繰り返しタスクの場合は、フィードバックの受信頻度を抑える (特に成功した場合) ことで、エージェントデータベースの同期回数と Mobile Linkサーバでの負荷が削減されます。
- 6. [完了] をクリックします。

エージェント AID_JOHN が次回にエージェントデータベースを同期すると、新しいタスク を受け取り、それを実行します。Hello World と Hello Again というテキストが表示された メッセージボックスで、[OK] をクリックします。

これで、[フォルダ] ビューのリストに Hello World タスクの2つのコピーが表示されます。 展開済みコピーは、[リモートタスク] » [展開済みタスク] の [フォルダ] ビューに表示されま す。これは、統合データベースにおけるコピーです。展開済みタスクのコピーは変更できま せん。デザイン時のタスクのコピーは、[リモートタスク] に引き続き表示されます。このタ スクは変更可能であり、新しい名前でもう一度展開できます。

展開済みタスクを右クリックして [受信者の追加] を選択すると、展開済みタスクをいつでも 他のエージェントに割り当てることができます。

結果

リモートタスクが実行されます。

次の手順

「レッスン11:リモートタスクのステータスのチェック」244ページに進みます。

レッスン11:リモートタスクのステータスのチェック

このレッスンでは、Sybase Central でリモートタスクのステータスをチェックします。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

◆タスク

- Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] » [リモー トタスク] » [展開済みタスク] の順に展開します。Hello World タスクの展開済みバージョン をクリックし、右ウィンドウ枠で [結果] タブを選択します。タブが自動的に再表示されるま で待つか、[F5] キーを押してすぐに再表示します。[結果] タブには、タスクのコマンドごと に行が表示され、コマンドが成功したか失敗したかを示す [結果コード] が表示されます。[結 果コード] の0は、成功を示します。
- 2. タスクの実行結果を異なる方法で表示するには、展開済みタスクの [受信者] タブを選択する か、タスクを実行したエージェントの [イベント] タブまたは [タスク] タブを見ます。

結果

タスクの実行結果が表示されます。

次の手順

「レッスン12:リモートデバイスでのリモートデータベースの作成」245ページに進みます。

レッスン 12: リモートデバイスでのリモートデータベースの作 成

このレッスンでは、リモートタスクを使用してリモートデバイスで新しいリモートデータベース を作成します。集中管理を既存の同期システムに追加している場合は、「レッスン13:同期のス ケジュール」247ページに進みます。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

◆タスク

1. 新しいリモートタスクを作成します。Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] を展開します。[リモートタスク] を右クリックし、[新規]»[リ モートタスク] をクリックします。

リモートタスク作成ウィザードが表示されます。

- [ようこそ] ページで、[名前] フィールドに Create DB と入力します。前に作成したタスクと 異なり、このタスクはリモートデータベースでの作成または操作になるため、[タスクにリ モートデータベースが必要、またはリモートデータベースを作成] をオンにし、リモートス キーマ名 [Tutorial Application v1.0] を選択します。これにより、このタスクで実行される データベースアクションが識別されます。[次へ] をクリックします。
- 3. [トリガのメカニズム] ページで、[エージェントで受信されたとき] をオンにし、[完了] をク リックしてウィザードを完了します。
- 4. [フォルダ] ビューで、新しく作成した [Create DB] タスクをクリックします。
- 5. 右側の [コマンド] ウィンドウ枠からリモートタスクにコマンドを追加します。
 - a. 最初のコマンドでは、リモートデバイスで新しい空のデータベースを作成します。コマ ンドタイプを **[データベースを作成]** に設定します。
 - b. ファイル名を {db_location}¥tutorial_v1.db に設定します。このファイル名は、エージェントの設定で指定した接続文字列内のファイル名に対応しています。
 - c. 新しいコマンドが表示されるまで [Tab] キーを押します。
 - d. 2 つ目のコマンドでは、新しいデータベースでスキーマを作成します。コマンドタイプ を [SQL を実行] に設定します。[インポート] をクリックします。
 - e. [開く] ウィンドウでファイル c:¥cadmin¥Central Admin Tutorial¥tutorial1_deploy
 ¥remote_setup.sql を選択し、[開く] をクリックします。同期モデルをコマンドに展開した
 ときに生成されたリモートデータベースを初期化する SQL がインポートされます。
- 6. これで、リモートタスクが完了しました。タスクを展開して、次の手順でエージェント AID_JOHN に割り当てます。
 - a. [フォルダ] ビューで Create DB タスクを右クリックし、[展開] をクリックします。 リモートタスク展開ウィザードが表示されます。
 - b. [タスクの名前と説明] ページでデフォルトをそのまま使用し、[次へ] をクリックします。
 - c. [受信者] ドロップダウンから [特定のエージェント] を選択します。[エージェント] リストで [AID_JOHN] を選択し、[次へ] をクリックします。
 - d. [配信オプション] ページで、[エージェントの次回の同期時] をオンにし、[次へ] をクリッ クします。
 - e. [結果とステータスのレポート] ページで、どちらの質問に対しても [すぐに結果とステー タスを送信] を選択します。
 - f. [完了] をクリックします。
- 7. タスクが成功したかどうかを確認します。
 - a. Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] » [統 合データベース] » [チュートリアル] » [エージェント] の順に展開し、[AID_JOHN] をク リックします。
 - b. [イベント] タブを選択し、[Create DB] タスクを検索します。タブが自動的に再表示され るまで待つか、[F5] キーを押してすぐに再表示します。
結果

リモートデバイスに新しいリモートデータベースが作成されます。

次の手順

「レッスン13:同期のスケジュール」247ページに進みます。

レッスン 13:同期のスケジュール

次の手順では、リモートデータベースを定期的に同期するように Mobile Link エージェントを設 定します。このことを行うには、スケジュールに基づいて実行されるリモートタスクを作成し、 リモートタスクが実行されるたびにデータベースを同期します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

これまでに作成した他のタスクは、1回のみ実行されることが、このタスクと異なります。この タスクは、リモートデバイスに残り、停止するまで定期的に実行されます。

◆ タスク

 新しいリモートタスクを作成します。Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] を展開します。[リモートタスク] を右クリックし、[新規]»[リ モートタスク] をクリックします。

リモートタスク作成ウィザードが表示されます。

- [ようこそ] ページで、[名前] フィールドに Sync と入力します。[タスクにリモートデータ ベースが必要、またはリモートデータベースを作成] をオンにし、リモートスキーマ名 [Tutorial Application v1.0] を選択します。これにより、このタスクで実行されるデータベー スアクションが識別されます。[次へ] をクリックします。
- 3. [トリガのメカニズム] ページで、[スケジュールに従う] を選択し、[次へ] をクリックします。
- 4. [開始日時] ページで、デフォルト値をそのまま使用します。これで、タスクの実行をすぐに 開始できます。[次へ] をクリックします。
- 5. [繰り返し] ページで、[次の間隔で繰り返し] をオンにし、間隔を1分に設定します。[完了] をクリックしてウィザードを完了します。

- 6. [フォルダ] ビューで、新しく作成した Sync タスクをクリックします。
- 7. 同期を開始する1つのコマンドをタスクに追加します。
 - a. [コマンド] タブで、最初のコマンドの [コマンドタイプ] を [同期] に設定します。
 - b. [同期プロファイル] に tutorial1_JOHN と入力します。これは、同期モデルを展開したときに作成した同期プロファイルです。
- 8. これで、同期タスクが完了しました。Sync を右クリックして、[展開] をクリックします。[次 へ] をクリックします。
- 9. [受信者] ドロップダウンで [特定のエージェント] をクリックし、タスクをエージェント AID_JOHN に割り当てます。[次へ] をクリックし、もう一度 [次へ] をクリックします。
- 10. [結果とステータスのレポート] ページで、[タスクが成功した場合] を [後でステータスのみ送 信] に設定し、[タスクが失敗した場合] を [すぐに結果とステータスを送信] に設定します。

このタスクは頻繁に繰り返されるため、要求するフィードバックを制限してパフォーマンス を向上させてください。

11. [完了] をクリックします。

結果

この新しいタスクをエージェントが受け取ると、リモートデータベースが1分ごとに同期されま す。

次の手順

「レッスン14:スケジュールされた同期の変更」248ページに進みます。

レッスン14:スケジュールされた同期の変更

前のレッスンでは、リモートデータベースを1分ごとに同期するリモートタスクを作成しました。このレッスンでは、同期間隔を1時間に1回に変更します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

リモートタスクを展開すると、展開済みバージョンは変更できなくなります。代わりに、必要な 変更を加えた新しいリモートタスクを作成してから既存のタスクをキャンセルし、新しいタスク を展開して置き換えます。

◆タスク

- 1. まず、既存の展開済みタスクをテンプレートとして使用して、希望の繰り返し間隔で新しい リモートタスクを作成します。
 - a. Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] » [リ モートタスク] » [展開済みタスク] の順に展開します。Sync を右クリックし、[コピー] を 選択してタスクをクリップボードにコピーします。
 - b. [リモートタスク] を右クリックし、[貼り付け] を選択します。リモートタスク名を変更 するかどうかを尋ねるウィンドウが表示されます。Sync every hour と入力し、[OK] をク リックします。
 - c. 新しい Sync every hour タスクを右クリックし、[プロパティ] を選択します。プロパティ ウィンドウの [繰り返し] ページで、[次の間隔で繰り返し] の値を [1 分] から [1 時間] に 変更し、[OK] をクリックします。
- 2. 次に、1分ごとに同期を行う既存のリモートタスクをキャンセルします。
 - a. [フォルダ] ビューで、展開済みバージョンの Sync タスクをクリックし、右ウィンドウ枠 で [受信者] タブをクリックします。
 - b. エージェント AID_JOHN のテーブル内のエントリを右クリックし、[キャンセル] をク リックします。
- 3. 最後に、新しい [Sync every hour] タスクを展開してエージェント AID_JOHN に割り当てま す。
 - a. Sync every hour を右クリックして、[展開] をクリックします。[次へ] をクリックします。
 - b. [受信者] ドロップダウンリストから [特定のエージェント] をクリックし、タスクをエージェント AID_JOHN に割り当てます。[次へ] をクリックし、もう一度 [次へ] をクリックします。
 - c. [結果とステータスのレポート] ページで、[タスクが成功した場合] を [後でステータスの み送信] に設定し、[タスクが失敗した場合] を [すぐに結果とステータスを送信] に設定し ます。
 - d. [完了] をクリックします。

