



# Oracle から SQL Anywhere へのマイグレーション

A whitepaper from Sybase iAnywhere

Date: April 2009

このホワイトペーパーは、SQL Anywhere 11 をベースに書かれていますが、ほとんどの部分で過去のバージョンや将来のバージョンでも利用可能です。また、英語のホワイトペーパーをそのまま翻訳しているため、製品画面ショットなどは英語版ソフトウェアのもので、SQL Anywhere は、日本語にローカライズされています。

## 目次

A whitepaper from Sybase iAnywhere .....	1
Date: April 2009 .....	1
はじめに .....	1
System Requirements システム要件 .....	1
SQL Anywhere のデータ移行機能 .....	1
データベース移行ウィザード .....	1
リモートデータアクセス .....	2
Oracle サンプルスキーマを移行する .....	2
ODBC データソースを作成する .....	2
新しい SQL Anywhere データベースを作成する .....	4
HR スキーマを移行する .....	4

## はじめに

SQL Anywhere には、Oracle データソースから、SQL Anywhere への容易かつ迅速なマイグレーションが可能なビルトインの機能が実装されています。直観的なウィザードと、リモートデータアクセスの機能を使用することで、Oracle データベースから SQL Anywhere へ、スキーマとデータを移行することが可能です。SQL Anywhere へマイグレーションすることで、SQL Anywhere の自動管理と自動チューニング機能を利用することが可能です。

このホワイトペーパーでは、Oracle データベースに含まれているサンプルスキーマを SQL Anywhere へ移行する方法について解説します。

備考：（ノート PC、スマートフォン、タブレット PC などの）リモートユーザーが、ローカルのデータベースと Oracle を同期できるよう、Oracle データベースをモバイル化する方法について興味がある方は、こちらをご参照ください。

([http://ftp2.iAnywhere.jp/tech/oracle\\_ml30.pdf](http://ftp2.iAnywhere.jp/tech/oracle_ml30.pdf))

## SYSTEM REQUIREMENTS システム要件

- SQL Anywhere 11.0.1 またはそれ以降
- Oracle Database 11g R1 サーバー
- Oracle Database 11g R1 クライアント (ODBC)
- [Oracle サンプルスキーマ](#) (通常 Oracle データベースとともにインストールされます)

データの移行について解説するにあたり、ここでは version 11g R1 を使用していますが、他のバージョンの Oracle データベースにも適用することが可能です。

## SQL ANYWHERE のデータ移行機能

他のソースから SQL Anywhere へデータを移行する場合、その方法には以下の 2 つのどちらかの選択肢があります。[データベース移行ウィザード](#) または [リモートデータアクセス](#) 機能です。

### データベース移行ウィザード

データベース移行ウィザードは、Oracle データベースから SQL Anywhere へのマイグレーションを実行する方法として最もシンプルなものです。このウィザードを使用すると、データベースに格納されている主キーと外部キー、全ての行を含めたテーブルを移行することができます。また、このウィザードを使用することで、Oracle のテーブルにユーザー定義のデータ

タイプ（抽象データタイプ）でない限りは、スキーマとデータが移行されます。Oracle と SQL Anywhere では、ユーザー定義のデータタイプの実装が大きく異なるため、データベース移行ウィザードでは、Oracle の抽象データタイプを、相当する SQL Anywhere のものに適切にマッチさせることはできません。このようなケースに直面した場合は、次のセクションで説明するリモートデータアクセスの機能を使用することで、移行が可能です。

このホワイトペーパーでは、データベース移行ウィザードを使用して、Oracle Human Relations (HR) サンプルスキーマを SQL Anywhere へマイグレーションします。

## リモートデータアクセス

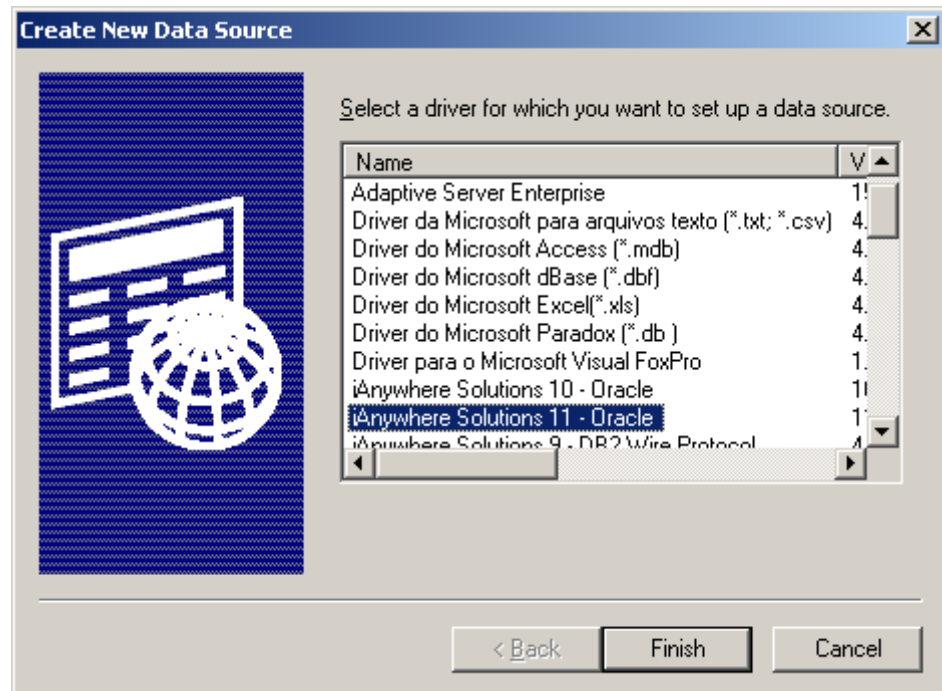
リモートデータアクセス (Remote Data Access 、 RDA) は、他のリレーショナルデータベースから SQL Anywhere へ、スキーマとデータをインポートするためのベースとなるメカニズムです。実際、RDA は、データベース移行ウィザードにおいても、データ移行プロセスを容易にするために使用されています。RDA を行う場合には、SQL Anywhere データベース内で「リモートサーバー」を定義します。次に、Oracle データベースにマッピングするリモートテーブルを表す「プロキシテーブル」を作成します。そして、プロキシテーブルを使って、Oracle データベースに対してクエリを投げ、SQL Anywhere テーブルに結果セットをインポートします。

# ORACLE サンプルスキーマを移行する

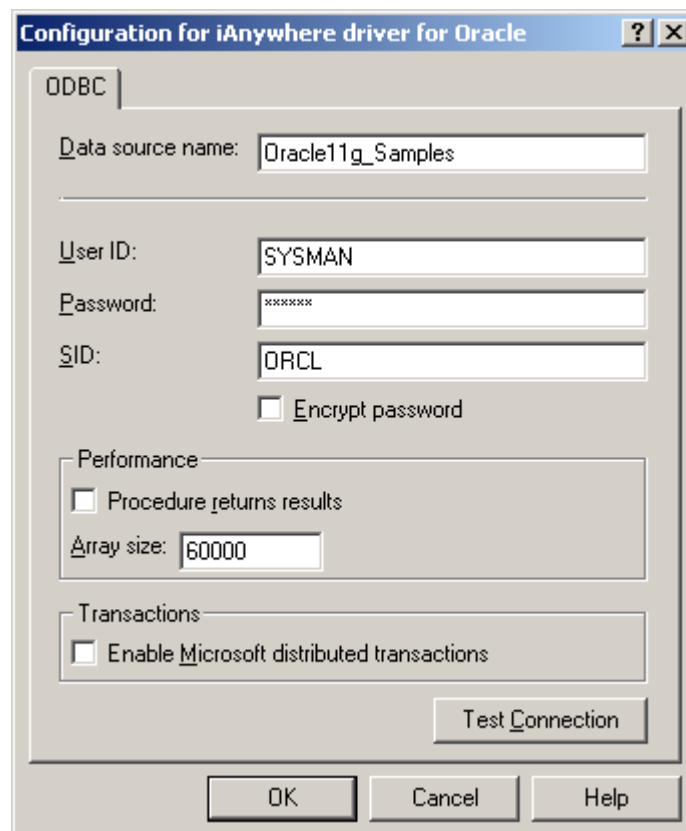
## ODBC データソースを作成する

データベース移行ウィザードとリモートデータアクセスの機能には、Oracle データベースの ODBC データソースが必要です。これには、2 種類の ODBC ドライバーの選択肢があります：一つは、Oracle ODBC クライアントで、もう一つは iAnywhere Solutions Oracle ドライバー（現 SQL Anywhere Oracle ドライバー、SQL Anywhere とともにインストールされます）です。どちらのケースでも、Oracle Net Manager アプレットを設定することで Oracle サンプルスキーマに確実にアクセスできることを確認してください。このホワイトペーパーでは、iAnywhere Solutions Oracle ドライバーを使用して、Oracle データベースサーバーにアクセスします。

1. ODBC Administrator を起動して、新たなユーザーと System DSN を追加してください。「iAnywhere Solutions 11 – Oracle」ドライバを使用します。



2. 「Oracle11g\_Samples」と入力し、Oracle データベースにログインするために必要なクレデンシャルを入力してください。HR、OE、SH のサンプルスキーマにアクセス可能な user ID を入れてください。.



