



INFORMATION ANYWHERE

iAnywhere
A SYBASE COMPANY

Technical Guide

Adaptive Server Anywhere 9への MySQLデータベースの移行



目次

はじめに	3
MySQL と Adaptive Server Anywhere の違い	4
データ形	5
MySQL 関数の Adaptive Server Anywhere へのマッピング	8
文字列関数	9
数値関数 11	12
日付および時刻関数	12
構文のマッピング	13
移行に関する他の問題	17
Adaptive Server Anywhere データベースへの MySQL データベースの移行	18
Adaptive Server Anywhere データベースの作成	19
MySQL データベースにデータソースを作成	21
MySQL データベースの Adaptive Server Anywhere への移行	25
MySQL データベースの Adaptive Server Anywhere への移行	26
新しい Adaptive Server Anywhere データベースの調整	39
MySQL から Adaptive Server Anywhere へのアプリケーションの移行	40
MySQL から Adaptive Server Anywhere への Perl アプリケーションの移行	41
MySQL から Adaptive Server Anywhere への PHP アプリケーションの移行	42
機能のマッピング	42
PHP 統合時の注意点	44
法的注意	45

はじめに

ソフトウェアバージョン

本書の内容は、SQL Anywhere Studioのバージョン9.0.1以降とMySQLのバージョン4.0以降を対象にしています。

Adaptive Server Anywhereには、MySQLやその他のRDBMからAdaptive Server Anywhereへスムーズに移行するための機能が組み込まれています。したがってMySQLからAdaptive Server Anywhereへのデータ移行は、データベースやアプリケーションにMySQL拡張を多用していないのであれば簡単です。

本書では、まず最初にAdaptive Server AnywhereとMySQLの違いについて、データ形式、フィーチャ、構文を含めて詳細に述べ、移行が困難なMySQLに固有の一部の機能に対処する方法を説明します。次に Sybase Centralのデータ統合ウィザードを使ってAdaptive MySQLデータベースからServer Anywhereデータベースへとデータを移行する方法を説明します。最後に、MySQLで起動する既存のアプリケーションをAdaptive Server Anywhereで起動するアプリケーションへと移行する方法を説明します。

MySQLとAdaptive Server Anywhereの違い

ここでは移行の際、問題となるMySQLとAdaptive Server Anywhereの違いについて述べ、また移行で生じる問題を解決するために有効と思われる対策を示します。MySQLにはないAdaptive Server Anywhereの機能によってコードを最適化する方法は多数ありますが、ここではいくつかの例を紹介します。

- サブクエリを使ってコードを簡略化することで、一時テーブルの使用を制限することができます。
MySQL 4.1がサブクエリをサポートするのは基本形式の場合に限られます。
- 行レベルロックがあるので、データベースに複数のユーザーがアクセスする場合でも更新時に全テーブルをロックする必要がなく、競合を減らすことができます。
- Adaptive Server Anywhereは外部キーをネイティブサポートしているので、MyIsamテーブルを使ってアプリケーションコードへ実装しなくてもサーバがjoinを実行できます。カスケード操作にも対応しています。
- ストアド・プロシージャやトリガによってアプリケーションコードを大幅に簡略化できるので、ビジネスロジックをアプリケーションログから独立して開発できます。
MySQLではストアド・プロシージャのサポートがバージョン5で追加されましたが、本書作成時点ではアルファ版で未対応です。
- ビューを使ったデータベースのスキーマや権限管理の簡素化と効率化が可能です。

SQL Anywhere Studioへの移行に先立ち、アイエニウェア・ソリューションズの[iAnywhere開発者リソース](#)で、Adaptive Server Anywhereに関連するドキュメント、サンプル、技術文書をご覧になることを強くお勧めします。

データ形

多くの場合、MySQLのデータ形式がそのままAdaptive Server Anywhereのデータ形になります。以下にいくつかの例を示します。

MySQLのデータ形	対応するAdaptive Server Anywhereのデータ形	備考
smallint unsigned smallint bool/Boolean double(m, n)	UNSIGNED SMALLINT TINYINT TINYINT or BIT DOUBLE(n)	MySQLの引数「m」はデータのフォーマットに関連しています。下記の注を参照してください。
float(m,n) unsigned	FLOAT(n)	MySQLにおける引数「m」はデータのフォーマットに関連しています。下記の注を参照してください。
fixed varchar(32) binary national char/varchar tiny/medium/long text	DECIMAL BINARY(32) CHAR/VARCHAR CHAR/VARCHAR/LONG VARCHAR	Adaptive Server Anywhereはテキスト値を保持するためのデータのみを格納するので、異なるBLOBでさらにBLOBサイズを指定する必要はありません。
tinyblob blob/mediumblob/longblob	BINARY(255) LONG BINARY	Adaptive Server Anywhereはテキスト値を保持するためのデータのみを格納するので、異なるBLOBでさらにBLOBサイズを指定する必要はありません。
year	DATE	

注：

データ形式自体に加えて、データ形式の宣言にも違いがあります。MySQLは数値形式に任意のパラメータを提供するので、整数値の表示桁数を指定できます（たとえint(4)は、<s>をスペースとすると、「1」を「<s><s><s><s>1」として返します）。形式の宣言時にオプションの「ZEROFILL」変更子を指定すると、前述の例ではスペースがゼロに置き換わります（「1」は「0001」として返されます）。形式の宣言時の表示形式とデータ値の移行は、Adaptive Server Anywhereではサポートされていません。cast関数やconvert関数、その他の各種ストリング操作関数によって、データベースから取得したデータ値のフォーマットが可能です。

以下のデータ形式は、構文上の違いというよりもさらに本質的な違いを示すものです。

MEDIUMINT

MEDIUMINTは3バイトの整数値です。希望の桁数に応じて、Adaptive Server Anywhereのinteger (4バイト) やsmallint (2バイト) を使用して容易にシミュレートできます。

Year

Yearは年度を表す2桁または4桁の値です。Adaptive Server Anywhereのデータ形DATEは年度値を保持するために使用できますが、より多くの保存スペースを必要とします。日付の計算と変換はAdaptive Server Anywhereの組み込み関数を使って行うことができます。詳細については「Adaptive Server Anywhere SQLリファレンス・マニュアル」の「日付と時刻データ型」をご覧ください。

「Adaptive Server Anywhere SQLリファレンス・マニュアル」はアイエニウェア・ソリューションズ株式会社の[ダウンロードページ](#)で入手できます。

以下のデータ形式は正確には一致せず、Adaptive Server Anywhereに移行するために作業が必要となります。

NCHAR / NVARCHAR

MySQLのバージョン4.1より前のバージョンでは、NCHARはCHARと同じなので、Adaptive Server Anywhereはこのデータ形式をそのままサポートできます。しかしMySQLバージョン4.1から、NCHAR値はUTF8文字セットでMySQLに格納されるようになりました。Adaptive Server AnywhereはUTF8を含めた各種の文字セットをサポートしています。データベースが適切に作成されている場合、国際文字コードの値を格納するための特別なデータ形を使う必要はありません。Adaptive Server Anywhereでサポートされている最新の国際文字コード系に関しては、「Adaptive Server Anywhere データベース管理ガイド」の「国際言語と文字セット」の章を参照してください。

「Adaptive Server Anywhereデータベース管理ガイド」はアイエニウェア・ソリューションズ株式会社の[ダウンロードページ](#)で入手できます。

ENUM

ENUM値は文字列オブジェクトであり、その値はテーブル生成時に列の定義に列挙された値から選択しなければなりません。値はENUMの定義中のインデックスポジションから挿入、検索することもできます。インデックス値「0」は空文字用に予約されています。ENUMデータ形は、Adaptive Server Anywhereでは、TINYINT列で示されます。MySQLのENUMと同じ動作をさせる方法は複数ありますが、ほぼ確実にクライアントアプリケーションに変更を加える必要が生じます。選択可能な方法は以下のとおりです。

クライアント側のアプリケーションを変更してENUM値を不要にする

ENUM値をクライアント側で変換する

ストアド・プロシージャやトリガ、計算列、ビュー、ENUM形用のマッピングテーブルなどを使い、MySQLによるENUM値の処理を模倣するようにサーバ側にロジックを追加する

たとえばENUM領域をもつテーブルにビューを生成して値を文字列として返す一方、通常のSELECTを使って値を数値として返します。ビューの使用例を以下に示します。

```
CREATE TABLE enumtbl( pkey INTEGER NOT NULL PRIMARY KEY, enumval TINYINT );

CREATE VIEW v_enumtable AS
  SELECT pkey,
         CASE enumval
           WHEN 0 THEN ''
           WHEN 1 THEN 'val1'
           WHEN 2 THEN 'val2'
           WHEN 3 THEN 'val3'
           ELSE NULL
         END
  FROM enumtbl;
```

クエリは以下のようになります。

```
SELECT pkey, enumval FROM v_enumtable;
```

あるいはenumtblからデータを検索するときは常にENUM文字列を含んだマッピングテーブルにjoinされるよう、ENUM値用にマッピングテーブルを生成することもできます。

```
CREATE TABLE enummap( enumval TINYINT NOT NULL PRIMARY KEY, enumstr
CHAR(16) );
```

クエリは以下のようになります。

```
SELECT pkey, enumstr FROM enumtbl, enummap
WHERE enumtbl.enumval = enummap.enumval;
```

ENUMのインデックス値を使う場合、テーブルへの挿入は直接行われます。それ以外の場合は、ストアード・プロシージャによりENUMを含んだテーブルへ列を挿入します。ストアード・プロシージャは、ENUM値をデコードするロジックを含むものとします。以下にAdaptive Server AnywhereでENUM列に相当する処理を行うストアード・プロシージャの実装例を示します（上記と同じテーブル定義を使用）。

```
CREATE PROCEDURE sp_insert_enumval( IN pkeyval int, IN enum CHAR(16) )
BEGIN
  DECLARE enum_map TINYINT;

  IF enum IS NOT NULL THEN
    CASE enum
      WHEN '' THEN SET enum_map = 0
      WHEN 'val1' THEN SET enum_map = 1
      WHEN 'val2' THEN SET enum_map = 2
      WHEN 'val3' THEN SET enum_map = 3
      ELSE SET enum_map = 0
    END CASE
  END IF;
```

```
INSERT INTO enumtbl VALUES( pkeyval, enum_map );  
END
```