結果

この新しいタスクをエージェントが受け取ると、リモートデータベースが1分ごとではなく1時 間ごとに同期されます。

次の手順

「レッスン15:即時同期の強制」249ページに進みます。

レッスン 15:即時同期の強制

前のレッスンでは、1時間ごとに同期するリモートデータベースを設定しました。このレッスンでは、サーバ起動のリモートタスク (SIRT)を使用して、1時間が経過する前に強制的に同期する

方法について説明します。この方法は、特定のタスクの実行時期を集中管理する場合に便利で す。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

◆タスク

● Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] » [リモー トタスク] » [展開済みタスク] の順に展開します。Sync every hour を右クリックし、[すべて の受信者について開始] をクリックします。

結果

タスクのすべての受信者は、次にサーバをポーリングするときに、タスクをすぐに実行するよう 指示されます。エージェントがサーバをポーリングする頻度は、エージェントの [管理ポーリン グ間隔] プロパティで制御されます。

次の手順

「レッスン16:リモートスキーマの変更」250ページに進みます。

レッスン16:リモートスキーマの変更

このチュートリアルでは、リモートデータベースに新しいテーブルを追加することによって、リモートスキーマを変更します。

このレッスンでは、リモートデータベースのスキーマを変更します。このチュートリアルでは、 データベースのリモートスキーマ名を変更するたびに、スキーマの変更が発生します。リモート スキーマ名の変更は、強制的に実行されるものではなく、常にユーザが任意で行います。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

1つのリモートデータベースに対して実行できるすべてのリモートタスクは、同じリモートス キーマ名を持つ他のリモートデータベースに対しても実行できるようにしてください。タスク が失敗するか成功するかに影響を及ぼす変更をデータベースに加えた場合は、必ずデータベース のリモートスキーマ名を変更してください。リモートデータベースの状態が影響するタスク内 のコマンドは、「[同期]」 コマンドと「[SQL を実行]」 コマンドのみです。

「[同期]」 コマンドは、リモートデータベースに同期プロファイルが存在するかどうかが影響す るため、同期プロファイルを追加または削除するたびにリモートスキーマ名を変更する必要があ ります。

「[SQL を実行]」 コマンドは、スキーマの一部として通常処理する多くのデータベースオブジェ クトの状態が影響します。たとえば、データベースへのテーブルの追加やデータベースからの テーブルの削除、データベース内のテーブルの定義の変更、ストアドプロシージャの追加や削除 を実行すると、「[SQL を実行]」 コマンドへの影響が発生し、これにより、リモートスキーマ名 を変更することが必要になります。

◆ タスク

 Sybase Central の [フォルダ] ビューに戻ります。[Mobile Link 16] で、Central Admin Tutorial を展開し、[リモートスキーマ名] を右クリックし、[新規]»[リモートスキーマ名] を クリックします。

リモートスキーマ名作成ウィザードが表示されます。

- 2. スキーマ名に Tutorial Application v2.0 と入力し、[完了] をクリックします。
- 3. 新しいリモートタスクを作成します。Sybase Central の [フォルダ] ビューの Central Admin Tutorial で、[リモートタスク] を右クリックし、[新規] »[リモートタスク] をクリックしま す。リモートタスク作成ウィザードが表示されます。
- 4. [ようこそ] ページで、[名前] フィールドに Schema Upgrade と入力します。
- 5. [タスクにリモートデータベースが必要、またはリモートデータベースを作成] をオンにし、 [リモートスキーマ名] を [Tutorial Application v1.0] に設定します。
- 6. [このタスクは、管理対象リモートデータベースのスキーマを更新] をオンにし、[新しいリ モートスキーマ名] を [Tutorial Application v2.0] に設定します。[完了] をクリックします。
- [コマンド] タブで、[コマンドタイプ] ドロップダウンリストから [SQL を実行] を選択します。[SQL] フィールドに次のように入力します。

```
CREATE TABLE product (
prod_id integer primary key,
name varchar( 100 )
);
```

これで、スキーマの変更タスクが完了しました。

新しいスキーマ変更タスクを展開する前に、リモートデバイスにすでに割り当てられている すべてのタスクを考慮する必要があります。Schema Upgrade タスクが完了すると、データ ベースのリモートタスク名は Tutorial Application v2.0 になります。古いリモートスキーマ 名 Tutorial Application v1.0 に関連付けられているリモートデバイス上のタスクは、実行しな くなり、エージェントによって破棄されます。これらのタスクに備えられている機能を保持 するため、新しいバージョンのタスクを作成し、それを新しいリモートスキーマ名に関連付 けてください。

8. [フォルダ] ビューの Central Admin Tutorial » [統合データベース] » [Tutorial] » [エージェ ント] で、AID_JOHN をクリックします。右ウィンドウ枠で、[タスク] タブを選択します。 アクティブタスクのみがエージェントで実行されています。これらのタスクに対してのみ、 新しいバージョンを作成する必要があります。この場合、アクティブタスクは Sync every hour タスクのみです。

[タスク] タブの [リモートスキーマ名] カラムをチェックすることで、このタスクが古いリ モートスキーマ名に関連付けられているかどうかを判断できます。このタスクは、Sync every hour タスクの [リモートスキーマ名] が Tutorial Application v1.0 であるため、古いリ モートスキーマ名に関連付けられていることを示しています。スキーマの変更後に引き続き 同期を行うには、このタスクの新しいバージョンを作成してエージェントに割り当てる必要 があります。

- 9. Sync every hour タスクを右クリックし、[タスクに移動] を選択します。
- 10. 展開済みタスク Sync every hour を右クリックし、[コピー] を選択します。
- 11. [リモートタスク] を右クリックし、[貼り付け] をクリックします。コピーしたタスクの名前 を尋ねられたら Sync every hour v2 と入力し、[OK] をクリックします。
- 12. 新しいスキーマに対する作業を続行するために、タスク内のコマンドを変更することが必要 かどうかを検討します。この場合、答えはいいえです。これは、コマンドは1つのみであり、 このコマンドはこのスキーマ変更で修正を加えていない tutorial1_JOHN 同期プロファイル のみに依存するためです。
- 新しいリモートスキーマ名に関連付けられたタスクであるとマーク付けします。Sync every hour v2 タスクを右クリックし、[プロパティ]を選択します。プロパティウィンドウの[一 般] ページで、[リモートスキーマ名] に Tutorial Application v2.0 を選択し、[OK] をクリック します。
- 14. 新しいタスクを展開するには、Sync every hour v2 タスクを右クリックし、[展開] をクリック します。[次へ] をクリックします。
- 15. [受信者] で [特定のエージェント] をクリックしてから、エージェント AID_JOHN を選択し ます。[次へ] をクリックしてから、[完了] をクリックします。
- 16. Schema Upgrade タスクを右クリックし、[展開] をクリックします。[次へ] をクリックしま す。
- 17. [受信者] ドロップダウンリストから [特定のエージェント] をクリックし、タスクをエージェ ント AID JOHN に割り当てます。[次へ] をクリックしてから、[完了] をクリックします。

結果

Schema Upgrade タスクが正常に実行されたことを確認してください。その後は、Sync every hour v2 タスクが1時間ごとに実行し、Sync every hour タスクは実行を停止します。

次の手順

「レッスン17:リモートデータベースの問い合わせ」253ページに進みます。

レッスン17:リモートデータベースの問い合わせ

このレッスンでは、リモートデータベースに対してクエリを行い、結果をサーバに返します。リ モートデータベースの状態を正確に把握できるため、トラブルシューティングに非常に便利で す。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

このチュートリアルでデータベースに追加したテーブルにはデータが含まれていないため、代わりにデータベースシステムテーブルに対して問い合わせます。この例ではシステムテーブルに対して問い合わせますが、ユーザテーブルに対する動作とまったく同じです。

前のレッスンで実行したスキーマの変更により、期待する操作(正しいカラムを持つ product テーブルを作成すること)が行われたかどうかを確認するとします。このことは、systable と systabcol の各システムテーブルに対してクエリを実行することによって確認できます。

◆ タスク

1. Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] を展開 し、[リモートタスク] を右クリックして [新規] » [リモートタスク] をクリックします。

リモートタスク作成ウィザードが表示されます。

- 2. [ようこそ] ページで、[名前] フィールドに Table Query と入力します。
- 3. [タスクにリモートデータベースが必要、またはリモートデータベースを作成] をオンにし、 [リモートスキーマ名] を [Tutorial Application v2.0] に設定して [次へ] をクリックします。
- [トリガのメカニズム] ページで、[エージェントで受信されたとき] をオンにし、[完了] をク リックします。
- 5. 次の SQL を使用して、[SQL を実行] コマンドをタスクに追加します。

SELECT * FROM systable WHERE table_name = 'product' go SELECT * FROM systabcol ORDER BY table_id

- 6. 新しい Table Query タスクを右クリックし、[展開] をクリックします。[次へ] をクリックします。
- 7. [受信者] に [特定のエージェント] を選択し、エージェント AID_JOHN を選択して [次へ] を クリックしてから、もう一度 [次へ] をクリックします。
- 8. [結果とステータスのレポート] ページで、[タスクが成功した場合] と [タスクが失敗した場合] をどちらも [すぐに結果とステータスを送信] に設定します。[完了] をクリックし、タスクが実行するまで待ちます。
- [フォルダ] ビューで Table Query タスクの展開済みコピーをクリックし、[結果] タブをク リックします。タブに結果が何も表示されない場合は、タブが自動的に再表示されるまで待 つか、[F5] キーを押してすぐに再表示します。
- 10. テーブルで [SQL を実行] 文の行を右クリックし、[詳細] を選択します。

[コマンドの結果] ウィンドウが表示されます。

11. ウィンドウで [結果] タブをクリックします。このタブには、実行されたクエリの結果が表示 されます。ウィンドウ枠上部の [結果] ドロップダウンを使用すると、2 つのクエリの結果を 切り替えることができます。[閉じる] をクリックします。

結果

リモートデータベースからの結果が表示されます。

次の手順

「レッスン 18 : SIRT を使用したファイルのアップロード」254 ページに進みます。

レッスン 18: SIRT を使用したファイルのアップロード

このレッスンでは、サーバ起動のリモートタスク (SIRT) を使用して、リモートデバイスからファ イルをアップロードします。ファイルに問題があるかどうかを管理者が検証できるため、リモー トデバイスからのファイルのアップロードはトラブルシューティングに便利です。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

内容と備考

Mobile Link エージェントをリモートデバイスで起動したとき、ファイル agent.txt にメッセージ を記録するよう指示しました。ここでは、そのファイルをリモートデバイスから取得して、検証 します。