3. **OK** をクリックし、**Finish** をクリックして ODBC DSN を保存します。

#### 新しい SQL ANYWHERE データベースを作成する

SQL Anywhere 管理プログラムである「Sybase Central」を使用してこのホワイトペーパーの残りのステップを実行します。

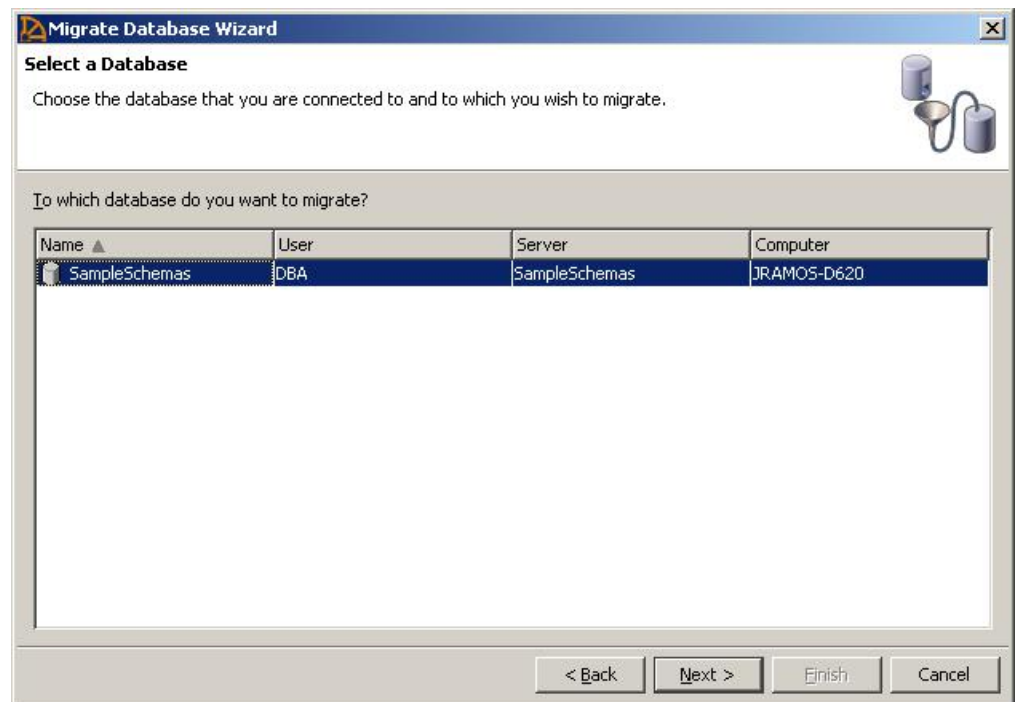
1. Sybase Central を起動します：以下の順でクリックします Start → Programs → SQL Anywhere 11 → Sybase Central
2. Tools メニューから、*SQL Anywhere | Create Database* を選択して、データベース作成ウィザードを起動します。
3. Welcome ページの **Next** をクリックします。
4. 「Select a database on this computer」を選択して **Next** をクリックします。
5. SQL Anywhere データベースファイルの名前と場所を指定します。例えば：  
C:¥MigrateToSA¥SampleSchemas.db.
6. データベース作成ウィザードでは、新しく SQL Anywhere データベースを作成するにあたり、例えば、暗号化の設定、ページサイズ、データベースの照合順など多数のオプションが用意されています。ここでは、新しいデータベースに対して追加のオプションは必要ないので、シンプルにデフォルトの設定を採用し、**Finish** をクリックしてウィザードを終了します。
7. Sybase Central で新しいデータベースを作成します。オペレーションが完了したら、**Close** をクリックします。

これで、新しく作成した SQL Anywhere データベース「SampleSchemas.db」に接続され、移行プロセスを開始することができます。

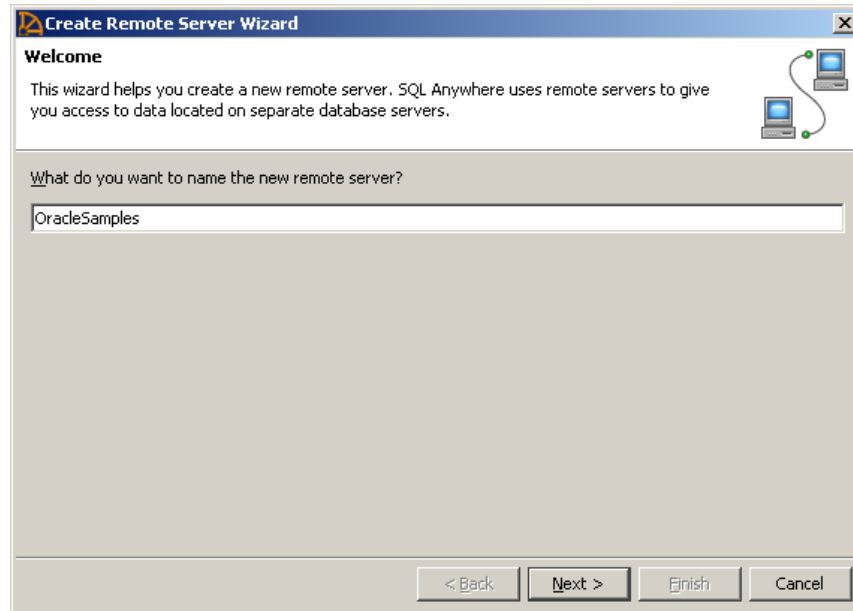
#### HR スキーマを移行する

データベース作成ウィザードを使用して HR スキーマを移行するのは、とても簡潔です。

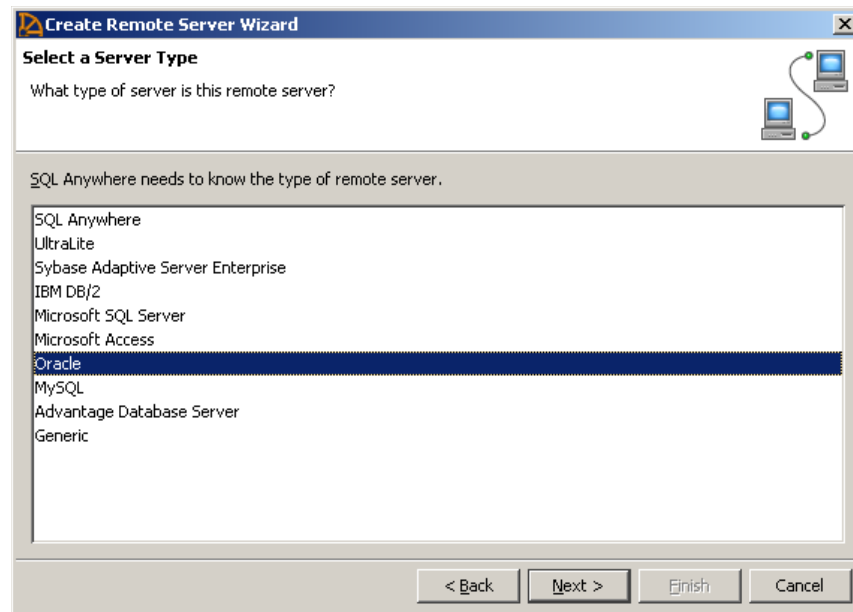
1. Tools メニューから、*SQL Anywhere | Migrate Database* を選択して、データベース作成ウィザードを起動します。
2. Welcome ページの **Next** をクリックします。
3. 「SampleSchemas」データベースを選択して、**Next** をクリックします。



4. Oracle データソースのリモートサーバーを作成する必要があります。 **Create Remote Server Now** をクリックして、リモート・サーバ作成ウィザードを起動します。
5. リモートサーバーの名前を指定します。例えば「OracleSamples」など。 **Next** をクリックします。

