



Mobile Link SQL パススルー

Joshua Savillt 著
2008/10/17

このホワイトペーパーは、SQL Anywhere 11を対象に書かれました。

目次

はじめに	3
SQL パススルー・スクリプトの作成	3
リモート・データベースの SQL パススルー・エントリの作成	4
SQL パススルー・スクリプトのダウンロードと実行	5
リモート・データベースでの SQL パススルー・スクリプトの手動実行	5
リモート・データベースでの SQL パススルー・スクリプトの自動実行	5
スクリプト結果の取得と確認	6
リモート・データベースで実行中にエラーが発生したスクリプトの修正	6
統合データベースにおけるスクリプト管理	7
SQL パススルーのための Mobile Link サーバ冗長性	7
SQL パススルー・スクリプトの削除	7
SQL パススルー・エントリの削除	8
SQL パススルー修正情報の削除	8
デモ - デモ・データベースを空のリモート・データベースに移動する	8
SQL Anywhere 統合データベースの作成	9
SQL Anywhere リモート・データベースの作成	9
統合データベースにおける SQL パススルーの構成	10
統合データベースで SQL パススルー・スクリプトの作成を確認する	14
リモート・データベースの同期	14
SQL パススルー・スクリプトの実行を確認する	16
デモの実行手順	16
デモ - SQL パススルー・スクリプトの実行中に発生したエラーを処理する	17
SQL Anywhere 統合データベースの作成	17
SQL Anywhere リモート・データベースの作成	17
リモート・データベース 1	18
リモート・データベース 2	18
統合データベースにおける SQL パススルーの構成	19
リモート・データベースの同期	20
統合データベースにおける SQL パススルー修正スクリプトの構成	21
デモの実行手順	23
まとめ	23

図

図 1 - ml_passthrough_script テーブル内の SQL パススルー・スクリプト	14
図 2 - ml_passthrough テーブル内の SQL パススルー・エントリ	14
図 3 - リモート・データベースにおける SQL パススルーの状況	15
図 4 - リモート・データベースの SQL パススルー・スクリプト	15
図 5 - リモート・データベースの SQL パススルーの進行状況	16
図 6 - SQL パススルー・スクリプトの結果	16
図 7 - スクリプトの失敗による Mobile Link クライアントのエラー	20
図 8 - リモート・データベース 1 における SQL パススルーの状況	20
図 9 - 統合データベースにおける SQL パススルーの状況	21
図 10 - 修正スクリプト実行後の リモート・データベース 1 における SQL パススルー・ステータス	22
図 11 - 修正スクリプト実行後の 統合データベースにおける SQL パススルー・ステータス	23

はじめに

新たな SQL パススルー機能を使用することにより、Mobile Link 同期環境においてリモート・データベースの強力かつ堅牢な管理を実現できます。管理者は、SQL パススルー機能を使用して、統合データベースから 1 つまたは複数のデータベースに SQL 文のスクリプトをダウンロードし、それらの SQL 文を適切な時間にリモート・データベースで実行することができます。スクリプトは、作成時に指定された定義済みの順序に従って、リモート・データベースで確実に実行されます。

SQL パススルー機能に加えて、リモート・データベースでのスクリプトの実行状況をモニタし、実行できないスクリプトを修正する機能も提供されます。

SQL パススルー・スクリプトの作成

SQL パススルー・スクリプトは統合データベースで作成され、同期時にリモート・データベースにダウンロードされます。新しい SQL パススルー・スクリプトは、`ml_add_passthrough_script` プロシージャを使用して作成され、統合データベースの `ml_passthrough_script` テーブルに格納されます。

`ml_add_passthrough_script` プロシージャを呼び出すときに、スクリプト名、フラグ、関連パブリケーション、スクリプト内容、および説明のパラメータを指定する必要があります。スクリプト名パラメータを使用して、SQL パススルー・スクリプトを指定します。スクリプト名には、NULL 以外のユニークな名前を指定します。flags パラメータは、リモート・データベースでの SQL パススルー・スクリプトの実行方法を指定します。フラグは、`manual`、`exclusive`、および `schema_diff` のオプションを 1 つ指定するかまたは組み合わせて指定できます。

各オプションの意味は以下のとおりです。

- Manual** - スクリプトは、リモート・データベースで手動でのみ実行できます。デフォルトでは、`manual` を指定しないかぎり、すべてのスクリプトが自動的に実行されます。
- Exclusive** - 同期に含まれるすべてのテーブルで排他ロックが取得される場合、同期の最後に自動でのみスクリプトを実行できます。このオプションは、SQL Anywhere リモート・データベースにのみ適用され、関連パブリケーションを定義する必要があります。
- Schema_diff** - スクリプトが `schemadiffing` モードで実行されます。`schemadiffing` モードでは、スクリプトに記述されているスキーマと一致するようにデータベース・スキーマが変更されます。たとえば、既存のテーブルの `create` 文が `alter` 文として処理されます。このオプションは、Ultra Light データベースで実行されるスクリプトにのみ適用されます。

セミコロンを区切り文字として使用することにより、2 つ以上のオプションを指定することができます。たとえば、`'manual;exclusive'` のように指定します。NULL が指定されている場合のデフォルト・フラグは、空の文字列と同じであるため、自動実行になります。

関連パブリケーション・パラメータは、スクリプトの実行前にアップロードする必要がないパブリケーションをカンマ区切りでリストします。関連パブリケーション・リストに指定されているすべてのパブリケーションが同期される場合にのみ、同期の最後に自動でスクリプトが実行されます。スクリプトに `ALTER TABLE` 文が含まれている場合、ダウンロード専用のパブリケーションを除いて、変更されるテーブルが含まれているすべてのパブリケーションを関連パブリケーションに含める必要があります。それにより、テーブルの変更時に、それらのパブリケーションがアップロードされるのを待つ必要がなくなります。なお、この作業を怠ると、スクリプトの適用後に、リモート・データベースが同期を実行できなくなる可能性があります。関連パブリケーション・リストには、スクリプトの実行前に同期する必要があるパブリケーションがないことを意味する NULL を指定することもできます。Ultra Light リモート・データベースは、関連パブリケーション・リストを無視します。

スクリプトの内容パラメータは、リモート・データベースで実行される SQL 文の集まりです。SQL パススルー・スクリプトのスクリプト内容に NULL を指定することはできません。スクリプト内容は、SQL Anywhere および Microsoft SQL Server の統合データベースの TEXT データ型のサイズに制限されます。ASE の場

合、スクリプト内容は、VARCHAR データ型の 16384 に制限されます。また、IBM DB2 UDB の場合、スクリプト内容は VARCHAR データ型の 4000 に制限され、Oracle の場合は CLOB のサイズに制限されます。