◆タスク

1. Sybase Central の [フォルダ] ビューの [Mobile Link 16] で、[Central Admin Tutorial] を展開 し、[リモートタスク] を右クリックして [新規] » [リモートタスク] をクリックします。

リモートタスク作成ウィザードが表示されます。

- 2. [ようこそ] ページで、[名前] フィールドに Upload Agent Log と入力します。
- 3. [タスクにリモートデータベースが必要、またはリモートデータベースを作成] がオンになっている場合はクリアし、[完了] をクリックします。
- [フォルダ] ビューで新しいタスクをクリックし、タスクにコマンドを追加します。[コマンド タイプ] を [ファイルをアップロード] に設定します。
- [サーバファイル名] を {agent_id}¥agent.txt に設定し、[リモートファイル名] を {agent_log} に 設定します。コマンドエディタで省略記号 (ピリオド3つ) を使用すると、マクロの値を簡単 に入力できます。

*{agent_log}*マクロは、Mobile Link エージェントがリモートデバイスで保持しているログファイルの名前に置き換えられます。

[サーバファイル名] フィールドには、*{agent_id}* マクロを使用してファイルが置かれるディレクトリを指定しました。このことは非常に重要です。マクロを使用しないでサーバファイル名を指定すると、タスクを実行するすべてのエージェントによって、アップロードファイルが同じ場所に配置されます。このとき、新しいエージェントは、前のエージェントによって書き込まれたファイルを毎回上書きします。マクロを使用すると、各エージェントはログファイルをサーバ上の異なる場所にアップロードするため、すべてのログファイルを表示できるようになります。

- 6. 新しい Upload Agent Log タスクを右クリックし、[展開] をクリックします。[次へ] をクリッ クします。
- 7. [受信者] で [特定のエージェント] をクリックしてから、エージェント AID_JOHN を選択し ます。[次へ] をクリックします。
- 8. [配信オプション] ページで、[エージェントの次回の同期時] をクリックし、[次へ] をクリックします。
- 9. [結果とステータスのレポート] ページで、[タスクが成功した場合] と [タスクが失敗した場合] をどちらも [すぐに結果とステータスを送信] に設定します。[完了] をクリックします。
- 10. 管理者は、Sybase Central でタスクを開始する必要があります。タスクを開始するには、[エー ジェント] の AID_JOHN に移動します。ウィンドウ枠で、[タスク] タブを選択し、Upload Agent Log タスクを右クリックし、[開始] をクリックします。タスクが実行するのを待ちま す。

アップロードしたファイルは、Mobile Link コマンドラインで-ftru オプションを使用して指定 された Mobile Link アップロードディレクトリに配置されます。アップロードディレクトリ には *c:¥cadmin demo¥consolidated¥upload* が指定されています。コマンドプロンプトまたは Windows エクスプローラーを使用して、このディレクトリを見てみます。*AID_JOHN* サブ フォルダがあります。このサブフォルダに、アップロードした *agent.txt* ファイルがあります。

結果

結果

次の手順

「クリーンアップ」256ページに進みます。

クリーンアップ

すべてのチュートリアルをコンピュータから削除します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成」232ページを参照してください。

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:リモートデータベース の集中管理の使用」231ページ

◆コンピュータからのチュートリアルの削除

- 1. タスクバー上で、Interactive SQL、Sybase Central、Mobile Link、同期クライアントの各ウィン ドウを右クリックし、[閉じる] を選択して閉じます。
- 2. 次の手順で、チュートリアルに関連するすべてのデータソースを削除します。
 - a. ODBC データソースアドミニストレータを起動します。
 - b. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [ODBC データソースア ドミニストレータ] をクリックします。
 - c. [ユーザデータソース] リストから cadmin_tutorial_consol を選択し、[削除] をクリックします。

結果

チュートリアルリソースが削除されます。

次の手順

なし。

チュートリアル:スクリプトバージョン句を使用したス キーマの変更

このチュートリアルでは、dbmlsync ScriptVersion 拡張オプションが使用されていない同期に関係 するリモートデータベースでスキーマの変更を実行する方法について説明します。このチュー トリアルでは、単一テーブルを同期する同期システムを設定し、スキーマを変更して同期対象 テーブルにカラムを追加して、同期を続行します。

必要なソフトウェア

このチュートリアルでは、チュートリアルを実行するローカルコンピュータに SQL Anywhere (Mobile Link を含む) が完全にインストールされていることを前提としています。

権限

統合データベースで次のロールおよび権限を持つ必要があります。

●SYS AUTH RESOURCE ROLE 互換ロール

●MONITOR システム権限

リモートデータベースで次のロールおよび権限を持つ必要があります。

●SYS REPLICATION ADMIN ROLE システムロール

●SYS RUN REPLICATION ROLE システムロール

概要

このチュートリアルでは、次の作業の方法について説明します。

- ●「レッスン1:統合データベースの作成と設定」
- ●「レッスン2:リモートデータベースの作成と設定」
- ●「レッスン3:リモートデータベースの同期」
- ●「レッスン4:リモートデータベースへのデータの挿入」
- ●「レッスン5:統合データベースでのスキーマ変更の実行」
- ●「レッスン6: リモートデータベースでのスキーマ変更の実行」
- ●「レッスン7:データの挿入」
- ●「レッスン8:同期」

参照

●「リモート Mobile Link クライアントでのスキーマの変更」『Mobile Link クライアント管理』

レッスン1:統合データベースの作成と設定

このレッスンでは、同期用に統合データベースを設定します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:スクリプトバージョン 句を使用したスキーマの変更」257ページ

◆ タスク

1. 次のコマンドを実行して、統合データベースを作成し、実行を開始します。

md c:¥cons cd c:¥cons dbinit -dba DBA,sql consol.db dbsrv16 consol.db

2. 次のコマンドを実行して、統合データベースの ODBC データソースを定義します。

dbdsn -w dsn_consol -y -c "UID=DBA;PWD=sql;DBF=consol.db;SERVER=consol"

 データベースを Mobile Link 統合データベースとして使用するには、Mobile Link で使用する システムテーブル、ビュー、ストアドプロシージャを追加する設定スクリプトを実行する必 要があります。次のコマンドを実行して、統合データベースとして consol.db を設定します。

dbisql -c "DSN=dsn_consol" %SQLANY16%¥MobiLink¥setup¥syncsa.sql

4. Interactive SQL を開き、dsn consol ODBC データソースを使用して consol.db に接続します。

dbisql -c "DSN=dsn_consol"

5. 次の SQL 文を実行します。統合データベースで customer テーブルが作成され、必要な同期ス クリプトが作成されます。

```
CREATE TABLE customer (

id unsigned integer primary key,

name varchar(256),

phone varchar(12)

);

CALL ml_add_column('my_ver1', 'customer', 'id', null );

CALL ml_add_column('my_ver1', 'customer', 'name', null );

CALL ml_add_table_script( 'my_ver1', 'customer', 'upload_insert',

'INSERT INTO customer ( id, name, phone ) '

|| 'VALUES ({ml r.id}, {ml r.name}, {ml r.phone} )' );

CALL ml_add_table_script( 'my_ver1', 'customer', 'download_cursor',

'SELECT id, name, phone from customer' );

CALL ml_add_table_script( 'my_ver1', 'customer', 'download_delete_cursor', '--{ml_ignore}' );

CMMIT;
```

チュートリアルに従って作業する間に、データベースに対してさらに SQL を実行するため、 この SQL の実行完了後も、引き続き Interactive SQL を実行し、データベースに接続した状態 にします。

6. 次のコマンドを実行して、Mobile Link サーバを起動します。

start mlsrv16 -c "DSN=dsn_consol" -v+ -ot mlsrv.txt -zu+

結果

統合データベースが作成され、Mobile Link で使用できるように設定されます。

次の手順

「レッスン2:リモートデータベースの作成と設定」259ページに進みます。

レッスン2:リモートデータベースの作成と設定

このレッスンでは、同期用にリモートデータベースを設定します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル:スクリプトバージョン 句を使用したスキーマの変更」257 ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成と設定」258ページを参照してください。

◆ タスク

1. 次のコマンドを実行して、リモートデータベースを作成し、実行を開始します。

cd.. md c:¥remote cd c:¥remote dbinit -dba DBA,sql remote.db dbsrv16 remote.db

2. Interactive SQL の別のインスタンスを開き、remote.db に接続します。

dbisql -c "SERVER=remote;DBF=remote.db;UID=DBA;PWD=sql"

3. Interactive SQL で次の SQL 文を実行し、同期させるテーブルを作成します。

```
CREATE TABLE customer (
id UNSIGNED INTEGER PRIMARY KEY,
name VARCHAR( 256),
phone VARCHAR( 12 )
);
```

4. リモートデータベースに接続された Interactive SQL インスタンスを引き続き使用して、パブ リケーション、Mobile Link ユーザ、サブスクリプションを作成します。スクリプトバージョ ンは、SCRIPT VERSION 句を使用したサブスクリプションに関連付けられています。この チュートリアルで示すスキーマのアップグレード手順は、SCRIPT VERSION 句を使用して設 定されたスクリプトバージョンがあるサブスクリプションでのみ動作するため、このことは 非常に重要です。

```
CREATE PUBLICATION p1 (
TABLE customer
);
```

CREATE SYNCHRONIZATION USER u1;

CREATE SYNCHRONIZATION SUBSCRIPTION my_sub TO p1 FOR u1 SCRIPT VERSION 'my ver1';

チュートリアルに従って作業する間に、データベースに対してさらに SQL を実行するため、 この SQL の実行完了後も、引き続き Interactive SQL を実行し、データベースに接続した状態 にします。

結果

リモートデータベースが作成され、設定されます。

次の手順

「レッスン3:リモートデータベースの同期」260ページに進みます。

レッスン3:リモートデータベースの同期

この時点では、同期システムの設定が動作しているはずです。このレッスンでは、リモートデー タベースにデータを挿入して同期をテストします。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:スクリプトバージョン 句を使用したスキーマの変更」257ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成と設定」258ページを参照してください。

◆ タスク

1. 統合データベースに接続された Interactive SQL のインスタンスを使用し、次の SQL 文を実行 して customer テーブルにローを挿入します。

INSERT INTO customer VALUES(100, 'John Jones', '519-555-1234'); COMMIT;

2. リモートデータベースに接続された Interactive SQL のインスタンスを使用し、次の SQL 文を 実行して customer テーブルにローを挿入します。 INSERT INTO customer VALUES(1, 'Willie Lowman', '705-411-6372'); COMMIT;