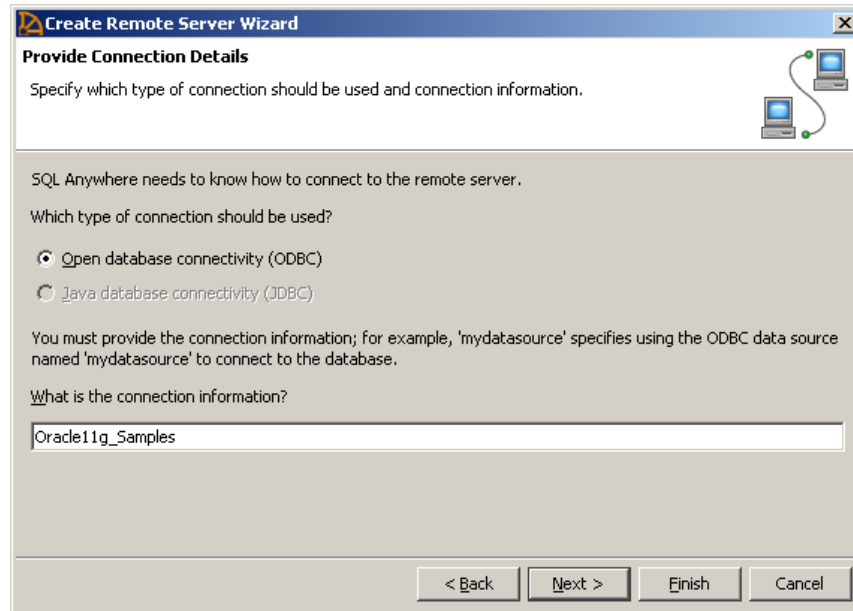


6. リモートサーバータイプのリストから Oracle を選択します。SQL Server、DB2、MySQL などの他のデータベースソースからも移行が可能です。**Next** をクリックします。

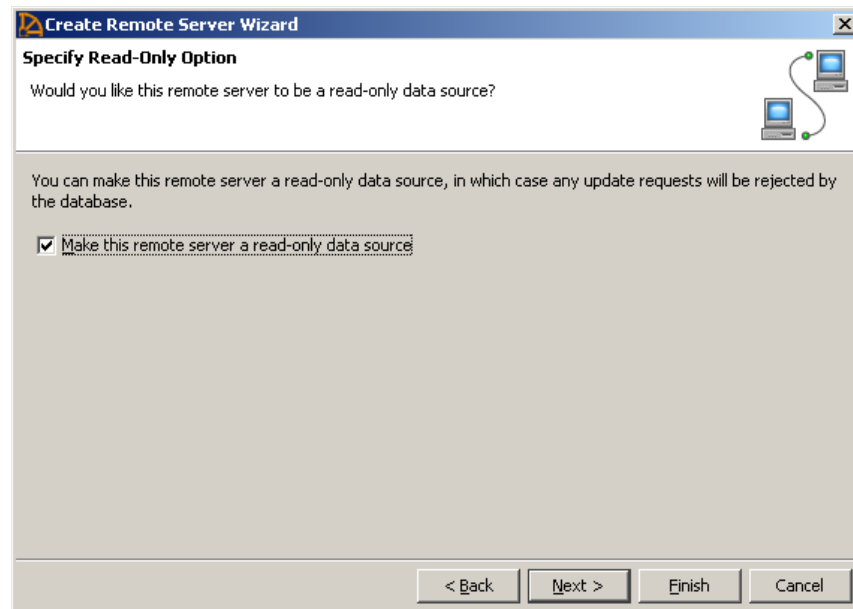


7. 接続タイプとして Select ODBC を選択します。接続情報には、先に作成した ODBC データソース名 c「Oracle11g\_Samples」を入力します。**Next** をクリックします。。





8. ここでは、Oracle データベースをアップデートする必要はないため、「Make this remote server a read-only data source」をクリックして、**Next** をクリックします。.



9. SQL Anywhere は、Oracle データベース内のサンプルスキーマにアクセスするために、外部ログインが必要です。「Create an external login for the current user」をクリックして、HR、OE SH にアクセスするための適切なクレデンシャルを入力してください。自由に接続をテストしてみてください。**Finish** をクリックします。.

**Create Remote Server Wizard**

**Specify an External Login**

Specify the username and password that will be used to connect to the remote server.

You are connected to this database as user 'DBA'. If the remote server does not define a user 'DBA' with the same password as defined in this database, then in order to connect to the remote server, you need to create an external login to assign an alternate login name and password for this user.

☒ **Create an external login for the current user**

Login name:

Password:

Confirm password:

You can test the remote server to ensure that the information provided results in a proper connection.

Click Finish to create the remote server.

< Back   Next >   Finish   Cancel

10. データベース移行ウィザードに戻ります。新しく作成した「OracleSamples」リモートサーバー を選択し、**Next** をクリックします。Sybase Central が Oracle データベースサーバーと通信する間数分かかることがあります。

**Migrate Database Wizard**

**Select a Remote Server**

Choose the remote server that you wish to migrate. You may create the remote server now, if needed.

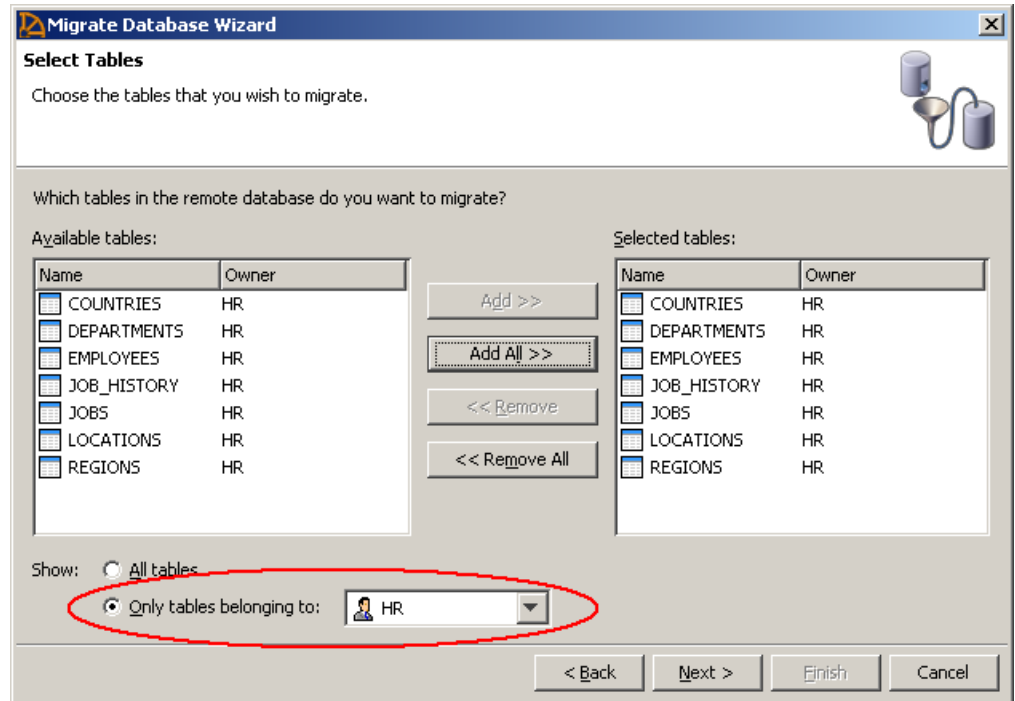
From which remote server do you want to migrate?

Name	Server Type	Conn. Type	Connection Info.
OracleSamples	Oracle	ODBC	Oracle11g_Samples

Optionally, what is the name of the remote database from which you want to migrate?