SQL Anywhere リモート・データベースを使用する場合、スクリプト内容は、通常は BEGIN END ブロックに存在する SQL 文に適用される適切な構文でなければなりません。Ultra Light リモート・データベースを使用する場合、SQL 文を 'go' という単語で区切り、文ごとに個別の行を使用する必要があります。

Description パラメータを使用して、SQL パススルー・スクリプトに関する追加情報を追加することができます。Description パラメータには NULL を指定できます。

リモート・データベースの SQL パススルー・エントリの作成

統合データベースで SQL パススルー・スクリプトを作成したら、スクリプトがダウンロードされるリモート・データベースとの関連付けを行う必要があります。ml_add_passthrough プロシージャを使用して、指定したスクリプトをダウンロードするデータベース（複数可）を定義します。このパススルー関係は、統合データベースの ml_passthrough テーブルに格納されます。

ml_add_passthrough プロシージャを呼び出すときに、リモート ID、スクリプト名、および実行順序のパラメータを指定する必要があります。

リモート ID パラメータは、スクリプトを受信するリモート・データベースを指定します。リモート ID は、ml_database テーブルに格納されている既存のリモート ID と一致しなければなりません。リモート ID に NULL を指定した場合、ml_database テーブルに格納されているすべてのリモート・データベースにスクリプトが適用されます。

スクリプト名パラメータは、リモート・データベースにダウンロードされるスクリプトを指定します。スクリプト名は、ml_passthrough_script テーブルに存在する有効なスクリプト名でなければなりません。

実行順序パラメータは、リモート・データベースでスクリプトが適用される順序を指定します。スクリプトは、必ず実行順序に従って適用されます。実行順序パラメータには、正の整数を指定します。NULL を指定した場合、以下の操作により実行順序が自動的に決定されます。

1. リモート ID が null の場合、ml_passthrough テーブルの実行順序カラムの最大値に 10 を足した値になります。
2. リモート ID が null でない場合、ml_passthrough テーブルに格納されている指定されたリモート ID の実行順序カラムの最大値に 10 を足した値になります。

指定されたリモート ID および実行順序のローが ml_passthrough テーブルにすでに存在する場合、ml_add_passthrough プロシージャは、ml_passthrough テーブルのローに指定されたスクリプト名を更新します。

SQL パススルー・スクリプトのダウンロードと実行

ml_passthrough テーブルに定義されている条件をリモート・データベースが満たしている場合、同期時に SQL パススルー・スクリプトがリモート・データベースに自動的にダウンロードされます。リモートからの ping が Mobile Link サーバで実行されたり、ダウンロードが自動的に再開されたようなファイルベースの同期では、スクリプトはダウンロードされません。

SQL Anywhere リモート・データベースの場合、SQL パススルー・スクリプトは dbo.sync_passthrough_script テーブルに格納されます。Ultra Light リモート・データベースの場合、スクリプトは syssql テーブルに格納されます。

SQL パススルー・スクリプトは、スクリプトの定義に基づいて自動または手動で実行できます。すべてのスクリプトが、実行順序に従って実行されます。SQL Anywhere リモート・データベースの場合、すべてのスクリプトが、DBO アカウントの DBA 権限を使用して実行されます。つまり、データベース・オブジェクトの参照、作成、または

変更を行う場合、データベース・オブジェクトの所有者を指定する必要があるということです。たとえば、t1 ではなく DBA.t1 のように指定する必要があります。

リモート・データベースでの SQL パススルー・スクリプトの手動実行

SQL パススルー・スクリプトを手動で実行するには、`sync_get_next_passthrough_script` および

`sync_execute_next_passthrough_script` 関数を使用します。

`sync_get_next_passthrough_script` 関数は、次に実行されるスクリプトの実行順序を返し、それ以上実行するスクリプトがない場合は NULL を返します。また、最後のスクリプトの実行中にエラーが発生し、そのエラーの処理に関する命令を統合データベースから受信していない場合も、NULL が生成されます。

`sync_execute_next_passthrough_script` 関数は、次のスクリプトを実行し、リモート・データベースに格納されている進行状況の情報を更新します。この情報は、次の同期時に統合データベースにアップロードされます。最後のスクリプトがエラーを返し、エラーの処理に関する命令を統合データベースから受信していない場合、スクリプトは実行されません。スクリプトが実行された場合、スクリプトの実行順序が返されます。スクリプトが実行されなかった場合は、NULL が返されます。

Ultra Light リモート・データベースの場合、ESQL API メソッド、C++ メソッド、.NET メソッド、および Pod インタフェースを使用して、SQL パススルー・スクリプトを手動で実行することができます。

リモート・データベースでの SQL パススルー・スクリプトの自動実行

SQL Anywhere リモート・データベースの場合、各同期の最後に SQL パススルー・スクリプトが自動的に実行されます。実行可能なスクリプトが実行順序に従って順序付けされます。すべてのスクリプトが実行されるか、実行に失敗するか、または自動的に実行されないスクリプトに到達するまで、スクリプトが 1 つずつ実行されます。

以下の条件があてはまる場合、スクリプトは自動的に実行されません。

1. スクリプトに対して `manual` フラグが指定されている。
2. スクリプトの関連パブリケーションに値が設定されており、以下のいずれかがあてはまる場合。
 - a. アップロードが実行されなかった。
 - b. アップロードに失敗した。
 - c. 関連アプリケーション・リストの 1 つまたは複数のパブリケーションが同期されなかった。
 - d. スクリプトを実行するのに十分なテーブル・ロックが取得されていない。

Mobile Link クライアントは、同期の最後にスクリプトが確実に実行されるように、同期の最初に LockTables 拡張オプションを使用して、要求よりも制限的なロックを同期テーブルで取得することができます。たとえば、LockTables を `shared` に設定したにもかかわらず、次に実行可能なスクリプトが排他ロックを必要とする場合、排他ロックを取得することができます。

Ultra Light リモート・データベースの場合、自動的に実行されるように設定されたスクリプトは、データベースが次回起動されるときに実行されます。実行されないようにするには、Ultra Light データベースで `dont_run_scripts` オプションを設定します。

スクリプト結果の取得と確認

SQL Anywhere リモート・データベースの場合、実行された SQL パススルー・スクリプトの結果は `dbo.sync_passthrough_status` テーブルに格納されます。結果には、リモート・データベースでスクリプトが実行された時間、成功したかどうか、および報告されたエラーが示されます。スクリプトの実行中にエラーが報告された場合、メッセージ文とともに SQL コードが保存されます。

Ultra Light リモート・データベースの場合、結果は `sysssl` テーブルに格納されます。結果には、リモート・データベースでスクリプトが実行された時間、成功したかどうか、および報告されたエラーが示されます。スクリプトの実行中にエラーが報告された場合、SQL コード、スクリプト内の失敗した行番号、およびエラー・パラメータのリストが表示されます。