SET

SET値は文字列オブジェクトであり、その値は列を定義する際に与えられるリストから選択しなければなりません。ENUM型とは異なり、有効な列の値を作り出すためにリストにある0あるいはそれ以外の値を組み合わせることができます。セットの各値にはバイナリ表現が与えられ、データは、設定された値の組み合わせにより割り当てられあるいは取り出されます。たとえば9という値を指定した場合、列にはSETから1番目と4番目の値が挿入されます。SETに存在する値（最大64）に応じて、tinyintからbigintまでのすべてについて、MySQLからAdaptive Server AnywhereにSET値をマッピングする必要があります。MySQLと同じ動作を得るためには、前述したENUMデータ型と同様の方法をとります。

MySQL関数のAdaptive Server Anywhereへのマッピング

MySQLとAdaptive Server Anywhereの関数には同じ名前が多くあります。異なる名前を持つMySQL関数でも、Adaptive Server Anywhereにはそれに相当する関数があります。MySQLにはAdaptive Server Anywhereに存在しない組み込み関数もいくつかありますが、これらの多くは同じ動作をするユーザ定義関数としてAdaptive Server Anywhereで作り出すことが可能です。このユーザ定義関数にAdaptive Server Anywhereデータベースと同じ名前をつければ、既存のクライアントアプリケーションのステートメントを修正する必要はありません。以下にMySQL組み込み関数と同じ動作をするAdaptive Server Anywhereユーザ定義関数を指定した例を示します。

```
CREATE FUNCTION FROM_UNIXTIME( IN fromdt bigint default 0,  
    IN fmt varchar(32) default 'Mmm dd, yyyy hh:mm:ss' )  
RETURNS datetime  
BEGIN  
    RETURN( dateformat( dateadd( second, fromdt, '1970/01/01 00:00:00' ),  
        fmt ) )  
END;  
  
CREATE FUNCTION SEC_TO_TIME( IN sec bigint default 0 )  
RETURNS time  
BEGIN  
    return( dateadd( second, sec, '1970/01/01 00:00:00' ) )  
END;
```

以下にMySQL関数とそれに相当するAdaptive Server Anywhere関数を示します。多くの関数を記載していますが、すべてを網羅しているわけではありません。これはAdaptive Server AnywhereとMySQLの関数のいずれもリリースにより変更されるためです。

ストリング関数

MySQL関数	Adaptive Server Anywhere 関数	備考
IFNULL(a, b)	IFNULL(a, b, a) or ISNULL(a, b)	Adaptive Server Anywhere の IFNULL関数は、MySQLのもの とは若干違う動作をします。
IF(cond, a, b)	IF cond THEN a ELSE b END IF	Adaptive Server Anywhereは、 埋め込み式と同様に処理ロジ ックでもセレクトリスト内の expressionとしてIFステート メントをサポートします。
CONCAT(a, b, ...)	STRING(a, b, ...)	MySQLでは「a, b, ...」のい ずれかがNULLの場合、リター ン値はNULLとなります。一方 Adaptive Server Anywhereでは 「a, b, ...」のいずれかがNULL の場合、連結用の空白文字列と して扱われます。
CONCAT_WS(sep, str1, str2, ...)	STRING(str1, sep, str2, sep, ...)	上記のコメント参照。
CONV(N, frombase, tobase)	INTOHEX(N), HEXTOINT(N)	Adaptive Server Anywhere の 関数は16進数への変換、または 16進数からの変換のみです。他 の変換は、UDFを使って手動で 行う必要があります。
HEX(arg)	INTOHEX(srg) if arg is numeric, HEXTOINT(arg) if arg is string	
CHAR(N,...)	CHAR(N)	Adaptive Server Anywhere CHAR()関数は、一つの引数の みサポートします。
STRCMP(expr1, expr2)	COMPARE(expr1, expr2)	
LENGTH(str)	BYTE_LENGTH(str)	Adaptive Server Anywhere の LENGTH()関数はstr内の文字列 を返します。バイト長である必 要はありません。
OCTET_LENGTH(str)	BYTE_LENGTH(str)	
CHARACTER_LENGTH(str)	CHAR_LENGTH(str)	
BIT_LENGTH(str)	BYTE_LENGTH(str) * 8	
LOCATE(substr, str[, pos])	LOCATE(str, substr[, pos])	引数の順番はAdaptive Server Anywhereとは異なります。

MySQL関数	Adaptive Server Anywhere 関数	備考
POSITION(substr IN str) INSTR(str, substr) SUBSTRING(str FROM pos[FOR len]) MID(str, pos, len) TRIM(str)	LOCATE(str, substr) LOCATE(str, substr) SUBSTRING(str, pos[, len]) SUBSTRING(str, pos, len) TRIM(str)	Adaptive Server Anywhereは他の形式のMySQL TRIM関数をサポートしていません。 trans_nameはAdaptive Server AnywhereとMySQLの文字セットで異なることがあります。詳細は「Adaptive Server Anywhere データベース管理ガイド」MySQLの関連ドキュメントを参照してください。
INSERT(str, pos, len, newstr) ELT(N, str1, str2, ...) CONVERT(expr USING trans_name)	STUFF(str, pos, len, newstr) ARGN(N, str1, str2, ...) CCONVERT(expr, trans_name)	

「Adaptive Server Anywhere **データベース管理ガイド**」はアイエニウェア・ソリューションズ株式会社の[ダウンロードページ](#)で入手できます。

数値関数

MySQL関数	Adaptive Server Anywhere 関数	備考
CEIL(x)	CEILING(x)	
ROUND(x)	ROUND(x, 0)	
x DIV y	FLOOR(x/y)	
LN(x)	LOG(x)	
LOG(x, y)	LOG(x) / LOG(b)	
LOG2(x)	LOG(x) / LOG(2)	
POW(x, y)	POWER(x, y)	
ATAN(x, y)	ATAN2(x, y)	

日付および時刻関数

MySQL関数	Adaptive Server Anywhere 関数	備考
TIME(expr)	CAST(expr as TIME)	CONVERT関数も使用可
TIMESTAMP(expr)	DATETIME(expr)	
DAYOFWEEK(expr)	DOW(expr)	
WEEKDAY(expr)	MOD(DOW(expr) - 1, 7)	
DAYOFMONTH(expr)	DAY(expr)	
WEEKOFYEAR(expr)	DATEPART(week, expr)	
PERIODADD(expr, N)	DATEFORMAT(DATEADD(month, 2, expr '/01'), 'YYYYMM')	
PERIOD_DIFF(P1, P2)	DATEDIFF(month, P2 '/01',P1 '/01')	
ADDDATE(date, numdays)	DATEADD(day, numdays, date)	
SUBDATE(date, numdays)	DATEADD(day, -numdays, date)	
EXTRACT(type FROM date)	DATEPART(type, date)	「 type 」引数は、Adaptive Server AnywhereとMySQLで異なるため、修正する必要があります。
TO_DAYS(date)	DAYS(date) - 58	Adaptive Server Anywhere は「 0000/01/01 」ではなく、「 0000/02/29 」からの差分を測定します。
MAKEDATE(year, dayofyear)	YMD(year, 0, dayofyear)	
UNIX_TIMESTAMP(date)	DATEDIFF(second, '1979/01/01', date)	

構文のマッピング

MySQLの構文のほとんどはAdaptive Server Anywhereで利用可能ですが、フィーチャにアクセスするための構文が違う場合もあります。それらの文およびAdaptive Server Anywhere側でそれらに相当するものを以下に説明します。

⇒以下に述べるAdaptive Server Anywhereの構文の例については「Adaptive Server Anywhere SQLリファレンス・マニュアル」の「SQL文」を参照してください。

「Adaptive Server Anywhere SQLリファレンス・マニュアル」はアイエニウェア・ソリューションズ株式会社の[ダウンロードページ](#)で入手できます。

演算子

MySQLには、2つ以上の任意の表現を比較して結果をbooleanで返す演算子があります。以下にそのような演算子を示します。Adaptive Server Anywhereに対応する演算子がある場合は一緒に記載しています。

MySQL演算子	Adaptive Server Anywhere演算子	備考
!= <=> ISNULL(expr) INTERVAL(N, N1, N2, ...)	<> (expr1 = expr2 OR ((expr1 IS NULL) AND (expr2 IS NULL))) IS NULL expr ビルトインとしては存在せず	<=>演算子は等式を表します。 NULL 値も含まれます。 (NULL=NULLがtrue) 同様に機能するユーザ定義関数を簡単に使用できます。 例: if (N < N1) then 0 elseif(N < N2) then 1 elseif ...
! && a XOR b	NOT AND OR ((a AND (NOT b)) OR ((NOT a) AND b))	Adaptive Server Anywhereの数式はXOR引数が多数ある場合は複雑になるので、アプリケーションシナリオによりますが、この表現の統合には他の方法をお勧めします。

データ操作書式

MySQL文	対応する Adaptive Server Anywhere文	備考
INSERT ... ON DUPLICATE KEY UPDATE	INSERT ... ON EXISTING UPDATE	Adaptive Server Anywhereには既存の行に対するERRORおよびSKIPオプションもあります。
SELECT ... INTO OUTFILE	UNLOAD SELECT ... DBISQL OUTPUT TO	
SELECT/UPDATE/DELETE ... LIMIT	FIRST or TOP n	
DEFAULT '0' NOT NULL auto_increment NOT NULL auto_increment	NOT NULL DEFAULT AUTOINCREMENT NOT NULL DEFAULT AUTOINCREMENT	
LIMIT offset, numRows	TOP numRows START AT offset	
Insert IGNORE	INSERT ... ON EXISTING SKIP	
Replace ...	INSERT ... ON EXISTING UPDATE	
FROM_DAYS()	DAYS()	
TO_DAYS()	DATEADD(day, ...)	
WEEKDAY()	DOW()	
GROUP_CONCAT	LIST	
STD	STDDEV	
CHARACTER_LENGTH Position()	LENGTH() function LOCATE() function	
LOCALTIME, LOCALTIMESTAMP	NOW() built in function	
DECODE	CASE statement	
INSERT INTO ... DEFAULT VALUES.	INSERT INTO ... VALUES(DEFAULT)	
LOAD DATA INFILE	LOAD TABLE	