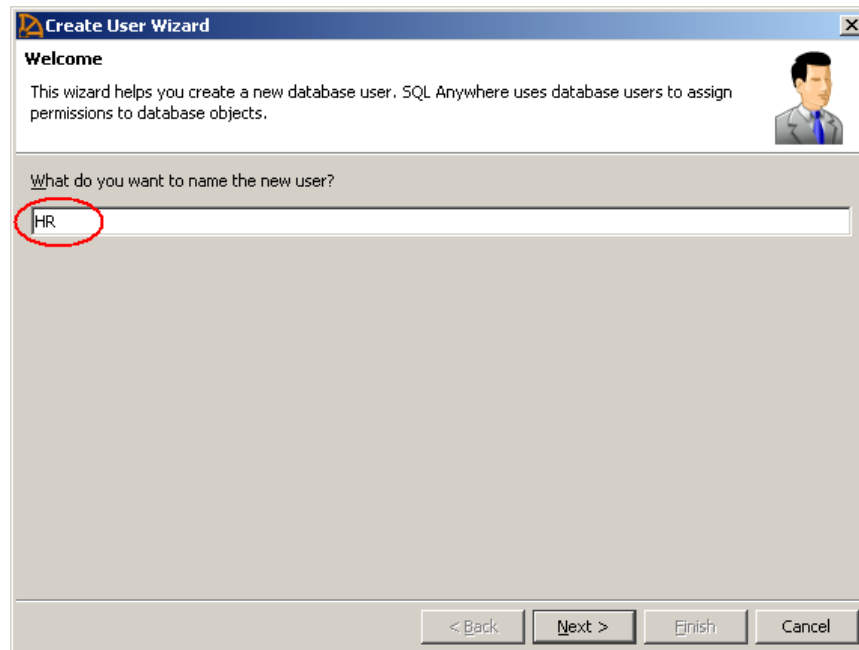
< Back   Next >   Finish   Cancel

11. Oracle データベースの全てのテーブルのリストが表示されます。「HR」に属するテーブルだけを選択し、**Add All** をクリックして、**Next** をクリックします。

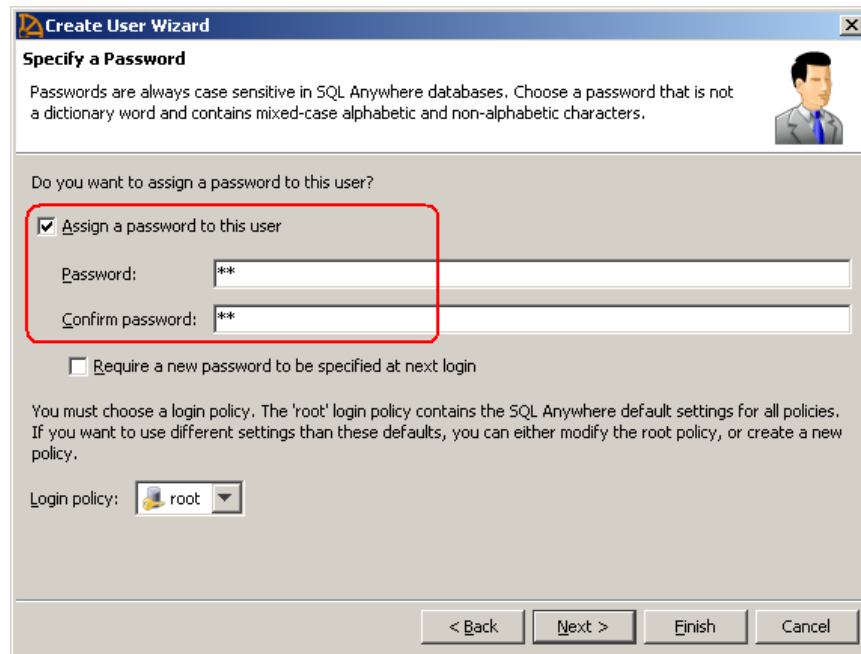


12. 特定のユーザーのテーブルを移行します。この例では、異なるスキーマの SQL Anywhere ユーザーを作成しようとしています。 **Create User Now** をクリックします。ユーザ作成ウィザードが表示されます。

13. 新しいユーザー名を「HR」とし、 **Next** をクリックします。



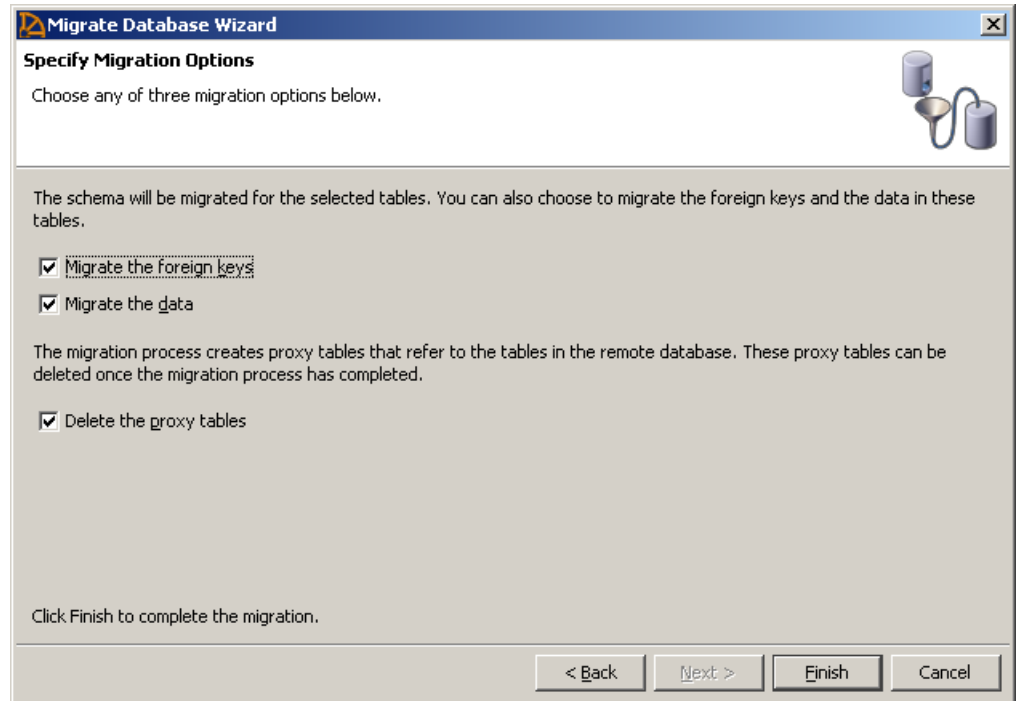
14. HR ユーザーにパスワードをアサインします。例えば、「hr」など。ここではデフォルトの設定を使用するので、**Finish** をクリックしてウィザードを終了します。



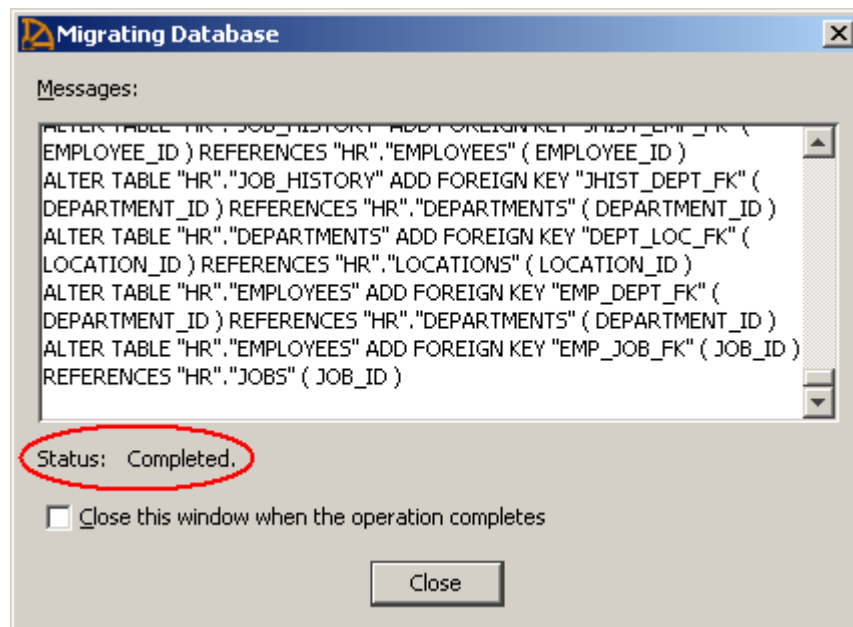
The screenshot shows the 'Create User Wizard' dialog box, specifically the 'Specify a Password' step. The title bar reads 'Create User Wizard'. Below the title bar, the section is titled 'Specify a Password'. A note states: 'Passwords are always case sensitive in SQL Anywhere databases. Choose a password that is not a dictionary word and contains mixed-case alphabetic and non-alphabetic characters.' To the right of this note is a small icon of a man in a suit. Below the note, a question asks: 'Do you want to assign a password to this user?'. There are two options: a checked checkbox labeled 'Assign a password to this user' and an unchecked checkbox labeled 'Require a new password to be specified at next login'. Below these checkboxes are two text input fields: 'Password:' and 'Confirm password:', both containing three asterisks (\*\*). Below these fields is a section for 'Login policy' with a dropdown menu currently set to 'root'. At the bottom of the dialog are four buttons: '< Back', 'Next >', 'Finish', and 'Cancel'. A red rectangle highlights the 'Assign a password to this user' checkbox and the password input fields.