すべてのリモート・データベースが、各同期セッションの一部として実行されたすべてのスクリプトの結果をアップロードします。Mobile Link サーバは、アップロードされた結果を統合データベースの `ml_passthrough_status` テーブルに格納します。このテーブルをモニタすることで、Mobile Link 同期環境内に分散されているすべての SQL パススルー・スクリプトの成否を判断することができます。

リモート・データベースで実行中にエラーが発生したスクリプトの修正

SQL パススルー・スクリプトの実行状況は、`ml_passthrough_status` テーブルに格納されます。スクリプトの実行中にエラーが発生した場合、手動で操作してエラーを解決する必要があります。スクリプトの実行エラーが発生したリモート・データベースは、解決の操作が完了するまでスクリプトを実行することはできません。

失敗した SQL パススルー・スクリプトを修正するには、`ml_add_passthrough_repair` プロシージャを使用します。`ml_add_passthrough_repair` プロシージャを使用して、特定のスクリプトの実行中にスクリプト・エラー・コードが発生した場合の対処方法を定義します。SQL パススルー修正情報は `ml_passthrough_repair` テーブルに格納されます。リモート・データベースでエラーを修正するための SQL パススルー・スクリプトは、`ml_passthrough_script` テーブルに格納されます。

`ml_add_passthrough_repair` プロシージャを呼び出すときに、失敗したスクリプト名、エラーコード、新しいスクリプト名、および対処方法のパラメータを指定する必要があります。

リモート・データベースで失敗したスクリプト名とエラー・コードが報告された場合、次の同期セッションの際に、そのスクリプト名とエラー・コードが `ml_passthrough_repair` テーブルのエントリと比較されます。リモート・データベースで報告された失敗したスクリプト名とエラー・コードが `ml_passthrough_repair` テーブルのエントリと一致した場合、テーブルに格納されている対処方法と新しいスクリプト名の値から、リモート・データベースのエラーの対処方法が決まります。

対処方法として、以下のいずれかを定義することができます。

R - (Replace :置換) 失敗したスクリプトを、新しいスクリプト名パラメータで指定されている新しいスクリプトと置き換えます。リモート・データベースで新しいスクリプトが実行されます。失敗したスクリプトを再実行するには、新しいスクリプト名に失敗したスクリプト名と同じ名前を設定します。

P - (Purge :ページ) リモート・データベースは、受信したすべてのスクリプトを破棄し、すべてのスクリプトのページ完了後に、スクリプトの実行を通常どおりに続行します。

S - (Skip :スキップ) リモート・データベースがエラーを無視し、失敗したスクリプトが成功したかのように、スクリプトの実行を続行します。

H - (Halt :停止) リモート・データベースは、命令を受け取るまでそれ以上のスクリプトを実行しません。

対処方法が R 以外の場合、新しいスクリプト名に NULL 値を設定する必要があります。対処方法が R の場合は、`ml_passthrough_script` テーブルにすでに存在する新しいスクリプト名を指定する必要があります。`ml_passthrough_repair` テーブルによって生成される対処方法は、エラーがアップロードされる同期セッションかまたはその次の同期セッションにおいてリモート・データベースにダウンロードされます。

統合データベースにおけるスクリプト管理

SQL パススルーは、スクリプトの管理と実行を行うための追加機能を提供します。これらの機能について、以下で説明します。

SQL パススルーのための Mobile Link サーバ冗長性

冗長ログギングをコンソール・ウィンドウに出力するかまたは `-o` オプションを使用して直接出力するように、Mobile Link サーバを設定することができます。この場合、Mobile Link サーバの起動時に、以下のコマンド・ライン・オプションを設定します。

```
mlsrv11 -sl java ( -DMLVerbosePassthrough=1 ) ...
```

ただし、このオプションを使用すると Mobile Link サーバのパフォーマンスが低下するため、運用環境では使用しないでください。

SQL パススルー・スクリプトの削除

`ml_delete_passthrough_script` プロシージャは、`ml_passthrough_script` テーブルから SQL パススルー・スクリプトを削除します。`ml_delete_passthrough_script` プロシージャを呼び出すときに、スクリプト名パラメータを指定する必要があります。スクリプト名パラメータは、`ml_passthrough_script` テーブルから削除するスクリプトを指定します。スクリプトが `ml_passthrough` テーブルまたは `ml_passthrough_repair` テーブルのローから参照されない場合にのみ、スクリプトが削除されます。

SQL パススルー・エントリの削除

`ml_delete_passthrough` プロシージャは、`ml_passthrough` テーブルから SQL パススルー・エントリを削除します。`ml_delete_passthrough` プロシージャを呼び出すときに、リモート ID、スクリプト名、および実行順序のパラメータを指定する必要があります。リモート ID は、どのリモート・データベースから `ml_passthrough` エントリを削除するかを指定します。リモート ID に NULL を指定した場合、対象スクリプト名のすべてのエントリが `ml_passthrough` テーブルから削除されます。スクリプト名は、`ml_passthrough` テーブルから削除されるスクリプトを指定します。このスクリプト名は、NULL 以外でなければなりません。実行順序が NULL の場合、実行順序にかかわらず、対応するリモート ID およびスクリプト名の `ml_passthrough` テーブルのすべてのローが削除されます。スクリプトの `ml_passthrough` エントリを削除しても、すでにそのスクリプトを受信したリモート・データベースからスクリプトが削除されることはありません。すでにスクリプトを受信したリモート・データベースは、`ml_passthrough` エントリが削除されているかどうかにかかわらず、そのスクリプトを実行します。

SQL パススルー修正情報の削除

`ml_delete_passthrough_repair` プロシージャは、`ml_passthrough_repair` テーブルから SQL パススルー修正情報を削除します。`ml_delete_passthrough_repair` プロシージャを呼び出すときに、失敗したスクリプト名およびエラー・コードのパラメータを指定する必要があります。失敗したスクリプト名は、`ml_passthrough_repair` テーブルから削除するスクリプト名であり、NULL を指定することはできません。エラー・コードが NULL の場合、対応する失敗したスクリプト名の `ml_passthrough_repair` テーブルのすべてのローが削除されます。

デモ - デモ・データベースを空のリモート・データベースに移動する

この SQL パススルーのデモンストレーションでは、SQL Anywhere 11 に付属しているデモ・データベースを使用します。このデモンストレーションでは、Anywhere 11 のインストール先の `%SQLANY11%\Scripts` ディレクトリに格納されている `mkdemo.sql` ファイルを呼び出すことにより、統合データベースとして使用される新しいバ