その他の構文

以下は、前述の分類には該当しませんが互換性のある項目のリストです。まったく同じというわけではありませんが、同一機能を提供する関数のマッピングも行っています。

MySQL構文	Adaptive Server Anywhere構文	備考
VERSION() mysql_insert_id() LAST_INSERT_ID variable mysql_affected_rows() ANALYZE TABLE	@@version global variable @@identity global variable @@identity global variable @@rowcount global variable sa_table_page_usage, sa_table_fragmentation	Adaptive Server Anywhere は property()関数を介して他の属性へのアクセスも提供します。
OPTIMIZE TABLE	CREATE STATISTICS	Adaptive Server Anywhereには最適化機能があり、自動的に統計を行うので、手動更新は不要です。
CHECK TABLE USE <i>database-name</i>	sa_validate() procedure	Adaptive Server Anywhereに同等の機能はありません。サーバ上の各データベースに接続する必要があります。
LOCK TABLES (<i>name</i>) WRITE	LOCK TABLE <i>table-name</i> IN EXCLUSIVE MODE	Adaptive Server Anywhereは行レベルのロック機能を持っているので、テーブルのロックは通常は不要です。
UNLOCK TABLES	COMMIT	WITH HOLD節を使ってカーソルを開いていない限り、COMMITはすべてのロックを解除します。
Create table(KEY...) DO	CREATE TABLE ... CREATE INDEX CALL	Adaptive Server Anywhereでは二つの文が必要です。

MySQL構文	Adaptive Server Anywhere 構文	備考
FLUSH/RESET	sa_flush_cache sa_flush_statistics	MySQLの他の書き出し可能な要素のほとんどはAdaptive Server Anywhereによって自動的に管理されるので、書き出しは不要です。
expr1 SOUNDS LIKE expr2	SOUNDEX(expr1) = SOUNDEX(expr2)	
REGEX/RLIKE	SIMILAR	SIMILARはmysql REGEX構文とは動作が異なりますが、同じ関数を実行します。 MySQL REGEXP関数が使われている箇所では使用可能のはずです。
BINARY str CURDATE() CURRENT_-DATE() CURTIME() CURRENT_-TIME() SYSDATE() LOCALTIME() CURRENT_TIMESTAMP() UTC_DATE() DATABASE() LOAD_FILE(file)	CAST str AS BINARY CURRENT DATE CURRENT TIME CURRENT TIMESTAMP CURRENT UTC TIMESTAMP CURRENT DATABASE xp_read_file(file)	Adaptive Server Anywhereでは、ファイルの内容は1つの長いバイナリフィールドとして返されます。MySQLでは1文字列として返されます。
CONNECTION_ID()	CONNECTION_-PROPERTY('Number')	

移行に関する他の問題

MySQLからAdaptive Server Anywhereに移行する際のその他の注意点を以下に示します。

- MySQLの識別子は、任意に一重引用符「`'`」で囲むことになっていますが、Adaptive Server Anywhereでは二重引用符「`"`」または大括弧「`[]`」を使用します。
- commentやsessionのように、Adaptive Server Anywhereでキーワードとなっても、MySQLではそうでないものがあります。これらのキーワードはAdaptive Server Anywhereで使用できるように「`"`」で囲まなければなりません。あるいは認識されたキーワードのリストを変更するために、Adaptive Server AnywhereのNON_KEYWORDS オプションを使うこともできます。

⇨NON_KEYWORDSオプションについては、「Adaptive Server Anywhereデータベース管理ガイド」の「NON_KEYWORDS オプション [互換性]」を参照してください。

「Adaptive Server Anywhere データベース管理ガイド」はアイエニウェア・ソリューションズ株式会社の[ダウンロードページ](#)で入手できます。

- Adaptive Server Anywhereでのタイムスタンプの最小値は「0001-01-01 00:00:00」で、MySQLでは「0000-00-00 00:00:00」です。
- MySQLでのタイムスタンプのフォーマットは「YYYY-MM-DD hh:mm:ss」です。Adaptive Server Anywhereは、小数点をタイムスタンプ値の一部として扱います。TIMESTAMP_FORMATオプションでは、日付/時刻値を返すためのフォーマットをそのまま指定できます。

⇨TIMESTAMP_FORMATオプションについては、「Adaptive Server Anywhereデータベース管理ガイド」の「TIMESTAMP_FORMAT オプション [互換性]」を参照してください。

「Adaptive Server Anywhere データベース管理ガイド」はアイエニウェア・ソリューションズ株式会社の[ダウンロードページ](#)で入手できます。

- MySQLでは、文字列を表すのに一重引用符も二重引用符も使用できますが、Adaptive Server Anywhereでは、デフォルトで一重引用符を使うようになっています。前述のように、二重引用符はデフォルトでデータベース・オブジェクトの識別に使用します。この機能の設定は、データベースのQUOTED_IDENTIFIERオプションで変更することができます。

⇨QUOTED_IDENTIFIERオプションについては、「Adaptive Server Anywhereデータベース管理ガイド」の「QUOTED_IDENTIFIER オプション [互換性]」を参照してください。

「Adaptive Server Anywhere データベース管理ガイド」はアイエニウェア・ソリューションズ株式会社の[ダウンロードページ](#)で入手できます。

Adaptive Server AnywhereデータベースへのMySQLデータベースの移行

MySQLからAdaptive Server Anywhereへのデータの移行は容易ですが、前述のように、MySQL特有のデータ形式を使っているときに限り、小さな問題が生じる可能性があります。データの移行はSybase Centralの一部であるデータ統合ウィザードを使って行います。またAdaptive Server Anywhereでsa_migrateストアド・プロシージャ・セットを使って統合をより一層カスタマイズすることもできます。Adaptive Server AnywhereのLOAD TABLE文とmysqldumpユーティリティの組み合わせも、データ統合に使うことができます。統合の際は、MySQLのSETやENUMなどのデータ形式がMySQLデータベースに使われているかどうかを確認してください。これらが使用されている場合は特別な注意が必要です。

⇒ MySQLのデータ形式とAdaptive Server Anywhereとの違いについては「[データ形](#)」を参照してください。

前提

- 本書は、サポートされたプラットフォーム上で動作するMySQLデータベースとWindowsにインストールされたAdaptive Server Anywhere 9.0.1が用意されていることを前提として記述されています。
- MySQLデータベースを作成していない場合、統合手順中でMySQLテストデータベースにテーブルをいくつか作成できます。
- Adaptive Server Anywhereデータベースが動作しているマシンに、MySQL ODBC パッケージ3.5.1以降のドライバもインストールしておく必要があります。

Adaptive Server Anywhereデータベースの作成

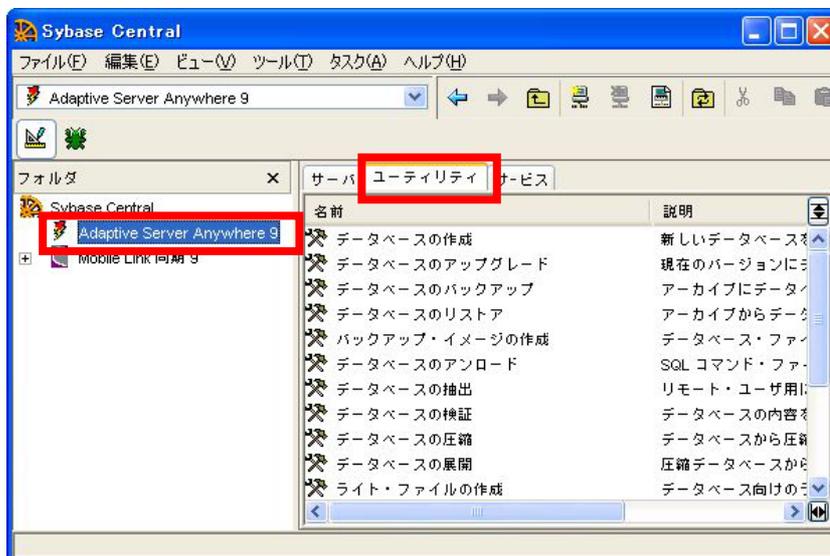
MySQLデータベースを統合するには、あらかじめAdaptive Server Anywhereデータベースを作成する必要があります。Sybase Centralを利用して新しいデータベースを作成する方法を以下に示します。

1. Sybase Centralを起動します。[スタート]メニューから、[プログラム] [SQL Anywhere 9] [Sybase Central]を選択します。

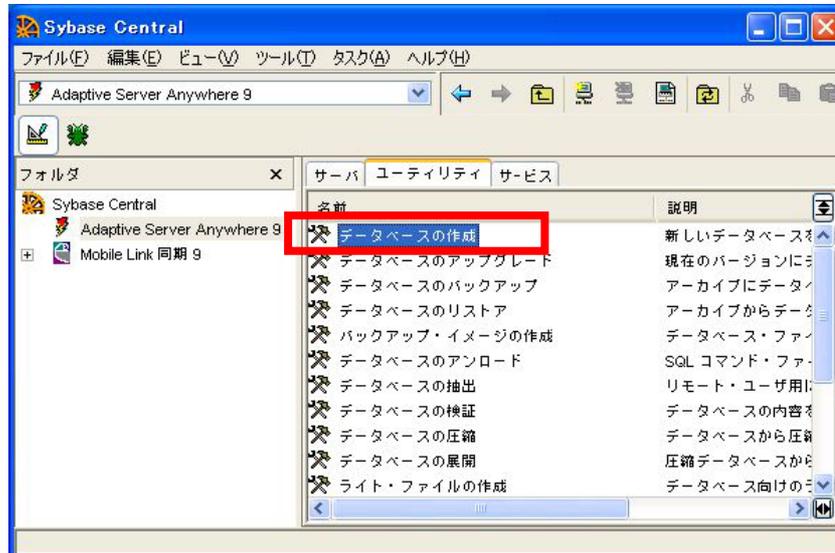


2. 新しいAdaptive Server Anywhere 9データベースを作成します。

- Sybase Centralの左側のペインからAdaptive Server Anywhere 9を選択します。右側のペインにある[ユーティリティ]タブをクリックします。