15. データベース移行ウィザードに戻り、新しく作成したユーザー「HR」を選択して、**Next** をクリックします。

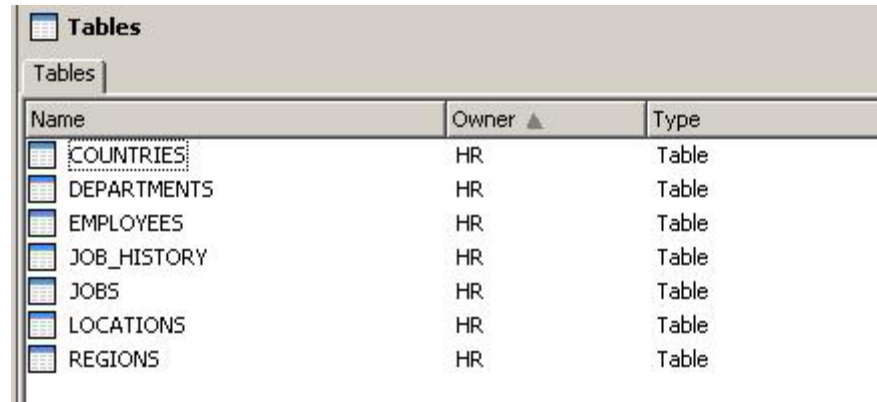
16. 外部キーとデータを移行します。移行後、アクセスしたい場合には生成されたプロキシテーブルをキープすることもできます。全てのオプションをクリックして、**Finish** をクリックします。



17. Sybase Central は、Oracle データベーススキーマをチェックし、適切なテーブルと外部キーレレーションを生成します。その後、Sybase Central は、Oracle インスタンスから SQL Anywhere データベースヘデータをインポートします。メッセージダイアログは、Sybase Central が実行する異なる SQL 文を表示します。移行が成功したら、**Close** をクリックします。



18. これで、Sybase Central 内に生成したテーブルを見ることができます。右側の窓で、「Tables」をダブルクリックします。HR Oracle スキーマに属するテーブルが現れます。SQL Anywhere では、これらは HR ユーザーに属します。



Name	Owner	Type
COUNTRIES	HR	Table
DEPARTMENTS	HR	Table
EMPLOYEES	HR	Table
JOB_HISTORY	HR	Table
JOBS	HR	Table
LOCATIONS	HR	Table
REGIONS	HR	Table

19. テーブル「EMPLOYEES」をダブルクリックし、異なるタブをナビゲートして、移行されたスキーマとデータをチェックします。

- Columns タブは、テーブルのカラム定義を表示します (主キー、名前、データタイプ、サイズ、など)。
- Constraints and Referencing Constraints タブは、外部キーと参照テーブルを表示します。
- Indexes タブは、テーブル内の全てのインデックスを表示します。デフォルトでは、インデックスはクラスタされていません。
- Data タブは、Oracle データベースから移行された全ての行を表示します。

EMPLOYEES (HR)

Columns

Constraints

Referencing Constraints

Indexes

Text Indexes

Triggers

Dependent Views

Data

	EMPLOYEE_ID	FIRST_NAME	LAST_NAME	EMAIL	PHONE_NUMBER	HIRE_DATE	JOB_ID
1	100	Steven	King	SKING	515.123.4567	1987-06-17 00:00:00.000	AD_PRES
2	101	Neena	Kochhar	NKOCHHAR	515.123.4568	1989-09-21 00:00:00.000	AD_VP
3	102	Lex	De Haan	LDEHAAN	515.123.4569	1993-01-13 00:00:00.000	AD_VP
4	103	Alexander	Hunold	AHUNOLD	590.423.4567	1990-01-03 00:00:00.000	IT_PROG
5	104	Bruce	Ernst	BERNST	590.423.4568	1991-05-21 00:00:00.000	IT_PROG
6	105	David	Austin	DAUSTIN	590.423.4569	1997-06-25 00:00:00.000	IT_PROG
7	106	Valli	Pataballa	VPATABAL	590.423.4560	1998-02-05 00:00:00.000	IT_PROG
8	107	Diana	Lorentz	DLORENTZ	590.423.5567	1999-02-07 00:00:00.000	IT_PROG
9	108	Nancy	Greenberg	NGREENBE	515.124.4569	1994-08-17 00:00:00.000	FI_MGR
10	109	Daniel	Faviet	DFAVIET	515.124.4169	1994-08-16 00:00:00.000	FI_ACCOUNT

これで、Oracle HR スキーマの SQL Anywhere 版ができました。このスキーマのテーブル内のカラムは、データベース移行ウィザードがマイグレーションを簡単に実行できるようにシンプルなデータタイプを使用して定義されています。抽象データタイプで定義されているカラムを取り扱う場合は、データベース移行ウィザードではなく、SQL Anywhere の Remote Data Access 機能を使用してください。



---

# Oracle から SQL Anywhere へのマイグレーションガイド

## ver. 1.1

アイエニウェアソリューションズ株式会社  
セールスエンジニアリング部

---



改定履歴

版	改定日付	内容
1.0	2007/10/19	初版
1.1	2008/10/29	カテゴリの細分化、関数／シンタックス等の追加対応

## 目次

はじめに .....	4
1. SQL .....	5
1. 1. データ型 .....	5
1. 1. 1. スカラーデータ型 .....	5
1. 1. 2. %TYPE 属性 .....	6
1. 1. 3. %ROWTYPE 属性 .....	7
1. 1. 4. RECORD 型 .....	8
1. 1. 5. TABLE 型 .....	9
1. 1. 6. REF CURSOR 型 .....	10
1. 2. 変数と定数 .....	11
1. 2. 1. 変数の宣言と代入 .....	11
1. 2. 2. 定数の宣言と代入 .....	11
1. 2. 3. デフォルト値の代入 .....	11
1. 3. 関数 .....	12
1. 3. 1. ADD_MONTHS .....	12
1. 3. 2. DECODE .....	13
1. 3. 3. MONTHS_BETWEEN .....	13
1. 3. 4. NVL .....	14
1. 3. 5. SYSDATE .....	14
1. 3. 6. TO_CHAR .....	15
1. 3. 7. TO_DATE .....	16
1. 3. 8. TO_NUMBER .....	16
1. 4. シンタックス .....	17
1. 4. 1. 結合演算子 .....	17
1. 4. 2. NULL 文 .....	17
1. 4. 3. テーブルのパーティションデータ一括削除 .....	18
1. 4. 4. INSERT コマンドの NOLOGGING 句 .....	18
1. 4. 5. UPDATE コマンドの構文 .....	18
1. 4. 6. コンソールに文字列を表示 .....	19
1. 4. 7. EXECUTE IMMEDIATE 文の INTO 句 .....	19
1. 5. プロシージャとファンクション .....	20
1. 5. 1. プロシージャの作成 .....	20
1. 5. 2. ファンクションの作成 .....	21
1. 5. 3. パッケージの作成 .....	22
1. 6. 制御構造 .....	23
1. 6. 1. IF-THEN-ELSIF-ELSE 文 .....	23
1. 6. 2. 基本 LOOP 文 (EXIT WHEN) .....	24
1. 6. 3. FOR-LOOP 文 .....	24
1. 6. 4. GOTO 文 .....	25
1. 7. カーソル .....	26

1. 7. 1. カーソル宣言 .....	26
1. 7. 2. カーソル FOR ループ .....	26
1. 7. 3. パラメータ付きカーソル .....	27
1. 8. 例外処理 .....	28
1. 8. 1. 例外ハンドラ .....	28
1. 8. 2. ユーザ定義例外 .....	29

## はじめに

この文書は Oracle データベースを SQL Anywhere にマイグレーションする場合に必要な変換項目についてまとめたものです。

なお、Oracle および SQL Anywhere のバージョンは以下の通りです。

- SQL Anywhere 12
- Oracle 10g

## 1. SQL

本章では Oracle と SQL Anywhere で使用法が異なる SQL について、以下の順で説明します。

1. データ型
2. 変数と定数
3. 関数
4. シンタックス
5. プロシージャとファンクション
6. 制御構造
7. カーソル処理
8. 例外処理

### 1. 1. データ型

Oracle と SQL Anywhere のデータ型の違いについて、以下に説明します。

#### 1. 1. 1. スカラーデータ型

Oracle と SQL Anywhere で異なるスカラーデータ型を次の表に示します。

Oracle データ型	SQL Anywhere データ型	備考
VARCHAR2[(n)]	VARCHAR(n)	可変長文字列型
NUMBER(1), NUMBER(2)	TINYINT	符号なし整数値型 (1 バイト)
NUMBER(1) ~ NUMBER(4)	SMALLINT	整数値型 (2 バイト)
NUMBER(5) ~ NUMBER(9)	INTEGER	整数値型 (4 バイト)
NUMBER(10) ~ NUMBER(18)	BIGINT	整数値型 (8 バイト)
DATE	DATETIME	日付時刻型
DATE	DATE	日付型 (時刻がデフォルトの場合)

### 1. 1. 2. %TYPE 属性

Oracle データ型	%TYPE	
変更内容	Oracle の %TYPE 属性を使用したデータ型の定義は SQL Anywhere ではサポートされていないため、適切なデータ型を定義するように対応	
使用例 1	dept 表の deptno 列の値を格納する変数を宣言	
	Oracle	<pre>DECLARE   var dept.deptno%TYPE; BEGIN   SELECT deptno INTO var FROM dept         WHERE loc = 'NEW YORK';   DBMS_OUTPUT.PUT_LINE(var); END; /</pre>
	SQL Anywhere	<pre>BEGIN   DECLARE var int;   SELECT deptno INTO var FROM dept         WHERE loc = 'NEW YORK';   MESSAGE var TO CLIENT; END</pre>

SQL Anywhere17 から%TYPE が使用可能になりました。

DECLARE var deptno%TYPE;