ージョンのデモ・データベースを作成します。

リモート・データベースは、同期で使用される単一の Temp_Table テーブルを持つシェルとして作成されます。パブリケーション pub1 には Temp_Table テーブルがあり、Mobile Link ユーザ rem100 のサブスクリプションがパブリケーション pub1 に対して作成されています。最初の同期時に、SQL パススルー・スクリプトがリモート・データベースにダウンロードされます。スクリプトが自動的に実行され、デモ・データベースの 4 つのテーブル (GROUPO.Customers、GROUPO.Products、GROUPO.SalesOrderItems、GROUPO.SalesOrders) の同期が設定されます。これらの 4 つのテーブルは、適切なユーザおよびドメイン定義を使用して、リモート・データベースに作成されます。また、これらの 4 つのテーブルを含む新しいパブリケーション pub2 が作成され、ユーザ rem100 のサブスクリプションが pub2 に対して作成されます。

上記の 4 つのテーブルについて、統合データベースで Mobile Link 同期が構成されます。4 つのテーブルには、タイムスタンプ・ベースの同期を実行するために、デフォルトの現在のタイムスタンプが格納される last_modified カラムが追加されます。各テーブルに download_cursor、download_delete_cursor、upload_insert、upload_update、および upload_delete テーブル同期スクリプトがあります。便宜上、ここでは同期スクリプトに競合解決は定義されていません。

パブリケーション pub2 の最初の同期時に SQL パススルー・スクリプトが実行され、シェルの Temp_Table テーブルおよびパブリケーション pub1 のすべての定義がクリーン・アップされます。これにより、パブリケーション pub2 のみがリモート・データベースに残り、完全に機能する同期システムが配備されます。

デモ用のファイルはすべて以下の iAnywhere Web サイトで入手することができます。

[・デモ・データベースを空のリモート・データベースに移動する](#)

[・SQL パススルー・スクリプトの実行中に発生したエラーを処理する](#)

SQL Anywhere 統合データベースの作成

SQL Anywhere 統合データベースを作成するには、

¥SQLPassThrough_EmptyRemote¥cons ディレクトリ内の setup.bat ファイルを実行します。setup.bat ファイルを実行すると、以下の処理が行われます。

- SQL Anywhere 統合データベース demo.db が初期化されます。
- demo データベース用のユーザ DSN demo_conswith 接続パラメータが作成されます。
- データベース・サーバで demo データベースが起動されます。
- demo データベースに対して syncsa.sql スクリプトが実行されます。
- demo データベースに対して mkdemo.sql スクリプトが実行されます。
- 統合データベース・スキーマが demo データベースに読み込まれます。
この統合データベース・スキーマは、
¥SQLPassThrough_EmptyRemote¥cons ディレクトリ内の
cons_schema.sql および modify_cons_schema.sql ファイルに配置されています。
- Mobile Link サーバが起動され、統合データベースに接続されます。

mkdemo.sql スクリプトは、%SQLANY11%\Scripts ディレクトリに配置されており、スクリプト内には大部分の統合データベース・スキーマが格納されています。cons_schema.sql および modify_cons_schema.sql スクリプトには、シェル・データベースの Mobile Link 同期に必要なデモ・データベース・スキーマに対する修正および追加が格納されています。

GROUPO.Customers、GROUPO.Products、GROUPO.SalesOrderItems、および GROUPO.SalesOrder の 4 つのテーブルを同期するために、SQL Anywhere 統合データベースを構成する必要があります。このため、各テーブルへの last_modified カラムの追加、統合データベースの削除ローを処理するためのシャドー・テーブルの作成、およびアップロードとダウンロードを処理するためのさまざまなテーブル・スクリプトの作成を行う必要があります。これらの機能は、modify_cons_schema.sql スクリプトにより実行されます。

SQL Anywhere リモート・データベースの作成

SQL Anywhere リモート・データベースを作成するには、¥SQLPassThrough_EmptyRemote¥rem ディレクトリ内の setup.bat ファイルを実行します。setup.bat ファイルを実行すると、以下の処理が行われます。

- SQL Anywhere リモート・データベース MLrem.db が初期化されます。
- MLrem データベース用のユーザ DSN demo_remwith 接続パラメータが作成されます。
- データベース・サーバで MLrem データベースが起動されます。
- ¥SQLPassThrough_EmptyRemote¥rem ディレクトリ内の rem_schema.sql ファイルが実行され、MLrem データベースにリモート・データベース・スキーマが作成されます。
- Mobile Link クライアントが起動され、Mobile Link サーバとの最初の同期が実行されます。

rem_schema.sql スクリプトには、シェル・データベースの作成および Mobile Link 同期の構成に必要な情報が格納されています。最初の同期で、SQL パススルーを使用して、デモ・データベースの Mobile Link 同期が構成されます。

ここで実行される最初の同期において、統合データベースの ml_database テーブルにリモート・データベースに関する情報が追加されます。この処理を行わないと、SQL パススルー・スクリプトを作成してこのリモート・データベースにおける実行をスケジュールすることはできません。

統合データベースにおける SQL パススルーの構成

SQL パススルー・ファイルには、create_users_and_domains、create_tables_and_schema、create_data_publication、create_data_subscription、および clean_up_remote_schema の 5 つの SQL パススルー・スクリプトが定義されます。SQL パススルー・スクリプトを作成するには、ml_add_passthrough_script プロシージャを呼び出します。

最初の 4 つの SQL パススルー・スクリプトは、リモート・データベース・ユーザの構成、リモート・データベース・テーブルの作成、パブリケーション pub2 の作成、および同期サブスクリプションの作成を行います。最初の 4 つの SQL パススルー・スクリプト用の関連パブリケーション・リストは NULL です。つまり、スクリプトが実行される前にパブリケーションを同期する必要はないということです。これらの SQL パススルー・スクリプトは、以下のように定義されます。

```
/*-----  
  
* Setup Passthrough Scripts  
*-----*/  
CALL ml_add_passthrough_script (  
  
    'create_users_and_domains',  
    null,  
    null,  
  
    ,  
  
    // *****  
    // Create groups, user IDs and grant user permissions  
    // *****  
  
    GRANT CONNECT, GROUP, RESOURCE TO "GROUP0";  
    ...  
    ',
```

```

,
This creates the users and the domains for the remote database schema
,
);

```

```

CALL ml_add_passthrough_script (
    'create_tables_and_schema',
    null,
    null,

```

```

,
// *****
// Create tables and grant permissions
// *****

```

```

CREATE TABLE GROUPO.Customers

```

```

(
    ID integer NOT NULL default autoincrement,
    Surname person_name_t NOT NULL,
    GivenName person_name_t NOT NULL,
    Street street_t NOT NULL,
    City city_t NOT NULL,
    State state_t NULL,
    Country country_t NULL,
    PostalCode postal_code_t NULL,
    Phone phone_number_t NOT NULL,
    CompanyName company_name_t NULL,
    CONSTRAINT CustomersKey PRIMARY KEY (ID)

```

```

);
...
,
,

```

This creates the tables on the remote database

```

,
);

```

```

CALL ml_add_passthrough_script (
    'create_data_publication',
    null,
    null,