- [データベースの作成]をダブルクリックします。
データベースの作成ウィザードが起動します。



- ウィザードの指示に従い、新しいデータベースを作成します。

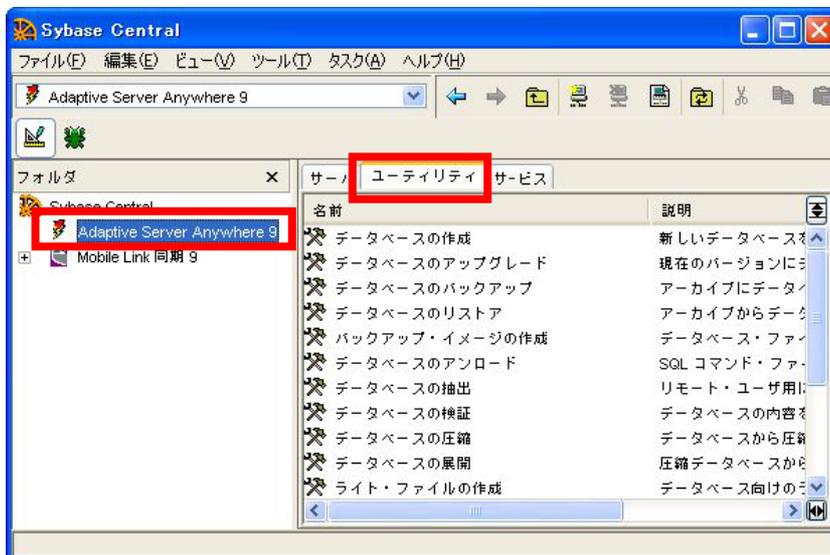
MySQLデータベースにデータソースを作成

統合処理にはデータソースへのODBC接続が必要です。したがってまずMySQLデータベースにODBCデータソース (DSN) を作成しなければなりません。

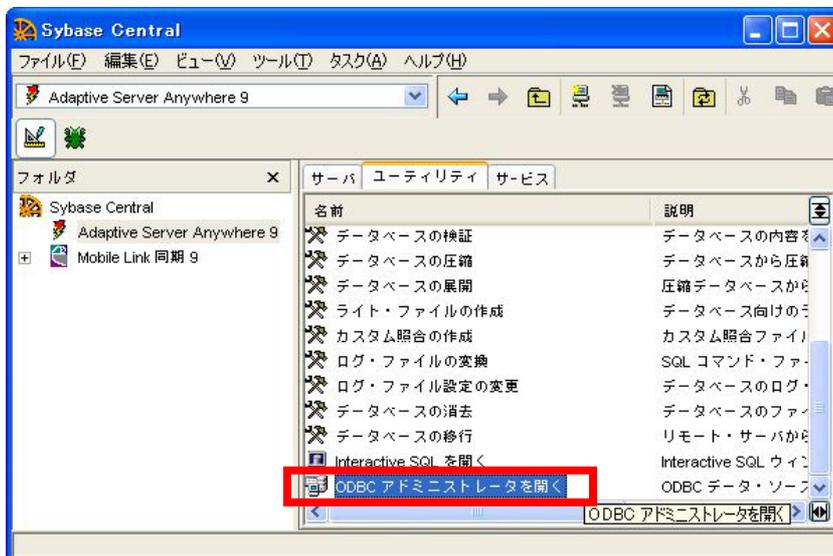
1. ODBC 3.5.1ドライバがない場合はダウンロードしてインストールしてください。
最新版は以下のサイトから入手できます。
<http://www.mysql.com/downloads/api-myodbc.html>
2. Sybase Centralを起動します。[スタート]メニューから、[プログラム] [SQL Anywhere 9] [Sybase Central]を選択します。



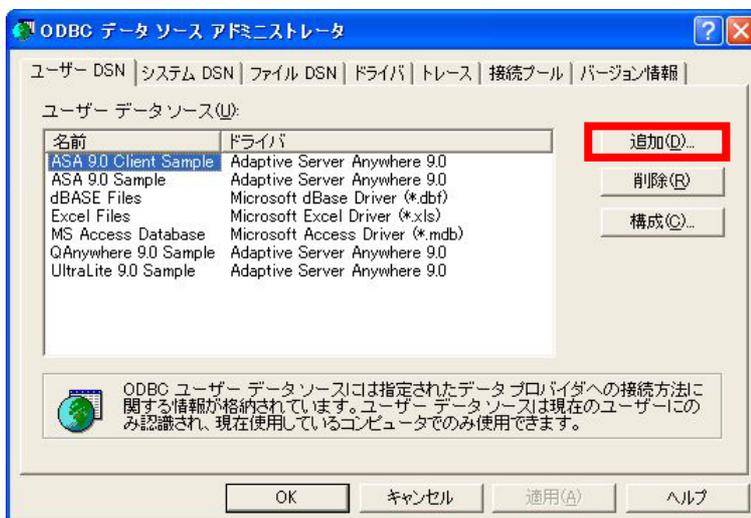
3. Sybase Centralの左側のペインからAdaptive Server Anywhere 9を選択し、右側のペインにある[ユーティリティ]タブをクリックします。



- [ODBCアドミニストレータを開く]をダブルクリックします。
ODBC データソース アドミニストレータ ダイアログが表示されます。



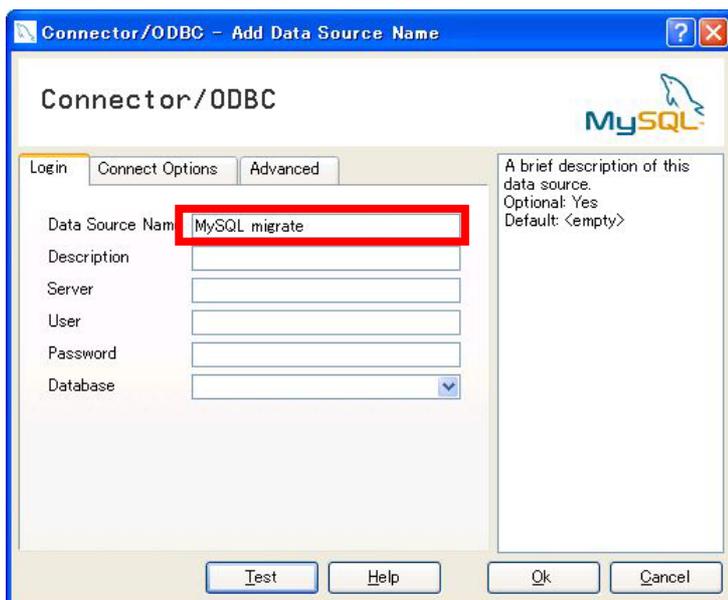
- [追加]をクリックします。
データベースの新規作成ウィザードが起動します。



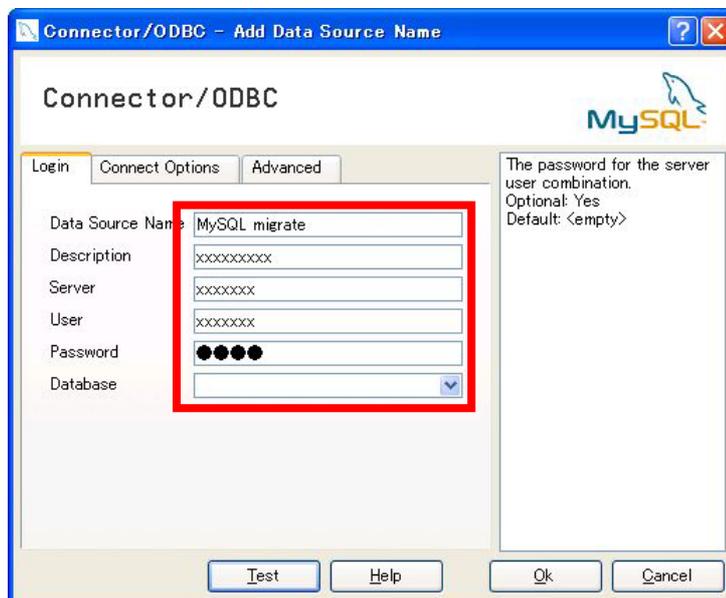
6. 利用可能なドライバのリストから「MySQL ODBC 3.5.1 Driver」を選択し、[完了]をクリックします。MySQL ODBC 3.5.1ドライバ – DSN設定ダイアログが表示されます。