として列型の参照指定が可能です。

### 1. 1. 3. %ROWTYPE 属性

Oracle データ型	%ROWTYPE	
変更内容	Oracle の %ROWTYPE 属性を使用した列名の参照は SQL Anywhere ではサポートされていないため、個別に変数を定義してデータを格納するように対応	
使用例 1	dept 表の全列の値を格納する変数を定義	
	Oracle	<pre>DECLARE   d_row dept%ROWTYPE; BEGIN   SELECT deptno,dname,loc INTO d_row FROM dept                                 WHERE deptno = 10;   DBMS_OUTPUT.PUT_LINE(d_row.dname); END; /</pre>
	SQL Anywhere	<pre>BEGIN   DECLARE deptno int;   DECLARE dname varchar(12);   DECLARE loc varchar(13);   SELECT deptno,dname,loc INTO deptno,dname,loc   FROM dept   WHERE deptno = 10;   MESSAGE dname TO CLIENT; END</pre>

SQL Anywhere17 から%ROWTYPE が使用可能になりました。

Oracle と同様に

```
DECLARE d_row dept%ROWTYPE
```

として行型を定義し d\_row.<カラム名> で列の参照が可能です。

## 1. 1. 4. RECORD 型

Oracle データ型	RECORD 型	
変更内容	Oracle の RECORD 型は SQL Anywhere ではサポートされていないため、個別に変数を定義してデータを格納するように対応	
使用例 1	2 つのフィールドを持つ RECORD 型を定義	
	Oracle	<pre> DECLARE   TYPE rec IS RECORD     (a NUMBER(2), b VARCHAR2(6));   record_type rec; BEGIN   record_type.a := 1;   record_type.b := 'Part1';   DBMS_OUTPUT.PUT_LINE(record_type.a    ' '   record_type.b); END; / </pre>
	SQL Anywhere	<pre> BEGIN   DECLARE record_type_a int;   DECLARE record_type_b varchar(6);   SET record_type_a = 1;   SET record_type_b = 'Part1';   MESSAGE record_type_a    ' ' record_type_b TO CLIENT; END </pre>



## 1. 1. 5. TABLE 型

Oracle データ型	TABLE 型	
変更内容	Oracle の TABLE 型は SQL Anywhere ではサポートされていないため、個別に変数を定義してデータを格納するように対応	
使用例 1	1 つのフィールドを持つ TABLE 型を定義	
	Oracle	<pre>DECLARE     TYPE tab IS TABLE OF VARCHAR2(10)     INDEX BY PLS_INTEGER;     col_table tab;  BEGIN     col_table(1) := 'Part1';     col_table(2) := 'Part2';     col_table(3) := 'Part3';     DBMS_OUTPUT.PUT_LINE(col_table(1));     DBMS_OUTPUT.PUT_LINE(col_table(2));     DBMS_OUTPUT.PUT_LINE(col_table(3));  END; /</pre>
	SQL Anywhere	<pre>BEGIN     DECLARE col_table_1 VARCHAR(10);     DECLARE col_table_2 VARCHAR(10);     DECLARE col_table_3 VARCHAR(10);     SET col_table_1 = 'Part1';     SET col_table_2 = 'Part2';     SET col_table_3 = 'Part3';     MESSAGE col_table_1 TO CLIENT;     MESSAGE col_table_2 TO CLIENT;     MESSAGE col_table_3 TO CLIENT;  END</pre>

## 1. 1. 6. REF CURSOR 型

Oracle データ型	REF CURSOR 型	
変更内容	Oracle の REF CURSOR 型はサポートしていないため、SELECT 文単位にカーソルを定義するように対応	
使用例 1	Oracle の REF CURSOR 型は未サポート	
	Oracle	<pre> CREATE OR REPLACE PROCEDURE   open_data(dept_no NUMBER) IS   TYPE cv_type IS REF CURSOR;   cur_cv cv_type;   emp_rec emp%ROWTYPE;   dept_rec dept%ROWTYPE; BEGIN   OPEN cur_cv FOR SELECT * FROM emp                                 WHERE deptno = dept_no;    LOOP     FETCH cur_cv INTO emp_rec;     EXIT WHEN cur_cv%NOTFOUND;     DBMS_OUTPUT.PUT_LINE('EMP '    emp_rec.ename);   END LOOP;   CLOSE cur_cv;   OPEN cur_cv FOR SELECT * FROM dept                                 WHERE deptno = dept_no;    LOOP     FETCH cur_cv INTO dept_rec;     EXIT WHEN cur_cv%NOTFOUND;     DBMS_OUTPUT.PUT_LINE('DEPT '    dept_rec.dname);   END LOOP;   CLOSE cur_cv; END; / </pre>
	SQL Anywhere	未対応

## 1. 2. 変数と定数

Oracle と SQL Anywhere の変数／定数の定義の違いについて、以下に説明します。

### 1. 2. 1. 変数の宣言と代入

Oracle シンタックス	変数の宣言と代入	
変更内容	Oracle の変数の宣言と代入を SQL Anywhere のシンタックスに変換する	
使用例 1	Oracle の変数の宣言と代入を SQL Anywhere のシンタックスに変換	
	Oracle	DECLARE var1 CHAR(2) := '01';
	SQL Anywhere	DECLARE var1 CHAR(2); SET var1 = '01';

### 1. 2. 2. 定数の宣言と代入

Oracle シンタックス	定数の宣言と代入	
変更内容	Oracle の定数の宣言と代入を SQL Anywhere のシンタックスに変換する	
使用例 1	Oracle の定数の宣言と代入を SQL Anywhere のシンタックスに変換	
	Oracle	DECLARE con1 CONSTANT CHAR(2) := '01';
	SQL Anywhere	DECLARE con1 CHAR(2); SET con1 = '01';

### 1. 2. 3. デフォルト値の代入

Oracle シンタックス	変数へのデフォルト値の代入	
変更内容	SQL Anywhere ではデフォルト値はサポートしていないため、変数の宣言後に値を代入することで対応	
使用例 1	Oracle のデフォルト値の代入を SQL Anywhere のシンタックスに変換	
	Oracle	DECLARE var1 VARCHAR2(10) DEFAULT 'SCOTT';
	SQL Anywhere	DECLARE var1 VARCHAR(10); SET var1 = 'SCOTT';

## 1. 3. 関数

Oracle と SQL Anywhere で使用法が異なる関数について、以下に説明します。

### 1. 3. 1. ADD\_MONTHS

Oracle 関数名	ADD_MONTHS	
説明	日付に月数を加算	
SQL Anywhere 対応関数名	DATE_ADD	
構文	Oracle	ADD_MONTHS(日付式, 月数)
	SQL Anywhere	DATEADD (日付要素(*1), 加算する値, 日付値)
使用例 1	col1 に格納されている日付から 12 ヶ月前の日付値を求める	
	Oracle	ADD_MONTHS(TO_DATE(col1,'yyyymmdd'),-12)
	SQL Anywhere	DATEADD(MONTH,-12,col1)

(\*1) 指定可能な日付要素は下表の通り。

日付要素	省略形	意味	有効値
Year	yy	年	0001 ~ 9999
Quarter	qq	四半期	1 ~ 4
Month	mm	月	1 ~ 12
Week	wk	週	1 ~ 54
Day	dd	日	1 ~ 31
Dayofyear	dy	年間通算日（年始からの日数）	1 ~ 366
Weekday	dw	日曜日～月曜日	1 ~ 7
Hour	hh	時	0 ~ 23
Minute	mi	分	0 ~ 59
Second	ss	秒	0 ~ 59
Millisecond	ms	ミリ秒	0 ~ 999
Calyearofweek	cyr		
Calweekofyear	cwk		
Caldayofweek	cdw		

### 1. 3. 2. DECODE

Oracle 関数名	DECODE	
説明	値の変換	
SQL Anywhere 対応関数名	CASE	
構文	Oracle	DECODE(式, 検索値 1, 戻り値 1 [, 検索値 2, 戻り値 2, ...] [, 規定値])
	SQL Anywhere	CASE WHEN 式 THEN 式 [WHEN 式 THEN 式 ...] [ELSE 式] END
使用例 1	col1 が NULL なら “不明” を NULL でなければ col2 の値を返す	
	Oracle	DECODE(col1, NULL, '不明', col2)
	SQL Anywhere	CASE WHEN col1 IS NULL THEN '不明' col2 END

### 1. 3. 3. MONTHS\_BETWEEN

Oracle 関数名	MONTHS_BETWEEN	
説明	2 つの日付の間の月数を求める	
SQL Anywhere 対応関数名	DATEDIFF	
構文	Oracle	MONTHS_BETWEEN(日付 1, 日付 2)
	SQL Anywhere	DATEDIFF(日付要素(*1), 開始日付, 終了日付)
使用例 1	date1 カラムの日付から date2 カラムの日付までの月数を返す	
	Oracle	MONTH_BETWEEN(date1, date2)
	SQL Anywhere	DATEDIFF(month, date1, date2)