```

```

,
CREATE PUBLICATION GROUPO.pub2 (
    TABLE GROUPO.Customers,
    TABLE GROUPO.SalesOrders,
    TABLE GROUPO.SalesOrderItems,
    TABLE GROUPO.Products);

```

```

,

```

```

,
This is to create publication pub2 with the 4 tables to be
synchronized
,
);

CALL ml_add_passthrough_script (
    'create_data_subscription',
    null,
    null,
,
CREATE SYNCHRONIZATION SUBSCRIPTION TO GROUPO.pub2 FOR rem100

    OPTIONS ScriptVersion='demo_v2';
,
,
This is to create synchronization subscription to pub2 for the remote
user
,
);

```

最後の SQL パススルー・スクリプトは、Temp_Table テーブルとパブリケーション pub1 を削除することにより、シェル・データベースをクリーン・アップします。このスクリプトの実行後には、パブリケーション pub2 のみが残り、リモート・データベースに Mobile Link 同期が構成されます。最後の SQL パススルー・スクリプト用の関連パブリケーション・リストには pub2 が含まれています。このパススルー・スクリプトは、パブリケーション pub2 の同期時のみ実行されます。この SQL パススルー・スクリプトは、以下のように定義されます。

```

CALL ml_add_passthrough_script (
    'clean_up_remote_schema',
    null,
    'pub2',
,
DROP SYNCHRONIZATION SUBSCRIPTION TO DBA.pub1 FOR rem100;
DROP PUBLICATION DBA.pub1;
DROP TABLE DBA.Temp_Table;
,
,
This cleans up the initial database schema in the remote
,
);

```

ml_add_passthrough プロシージャを使用して、SQL パススルー・スクリプトに、そのスクリプトがダウンロードされて実行されるリモート・データベースを関連付けます。以下のように、5 つのすべての SQL パススルー・スクリプトを、すべてのリモート・データベースで実行されるように構成して関連付けます。

```

/*-----
* Add the Passthrough Scripts
*-----*/
CALL ml_add_passthrough (

```

```

        null,
        'create_users_and_domains',
        10
    );

CALL ml_add_passthrough (
    null,
    'create_tables_and_schema',
    20
);

CALL ml_add_passthrough (
    null,
    'create_data_publication',
    30
);

CALL ml_add_passthrough (
    null,
    'create_data_subscription',
    40
);

CALL ml_add_passthrough (
    null,
    'clean_up_remote_schema',
    50
);

```

この構成を行うには、¥SQLPassThrough_EmptyRemote¥cons ディレクトリ内の passthrough.bat ファイルを実行します。

パフォーマンスを改善するために、Mobile Link サーバは、スクリプトが初めて ml_passthrough_script テーブルに追加されるまでは SQL パススルー・スクリプトの処理を有効にしません。Mobile Link サーバは、スクリプトが追加されるのを待ちながら、一定の間隔でこのテーブルを照会します。Mobile Link コンソール・ウィンドウに以下のメッセージが表示されたら、SQL パススルー・スクリプトが Mobile Link サーバによって読み込まれ、リモート・データベースにダウンロードされる準備が整っていることを意味します。

```

I. 2008-03-05 15:26:01. <Main> Server passthrough support has been
enabled.

```

統合データベースで SQL パススルー・スクリプトの作成を確認する

SQL Anywhere 統合データベースでは、ml_passthrough_script テーブルで SQL パススルー・スクリプトを確認することができます。以下の図では、Interactive SQL に ml_passthrough_script テーブルが表示されています。

SQL Statements				
1	SELECT * FROM "DBA"."ml_passthrough_script"			
2				
3				

Results				
script_id	script_name	flags	affected_pubs	script
1	create_users_and_domains	(NULL)	(NULL)	// %%
2	create_tables_and_schema	(NULL)	(NULL)	// %%
3	create_data_publication	(NULL)	(NULL)	CREATE PUBLICATION GROUPO_pub2 (TABLE GROUPO_Customers, TABLE GROUPO_SalesOrders, TABLE GROUPO_S
4	create_data_subscription	(NULL)	(NULL)	CREATE SYNCHRONIZATION SUBSCRIPTION TO GROUPO_pub2 FOR rem100 OPTIONS ScriptVersion='demo_v2';
5	clean_up_remote_schema	(NULL)	pub2	DROP SYNCHRONIZATION SUBSCRIPTION TO DBA_pub1 FOR rem100; DROP PUBLICATION DBA_pub1; DROP TABLE

図 1 - ml_passthrough_script テーブル内の SQL パススルー・スクリプト

また、SQL パススルー・エントリは、ml_passthrough テーブルで確認することができます。以下の図では、Interactive SQL に ml_passthrough テーブルが表示されています。

SQL Statements				
1	SELECT * FROM "DBA"."ml_passthrough"			
2				
3				

Results				
	remote_id	run_order	script_id	last_modified
1	rem100	10	1	2008-03-12 13:09:41.103
2	rem100	20	2	2008-03-12 13:09:41.134
3	rem100	30	3	2008-03-12 13:09:41.134
4	rem100	40	4	2008-03-12 13:09:41.134
5	rem100	50	5	2008-03-12 13:09:41.134

図 2 - ml_passthrough テーブル内の SQL パススルー・エントリ

リモート・データベースの同期

統合データベースに SQL パススルーが構成されている場合、そのスクリプトをダウンロードして実行するために、リモート・データベースを同期する必要があります。リモート・データベースを同期するには、¥SQLPassThrough_EmptyRemote¥rem ディレクトリ内の sync_pub1.bat ファイルを実行します。この同期において、関連パブリケーション・リストが NULL であるすべての SQL パススルー・スクリプト (create_users_and_domains、create_tables_and_schema、create_data_publication、create_data_subscription) が、リモート・データベースで実行されます。関連パブリケーション・リストが NULL である SQL パススルー・スクリプトの実行後には、新しいパブリケーション pub2 がリモート・データベースに存在します。SQL パススルー・スクリプトの実行状況を判断するには、リモート・データベースの dbo.sync_passthrough_status テーブルを確認します。この情報は、次回の同期時に統合データベースに戻されます。

SQL Statements						
1	SELECT * FROM "dbo"."sync_passthrough_status"					
2						
3						

Results						
	run_order	script_id	script_status	error_code	error_text	remote_run_time
1	10	15		0		2008-03-12 13:14:04.030
2	20	25		0		2008-03-12 13:14:04.092
3	30	35		0		2008-03-12 13:14:04.827
4	40	45		0		2008-03-12 13:14:04.842

図 3 - リモート・データベースにおける SQL パススルーの状況

dbo.sync_passthrough_script テーブルには、リモート・データベースで実行されたスクリプトが格納されます。dbo.sync_passthrough_progress テーブルには、最後に実行が成功したスクリプトの実行順序、最後に試行されたスクリプトの実行順序、および（統合データベースにおける）スクリプトが最後にダウンロードされた時間が格納されます。