3. 次のコマンドを実行して同期します。

dbmlsync -v+ -ot sync1.txt -c UID=DBA;PWD=sql;SERVER=remote -s my_sub -k

結果

リモートデータベースが同期されます。

customer テーブルの内容と統合データベースを比較することによって、同期が成功したことを確認できます。dbmlsync ログの syncl.txt を調べて、エラーがないかを確認することもできます。

次の手順

「レッスン4:リモートデータベースへのデータの挿入」261ページに進みます。

レッスン4:リモートデータベースへのデータの挿入

このレッスンでは、リモートデータベースにデータを挿入して、アップロードする必要があるリ モートデータベースで操作が実行されている場合でも、スキーマの変更を処理できることを示し ます。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:スクリプトバージョン 句を使用したスキーマの変更」257ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成と設定」258ページを参照してください。

◆ タスク

● リモートデータベースに接続された Interactive SQL のインスタンスを使用し、次の SQL 文を 実行して customer テーブルにローを挿入します。

INSERT INTO customer VALUES(2, 'Sue Slow', '602-411-5467'); COMMIT;

結果

データがリモートデータベースに挿入されます。

次の手順

「レッスン5:統合データベースでのスキーマ変更の実行」262ページに進みます。

レッスン5:統合データベースでのスキーマ変更の実行

このレッスンでは、統合データベースでスキーマを変更します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル:スクリプトバージョン 句を使用したスキーマの変更」257ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成と設定」258ページを参照してください。

◆ タスク

 新しいカラムを customer テーブルに追加して、顧客の携帯電話番号を保存します。まず、統 合データベースに接続されている Interactive SQL のインスタンスで次の SQL 文を実行して、 統合データベースに新しいカラムを追加します。

ALTER TABLE customer ADD cell_phone VARCHAR(12) DEFAULT NULL;

 my_ver2 という新しいスクリプトバージョンを作成して、リモートデータベースから新しい スキーマで同期を処理します。古いスキーマを使用するリモートデータベースでは、古いス クリプトバージョン my_ver1 が引き続き使用されます。統合データベースで次の SQL 文を 実行します。

CALL ml_add_column('my_ver2', 'customer', 'id', null); CALL ml_add_column('my_ver2', 'customer', 'name', null); CALL ml_add_column('my_ver2', 'customer', 'phone', null); CALL ml_add_column('my_ver2', 'customer', 'cell_phone', null);

CALL ml_add_table_script('my_ver2', 'customer', 'upload_insert', 'INSERT INTO customer (id, name, phone, cell_phone) ' || 'VALUES ({ml r.id}, {ml r.name}, {ml r.phone}, {ml r.cell_phone})');

CALL ml_add_table_script('my_ver2', 'customer', 'download_cursor', 'SELECT id, name, phone, cell_phone from customer');

CALL ml_add_table_script('my_ver2', 'customer', 'download_delete_cursor', '--{ml_ignore}'); COMMIT;

結果

統合データベースのスキーマが更新されます。

次の手順

「レッスン6:リモートデータベースでのスキーマ変更の実行」262ページに進みます。

レッスン6:リモートデータベースでのスキーマ変更の実行

このレッスンでは。リモートデータベースを変更して、新しいカラムを customer テーブルに追加し、同期に使用されるスクリプトバージョンを変更します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:スクリプトバージョン 句を使用したスキーマの変更」257ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成と設定」258ページを参照してください。

◆ タスク

 同期スキーマ変更を開始します。これは、同期対象テーブルに影響を及ぼすほとんどのス キーマ変更で必要です。この文によって、サブスクリプションの同期に使用されるスクリプ トバージョンが変更され、スキーマの変更を安全に実行できるように、影響を受けるテーブ ルがロックされます。

リモートデータベースに接続されている Interactive SQL のインスタンスを使用して、リモー トデータベースで次の SQL 文を実行します。

START SYNCHRONIZATION SCHEMA CHANGE FOR TABLES customer SET SCRIPT VERSION = 'my ver2';

2. 次の SQL 文を実行して、新しいカラムを customer テーブルに追加します。

ALTER TABLE customer ADD cell_phone VARCHAR(12) DEFAULT NULL;

スキーマの変更を閉じます。これにより、テーブルのロックが解除されます。

STOP SYNCHRONIZATION SCHEMA CHANGE;

結果

リモートデータベースで同期スキーマの変更が実行されます。

次の手順

「レッスン7:データの挿入」263ページに進みます。

レッスン7:データの挿入

このレッスンでは、新しいスキーマを使用して、リモートデータベースと統合データベースにさらにデータを挿入します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:スクリプトバージョン 句を使用したスキーマの変更」257ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成と設定」258ページを参照してください。

◆リモートデータベースへのデータの挿入

- 1. Interactive SQL を使用して、リモートデータベースで次の SQL 文を実行します。 INSERT INTO customer VALUES(3, 'Mo Hamid', '613-411-9999', '613-502-1212'); COMMIT:
- 2. Interactive SQL を使用して、統合データベースで次の SQL 文を実行します。

INSERT INTO customer VALUES(101, 'Theo Tug', '212-911-7677', '212-311-3900'); COMMIT;

結果

新しいスキーマを使用してリモートデータベースと統合データベースにデータが挿入されます。

次の手順

「レッスン8:同期」264ページに進みます。

レッスン8:同期

このレッスンでは、スキーマの変更ともう一度同期します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。「チュートリアル:スクリプトバージョン 句を使用したスキーマの変更」257ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:統合データベースの作成と設定」258ページを参照してください。

◆タスク

● 次のコマンドを実行して、もう一度同期します。

dbmlsync -v+ -ot sync2.txt -c UID=DBA;PWD=sql;SERVER=remote -s my_sub -k

スキーマの変更前に挿入されたロー Sue Slow が、スクリプトバージョン my_ver1 を使用して アップロードされます。スキーマの変更後に挿入されたロー Mo Hamid が、スクリプトバー ジョン my_ver2 を使用してアップロードされます。my_ver2 のダウンロードカーソルを使 用して、ローがダウンロードされます。

結果

これで、スキーマの変更が完了し、正常に同期を続行できます。

次の手順

なし。

チュートリアル:ScriptVersion 拡張オプションを使用 したスキーマの変更

このチュートリアルでは、ScriptVersion 拡張オプションを使用している場合にスキーマを変更する方法について説明します。

注意

可能な場合は、ScriptVersion 拡張オプションを使用しないことをおすすめします。代わりに、 CREATE SYNCHRONIZATION SUBSCRIPTION 文の SCRIPT VERSION 句または ALTER SYNCHRONIZATION SUBSCRIPTION 文の SET SCRIPT VERSION 句を使用して、スクリプト バージョンをサブスクリプションに関連付けます。これらを実装すると、スキーマを柔軟にアッ プグレードできるようになります。

必要なソフトウェア

このチュートリアルでは、チュートリアルを実行するローカルコンピュータに SQL Anywhere (Mobile Link を含む) が完全にインストールされていることを前提としています。

権限

統合データベースで次のロールおよび権限を持つ必要があります。

●SYS AUTH RESOURCE ROLE 互換ロール

●MONITOR システム権限

リモートデータベースで次のロールおよび権限を持つ必要があります。

●SYS REPLICATION ADMIN ROLE システムロール

●SYS RUN REPLICATION ROLE システムロール

概要

このチュートリアルでは、次の作業の方法について説明します。

- ●「レッスン1:統合データベースの作成と設定」
- ●「レッスン2:リモートデータベースの作成と設定」
- ●「レッスン3:リモートデータベースの同期」
- ●「レッスン4:統合データベースでのスキーマ変更の実行」
- ●「レッスン5:リモートデータベースでのスキーマ変更の実行」

レッスン1:統合データベースの作成と設定

このレッスンでは、同期用に統合データベースを設定します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル: ScriptVersion 拡張オプ ションを使用したスキーマの変更」265 ページ

◆ 統合データベースの作成と設定

1. 次のコマンドを実行して、統合データベースを作成して起動します。

md c:¥cons cd c:¥cons dbinit -dba DBA,sql consol.db dbsrv16 consol.db

2. 次のコマンドを実行して、統合データベースの ODBC データソースを定義します。

dbdsn -w dsn_consol -y -c "UID=DBA;PWD=sql;DBF=consol.db;SERVER=consol"

 データベースを Mobile Link 統合データベースとして使用するには、Mobile Link で使用する システムテーブル、ビュー、ストアドプロシージャを追加する設定スクリプトを実行する必 要があります。次のコマンドを実行して、統合データベースとして consol.db を設定します。

dbisql -c "DSN=dsn_consol" %SQLANY16%¥MobiLink¥setup¥syncsa.sql

4. Interactive SQL を開き、dsn consol DSN を使用して consol.db に接続します。

dbisql -c "DSN=dsn_consol"

5. Interactive SQL で次の SQL 文を実行します。統合データベースで customer テーブルが作成 され、必要な同期スクリプトが作成されます。

CREATE TABLE customer (id unsigned integer primary key, name varchar(256), phone varchar(12)); CALL ml_add_column('my_ver1', 'customer', 'id', null); CALL ml_add_column('my_ver1', 'customer', 'name', null); CALL ml_add_column('my_ver1', 'customer', 'phone', null); CALL ml_add_table_script('my_ver1', 'customer', 'upload_insert', 'INSERT INTO customer (id, name, phone) ' || 'VALUES ({ml r.id}, {ml r.name}, {ml r.phone})');

CALL ml_add_table_script('my_ver1', 'customer', 'download_cursor', 'SELECT id, name, phone from customer');

CALL ml_add_table_script('my_ver1', 'customer', 'download_delete_cursor', '--{ml_ignore}'); COMMIT;

チュートリアルに従って作業する間に、データベースに対してさらに SQL を実行するため、 この SQL 文の実行完了後も、引き続き Interactive SQL を実行し、データベースに接続した状 態にします。

6. 次のコマンドを実行して、Mobile Link サーバを起動します。

start mlsrv16 -c "DSN=dsn_consol" -v+ -ot mlsrv.txt -zu+

結果

統合データベースが作成され、同期用に設定されます。

次の手順

「レッスン2:リモートデータベースの作成と設定」267ページに進みます。

レッスン2:リモートデータベースの作成と設定

このレッスンでは、同期用にリモートデータベースを設定します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル: ScriptVersion 拡張オプ ションを使用したスキーマの変更」265ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1:統合データベースの作成と設定」265ページを参照してください。