(\*1) 指定可能な日付要素については「1. 3. 1. ADD\_MONTHS」の章を参照

### 1. 3. 4. NVL

Oracle 関数名	NVL	
説明	NULL 値を別の値に変換	
SQL Anywhere 対応関数名	ISNULL	
構文	Oracle	NVL(式, 値)
	SQL Anywhere	ISNULL(式, 値)
使用例 1	col1 カラムの値が NULL 値ならば “0” を返す	
	Oracle	NVL(col1, 0)
	SQL Anywhere	ISNULL(col1,0)

### 1. 3. 5. SYSDATE

Oracle 関数名	SYSDATE	
説明	現在の日時を求める	
SQL Anywhere 対応関数名	NOW, GETDATE	
構文	Oracle	SYSDATE
	SQL Anywhere	NOW() GETDATE()
使用例 1	現在の日時を求める	
	Oracle	SYSDATE
	SQL Anywhere	NOW() GETDATE()

## 1. 3. 6. TO\_CHAR

Oracle 関数名	TO_CHAR	
説明	数値／日付を文字列型に変換	
SQL Anywhere 対応関数名	CONVERT	
構文	Oracle	TO_CHAR(式 [, フォーマット [, NLS パラメータ] ])
	SQL Anywhere	CONVERT(データ型 [(長さ)], 値 [, 日付形式(*2)])
使用例 1	数値を文字列型に変換する	
	Oracle	TO_CHAR(12345)
	SQL Anywhere	CONVERT(CHAR(20), 12345)
使用例 2	今日の日付をフォーマットを指定して文字列型に変換する	
	Oracle	TO_CHAR(SYSDATE, 'MON DD, YYYY')
	SQL Anywhere	CONVERT(CHAR(20), NOW(), 107)

(\*2) 指定可能な日付形式の一覧は下表の通り。

世紀なし (yy)	世紀あり(yyyy)	出力形式
-	0 または 100	mmm dd yyyy hh:nnAM ( または PM)
1	101	mm/dd/yy[yy]
2	102	[yy]yy.mm.dd
3	103	dd/mm/yy[yy]
4	104	dd.mm.yy[yy]
5	105	dd-mm-yy[yy]
6	106	dd mmm yy[yy]
7	107	mmm dd, yy[yy]
8	108	hh:nn:ss
-	9 または 109	mmm dd yyyy hh:nn:ss:sssAM ( または PM)
10	110	mm-dd-yy[yy]
11	111	[yy]yy/mm/dd
12	112	[yy]yyymmdd
13	113	d mmm yyyy hh:nn:ss:sss (24 時間形式、欧州デフォルト + ミリ秒、年 4 桁)
14	114	hh:nn:ss (24 時間形式)
20	120	yyyy-mm-dd hh:nn:ss (24 時間表示、ODBC 標準、年 4 桁)
21	121	yyyy-mm-dd hh:nn:ss:sss (24 時間表示、ODBC 標準 + ミリ秒、年 4 桁)
-	365	yyyyjjj ( 文字列または整数で指定。jjj は年内のユリウス日の番号 (1 ~ 366))

### 1. 3. 7. TO\_DATE

Oracle 関数名	TO_DATE	
説明	文字列を日付型に変換	
SQL Anywhere 対応関数名	DATE	
構文	Oracle	TO_DATE(文字列 [, フォーマット [, NLS パラメータ]])
	SQL Anywhere	DATE(文字列)
使用例 1	文字列を日付型に変換する	
	Oracle	TO_DATE('20071010')
	SQL Anywhere	DATE('20071010')

### 1. 3. 8. TO\_NUMBER

Oracle 関数名	TO_NUMBER	
説明	文字列を数値型に変換	
SQL Anywhere 対応関数名	CONVERT	
構文	Oracle	TO_NUMBER(文字列 [, フォーマット [, NLS パラメータ]])
	SQL Anywhere	CONVERT(データ型 [(長さ)], 値 [, 日付形式(*2)])
使用例 1	文字列を数値型に変換する	
	Oracle	TO_NUMBER('12345')
	SQL Anywhere	CONVERT(INTEGER, '12345')

(\*2) 指定可能な日付形式については「1. 3. 6. TO\_CHAR」の章を参照



## 1. 4. シンタックス

Oracle と SQL Anywhere で使用法が異なるシンタックスについて、以下に説明します。

### 1. 4. 1. 結合演算子

Oracle シンタックス	外部結合(+)	
変更内容	Oracle 固有の記述方式を ANSI 標準構文へ変換	
使用例 1	テーブル t1 と t2 を col1, col2 カラムを元に左外部結合を行う	
	Oracle	<pre>FROM tbl1, tbl2 WHERE tbl1.col1 = tbl2.col1(+)       AND tbl1.col2 = tbl2.col2(+)</pre>
	SQL Anywhere	<pre>FROM tbl1 LEFT OUTER JOIN tbl2 ON  tbl1.col1 = tbl2.col1    AND tbl1.col2 = tbl2.col2</pre>

### 1. 4. 2. NULL 文

Oracle シンタックス	NULL 文	
変更内容	SQL Anywhere では Oracle の NULL 文はサポートしていないため、コメントアウトするなどの対応が必要	
使用例 1	Oracle の NULL 文をコメントアウト	
	Oracle	<pre>IF var = 'OK' THEN   DBMS_OUTPUT.PUT_LINE('OK'); ELSE   NULL; END IF;</pre>
	SQL Anywhere	<pre>IF var = 'OK' THEN   MESSAGE 'OK' TO CLIENT; ELSE   --NULL; END IF;</pre>

## 1. 4. 3. テーブルのパーティションデータ一括削除

Oracle シンタックス	テーブルのパーティションデータ一括削除	
変更内容	Oracle のパーティションデータ一括削除（TRUNCATE PARTITION）文をパーティションキーによる DELETE に変更（SQL Anywhere ではパーティションはサポートされていないため）	
使用例 1	Oracle におけるパーティション・データ一括削除文を SQL Anywhere の DELETE 文に変換	
	Oracle	ALTER TABLE tbl1 TRUNCATE PARTITION partition1
	SQL Anywhere	DELETE FROM tbl1 WHERE <i>partition_key</i> = <i>value</i> ;

## 1. 4. 4. INSERT コマンドの NOLOGGING 句

Oracle シンタックス	INSERT コマンドの NOLOGGING 句	
変更内容	NOLOGGING 句を削除（SQL Anywhere では NOLOGGING モードがサポートされていないため）	
使用例 1	Oracle の NOLOGGING 句を削除	
	Oracle	INSERT INTO tbl1 NOLOGGING VALUES ( <i>xxx</i> );
	SQL Anywhere	INSERT INTO tbl1 VALUES ( <i>xxx</i> );

## 1. 4. 5. UPDATE コマンドの構文

Oracle シンタックス	UPDATE コマンド	
変更内容	ANSI 標準シンタックスへ変換	
使用例 1	Oracle の UPDATE コマンドを ANSI 標準のシンタックスに変換	
	Oracle	<pre> UPDATE TBL1   SET (col1, col2, col3)     = (SELECT col1, col2, col3 FROM tbl2       WHERE...)   WHERE tbl1.col1=xxx </pre>
	SQL Anywhere	<pre> UPDATE tbl1   SET col1= tbl2.col1     col2= tbl2.col2     col3= tbl2.col3   FROM tbl1, tbl2  WHERE tbl1.col1 = tbl2.col1     AND tbl1.col1 = <i>xxx</i> </pre>

## 1. 4. 6. コンソールに文字列を表示

Oracle シンタックス	コンソールに文字列を表示	
変更内容	Oracle の DBMS_OUTPUT.PUT_LINE 関数を SQL Anywhere の MESSAGE 文に変更	
使用例 1	コンソールに文字列 'OK' を表示	
	Oracle	<pre>BEGIN   DBMS_OUTPUT.PUT_LINE('OK'); END; /</pre>
	SQL Anywhere	<pre>BEGIN   MESSAGE 'OK' TO CLIENT; END</pre>

## 1. 4. 7. EXECUTE IMMEDIATE 文の INTO 句

Oracle シンタックス	コレクション	
変更内容	Oracle では EXECUTE IMMEDIATE 文の コレクションはサポートしていないため、個別に変数を定義してデータを出し入れするように対応	
使用例 1	Oracle の コレクションは未サポート	
	Oracle	<pre>DECLARE   TYPE tab IS TABLE OF dept%ROWTYPE   INDEX BY PLS_INTEGER;   table_rec      tab;   CURSOR dept_cl IS SELECT * FROM dept;   n NUMBER := 0;  BEGIN   FOR r IN dept_cl LOOP     n := n + 1;     table_rec(n) := r;     DBMS_OUTPUT.PUT_LINE       (table_rec(n).deptno    ' '           table_rec(n).dname    ' '           table_rec(n).loc);   END LOOP;  END; /</pre>
	SQL Anywhere	未対応