7. Data Source Nameにデータソースの名前を入力します。例としてデータソースの名前を「MySQL migrate」とします。



8. MySQLデータベースに関する必要な項目に適切な値を入力します。

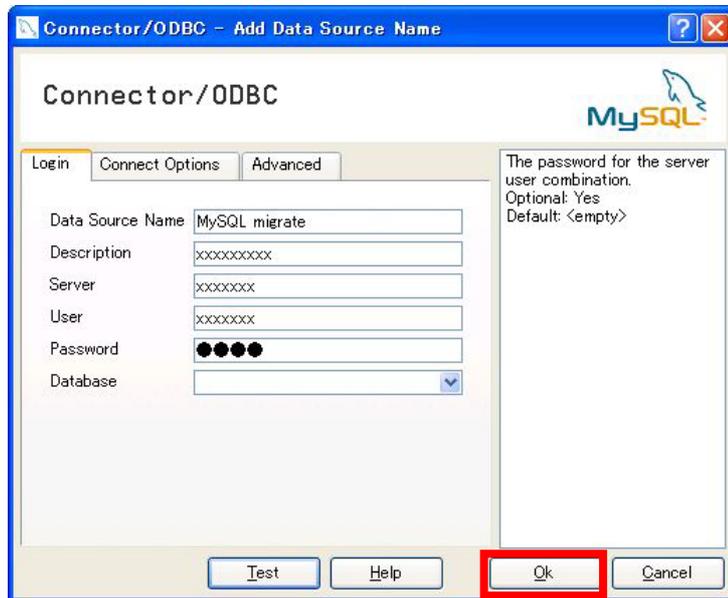


9. [Test]ボタンをクリックし、データソースに間違いがないか確認してください。

データソースに問題なければ以下のダイアログボックスが表示されます。



10. [OK]をクリックします。

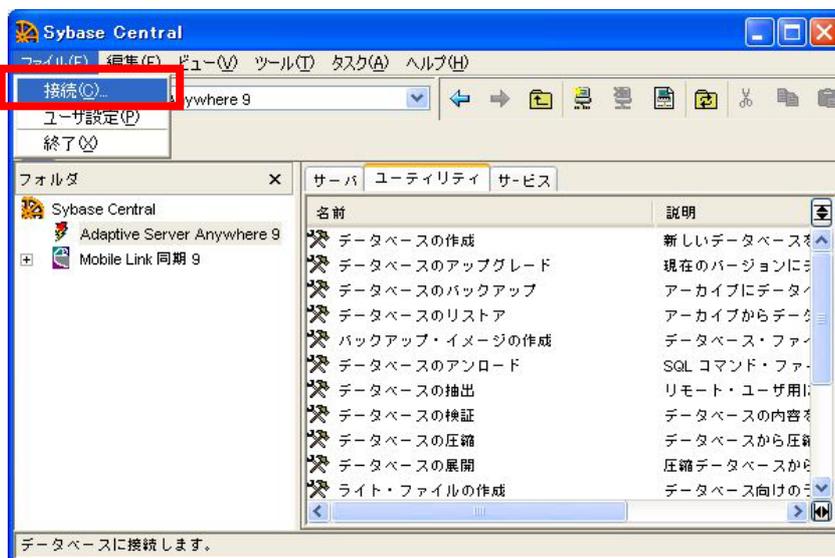


MySQLデータベースのAdaptive Server Anywhereへの移行

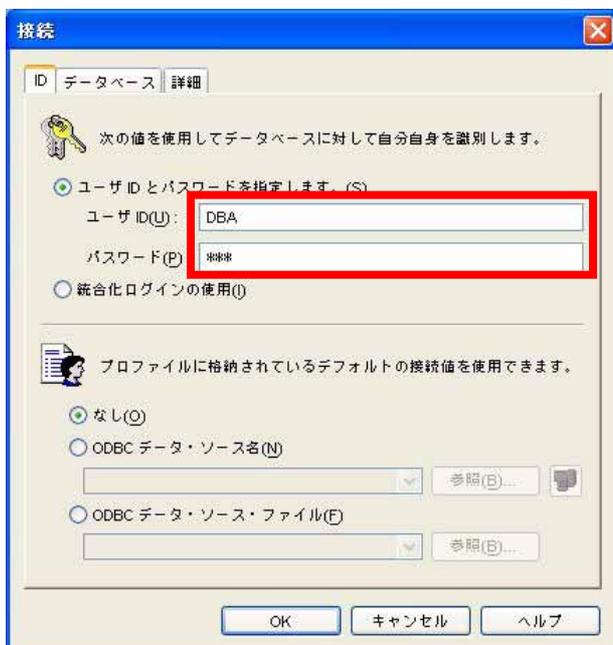
新しいAdaptive Server Anywhereデータベースへの統合を実行するためには、まずAdaptive Server Anywhereデータベースに接続しなければなりません。データベースファイルが格納されている場所への接続方法を以下に説明します。

Adaptive Server Anywhereデータベースへの接続

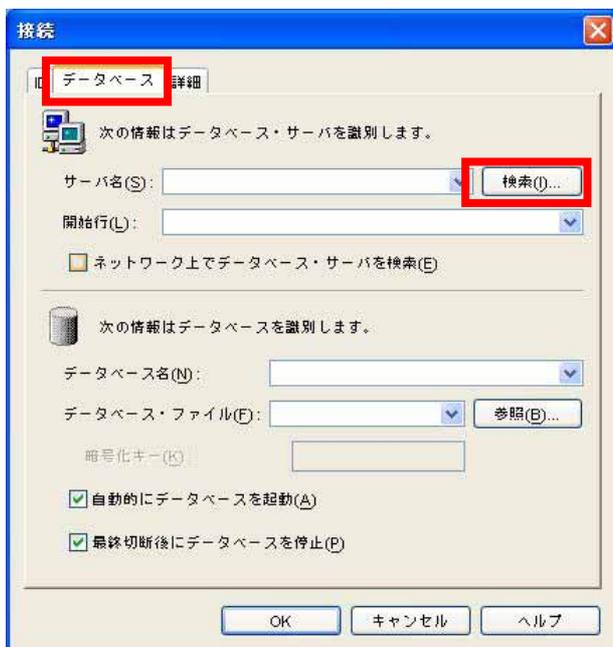
1. Sybase Centralの左側のペインからAdaptive Server Anywhere 9を選択し、[ファイル]メニューから[接続]を選択します。
接続ダイアログが表示されます。



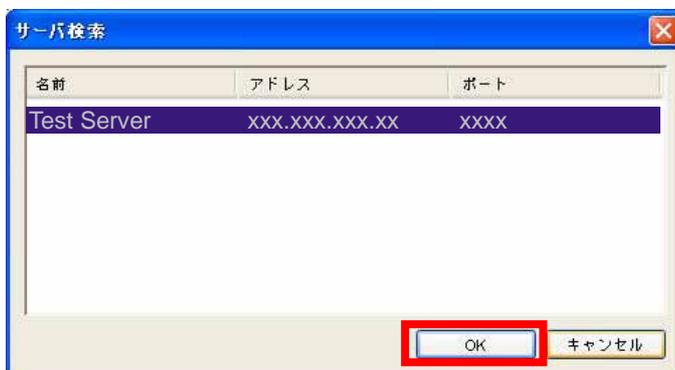
2. [ID]タブで、データベース用のユーザIDとパスワードを入力します。
デフォルトでは、すべてのAdaptive Server AnywhereデータベースはユーザIDが「DBA」、パスワードが「SQL」になっています。



3. [データベース]タブから[検索]ボタンをクリックし、作成したAdaptive Server Anywhereデータベースファイルを選択します。



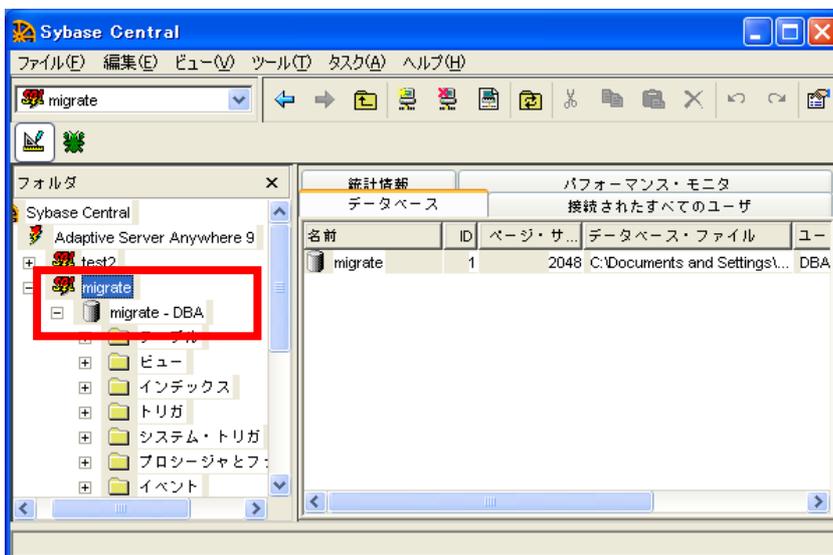
4. [OK]をクリックします。
Adaptive Server Anywhereデータベースサーバが自動的に起動します。



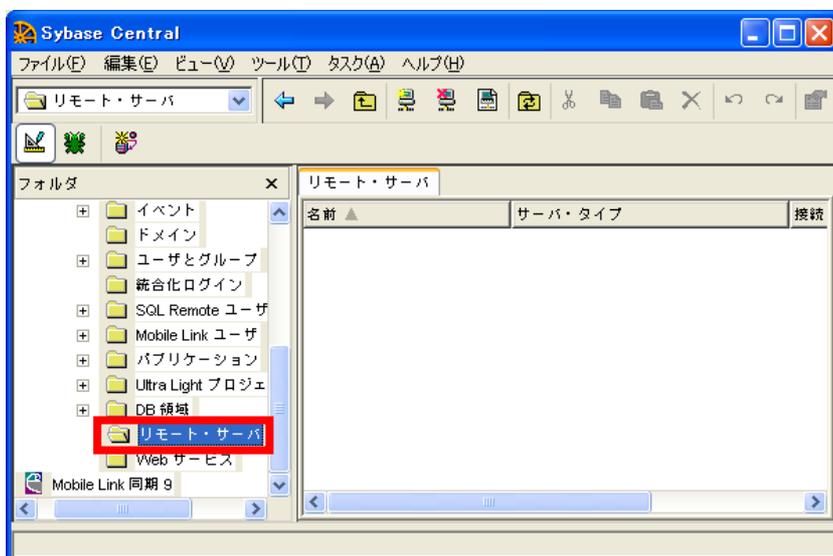
次の手順で、Sybase CentralでMySQLデータベースを検索します。これはリモートサーバを生成して行います。

リモートサーバの作成

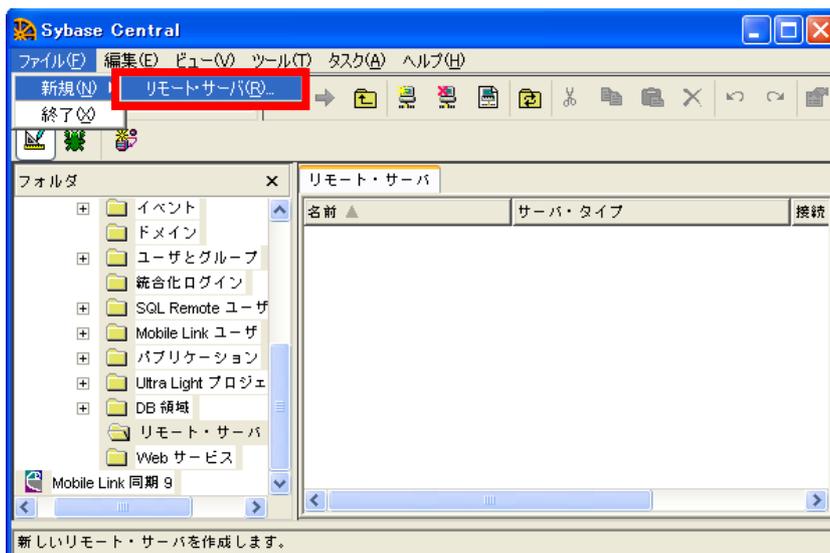
1. Sybase Centralの左側のペインで、データベースサーバとデータベースアイコンを開きます。
下の例では、データベースmigrateは、同じ名前のデータベースサーバmigrate上で起動しています。