SQL Statements						
1	SELECT * FROM "dbo"."sync_passthrough_script"					
2						
3						

Results						
	run_order	script_id	script_name	flags	affected_pubs	script
1	10	1	create_users_and_domains	0 (NULL)		// %%
2	20	2	create_tables_and_schema	0 (NULL)		// %%
3	30	3	create_data_publication	0 (NULL)		CREATE PUBLICATION GROUP pub2 (TABLE GROUP Customers, TABLE GROUP SalesOrders, TABLE GROUP ...)
4	40	4	create_data_subscription	0 (NULL)		CREATE SYNCHRONIZATION SUBSCRIPTION TO GROUP pub2 FOR rem100 OPTIONS ScriptVersion='demo_v2';
5	50	5	clean_up_remote_schema	0 (NULL)		DROP SYNCHRONIZATION SUBSCRIPTION TO DBA.pub1 FOR rem100; DROP PUBLICATION DBA.pub1; DROP TABLE ...

図 4 - リモート・データベースの SQL パススルー・スクリプト

SQL Statements				
1	SELECT * FROM "dbo"."sync_passthrough_progress"			
2				
3				

Results				
	progress_id	last_download_time	last_script_applied	last_script_attempted
1	0	2008-03-12 13:09:41.134	40	40

図 5 - リモート・データベースの SQL パススルーの進行状況

パブリケーション pub2 の最初の同期が実行されると、最後の SQL パススルー・スクリプトである clean_up_remote_schema がリモート・データベースに対して実行されます。パブリケーション pub2 を使用してリモート・データベースを同期するには、¥SQLPassThrough_EmptyRemote¥rem ディレクトリ内の sync_pub2.bat ファイルを実行します。パブリケーション pub1 に関連するスキーマ全体がクリーン・アップされて削除され、リモート・データベースにパブリケーション pub2 ののみが残ります。

SQL パススルー・スクリプトの実行を確認する

sync_pub2.bat を実行してパブリケーション pub2 の同期をもう一度実行すると、すべての SQL パススルー結果が統合データベースにアップロードされます。統合データベースの ml_passthrough_status テーブルに、実

行されたすべての SQL パススルー・スクリプトの状態と、スクリプトが実行されたリモート・データベースが報告されます。

SQL Statements								
1	SELECT * FROM "DBA"."ml_passthrough_status"							
2								
3								
Results								
	status_id	remote_id	run_order	script_id	script_status	error_code	error_text	remote_run_time
1	1	rem100	10	1	S	0		2008-03-12 13:14:04.030
2	2	rem100	20	2	S	0		2008-03-12 13:14:04.092
3	3	rem100	30	3	S	0		2008-03-12 13:14:04.827
4	4	rem100	40	4	S	0		2008-03-12 13:14:04.842
5	5	rem100	50	5	S	0		2008-03-12 13:20:43.089

図 6 - SQL パススルー・スクリプトの結果

SQL パススルー・スクリプトの実行中にエラーが発生した場合、ml_passthrough_status テーブルに script_status と error_code が表示されます。

デモの実行手順

以下の手順により、指定されたバッチ・ファイルを使用してデモを実行します。

1. %SQLPassThrough_EmptyRemote%cons%setup.bat
2. %SQLPassThrough_EmptyRemote%rem%setup.bat
3. %SQLPassThrough_EmptyRemote%cons%passthrough.bat

SQL パススルー・スクリプトが Mobile Link サーバによって読み込まれたことを示す以下のメッセージが Mobile Link コンソール・ウィンドウに表示されるまで待ちます。

```
I. 2008-03-05 15:26:01. <Main> Server passthrough support has been enabled.
```

4. %SQLPassThrough_EmptyRemote%rem%sync_pub1.bat
5. %SQLPassThrough_EmptyRemote%rem%sync_pub2.bat
6. %SQLPassThrough_EmptyRemote%rem%sync_pub2.bat

デモ - SQL パススルー・スクリプトの実行中に発生したエラーを処理する

この SQL パススルーのデモンストレーションでは、2 つのリモート・データベースと、単純なスキーマ定義を持つ統合データベースを使用します。このデモンストレーションでは、SQL パススルー修正機能を使用して、リモート・データベースで実行に失敗したスクリプトを修正する方法を示します。SQL パススルー・スクリプトは、2 つのデータベースで create view 文を実行します。ただし、一方のデータベースには、同じ名前のビューがすでに定義されています。ここでは、SQL パススルー修正機能を使用して、すでにビューが定義されているリモート・データベースのビューを変更します。

SQL Anywhere 統合データベースの作成

SQL Anywhere 統合データベースを作成するには、%SQLPassThrough_View_Error%cons ディレクトリ内の setup.bat ファイルを実行します。setup.bat ファイルを実行すると、以下の処理が行われます。

- ・ SQL Anywhere 統合データベース MLcons.db が初期化されます。
- ・ MLcons データベース用のユーザ DSN viewerror_conswith 接続パラメータが作成されます。
- ・ データベース・サーバで MLcons データベースが起動されます。

- ・ デモ・データベースに対して syncsa.sql スクリプトが実行されます。
- ・ 統合データベース・スキーマが MLcons データベースに読み込まれます。この統合データベース・スキーマは、¥SQLPassThrough_View_Error¥cons ディレクトリ内の cons_schema.sql ファイルに配置されています。
- ・ MLcons データベースにテスト・データが挿入されます。
- ・ Mobile Link サーバが起動され、統合データベースに接続されます。

con_schema.sql スクリプトは、Table1 および Table2 の 2 つのテーブルと、Mobile Link 同期に必要なすべての同期スクリプトを作成します。

SQL Anywhere リモート・データベースの作成

このデモンストレーションでは、MLrem.db と MLrem2.db の 2 つのリモート・データベースを使用します。それぞれのリモート・データベースには、Table1 と Table2 の 2 つのテーブルが格納されています。

リモート・データベース 1

リモート・データベース 1 を作成するには、¥SQLPassThrough_View_Error¥rem ディレクトリ内の setup.bat ファイルを実行します。setup.bat ファイルを実行すると、以下の処理が行われます。

- ・ SQL Anywhere リモート・データベース MLrem.db が初期化されます。
- ・ MLrem データベース用のユーザ DSN viewerror_remwith 接続パラメータが作成されます。
- ・ データベース・サーバで MLrem データベースが起動されます。
- ・ リモート・データベース・スキーマが MLrem データベースに読み込まれます。このリモート・データベース・スキーマは、¥SQLPassThrough_View_Error¥rem ディレクトリ内の rem_schema.sql ファイルに配置されています。rem_schema.sql は、以下の構文を使用して、Table1 と Table2 の間にビューを作成します。

```
CREATE VIEW DBA.v_TableData
AS SELECT Table1.Table1_id Table1ID, Table2.Table2_id Table2ID
FROM DBA.Table1, DBA.Table2
WHERE Table1.Table1_id = Table2.Table2_id;
```