◆タスク

1. 次のコマンドを実行して、リモートデータベースを作成して起動します。

cd.. md c:¥remote cd c:¥remote dbinit -dba DBA,sql remote.db dbeng16 remote.db

2. Interactive SQL の別のインスタンスを開き、remote.db に接続します。

dbisql -c "SERVER=remote;DBF=remote.db;UID=DBA;PWD=sql"

3. Interactive SQL で次の SQL 文を実行し、同期させるテーブルを作成します。

```
CREATE TABLE customer (
id unsigned integer primary key,
name varchar(256),
phone varchar(12));
```

4. パブリケーション、Mobile Link ユーザ、サブスクリプションを作成します。

```
CREATE PUBLICATION p1 (
TABLE customer
);
CREATE SYNCHRONIZATION USER u1;
CREATE SYNCHRONIZATION SUBSCRIPTION my_sub
TO p1
```

FOR u1 OPTION ScriptVersion='my ver1';

チュートリアルに従って作業する間に、データベースでさらに SQL 文を実行するため、この SQL 文の実行完了後も、引き続き Interactive SQL を実行し、データベースに接続した状態 にします。

結果

リモートデータベースが作成され、同期用に設定されます。

次の手順

「レッスン3:リモートデータベースの同期」268ページに進みます。

レッスン3:リモートデータベースの同期

この時点では、同期システムの設定が動作しているはずです。このレッスンでは、リモートデー タベースにデータを挿入して同期をテストします。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル: ScriptVersion 拡張オプ ションを使用したスキーマの変更」265ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成と設定」265ページを参照してください。

◆タスク

1. 統合データベースに接続された Interactive SQL のインスタンスを使用し、次の SQL 文を実行 して customer テーブルにローを挿入します。

INSERT INTO customer VALUES(100, 'John Jones', '519-555-1234'); COMMIT;

2. リモートデータベースに接続された Interactive SQL のインスタンスを使用し、次の SQL 文を 実行して customer テーブルにローを挿入します。

INSERT INTO customer VALUES(1, 'Willie Lowman', '705-411-6372'); COMMIT;

3. 次のコマンドを実行して同期します。

dbmlsync -v+ -ot sync1.txt -c UID=DBA;PWD=sql;SERVER=remote -s my_sub -k

customer テーブルの内容と統合データベースを比較することによって、同期が成功したこと を確認できます。dbmlsync ログの *syncl.txt* を調べて、エラーがないかを確認することもでき ます。

結果

統合データベースとリモートデータベースが同期されます。

次の手順

「レッスン4:統合データベースでのスキーマ変更の実行」269ページに進みます。

レッスン4:統合データベースでのスキーマ変更の実行

このレッスンでは、新しいカラムを customer テーブルに追加して、顧客の携帯電話番号を保存 します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル: ScriptVersion 拡張オプ ションを使用したスキーマの変更」265 ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成と設定」265ページを参照してください。

◆タスク

 統合データベースに接続された Interactive SQL のインスタンスを使用し、次の SQL 文を実行 して customer テーブルにローを挿入します。

ALTER TABLE customer ADD cell_phone VARCHAR(12) DEFAULT NULL;

 my_ver2 という新しいスクリプトバージョンを作成して、リモートデータベースから新しい スキーマで同期を処理します。古いスキーマを使用するリモートデータベースでは、古いス クリプトバージョン my ver1 が引き続き使用されます。

統合データベースで次の SQL 文を実行します。

CALL ml_add_column('my_ver2', 'customer', 'id', null); CALL ml_add_column('my_ver2', 'customer', 'name', null); CALL ml_add_column('my_ver2', 'customer', 'phone', null); CALL ml_add_column('my_ver2', 'customer', 'cell_phone', null);

- CALL ml_add_table_script('my_ver2', 'customer', 'upload_insert', 'INSERT INTO customer (id, name, phone, cell_phone) ' || 'VALUES ({ml r.id}, {ml r.name}, {ml r.phone}, {ml r.cell phone})');
- CALL ml_add_table_script('my_ver2', 'customer', 'download_cursor', 'SELECT id, name, phone, cell_phone from customer');

CALL ml_add_table_script('my_ver2', 'customer', 'download_delete_cursor', '--{ml_ignore}');

COMMIT;

結果

統合データベースに変更が加えられ、スキーマの変更を処理する新しいスクリプトバージョンが 作成されます。

次の手順

「レッスン5:リモートデータベースでのスキーマ変更の実行」270ページに進みます。

レッスン5:リモートデータベースでのスキーマ変更の実行

このレッスンでは。リモートデータベースを変更して、新しいカラムを customer テーブルに追加し、同期に使用されるスクリプトバージョンを変更します。

前提条件

このレッスンでは、このチュートリアルの冒頭の「権限」セクションに一覧になっているロール および権限を持っていることを前提としています。 「チュートリアル: ScriptVersion 拡張オプ ションを使用したスキーマの変更」265 ページ

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1:統合データベースの作成と設定」265ページを参照してください。

リモートデータベースを変更し同期用のスクリプトバージョンを作成する前に、アップロードを 必要とする customer テーブルに対する操作がないことを確認してください。このための最適な 方法として、sp_hook_dbmlsync_schema_upgrade フックでスキーマの変更を実行することが挙げ られます。このフックを使用すると、同期の開始時点で同期対象テーブルがロックされ、スキー マの変更が完了するまでロック状態が保持されるため、スキーマの変更が安全に実行されたこと を dbmlsync で確認できます。

内容と備考

警告

アップロードする操作があるときにスキーマを変更すると、スキーマ変更後にリモートデータ ベースで常に同期できなくなります。

◆タスク

 リモートデータベースで次の SQL 文を実行して、sp_hook_dbmlsync_schema_upgrade フック を作成します。このフックにより、新しいカラムが customer テーブルに追加され、サブスク リプションで格納された ScriptVersion 拡張オプションの値が変更されます。このフックは、 実行後に dbmlsync によって削除されます。

CREATE PROCEDURE sp_hook_dbmlsync_schema_upgrade() BEGIN ALTER TABLE customer ADD cell_phone varchar(12) default null;

ALTER SYNCHRONIZATION SUBSCRIPTION my_sub ALTER OPTION ScriptVersion='my_ver2'; UPDATE #hook_dict SET value = 'always' WHERE name = 'drop hook'; END;

2. sp_hook_dbmlsync_schema_change フックを実行して、アップロードを必要とする操作のアップロードを同期し、スキーマを変更します。次のコマンドを実行します。

dbmlsync -v+ -ot sync2.txt -c UID=DBA;PWD=sql;SERVER=remote -s my_sub -k

この同期が完了したら、dbmlsync ログ sync2.txt を調べて、スキーマの変更が完了しなかった ことを示すエラーがないことを確認してください。

結果

これで、スキーマの変更が完了し、正常に同期を続行できます。

次の手順

なし。

チュートリアル:Mobile Link リプレイユーティリティ を使った複数の Mobile Link クライアントのシミュ レート

このチュートリアルでは、mlreplay ユーティリティを使って複数の Mobile Link クライアントを 単一のコンピュータ上でシミュレートする方法について説明します。

必要なソフトウェア

•SQL Anywhere 16

権限

統合データベースで次のロールおよび権限を持つ必要があります。

●SYS AUTH RESOURCE ROLE 互換ロール

●MONITOR システム権限

リモートデータベースで次のロールおよび権限を持つ必要があります。

●SYS REPLICATION ADMIN ROLE システムロール

●SYS RUN REPLICATION ROLE システムロール

前提知識と経験

次の知識と経験が必要です。

●Mobile Link イベントスクリプトの基本的な知識

概要

- このチュートリアルでは、次の作業の方法について説明します。
- ●Mobile Link 統合データベースの設定
- ●Mobile Link サーバの起動による同期の記録とリプレイ
- ●mlreplay ユーティリティを使用した Mobile Link クライアントのシミュレート

参照

●「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』

レッスン1: Mobile Link 統合データベースの設定

このレッスンでは、Mobile Link 統合データベースを設定します。

前提条件

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link リプレイユーティ リティを使った複数の Mobile Link クライアントのシミュレート」271 ページ

◆タスク

1. 新しい作業ディレクトリを作成して、このチュートリアルで作成するすべてのサンプルファ イルを格納します。

このチュートリアルでは、パスを c:¥mlreplay とします。

2. コマンドプロンプトで、作業ディレクトリを c:¥mlreplay に変更します。

このチュートリアルでは、すべてのコマンドがこのディレクトリから実行されることを前提とします。

3. 次のコマンドを実行して、*cons.db* という名前の SQL Anywhere 統合データベースを作成しま す。

dbinit -dba DBA,sql cons.db

4. 次のコマンドを入力して、統合データベースを起動します。

dbsrv16 cons.db

- 5. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [ODBC データソースアドミ ニストレータ] をクリックします。
- 6. [ユーザ DSN] タブをクリックし、[追加] をクリックします。

- [データソースの新規作成] ウィンドウで、[SQL Anywhere 16] をクリックし、[完了] をクリックします。
- 8. [SQL Anywhere の ODBC 設定] ウィンドウで、次の操作を行います。
 - a. [ODBC] タブをクリックします。
 - b. [データソース名] フィールドに cons と入力します。
 - c. [ログイン] タブをクリックします。
 - d. [ユーザ ID] フィールドに DBA と入力します。
 - e. [パスワード] フィールドに sql と入力します。
 - f. **[アクション]** ドロップダウンリストから、**[このコンピュータで稼働しているデータベー** スに接続] を選択します。
 - g. [サーバ名] フィールドに、cons と入力します。
 - h. [データベース名] フィールドに cons と入力します。
 - i. [OK] をクリックします。
- 9. ODBC データソースアドミニストレータを閉じます。

[ODBC データソースアドミニストレータ] ウィンドウで [OK] をクリックします。

10. Interactive SQL の統合データベースに接続します。

次のコマンドを実行します。

dbisql -c "DSN=cons"

11. Interactive SQL で次の文を実行して、*syncsa.sql* 設定スクリプトを使用して Mobile Link のシス テムテーブルとストアドプロシージャを作成します。*C:¥Program Files¥SQL Anywhere 16¥* は、SQL Anywhere 16 インストール環境のロケーションに置き換えてください。

READ "C:¥Program Files¥SQL Anywhere 16¥MobiLink¥setup¥syncsa.sql";