2. Sybase Centralの左側のペインからリモート・サーバを選択します。

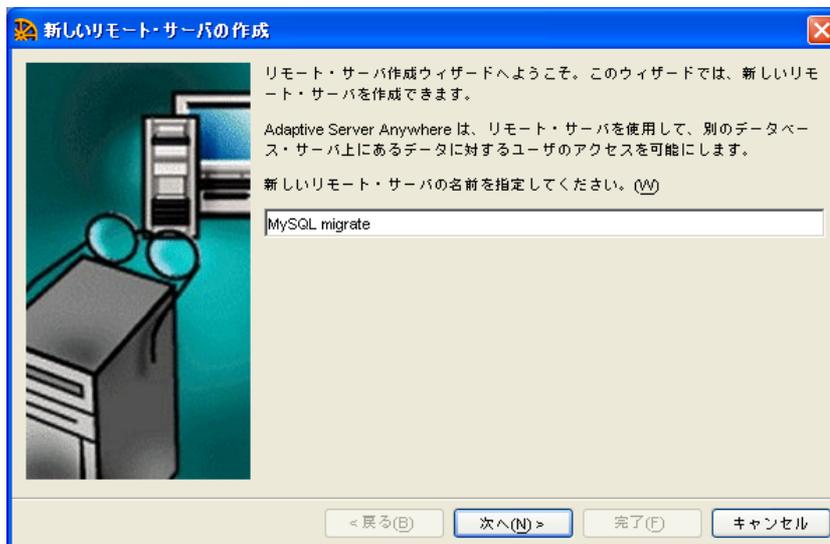


3. [ファイル]メニューから、[新規] [リモート・サーバ]を選択します。
新しいリモート・サーバの作成ウィザードが起動します。



4. ウィザードの指示に従い、MySQLデータベースに接続するリモートサーバを作成します。

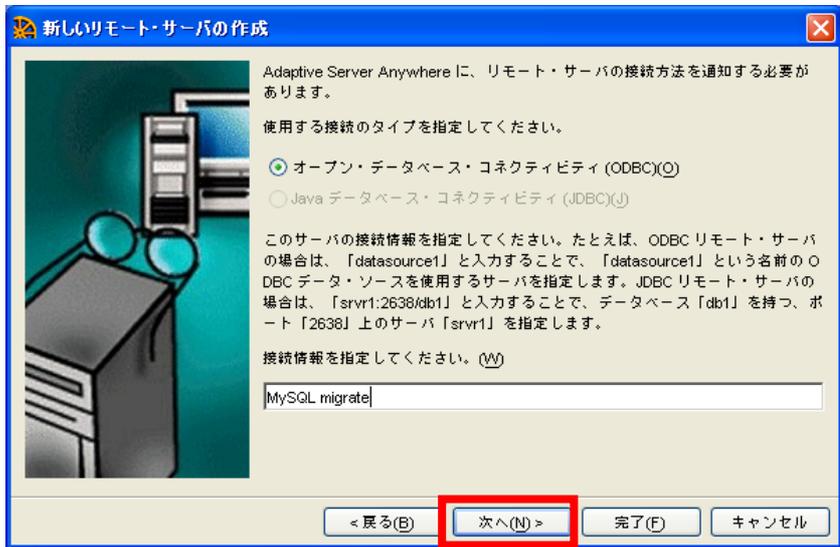
- ウィザードの最初の画面でリモートサーバ名を入力します。例として名前を「MySQL migrate」とします。[次へ]をクリックします。



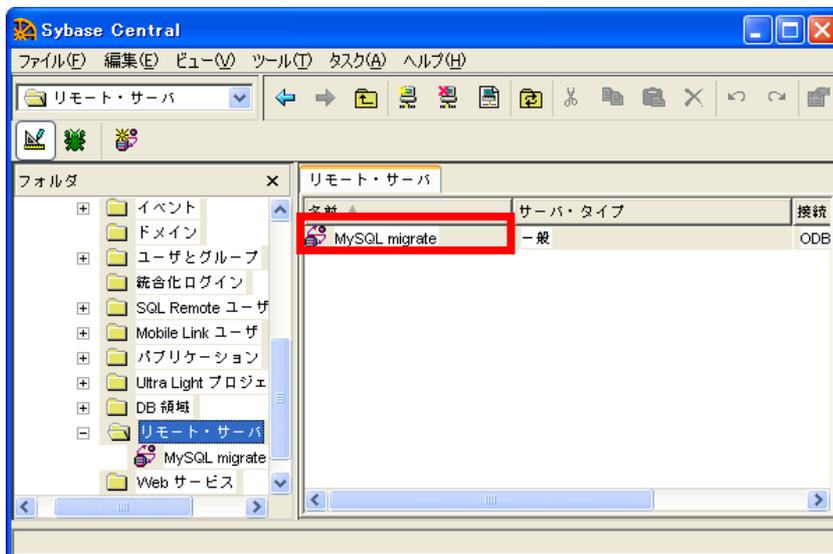
- リモート・サーバのタイプとして「一般」を選択します。[次へ]をクリックします。



- Open Database Connectivity (ODBC) オプションを選択し、接続情報フィールドに MySQL データベースの ODBC データソース名を入力します。たとえば作成時に ODBC データソースに「MySQL migrate」と名付けたら、接続情報フィールドに「MySQL migrate」と入力します。[次へ]をクリックします。



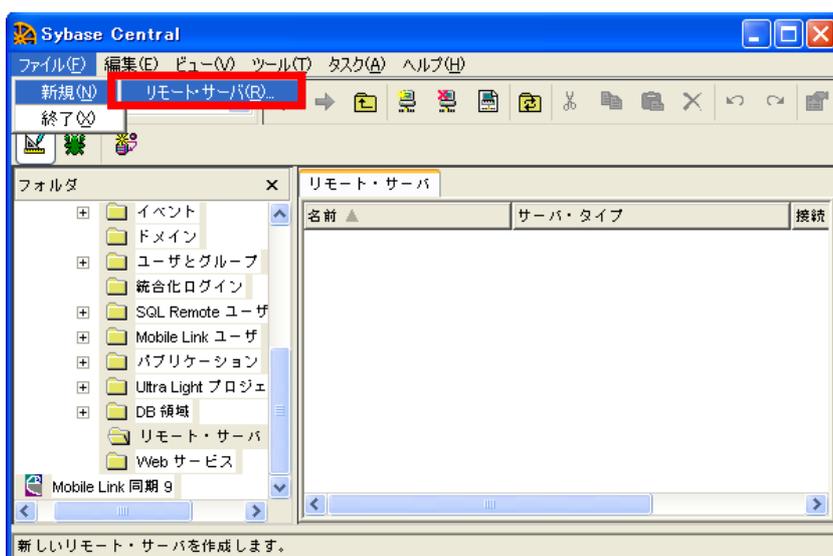
5. [完了]をクリックします。
新しいリモートサーバがSybase Central内で起動します。



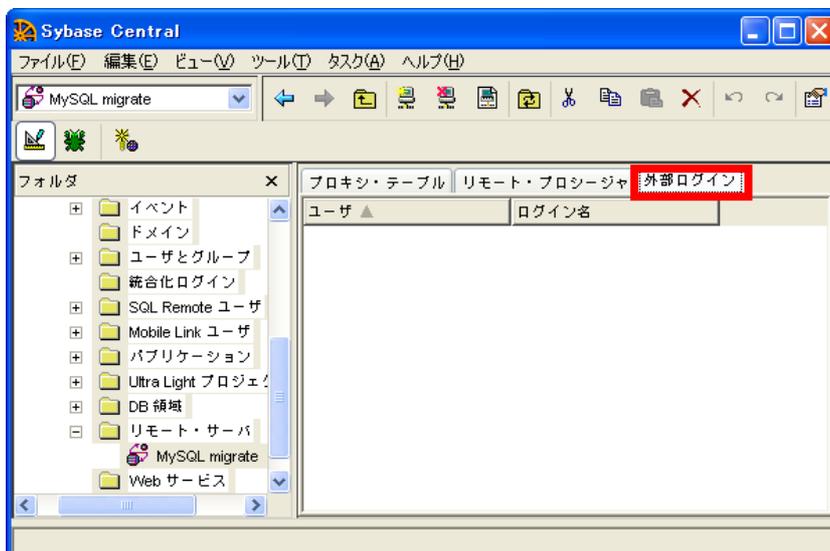
リモートサーバが、Adaptive Server Anywhereデータベースに接続したユーザを同じユーザとして認識しない場合は、現在のユーザに対して外部ログインを作成する必要があります。たとえばAdaptive Server AnywhereデータベースにユーザID、DBAで接続し、MySQLデータベースがユーザID、DBAを拒否する場合は、外部ログインを作成しなければなりません。

外部ログインの作り方

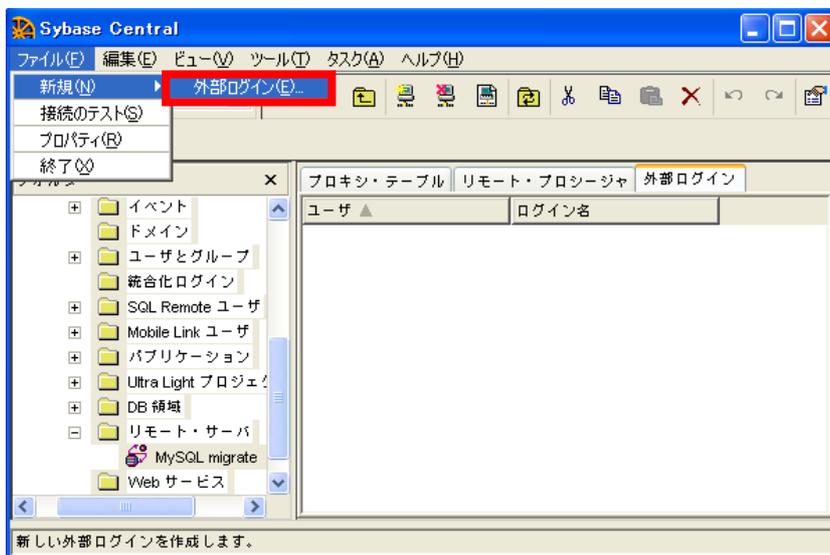
1. Sybase Centralの左側のペインから[リモート・サーバ]フォルダを開き、リモートサーバを選択します。