- ・ MLrem データベースにテスト・データが挿入されます。
- ・ Mobile Link クライアントが起動され、Mobile Link サーバとの最初の同期が実行されます。SQL パススルー修正機能のデモンストレーション用に、リモート・データベース 1 にビューが作成されます。

リモート・データベース 2

リモート・データベース 2 を作成するには、¥SQLPassThrough_View_Error¥rem2 ディレクトリ内の setup.bat ファイルを実行します。setup.bat ファイルを実行すると、以下の処理が行われます。

- ・ SQL Anywhere リモート・データベース MLrem2.db が初期化されます。
- ・ MLrem2 データベース用のユーザ DSN viewerror_rem2with 接続パラメータが作成されます。
- ・ データベース・サーバで MLrem2 データベースが起動されます。
- ・ リモート・データベース・スキーマが MLrem2 データベースに読み込まれます。このリモート・データベース・スキーマは、¥SQLPassThrough_View_Error¥rem2 ディレクトリ内の rem2_schema.sql ファイルに配置されています。
- ・ MLrem2 データベースにテスト・データが挿入されます。
- ・ Mobile Link クライアントが起動され、Mobile Link サーバとの最初の同期が実行されます。リモート・データベース 2 にはビューは作成されません。ビューは、SQL パススルー・スクリプトによって作成されます。両方のデータベースの最初の同期において、統合データベースの ml_database テーブルにリモート・データベースに関する情報が格納されます。この処理を行わないと、SQL パススルー・スクリプトを作成し

てこのリモート・データベースにおける実行をスケジュールすることはできません。

統合データベースにおける SQL パススルーの構成

SQL パススルー・スクリプトを作成するには、`¥SQLPassThrough_View_Error¥cons` ディレクトリ内の `passthrough.bat` スクリプトを実行します。`passthrough.bat` ファイルは `passthrough.sql` スクリプトを実行します。SQL パススルー・スクリプトは、以下のように定義されます。

```
/*-----  
  
* Setup Passthrough Scripts  
*-----*/  
CALL ml_add_passthrough_script (  
  
    'create_view',  
    'automatic',  
    'pub1',  
    '  
CREATE VIEW DBA.v_TableData  
AS SELECT Table1.Table1_id  
TableID, Table1.Data Data1,  
Table2.Data Data2  
        FROM DBA.Table1, DBA.Table2  
        WHERE Table1.Table1_id = Table2.Table2_id;  
    '  
    '  
Create a view TableData on the remote databases  
    '  
);  
COMMIT;  
/*-----  
* Add the Passthrough Scripts  
*-----*/  
CALL ml_add_passthrough (  
    null,  
    'create_view',  
    10  
);  
COMMIT;
```

リモート・データベースの同期

リモート・データベースを同期するには、当該リモート・データベースのディレクトリに配置されている `sync.bat` ファイルを実行します。リモート・データベース 1 で SQL パススルー・スクリプトの実行に失敗した場合、データベースの同期も失敗します。ただし、スクリプト・エラーを解決するための手動操作を行うまでは、リモート・データベース 1 用の SQL パススルー・スクリプトを実行しないかぎり、その後の同期は成功します (これは正しいか、または「成功しません」と言うべきか ?)。リモート・データベース 1 における同期の失敗は、以下のように表示されます。

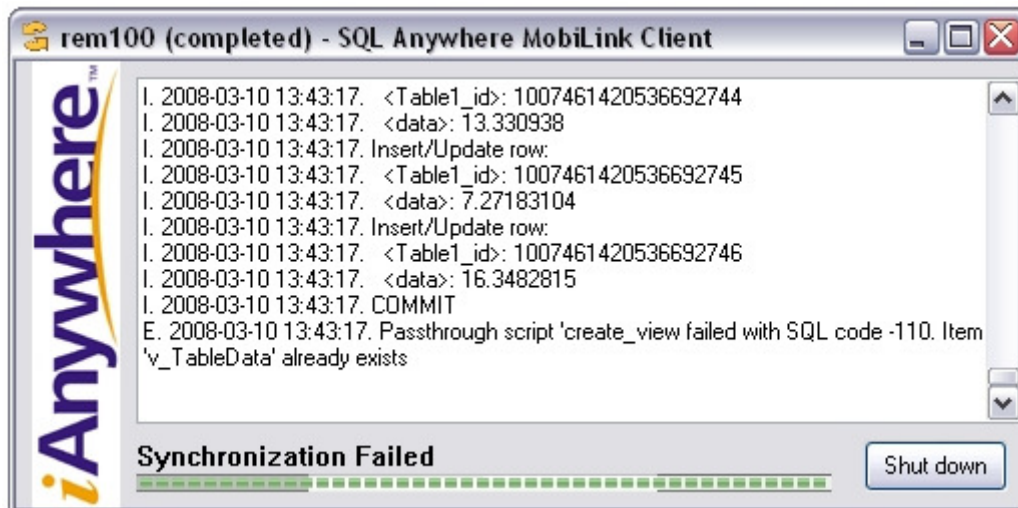


図 7 - スクリプトの失敗による Mobile Link クライアントのエラー

リモート・データベース 1 で SQL パススルー・スクリプトが実行されると、dbo.sync_passthrough_status テーブルが更新されて実行状況が表示されます。

SQL Statements						
1	SELECT * FROM "dbo"."sync_passthrough_status"					
2						
3						

Results						
run_order	script_id	script_status	error_code	error_text	remote_run_time	
1	10	1E	-110	Item 'v_TableData' already exists	2008-03-12 13:37:04.327	

図 8 - リモート・データベース 1 における SQL パススルーの状況

各リモート・データベースの次回の同期時に、SQL パススルーの状況の結果が統合データベースにアップロードされます。この情報は、Mobile Link テーブル ml_passthrough_status に格納されます。

SQL Statements							
1	SELECT * FROM "DBA"."ml_passthrough_status"						
2							
3							

Results							
status_id	remote_id	run_order	script_id	script_status	error_code	error_text	remote_run_time
1	1 rem100	10	1E		-110	Item 'v_TableData' already exists	2008-03-12 13:37:04.327
2	2 rem200	10	1S		0		2008-03-12 13:37:07.248

図 9 - 統合データベースにおける SQL パススルーの状況

この時点では、スクリプト・エラーを解決するための手動操作を行うまでは、リモート・データベース 1 で SQL パススルー・スクリプトを実行することはできません。

統合データベースにおける SQL パススルー修正スクリプトの構成

リモート・データベース 1 で SQL パススルー・スクリプトの実行に失敗した場合、手動操作を行わないと、それ以上スクリプトを実行することはできません。スクリプト・エラーを解決するには、SQL パススルー修正スクリプトを作成し、リモート・データベース 1 でそれを実行します。エラーを無視する方法もありますが、このデモンストレ