Interactive SQL によって syncsa.sql が統合データベースに適用されます。

*syncsa.sql*を実行すると、前に **ml**_が付いた一連のシステムテーブルとストアドプロシージャ が作成されます。これらのテーブルとストアドプロシージャは、同期処理中に Mobile Link サーバによって使用されます。

12. Interactive SQL で次の SQL 文を実行し、T1 テーブルを作成します。

CREATE TABLE T1 (pk1 INTEGER, pk2 INTEGER, c1 VARCHAR(30000), PRIMARY KEY(pk1, pk2));

Interactive SQL によって、統合データベースに T1 テーブルが作成されます。

13. Interactive SQL を閉じます。

結果

統合データベースが設定されます。

次の手順

「レッスン2: Mobile Link プロジェクトの作成」274ページに進みます。

レッスン2: Mobile Link プロジェクトの作成

このレッスンでは、新しい Mobile Link プロジェクトを作成して、統合データベースに接続します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」272 ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link リプレイユーティ リティを使った複数の Mobile Link クライアントのシミュレート」271 ページ

◆タスク

- 1. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [Sybase Central] をクリッ クします。
- 2. [ツール] » [Mobile Link 16] » [新しいプロジェクト] をクリックします。
- 3. [新しいプロジェクトの名前を指定してください。] フィールドに mlreplay_project と入力します。
- 4. [**ロケーション**] フィールドに C:¥mlreplay と入力し、[次へ] をクリックします。
- 5. [データベースの表示名] フィールドに cons と入力します。
- 6. [編集] をクリックします。[汎用 ODBC データベースに接続] ウィンドウが表示されます。
- 7. [ユーザ ID] フィールドに、DBA と入力します。
- 8. [パスワード] フィールドに、sql と入力します。
- 9. [ODBC データソース名] フィールドで、[参照] をクリックして cons を選択します。
- 10. [OK] をクリックし、[保存] をクリックします。
- 11. [パスワードを記憶] オプションをオンにし、ウィザードの最後に達するまで [次へ] をクリックしてすべてのデフォルトをそのまま使用します。

チュートリアル : Mobile Link リプレイユーティリティを使った複数の Mobile Link クライアントのシ ミュレート

12. [完了] をクリックします。

結果

Mobile Link プロジェクトが作成されます。

次の手順

「レッスン3:同期スクリプトの追加」275ページに進みます。

レッスン3:同期スクリプトの追加

スクリプトは、それぞれ指定のスクリプトバージョンに属しています。スクリプトは、統合デー タベースにスクリプトバージョンを追加した後に追加してください。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」272 ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link リプレイユーティ リティを使った複数の Mobile Link クライアントのシミュレート」271 ページ

内容と備考

Sybase Central を使用して同期スクリプトを表示、作成、修正できます。このレッスンでは、次の同期スクリプトを作成します。

- upload_insert このイベントは、新しいクライアント側データを統合データベースに適用す る方法を定義します。
- download_cursor このイベントは、リモートクライアントにダウンロードするデータを定 義します。
- download_delete_cursor このイベントは、アップロード専用の同期スクリプトを使用する 場合に必要です。このチュートリアルでは、このイベントを無視するように Mobile Link サー バを設定します。

◆ タスク

- 1. [ビュー] »[フォルダ] をクリックします。
- 2. Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、mlreplay_project、[統合データベース]、cons DBA の順に展開します。
- 3. [バージョン] を右クリックし、[新規]»[バージョン] を選択します。
- 4. [新しいスクリプトバージョンの名前を指定してください。] フィールドに MLReplayDemo と 入力します。

- 5. [完了] をクリックします。
- 6. Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、mlreplay_project、[統合データベース]、cons DBA の順に展開します。
- 7. [同期テーブル] を右クリックし、[新規] » [同期テーブル] をクリックします。
- 8. [リモートテーブルと同じ名前のテーブルを統合データベースで選択する] オプションをク リックします。
- 9. [同期するテーブルの所有者を指定してください。] リストで [DBA] をクリックします。
- 10. [同期するテーブルを指定してください。] リストで [T1] をクリックします。
- 11. [完了] をクリックします。

T1 テーブルは同期テーブルとして登録され、そのテーブルにスクリプトを追加できます。

- 12. Sybase Central の左ウィンドウ枠の [Mobile Link 16] で、mlreplay_project、[統合データベース]、cons DBA、[同期テーブル] の順に展開します。
- 13. [T1] を右クリックし、[新規] » [テーブルスクリプト] をクリックします。
- 14. [テーブルスクリプトを作成するバージョンを指定してください。] リストで MLReplayDemo をクリックします。
- 15. [テーブルスクリプトを実行するイベントを指定してください。] リストで upload_insert をクリックし、[次へ] をクリックします。
- 16. [完了] をクリックします。
- 17. Sybase Central の右ウィンドウ枠で、upload_insert イベント用の次の SQL スクリプトを使用 します。

INSERT INTO T1 VALUES(cast({ml s.remote_id} as INTEGER), {ml r.2}, {ml r.3});

upload_insert イベントは、リモートデータベースに挿入されたデータが統合データベースに どのように適用されるかを決定します。「upload_insert テーブルイベント」『Mobile Link サー バ管理』を参照してください。

- 18. [ファイル]» [保存] をクリックします。
- 19. 手順 13 ~ 16 を繰り返して、手順 15 の upload_insert イベントの代わりに download_cursor イベントを指定します。
- 20. Sybase Central の右ウィンドウ枠で、download_cursor イベント用の次の SQL スクリプトを使用します。

SELECT pk1, pk2, c1 FROM T1;

download_cursor スクリプトは、ダウンロードしてリモートデータベースに(挿入または更新 によって)取り込む必要があるローを、統合データベースから選択するためのカーソルを定義 します。 download_cursor の詳細については、「download_cursor テーブルイベント」『Mobile Link サーバ管理』を参照してください。

- 21. [ファイル]»[保存] をクリックします。
- 22. 手順 13 ~ 16 を繰り返して、手順 15 の upload_insert イベントの代わりに download_delete_cursor イベントを指定します。
- 23. Sybase Central の右ウィンドウ枠で、download_delete_cursor イベント用の次の SQL スクリプ トを使用します。

--{ml_ignore}

24. [ファイル]»[保存] をクリックします。

結果

同期スクリプトが作成されます。

次の手順

「レッスン4:記録のための Mobile Link サーバの起動」277 ページに進みます。

参照

- 「Mobile Link イベントの概要」『Mobile Link サーバ管理』
 「スクリプトの追加と削除」『Mobile Link サーバ管理』
 「ローをアップロードするスクリプト」『Mobile Link サーバ管理』
 「ローをダウンロードするスクリプト」『Mobile Link サーバ管理』
 「upload_insert テーブルイベント」『Mobile Link サーバ管理』
 「upload_update テーブルイベント」『Mobile Link サーバ管理』
 「upload_delete テーブルイベント」『Mobile Link サーバ管理』
 「download_cursor テーブルイベント」『Mobile Link サーバ管理』
 「download_delete_cursor テーブルイベント」『Mobile Link サーバ管理』
 「ダイレクトローハンドリング」『Mobile Link サーバ管理』
 「ダイレクトアップロード」『Mobile Link サーバ管理』
 「ダイレクトダウンロード」『Mobile Link サーバ管理』
- ●「タイムスタンプベースのダウンロードの実装」『Mobile Link サーバ管理』
- ●「リモートデータベース間でローを分割する」『Mobile Link サーバ管理』

レッスン4:記録のための Mobile Link サーバの起動

このレッスンでは、-c オプションを使って Mobile Link サーバ (mlsrv16) を起動し、統合データ ベースに接続します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」272 ページを参照してください。 このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link リプレイユーティ リティを使った複数の Mobile Link クライアントのシミュレート」271 ページ

◆ タスク

● 次のコマンドを実行して、統合データベースに接続します。

mlsrv16 -c "DSN=cons" -zu+ -zs mlreplay_svr -x tcpip -ot mlsrv.mls -v+ -rp .

使用している Mobile Link サーバの各オプションの説明を次に示します。-ot および-v オプションは、デバッグとトラブルシューティングの情報を提供します。これらのロギングオプションは、開発環境での使用に適しています。一般に、パフォーマンス上の理由から、-v は運用環境では使用しません。

オプション	説明
-c	続いて接続文字列を指定します。
-ot	メッセージログファイル mlsrv.mls を指定します。
-v+	ログを取る対象となる情報を指定します。- v+を使用して、最大冗長ロギングをオンに 設定します。
-rp	同期がプレイバック用に記録されるディレ クトリを指定します。
-X	同期要求を受信するために使用するプロト コルを設定します。
-ZS	Mobile Link サーバ名を設定します。
-zu+	自動的に新しいユーザを追加します。

結果

Mobile Link サーバが起動し、統合データベースに接続されます。Mobile Link サーバメッセージ ウィンドウが表示されます。

次の手順

「レッスン5: Mobile Link クライアントデータベースの設定」279ページに進みます。

参照

●「Mobile Link サーバオプション」『Mobile Link サーバ管理』

レッスン 5: Mobile Link クライアントデータベースの設定

Mobile Link は、統合データベースサーバと多数のモバイルデータベースとの間で同期を実行で きるように設計されています。このレッスンでは、リモートデータベースを作成し、T1 テーブ ル(このテーブルは統合データベースと同期する)を作成し、同期パブリケーション、ユーザ、 サブスクリプションを作成します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」272 ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link リプレイユーティ リティを使った複数の Mobile Link クライアントのシミュレート」271 ページ

内容と備考

このレッスンでは、SQL Anywhere データベースを統合データベースと Mobile Link クライアント に使用します。また、このチュートリアルの目的上、Mobile Link クライアント、統合データベー ス、および Mobile Link サーバはすべて同じコンピュータに置きます。

Mobile Link クライアントデータベースを設定するために、リモートデータベースに対して T1 テーブルを作成します。T1 テーブルは、統合データベースの T1 テーブルに対応します。Mobile Link サーバでは、SQL ベースのスクリプトを使用して製品数が同期されます。

テーブルの作成後に、クライアントデータベースで同期ユーザ、パブリケーション、サブスクリ プションを作成します。パブリケーションは、リモートデータベース上の同期対象となるテーブ ルとカラムを識別します。これらのテーブルとカラムを「アーティクル」と呼びます。同期サブ スクリプションは、パブリケーションに対する Mobile Link ユーザのサブスクリプションです。