2. 右側のペインで、[外部ログイン]タブをクリックします。



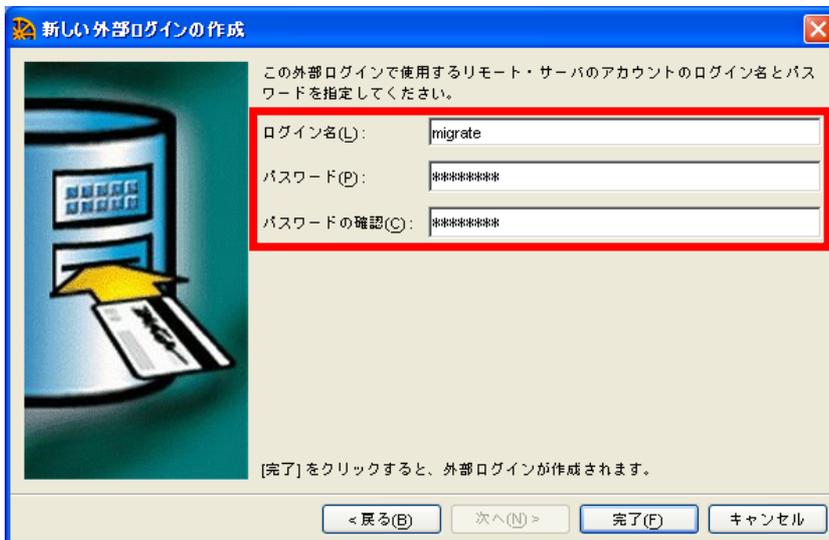
3. [ファイル]メニューから、[新規] [外部ログイン]を選択します。
外部ログインの作成ウィザードが起動します。



4. ユーザリストから現在接続しているユーザを選択します。



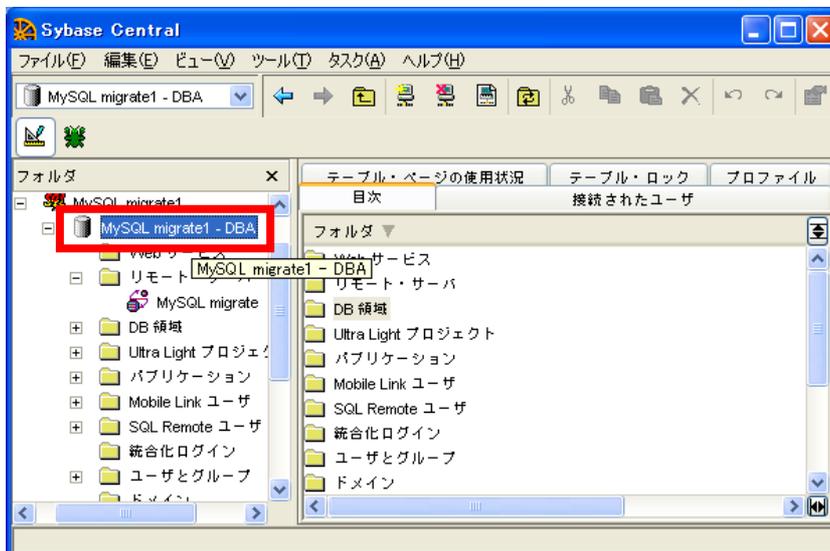
5. [ログイン名]にMySQLデータベースのユーザ名を入力します。[パスワード]にこのユーザのパスワードを入力します。[完了]をクリックします。



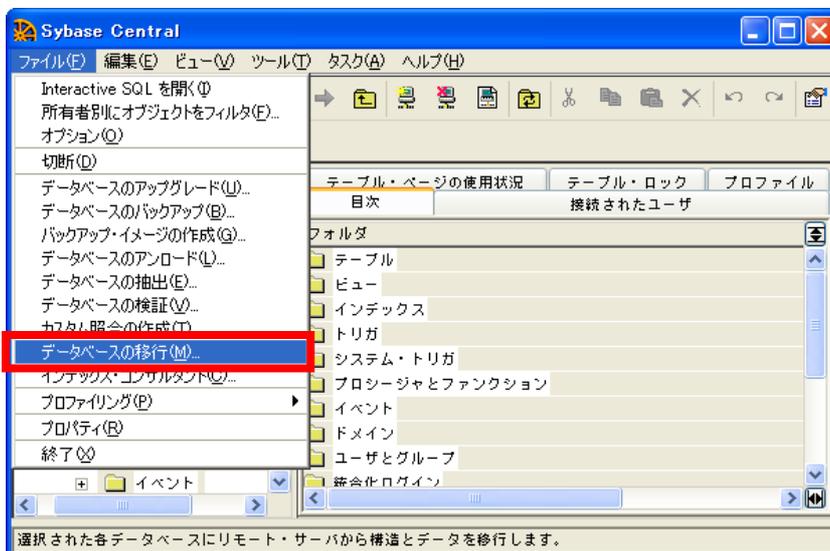
これでMySQLデータベースを統合する準備が整いました。Adaptive Server Anywhereが起動し、ODBCを介してMySQLデータベースに接続できます。次の手順では、統合ウィザードを使って移行を行います。

MySQLデータベースの移行

1. Sybase Centralの左側のペインから、Adaptive Server Anywhereデータベースを選択します。



2. [ファイル]メニューから、[データベースの移行]を選択します。
データベースの統合ウィザードが起動します。



3. はじめのダイアログで[次へ]をクリックします。



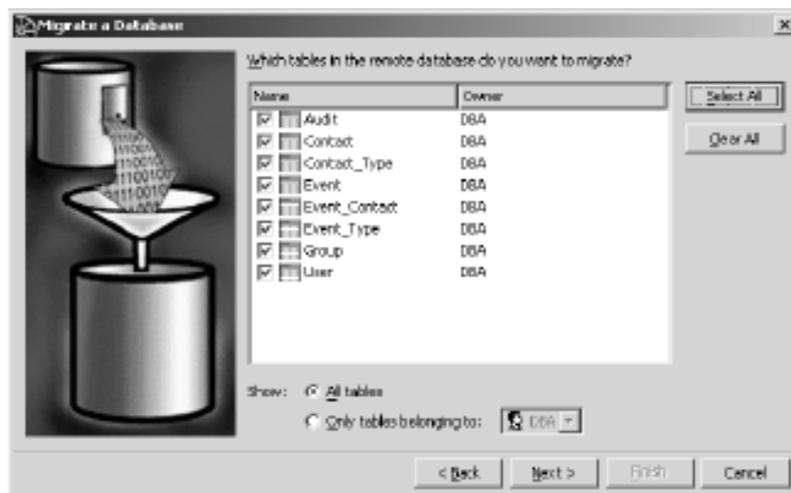
4. 該当のデータベースを選択し、[次へ]をクリックします。



5. 作成したMySQLリモートサーバ、ここではMySQL migrateを選択し、[次へ]をクリックします。

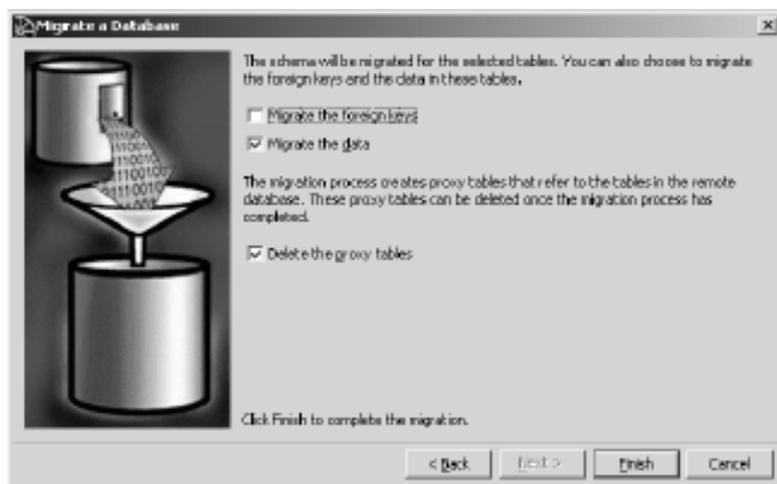


6. Adaptive Server Anywhere データベースにすべてのMySQL テーブルを移行するため、[全て選択] ボタンをクリックし、[次へ]をクリックします。



7. 所有したいテーブルのAdaptive Server Anywhereデータベースユーザを選択します。[次へ]をクリックします。

8. 統合したいオプションを選択します。MySQL ODBCドライバの制限により、外部キーを統合できません。後でエラーが起こるのを避けるため、[外部キーを統合する]のチェックを外します。



9. [完了]をクリックして、移行を開始します。
データベースの統合ウィンドウが表示されます。状態が「完了」に変わったら、このウィンドウを閉じます。

新しい Adaptive Server Anywhere データベースの調整

Adaptive Server Anywhere データベースへの MySQL スキーマとデータの統合が完了したので、Adaptive Server Anywhere のメリットを活用できるようになりました。すぐに活用できるのは処理サポートです。

ここで簡単な調整を加えることで、既存のスキーマをさらに改善できます。

すべての MySQL テーブルが参照整合性をサポートしているわけではないので、MySQL スキーマが外部キーを持っていない場合があります。MySQL ODBC ドライバは、InnoDB ユーザに対しても、外部キーのチェック機能をサポートしていません。そのため新しい Adaptive Server Anywhere データベースは外部キーを持たないのです。

参照の整合性をサポートする方法

1. MySQL データベースに対して以下の SQL 文を発行し、MySQL データベースの外部キーをリストアップします。

```
SHOW TABLE STATUS FROM database_name
```

SHOW CREATE TABLE table_name を使っても外部キーの関係を知ることができます。参照の制約がテーブルごとのコメント列に以下の形でリストアップされます。

```
(column_name) REFER ref_db_name/ref_table_name(ref_column_name)
```

2. 参照の整合性制約を指定します。

Sybase Central を用いて、データベースに外部キーを追加します。

あるいは外部キーごとに、Adaptive Server Anywhere データベースに対して以下の SQL 文を発行します。(インタラクティブ SQL ユーティリティ (dbisql) を使用)

```
ALTER TABLE "table_name"  
ADD FOREIGN KEY "foreign_key_name" ("column_name")  
REFERENCES "ref_table_name" ("ref_column_name");
```

新しい外部キーの制約を設定することで、Adaptive Server Anywhere データベースは自動的に参照の整合性をチェックし、データの整合性を大幅に高めます。