ーションではスクリプトを修正する方法を採用します。

図 9 では、ml_passthrough_status テーブルの error_code と error_text に以下のようにエラーが示されています。

```
-110  
Item 'v_TableData' already exists
```

このエラーは、ビュー v_TableData がリモート・データベース 1 にすでに存在しているために発生したものです。この SQL パススルー・エラーの最も簡単な解決方法として、このビューを目的のビューに変更するスクリプトを新たに作成します。

新しい SQL パススルー・スクリプトは、以下のように定義されます。

```
/*-----  
* Setup Passthrough Repair Scripts  
*-----*/  
CALL ml_add_passthrough_script (  
    'alter_view',  
    'automatic',  
    'pub1',  
    '  
ALTER VIEW DBA.v_TableData  
AS SELECT Table1.Table1_id  
TableID, Table1.Data Data1,  
  
Table2.Data Data2  
FROM DBA.Table1, DBA.Table2  
WHERE Table1.Table1_id = Table2.Table2_id;  
,  
,  
Create a view TableData on the remote databases  
,  
);  
COMMIT;
```

次に、ml_add_passthrough_repair プロシージャを呼び出します。ml_add_passthrough_repair プロシージャは、create_view スクリプトを実行したときにエラー・コード 110 の受信が報告されたすべてのリモート・サーバに alter_view スクリプトをダウンロードして実行するように Mobile Link サーバに命令します。ml_add_passthrough_repair プロシージャは、以下のように呼び出されます。

```
/*-----  
* Add the Passthrough Scripts  
*-----*/  
CALL ml_add_passthrough_repair (  
    'create_view',  
    -110,  
    'alter_view',  
    'R'  
);  
COMMIT;
```

リモート・データベース 1 の次回の同期時に、ALTER VIEW スクリプトがダウンロードされて実行されます。このスクリプトが実行されると、dbo.sync_passthrough_status テーブルが更新されて実行状況が表示されます。

SQL Statements						
1	SELECT * FROM "dbo"."sync_passthrough_status"					
2						
3						

Results						
	run_order	script_id	script_status	error_code	error_text	remote_run_time
1	10	25		0		2008-03-12 13:41:59.344

図 10 - 修正スクリプト実行後の リモート・データベース 1 における SQL パススルー・ステータス

リモート・データベース 1 をもう一度同期すると、SQL パススルーの状況の結果が統合データベースにアップロードされます。この情報は、Mobile Link テーブル ml_passthrough_status に格納されます。

SQL Statements							
1	SELECT * FROM "DBA"."ml_passthrough_status"						
2							
3							

Results								
	status_id	remote_id	run_order	script_id	script_status	error_code	error_text	remote_run_time
1	1	rem100	10	1E		-110	Item 'v_TableData' already exists	2008-03-12 13:37:04.327
2	2	rem200	10	15		0		2008-03-12 13:37:07.248
3	3	rem100	10	25		0		2008-03-12 13:41:59.344

図 11 - 修正スクリプト実行後の 統合データベースにおける SQL パススルー・ステータス

SQL パススルー・スクリプトを使用して両方のリモート・データベースに正しくビューが作成されました。リモート・データベース 1 では、以降の SQL パススルー・スクリプトを通常どおりに実行できるようになりました。

デモの実行手順

以下の手順により、指定されたバッチ・ファイルを使用してデモを実行します。

1. %SQLPassThrough_View_Error%cons%setup.bat
2. %SQLPassThrough_View_Error%rem%setup.bat
3. %SQLPassThrough_View_Error%rem2%setup.bat
4. %SQLPassThrough_View_Error%cons%passthrough.bat

SQL パススルー・スクリプトが Mobile Link サーバによって読み込まれたことを示す以下のメッセージが Mobile Link コンソール・ウィンドウに表示されるまで待ちます。

I. 2008-03-05 15:26:01. <Main> Server passthrough support has been enabled.

5. %SQLPassThrough_View_Error%rem%sync.bat
6. %SQLPassThrough_View_Error%rem2%sync.bat
7. %SQLPassThrough_View_Error%rem%sync.bat
8. %SQLPassThrough_View_Error%rem2%sync.bat
9. %SQLPassThrough_View_Error%cons%repair_passthrough.bat
10. %SQLPassThrough_View_Error%rem%sync.bat
11. %SQLPassThrough_View_Error%rem2%sync.bat
12. %SQLPassThrough_View_Error%rem%sync.bat
13. %SQLPassThrough_View_Error%rem2%sync.bat

まとめ

このマニュアルでは、SQL Anywhere 11.0.0 の新しい SQL パススルー機能について説明し、新機能の使用方法を示す 2 つのデモンストレーションを紹介しました。SQL パススルー機能は、Mobile Link 同期環境における非常に強力かつ堅牢な管理ツールです。

法的注意

Copyright (C) 2008 iAnywhere Solutions, Inc. All rights reserved.

iAnywhere Solutions、iAnywhere Solutions（ロゴ）は、iAnywhere Solutions, Inc.とその系列会社の商標です。その他の商標はすべて各社に帰属します。

本書に記載された情報、助言、推奨、ソフトウェア、文書、データ、サービス、ロゴ、商標、図版、テキスト、写真、およびその他の資料（これらすべてを"資料"と総称する）は、iAnywhere Solutions, Inc.とその提供元に帰属し、著作権や商標の法律および国際条約によって保護されています。また、これらの資料はいずれも、iAnywhere Solutionsとその提供元の知的所有権の対象となるものであり、iAnywhere Solutionsとその提供元がこれらの権利のすべてを保有するものとします。

資料のいかなる部分も、iAnywhere Solutionの知的所有権のライセンスを付与したり、既存のライセンス契約に修正を加えることを認めるものではないものとします。

資料は無保証で提供されるものであり、いかなる保証も行われません。iAnywhere Solutionsは、資料に関するすべての陳述と保証を明示的に拒否します。これには、商業性、特定の目的への整合性、非侵害性の黙示的な保証を無制限に含みます。

iAnywhere Solutionsは、資料自体の、または資料が依拠していると思われる内容、結果、正確性、適時性、完全性に関して、いかなる理由であろうと保証や陳述を行いません。iAnywhere Solutionsは、資料が途切れていないこと、誤りがいないこと、いかなる欠陥も修正されていることに関して保証や陳述を行いません。ここでは、「iAnywhere Solutions」とは、iAnywhere Solutions, Inc.またはSybase, Inc.とその部門、子会社、継承者、および親会社と、その従業員、パートナー、社長、代理人、および代表者と、さらに資料を提供した第三者の情報元や提供者を表します。