◆ タスク

1. dbinit コマンドラインユーティリティを使用して、Mobile Link クライアントデータベースを 作成します。

次のコマンドを実行して、remote データベースを作成します。

dbinit -dba DBA,sql remote.db

2. dbeng16 コマンドラインユーティリティを使用して、Mobile Link クライアントデータベース を起動します。

次のコマンドを実行して、remote データベースを起動します。

dbeng16 remote

3. Interactive SQL を使用して、リモートデータベースに接続します。

次のコマンドを実行します。

dbisql -c "SERVER=remote;UID=DBA;PWD=sql"

4. remote データベース用に T1 テーブルを作成します。

Interactive SQL で次の SQL 文を実行します。

```
CREATE TABLE T1 (

pk1 INTEGER,

pk2 INTEGER,

c1 VARCHAR(30000),

PRIMARY KEY(pk1,pk2)

);
```

SET OPTION PUBLIC.ml_remote_id = '0';

5. remote データベース用に Mobile Link 同期ユーザ、パブリケーション、サブスクリプション を作成します。

Interactive SQL で次の SQL 文を実行します。

CREATE PUBLICATION P1 (TABLE T1); CREATE SYNCHRONIZATION USER U1; CREATE SYNCHRONIZATION SUBSCRIPTION TO P1 FOR U1 TYPE 'TCPIP' ADDRESS 'host=localhost;port=2439';

6. 次のレッスンのために Interactive SQL を開いたままにします。

結果

リモートデータベース、**T1** テーブル、同期パブリケーション、ユーザ、サブスクリプションが 作成されます。

次の手順

「レッスン6:同期の記録」280ページに進みます。

参照

- ●「初期化ユーティリティ (dbinit)」『SQL Anywhere サーバ データベース管理』
- ●「Mobile Link クライアント」『Mobile Link クライアント管理』
- ●「CREATE SYNCHRONIZATION USER 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファ レンス』
- ●「CREATE PUBLICATION 文 [Mobile Link] [SQL Remote]」『SQL Anywhere サーバ SQL リファ レンス』
- ●「CREATE SYNCHRONIZATION SUBSCRIPTION 文 [Mobile Link]」『SQL Anywhere サーバ SQL リファレンス』

レッスン6:同期の記録

このレッスンでは、dbmlsync ユーティリティを実行して SQL Anywhere リモートデータベース用の Mobile Link 同期を開始します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」272 ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link リプレイユーティ リティを使った複数の Mobile Link クライアントのシミュレート」271 ページ

◆タスク

1. スキーマが Mobile Link サーバ上でキャッシュされるように、最初に記録された同期を実行します。

次のコマンドを実行して、remote データベースを同期します。

dbmlsync -c "SERVER=remote;UID=DBA;PWD=sql" -ot remote1.mls -e "sv=MLReplayDemo" -v+

次の表は、使用されている各 dbmlsync オプションを説明しています。

オプション	説明
-c	接続文字列を指定します。
-ot	メッセージのログの記録先ファイルを指定します。
-е	同期先のスクリプトバージョンを指定します。
-v+	ログを取る対象となる情報を指定します。-v+を使用して、最大冗長ロギ ングをオンに設定します。

同期が成功したことを示す出力画面が表示されます。SQLベースの同期によって、クライアントのT1テーブル内のローが、統合データベース内のT1テーブルに転送されました。

2.2回目の同期が発生するように、データ挿入のためにリモートデータベースを準備します。

Interactive SQL で引き続き remote データベースに接続されている必要があります。接続されていない場合は、次のコマンドを実行して、remote データベースに接続します。

dbisql -c "SERVER=remote;UID=DBA;PWD=sql"

3. リプレイセッション中に Mobile Link サーバにアップロードするように、データを remote データベースにロードします。

Interactive SQL で次の SQL 文を実行します。

INSERT INTO T1 (pk1,pk2,c1) values (0,1,'data1'); INSERT INTO T1 (pk1,pk2,c1) values (0,2,'data2'); INSERT INTO T1 (pk1,pk2,c1) values (0,3,'data3'); INSERT INTO T1 (pk1,pk2,c1) values (0,4,'data4'); INSERT INTO T1 (pk1,pk2,c1) values (0,5,'data5'); INSERT INTO T1 (pk1,pk2,c1) values (0,6,'data6'); INSERT INTO T1 (pk1,pk2,c1) values (0,7,'data7'); INSERT INTO T1 (pk1,pk2,c1) values (0,8,'data8'); INSERT INTO T1 (pk1,pk2,c1) values (0,9,'data9'); INSERT INTO T1 (pk1,pk2,c1) values (0,10,'data10'); COMMIT;

4. 2回目に記録した同期を実行します。これは、リプレイされるプロトコルです。

次のコマンドを実行して、remote データベースを同期します。

dbmlsync -c "SERVER=remote;UID=DBA;PWD=sql" -ot remote2.mls -e "sv=MLReplayDemo" -v+

結果

データベースが同期されます。

次の手順

「レッスン7:リプレイのための Mobile Link サーバの再起動」282ページに進みます。

参照

- ●「SQL Anywhere クライアント」『Mobile Link クライアント管理』
- ●「Mobile Link SQL Anywhere クライアントユーティリティ (dbmlsync)」『Mobile Link クライア ント管理』

レッスン7:リプレイのための Mobile Link サーバの再起動

このレッスンでは、記録を停止するために Mobile Link サーバを停止してから、-rp オプションを 指定せずにサーバを再起動して、サーバでリプレイの準備をします。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」272 ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link リプレイユーティ リティを使った複数の Mobile Link クライアントのシミュレート」271 ページ

◆ タスク

1. 次のコマンドを実行して、Mobile Link サーバである mlreplay_svr を停止します。

mlstop -w -t 1m mlreplay_svr

Mobile Link サーバは、同期記録とともに停止します。

次の表は、使用されている各オプションを説明しています。
オプション	説明
-W	コマンドプロンプトに戻る前に、サーバがシャットダウンされるのを待機 します。
-t	1 分後、または現在の同期が完了した後のどちらか早い時期にサーバを シャットダウンするように指定します。

2. 次のコマンドを実行して、統合データベースに接続します。

mlsrv16 -c "DSN=cons" -zu+ -zs mlreplay_svr -x tcpip -ot server_replay.mls -v+

Mobile Link サーバメッセージウィンドウが表示されます。

使用している Mobile Link サーバの各オプションの説明を次に示します。-ot および-v オプションは、デバッグとトラブルシューティングの情報を提供します。これらのロギングオプションは、開発環境での使用に適しています。一般に、パフォーマンス上の理由から、-v は運用環境では使用しません。

オプション	説明
-с	接続文字列を指定します。
-ot	メッセージログファイル server_replay.mls を指定します。
-v+	ログを取る対象となる情報を指定します。-v+ を使用して、最大冗長ロ ギングをオンに設定します。
-X	同期要求を受信するために使用するプロトコルを設定します。
-ZS	Mobile Link サーバ名を設定します。
-zu+	自動的に新しいユーザを追加します。

結果

Mobile Link サーバが停止してから再度起動します。

次の手順

「レッスン8:同期のリプレイ」284ページに進みます。

参照

●「Mobile Link サーバオプション」『Mobile Link サーバ管理』

レッスン8:同期のリプレイ

このレッスンでは、スキーマが Mobile Link サーバにキャッシュされるように同期を実行します。 シミュレートされたクライアント情報ファイルを作成し、シミュレートされたクライアントで Mobile Link プロトコル情報をリプレイします。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としていま す。「レッスン1: Mobile Link 統合データベースの設定」272 ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link リプレイユーティ リティを使った複数の Mobile Link クライアントのシミュレート」271 ページ

内容と備考

シミュレートされたクライアント情報ファイルは、記録されたプロトコルを、複数のシミュレートされたクライアント間で同時にリプレイするときにだけ必要です。

◆ タスク

1. 次のコマンドを実行して、remote データベースを同期します。

dbmlsync -c "SERVER=remote;UID=DBA;PWD=sql" -ot remote3.mls -e "sv=MLReplayDemo" -v+

次の表は、使用されている各 dbmlsync オプションを説明しています。

オプション	説明
-c	接続文字列を指定します。
-ot	メッセージのログの記録先ファイルを指定します。
-е	同期先のスクリプトバージョンを指定します。
-v+	ログを取る対象となる情報を指定します。-v+を使用して、最大冗長ロギ ングをオンに設定します。

同期が成功したことを示す出力画面が表示されます。SQL ベースの同期によって、クライアントのT1 テーブル内のローが、統合データベース内のT1 テーブルに転送されました。

2. mlreplay ユーティリティで使用するために、シミュレートされたクライアント情報ファイル を作成します。

新しいテキストファイルを作成して、表示されているように次のカンマ区切りリストに書き 込みます。

mlreplay1,,1, mlreplay2,,2, mlreplay3,,3, mlreplay4,,4, mlreplay5,,5, mlreplay6,,6, mlreplay7,,7, mlreplay8,,8, mlreplay9,,9, mlreplay10,,10,

3. 作業ディレクトリに、そのファイルを mlreplay.csv として保存します。

クライアント情報ファイルは、10個のリモートクライアントをシミュレートするために使用 できます。

4. シミュレートされクライアントで、記録した同期をリプレイします。

次のコマンドを実行します。

mlreplay -ap -x tcpip -ot mlreplay.mls -sci mlreplay.csv recorded_protocol_mlreplay_svr_2.mlr

次の表は、使用	されている各オプショ	ンを説明しています。
---------	------------	------------

オプション	説明
-ap	mlreplay ユーティリティによって Mobile Link サーバ上で進行オフセット 不一致警告が生成されないように、リプレイセッションでリプレイされて いる同期の進行を調整します。
-X	同期要求を受信するために使用するプロトコルを設定します。
-ot	メッセージのログを取るファイルを指定します。
-sci	クライアント情報ファイルのロケーションを指定します。

mlreplay ユーティリティは、*recorded_protocol_mlreplay_svr_2.mlr* という記録されたプロトコルファイルに、接続の開始から接続の最後までの情報を格納します。

5. テキストエディタで *mlreplay.mls* ログファイルを開き、Mobile Link リプレイの結果を確認し ます。

結果

同期が実行され、スキーマが Mobile Link サーバ上でキャッシュされ、シミュレートされたクラ イアントに関する Mobile Link プロトコル情報をリプレイするためにシミュレートされたクライ アント情報ファイルが作成されます。

次の手順

「クリーンアップ」286ページに進みます。

参照

●「Mobile Link Replay ユーティリティ (mlreplay)」『Mobile Link サーバ管理』

クリーンアップ

チュートリアルをコンピュータから削除します。

前提条件

このレッスンは、受講者がこれまでのすべてのレッスンを終了していることを前提としています。「レッスン1: Mobile Link 統合データベースの設定」272 ページを参照してください。