正しく配置されたインデックスはデータベースのパフォーマンスを著しく高めますが、正確に配置されていない場合、パフォーマンスは大幅に低下します。SQL Anywhere Studio 9 では、データベースの利用度と作業負荷を調べ、必要に応じてインデックス構造の変更を推奨するインデックス・コンサルタントが導入されました。MySQL では、外部キーの列にインデックスが明確に定義されていなくてはなりません。Adaptive Server Anywhere ではその必要はありません。また MySQL は各プライマリ・キーに対してプライマリインデックスを生成しますが、Adaptive Server Anywhere では不要です。統合の過程で、インデックス・コンサルタントが MySQL データベースからコピーされた余分なインデックスの削除を促すはずですが、インデックス・コンサルタントは、統合の完了後も Adaptive Server Anywhere データベースのパフォーマンスを高める便利なツールとしてお役に立つことでしょう。

⇨ スキーマの最適化については、[iAnywhere 開発者リソース](#)で入手できる SQL Anywhere Studio 関連の資料を参照してください。

MySQLからAdaptive Server Anywhereへのアプリケーションの移行

MySQLからAdaptive Server Anywhereへとアプリケーションを移行する方法は、MySQLにアクセスするインタフェースにより異なります。移行する際に最小限の作業が必要となる汎用インタフェースを以下に示します。

ODBC

Adaptive Server AnywhereとMySQLはどちらもODBC 3.51 API仕様をサポートしています。一般にこれらのアプリケーションの移行には、Adaptive Server Anywhereに合わせてODBCデータソースを変更する必要があります。特定のAPI機能の実装に明確な違いがあることもありますが、ODBCの仕様は十分に確立されているので、違いは小さいはずです。

JDBC

MySQLはタイプ4JDBCドライバを使います（100%Java実装）。これをAdaptive Server Anywhereへ移行するにはSybase jConnectドライバを使う必要があります。ただしAdaptive Server Anywhereのパフォーマンスをの最大限に引き出すためには、iAnywhere JDBCドライバを使用することをお勧めします。iAnywhere JDBCドライバはタイプ2JDBCドライバです。iAnywhere JDBCドライバはJDBC2.0仕様の全コア・エレメントとオプションの一部をサポートします。

Perl

Perlアプリケーションの移行については、「[MySQLからAdaptive Server AnywhereへのPerlアプリケーションの移行](#)」をご覧ください。

PHP

PHPアプリケーションの移行については、「[MySQLからAdaptive Server AnywhereへのPHPアプリケーションの移行](#)」をご覧ください。

MySQLでサポートされている他のインタフェースを使って書かれたアプリケーションは、Adaptive Server Anywhereのドライバではサポートされていないので、移行するにはさらに多くの作業が必要です。これにはMySQL C/C++ API、Python、Tcl、Eiffelのアクセスドライバが含まれます。ただしODBCにブリッジできたり、ネイティブでAdaptive Server Anywhereにアクセスできるサードパーティ製のドライバも存在します。

MySQLからAdaptive Server AnywhereへのPerlアプリケーションの移行

MySQLからAdaptive Server AnywhereへのPerlアプリケーションの移行はとても簡単です。DBD::ODBCドライバを使ってODBCに接続してもよいし、Adaptive Server Anywhereに同梱されているネイティブのAdaptive Server Anywhereドライバ（名称DBD::ASANY）を使うという選択肢もあります。すでにDBD::ODBCドライバをご使用の場合は、アプリケーションの移行は単にAdaptive Server Anywhereを参照する接続文字列を変更するだけでできます。

ただし前述のように、移行の完了後には、Adaptive Server AnywhereとMySQLの違いを処理するための小さな調整作業が必要となります。MySQL固有のメソッドの一部は、クエリまたは標準DBD機能により、対応するAdaptive Server Anywhereに移行できます。以下に例を示します。

MySQL	Adaptive Server Anywhere	備考
mysql_insertid is_blob, is_num, is_not_null, length, name, table, type is_key, is_pri_key	SELECT @@identity NAME, TYPE, SCALE, PRECISION, NULLABLE SELECT ... FROM syscolumn WHERE ...	これらの属性項目はすべてDBIの標準要素です。 インデックスやキーの検索は、システムテーブル内のテーブルおよび列の定義を参照して行います。

MySQLからAdaptive Server AnywhereへのPHPアプリケーションの移行

MySQLからAdaptive Server AnywhereへのPHPアプリケーションの移行はとても簡単です。これにはODBCを使ってAdaptive Server Anywhereに接続する方法と、アイエニウェア・ソリューションズのAdaptive Server Anywhere-PHPモジュール（アイエニウェアの[開発者用Webサイト](#)からダウンロード可能）を使う方法があります。

Windowsユーザには、ODBC APIを使った方法の方が好まれるかもしれません。ODBC用にWindowsのDSNを設定するのは簡単です。また、PHP用のWindowsバイナリにはODBC対応機能が組み込まれています。

一方でLinuxユーザは、PHPモジュールを設定するほうが簡単に感じるかもしれません。configureスクリプトを呼び出すと、`-with sqlanywhere[=path_to_asa]` フラグを使って、Adaptive Server AnywhereサポートがPHPにコンパイルされます。モジュールの詳細は、以下のサイトでご覧になれます。
www.sybase.com/detail?id=1019698

PHPアプリケーションがMySQLデータベースに接続するためすでにODBCを使用している場合、関数コールを変更する必要はありません。次項を飛ばして「[PHP統合時の注意](#)」に進んでください。

機能のマッピング

MySQL、ODBC、Adaptive Server AnywhereのAPIはとても似ています。一方の機能を他方の機能に直接マッピングできることもありますが、一方の機能に相当する機能が他方ない場合もあります。そのときは、開発者自身が同様の結果をもたらす代替コードを作らなければなりません。場合によっては、Adaptive Server Anywhereの高度な機能を活用するため、コードの一部を書き直すことをお勧めします。

以下の表に、広く利用されているいくつかのMySQLの機能と、それに相当するODBCおよびAdaptive Server Anywhereの機能を示します。

MySQL	Adaptive Server Anywhere (ODBC)	Adaptive Server Anywhere (PHPモジュール)
mysql_close	odbc_close	sqlanywhere_disconnect
mysql_connect	odbc_connect	sqlanywhere_connect
mysql_erro	odbc_error	「 mysql_erro 」を参照
mysql_error	odbc_errormsg	なし
mysql_escape_string	「mysql_escape_string」を参照。	「 mysql_escape_string 」を参照
mysql_fetch_row	odbc_fetch_row	sqlanywhere_fetch_row
mysql_insert_id	「mysql_insert_id」を参照。	「 mysql_insert_id 」を参照

MySQL	Adaptive Server Anywhere (ODBC)	Adaptive Server Anywhere (PHPモジュール)
mysql_num_fields	odbc_num_fields	sqlanywhere_num_fields
mysql_num_rows	odbc_num_rows	sqlanywhere_num_rows
mysql_query	odbc_exec	sqlanywhere_query
mysql_select_db	なし	なし

mysql_connect

ODBCを介した接続は簡単です。odbc_connect関数は、DSN、ユーザ名、パスワードを最低限必要とします。PHPモジュールを介して接続するには、Adaptive Server Anywhere接続文字列が必要になります。通常は次の関数を呼び出すことになります。

```
sqlanywhere_connect("uid=DBA;pwd=SQL;eng=eng_name")
```

mysql_errno

この関数は前回のクエリのエラー数を返します。次のSQL文を発行しても同じ結果が得られます。

```
SELECT @@error
```

mysql_escape_string

ODBCとPHPモジュールのどちらもSQL文字列をエスケープする機能を提供していませんが、一重引用符を二重引用符で置き換えることで簡単に実行できます。

mysql_insert_id

この関数はオートインクリメント列の最後に挿入されたIDを返します。次のSQL文を発行しても同じ結果が得られます。

```
SELECT @@identity
```

上記のようにMySQL関数の多くは、ODBCやAdaptive Server Anywhere呼び出しに直接変換できます。残りの関数についても簡単な代替法があります。ただし残念ながらMySQLとAdaptive Server Anywhereには無視できない違いが存在します。次項ではこの点に関して説明します。

PHP統合時の注意点

各データベースベンダーの間には、SQL文字列を処理する方法にわずかな違いがあります。たとえばMySQLのタイムスタンプのフォーマットはYYYY-MM-DD hh:mm:ssですが、Adaptive Server Anywhereでは、小数点以下の秒までのタイムスタンプがサポートされています。PHPのstrtotime関数はAdaptive Server Anywhereのタイムスタンプを認識できませんので、Adaptive Server Anywhereのタイムスタンプから秒の小数点部分を削除する作業が必要になります。

Adaptive Server Anywhereは、ODBCを介したトランザクションとprepared statementもサポートしています。odbc_commit関数とodbc_rollback関数は、ユーザの希望通りにトランザクションを終了します。PHPではautocommitがデフォルトで設定されている点に注意しなくてはなりません。これは各文がただちに実行されることを意味します。odbc_autocommit関数を使ってautocommitを設定すれば、大きなトランザクションが可能になります。同じクエリを違うパラメータで何度も実行するときには、prepared statementが便利です。これは各動的SQL文が一度だけエンジンに組み込まれることで効率を高めることによります。prepared statementの実行にはodbc_prepare関数とodbc_execute関数が使われます。

要約すれば、PHPアプリケーションをMySQLからAdaptive Server Anywhereへと移行するには、データベースを統合し、MySQL関数の呼び出しをAdaptive Server Anywhere関数の呼び出しへと変更し、構文とSQL文を調整してデータベース間の違いを解決することが必要になります。通常は、Adaptive Server Anywhereで利用できる高度な機能を使ってパフォーマンスの向上を図ることも可能です。

法的注意

Copyright(C) 2005 iAnywhere Solutions, Inc. All rights reserved.

iAnywhere、iAnywhere Solutions、iAnywhere Solutions(ロゴ)、Adaptive Server、SQL Anywhereは iAnywhere Solutions, Inc.またはSybase, Inc.とその系列会社の米国または日本における登録商標または商標です。その他の商標はすべて各社に帰属します。

Mobile Linkの技術には、Certicom, Inc.より供給を受けたコンポーネントが含まれています。これらのコンポーネントは特許によって保護されています。

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.とその供給元に帰属し、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, Inc.とその供給元の知的所有権の対象となるものであり、iAnywhere Solutions, Inc.とその供給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 Solutions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。Sybaseは、資料が途切れていないこと、誤りがないこと、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。



アイエニウェア・ソリューションズ株式会社

<http://www.ianywhere.jp/>