このレッスンでは、このチュートリアルの開始時に、権限のセクションで一覧されているロール と権限を持っていることを前提としています。「チュートリアル: Mobile Link リプレイユーティ リティを使った複数の Mobile Link クライアントのシミュレート」271 ページ

◆ タスク

- 1. タスクバー上で、Interactive SQL、SQL Anywhere、Mobile Link、同期クライアントの各ウィンドウを右クリックし、[閉じる] を選択して閉じます。
- 2. 次の手順で、チュートリアルに関連するすべてのデータソースを削除します。
 - a. ODBC データソースアドミニストレータを起動します。
 - b. [スタート] » [プログラム] » [SQL Anywhere 16] » [管理ツール] » [ODBC データソースア ドミニストレータ] をクリックします。
 - c. [ユーザデータソース] リストから cons を選択し、[削除] をクリックします。
- 統合データベースとリモートデータベースが保存されているディレクトリを削除します。

結果

チュートリアルがコンピュータから削除されます。

索引

記号

.NET Mobile Link サーバ API の利点, 11 .NET 同期論理 説明, 11 .NET 用 Mobile Link サーバ API 利点, 13

A

authenticate_user Sybase Central, 46

С

CodeXchange Mobile Link $\forall \mathcal{V}\mathcal{P}\mathcal{N}, 7$ COMMIT 文 Mobile Link 警告, 17 Contact Mobile Link サンプル Contact テーブル,80 Customer テーブル,78 Product テーブル,81 SalesRep テーブル,77 構築,73 実行,73 テーブル,75 統計のモニタリング,83 ユーザ,77 custase.sql ロケーション、59 CustDB Mobile Link ULProduct テーブル, 70 Mobile Link サンプルアプリケーション, 57 Mobile Link サンプルテーブル, 63 Mobile Link ユーザ, 66 SQL Anywhere CustDB 統合データベース, 59 リストア,72 custdb.sqc ロケーション,62 CustDB Mobile Link サンプル ULCustomer テーブル, 69 ULOrder テーブル, 67 CustDB アプリケーション DB2, 59 同期スクリプト,59

```
custmss.sql
ロケーション,59
custora.sql
ロケーション,59
```

D

DB2 CustDB アプリケーション, 59

Ε

Excel Mobile Link チュートリアル, 190

IBM DB2 CustDB アプリケーション, 59 IMAP 認証 同期モデル, 46

J

```
Java
Mobile Link サーバ API の利点, 11
同期論理, 11
Java 同期論理
説明, 11
Java 用 Mobile Link サーバ API
利点, 12
```

L

LDAP 同期モデル,46

Μ

Mobile Link と SQL Anywhere のチュートリア ル,105 Oracle チュートリアル, 118 アーキテクチャ,1 機能,2 クイックスタート,5 サンプル,7 システム設定のチェック,26 集中管理のチュートリアル,231 処理の概要,13 説明,1 同期の基本,1 同期論理の作成オプション,11 Mobile Link Contact サンプル 説明,72 Mobile Link アップロード 処理,18 定義,14 Mobile Link アプリケーション 開発方法,11 設計,7 Mobile Link イベント 概要,15 Mobile Link サーバ はじめに、5 Mobile Link サーバ API 利点,12 Mobile Link システム設定 チェック,26 Mobile Link スクリプト 説明,15 Mobile Link ダイレクトローハンドリング チュートリアル,164 Mobile Link ダウンロード 定義,14 Mobile Link 同期 custdb サンプルアプリケーション,57 Mobile Link 同期処理 説明,5 Mobile Link の機能 説明,2 Mobile Link プロジェクト 作成,23 説明,22 Mobile Link プロジェクトの作成 Sybase Central タスク, 23

0

Oracle Mobile Link チュートリアル, 118

Ρ

POP3 認証 同期モデル,46

S

ScriptVersion 拡張オプション リモートスキーマの変更チュートリアル,265 SCRIPT VERSION 句 リモートスキーマの変更チュートリアル,257 SQLE_ROW_DELETED_TO_MAINTAIN_REFER ENTIAL_INTEGRITY Ultra Light 同期, 19 SQL 同期論理 代替手段, 11 Sybase Central の Mobile Link 16 プラグイン 説明,22

U

upload_delete Contact サンプル, 81 CustDB サンプル, 70

Χ

XML の Mobile Link ダイレクトローハンドリン グ チュートリアル, 210

あ

アップロード
Mobile Link 処理, 18
Mobile Link 定義, 14
Mobile Link トランザクション, 17
アップロードの処理方法
説明, 18
アプリケーション
随時接続, 7
スマートクライアント, 7

い

イベント Mobile Link の概要, 15 [イベント] タブ 同期モデル,45

お

オンラインアプリケーション Mobile Link, 7

か

外部サーバ
 同期モデル内の認証,46
 外部サーバに対する認証
 同期モデル,46
 カスケード削除
 Mobile Link 同期,19

き

競合解決
 Contact サンプル, 81
 CustDB サンプル, 70
 競合の検出と解決
 同期モデルでの変更, 44
 同期モデル内, 43

<

クイックスタート Mobile Link, 5

さ

再展開 同期モデル,51 削除 同期モデル内の変更,38 サブセット 説明,39 同期モデルでの変更,40 サブセットのダウンロード 説明,39 参照整合性 Mobile Link 同期, 19 参照整合性と同期 Mobile Link $2 \overline{7}$, 19 サンプル Contact Mobile Link サンプル, 72 Mobile Link, 7 Mobile Link CustDB アプリケーション, 57 サンプルアプリケーション Mobile Link CustDB アプリケーション, 57 サンプルデータベース

Mobile Link CustDB アプリケーション, 57

L

システム設定 Mobile Link のチェック, 26 システム設定のチェック Mobile Link, 26 障害 Mobile Link 同期リカバリ, 17

す

```
随時接続アプリケーション
 Mobile Link, 7
スキーマ
 同期モデルの再展開,51
スキーマ更新ウィザード
 説明,47
スキーマの更新
 スキーマ更新ウィザード,47
 同期モデルの再展開、51
スキーマの変更
 同期モデルの再展開,51
スクリプト
 Mobile Link の概要, 15
 テーブルの検索,45
 同期モデルでの変更,45
スマートクライアントアプリケーション
 Mobile Link, 7
```

せ

セキュリティ Mobile Link の概要, 21 設定 Mobile Link 同期, 5 同期モデル作成ウィザードを使用した Mobile Link, 30

た

ダイレクトローハンドリング Microsoft Excel のチュートリアル, 190 XML のチュートリアル, 210 チュートリアル, 164 ダウンロード Mobile Link 定義, 14 Mobile Link トランザクション, 17 ダウンロードサブセット 同期モデルでの変更, 40 ダウンロードタイプ 同期モデル内の変更,37

ち

チュートリアル Microsoft Excel の Mobile Link ダイレクトロー ハンドリング,190 Mobile Link (ASE), 136 Mobile Link (Oracle), 118 Mobile Link スクリプトと競合解決のモニタリ ング,85 Mobile Link $\forall \mathcal{I} \lor \mathcal{I} \land \mathcal{I} \land$ Mobile Link & SQL Anywhere, 105 ScriptVersion 拡張オプションを使用したス キーマの変更,265 XML の Mobile Link ダイレクトローハンドリ ング、210 カスタムユーザ認証用の Java または .NET, 155 スクリプトバージョン句を使用したスキーマ の変更チュートリアル,257 複数の Mobile Link クライアントのシミュレー F. 271 リモートデータベースの集中管理.231

つ

通信フォールト Mobile Link 同期リカバリ, 17

τ

```
テストウィンドウ
Mobile Link, 49
デッドロック
Mobile Link アップロードの処理, 18
展開
同期モデルでのバッチファイル, 52
データ移動テクノロジ
Mobile Link 同期, 1
データベース
Mobile Link との同期, 1
テーブルマッピング
説明, 33
```

と

同期 Mobile Link ASE チュートリアル, 136 Mobile Link Oracle チュートリアル, 118

Mobile Link システムのアーキテクチャ,1 Mobile Link 処理の概要,13 Mobile Link と SQL Anywhere のチュートリア ル,105 Mobile Link トランザクション, 17 Mobile Link の説明, 1 Mobile Link $\mathcal{O}\mathcal{P}\mathcal{A}\mathcal{A}\mathcal{P}\mathcal{P}$, 17 Mobile Link $n 7 \pi - 7 \lambda$, 19 クイックスタート,5 展開した同期モデル,52 同期論理の作成オプション,11 同期システム コンポーネント、1 同期処理 説明,13 同期テーブル マッピングの追加,33 同期のアップロード Mobile Link 処理, 18 同期の基本 説明,1 同期の障害処理の方法 Mobile Link, 17 同期の方法 custdb サンプルアプリケーション,57 Mobile Link Contact $\forall \mathcal{V}\mathcal{I}\mathcal{V}$, 72 同期モデル 概要,29 削除,31 作成,30 制限事項,54 説明,29 タスク,32 テスト、49 展開,48 リモートデータベース,31 同期モデル作成ウィザード 使用法,30 同期モデル展開ウィザード 使用法,48 同期モデルの作成 Sybase Central タスク, 29 同期論理 作成オプション,11 同期論理の作成オプション 説明,11 統合データベース 追加,25

変更, 26 同時実行性 Mobile Link アップロードの処理, 18 トラブルシューティング Mobile Link 同期の障害, 17 トランザクション Mobile Link 同期中, 17 Mobile Link のコミットとロールバック, 17

は

はじめに Mobile Link, 5 バッチファイル 同期モデルの展開, 52 パフォーマンス Mobile Link アップロードの処理, 19

ふ

フォールト Mobile Link 同期リカバリ, 17
フォールトトレラント 説明, 17
プラグイン Mobile Link, 22
プロジェクト 作成, 23 統合データベースの追加, 25
プロトコル Mobile Link 同期, 1
分散データベース Mobile Link 同期, 1

ま

マッピング テーブルとカラム,33

ŧ

モデル Mobile Link 同期, 29 モデル、Mobile Link 制限事項, 54

り

リモートスキーマ 同期モデル,31 リモートスキーマの変更チュートリアル ScriptVersion 拡張オプション,265 SCRIPT VERSION 句, 257 リモートデータベースの中央管理 ロールと権限, 231

ろ

ログファイル Mobile Link, 100 ロールバック Mobile Link 警告, 17