## 1. 5. プロシージャとファンクション

Oracle と SQL Anywhere におけるプロシージャおよびファンクション作成上の相違点について、以下に説明します。

### 1. 5. 1. プロシージャの作成

Oracle シンタックス	プロシージャの作成	
変更内容	Oracle のプロシージャを SQL Anywhere の形式に変換	
使用例 1	Oracle のプロシージャを SQL Anywhere の形式に変換する	
	Oracle	<pre>CREATE OR REPLACE PROCEDURE inout_sample (     no IN OUT NUMBER ) IS BEGIN     no := no + 100; END; /</pre>
	SQL Anywhere	<pre>IF EXISTS ( SELECT name FROM sysobjects             WHERE name = 'inout_sample' AND type = 'P')     DROP FUNCTION inout_sample; CREATE PROCEDURE inout_sample (     INOUT no INT ) BEGIN     SET no = no + 100; END;</pre>

## 1. 5. 2. ファンクションの作成

Oracle シンタックス	ファンクションの作成	
変更内容	Oracle のファンクションを SQL Anywhere の形式に変換	
使用例 1	Oracle のファンクションを SQL Anywhere の形式に変換する	
	Oracle	<pre>CREATE OR REPLACE FUNCTION triangle (   base IN NUMBER DEFAULT 10,   height IN NUMBER DEFAULT 20 ) RETURN NUMBER IS BEGIN   RETURN((base * height)/2); END; /</pre>
	SQL Anywhere	<pre>IF EXISTS ( SELECT name FROM sysobjects             WHERE name = 'triangle' AND type = 'P')   DROP FUNCTION triangle; CREATE FUNCTION triangle (   base INT DEFAULT 10,   IN height INT DEFAULT 20 ) RETURNS INT BEGIN   RETURN((base * height)/2); END;</pre>

### 1. 5. 3. パッケージの作成

Oracle シンタックス	パッケージの宣言	
変更内容	Oracle のパッケージをプロシージャまたはファンクションに分解 (SQL Anywhere にはパッケージという概念がないため)	
使用例 1	Oracle のプロシージャを含んだパッケージを SQL Anywhere のプロシージャに変換する	
	Oracle	<pre>CREATE OR REPLACE PACKAGE pkg1 AS     PROCEDURE proc1 (arg1 char(2) ) END pkg1; CREATE OR REPLACE PACKAGE BODY pkg1 AS     PROCEDURE proc1 (arg1 char(2) ) IS     BEGIN         . . . .     END</pre>
	SQL Anywhere	<pre>IF EXISTS ( SELECT name FROM sysobjects             WHERE name = 'proc1' AND type = 'P')     DROP PROCEDURE proc1; CREATE PROCEDURE proc1 @arg1 char(2) AS     BEGIN         . . .     END</pre>
使用例 2	Oracle のファンクションを含んだパッケージを SQL Anywhere のファンクションに変換する	
	Oracle	<pre>CREATE OR REPLACE PACKAGE pkg1 AS     FUNCTION func1 (arg1 char(2) ) END pkg1; CREATE OR REPLACE PACKAGE BODY pkg1 AS     FUNCTION func1 (arg1 char(2)) IS     BEGIN         . . . .     END</pre>
	SQL Anywhere	<pre>IF EXISTS ( SELECT name FROM sysobjects             WHERE name = 'func1' AND type = 'P')     DROP FUNCTION func1; CREATE FUNCTION func1 (arg1 char(2) ) RETURNS int     BEGIN         . . .     END</pre>

## 1. 6. 制御構造

Oracle と SQL Anywhere における制御構造の相違点について、以下に説明します。

### 1. 6. 1. IF-THEN-ELSIF-ELSE 文

Oracle シンタックス	条件制御 IF-THEN-ELSIF-ELSE 文	
変更内容	Oracle の ELSIF を ELSE IF に変更し、また IF に対応する END IF を追加	
使用例 1	Oracle の IF-THEN-ELSIF-ELSE 文を SQL Anywhere のシンタックスに変換	
	Oracle	<pre>IF var = 10 THEN     DBMS_OUTPUT.PUT_LINE('値は 10'); ELSIF var = 20 THEN     DBMS_OUTPUT.PUT_LINE('値は 20'); ELSE     DBMS_OUTPUT.PUT_LINE('値はそれ以外'); END IF;</pre>
	SQL Anywhere	<pre>IF var = 10 THEN     print '値は 10'; ELSE     IF var = 20 THEN         print '値は 20';     ELSE         print '値はそれ以外';     END IF; END IF;</pre>

### 1. 6. 2. 基本 LOOP 文 (EXIT WHEN)

Oracle シンタックス	基本 LOOP (EXIT WHEN) 文	
変更内容	SQL Anywhere では Oracle の EXIT WHEN 文をサポートしていないため、ラベルと LEAVE 文を使用することで対応	
使用例 1	Oracle の EXIT WHEN 文を SQL Anywhere の LEAVE 文を使用するように変換	
	Oracle	<pre>LOOP   EXIT WHEN c_count = 3;   DBMS_OUTPUT.PUT_LINE('OK');   c_count := c_count + 1; END LOOP;</pre>
	SQL Anywhere	<pre>lbl: LOOP   IF c_count = 3 THEN LEAVE lbl; END IF; MESSAGE 'OK' TO CLIENT; SET c_count = c_count + 1; END LOOP lbl;</pre>

### 1. 6. 3. FOR-LOOP 文

Oracle シンタックス	FOR-LOOP 文	
変更内容	SQL Anywhere では FOR-LOOP 文をサポートしていないため、ラベルと LEAVE 文を使用することで対応	
使用例 1	Oracle の FOR-LOOP 文を SQL Anywhere の LEAVE 文を使用するように変換	
	Oracle	<pre>FOR c_count IN 1..3 LOOP   DBMS_OUTPUT.PUT_LINE('OK'); END LOOP;</pre>
	SQL Anywhere	<pre>lbl: LOOP   IF c_count = 3 THEN LEAVE lbl; END IF; MESSAGE 'OK' TO CLIENT; SET c_count = c_count + 1; END LOOP lbl;</pre>



## 1. 6. 4. GOTO 文

Oracle シンタックス	GOTO 文	
変更内容	SQL Anywhere では Oracle の GOTO 文はサポートしていない	
使用例 1	Oracle の GOTO 文はそのままでは対応不可のため、どうしても必要な場合はロジックの見直しが必要	
	Oracle	<pre>GOTO ravel; DBMS_OUTPUT.PUT_LINE('処理 1'); DBMS_OUTPUT.PUT_LINE('処理 2'); &lt;&lt;ravel&gt;&gt; DBMS_OUTPUT.PUT_LINE('ラベル処理');</pre>
	SQL Anywhere	未対応

## 1. 7. カーソル

Oracle と SQL Anywhere におけるカーソル処理の相違点について、以下に説明します。

### 1. 7. 1. カーソル宣言

Oracle シンタックス	カーソル宣言	
変更内容	ANSI 標準シンタックスへ変換	
使用例 1	Oracle のカーソル宣言文を ANSI 標準のシンタックスに変換	
	Oracle	CURSOR cur1 IS SELECT * FROM tbl1
	SQL Anywhere	DECLARE cur1 CURSOR FOR SELECT * FROM tbl1

### 1. 7. 2. カーソル FOR ループ

Oracle シンタックス	カーソル FOR ループ	
変更内容	Oracle のカーソル FOR ループを SQL Anywhere のカーソル FOR ループに変換	
使用例 1	Oracle のカーソル FOR ループを SQL Anywhere のカーソル FOR ループに変換	
	Oracle	DECLARE CURSOR emp_cur IS SELECT empno, ename FROM emp WHERE deptno = 10; BEGIN FOR emp_rec IN emp_cur LOOP DBMS_OUTPUT.PUT_LINE(emp_rec.empno    ' '    emp_rec.ename); END LOOP; END; /
	SQL Anywhere	BEGIN FOR cursor_loop AS emp_cur CURSOR FOR SELECT empno,ename FROM emp WHERE deptno = 10 DO MESSAGE empno    ' '    ename TO CLIENT; END FOR; END

### 1. 7. 3. パラメータ付きカーソル

Oracle シンタックス	パラメータ付きカーソル	
変更内容	SQL Anywhere では Oracle のパラメータ付きカーソルはサポートしていないため、プロシージャレベルでパラメータ化する等の対応が必要	
使用例 1	SQL Anywhere では Oracle のパラメータ付きカーソルは未サポート	
	Oracle	<pre>DECLARE   CURSOR emp_cur(d_no NUMBER) IS     SELECT empno, ename FROM emp     WHERE deptno = d_no;   d_var NUMBER; BEGIN   d_var := &amp;DEPTNO;   FOR emp_rec IN emp_cur(d_var) LOOP     DBMS_OUTPUT.PUT_LINE(emp_rec.empno    ' '    emp_rec.ename);   END LOOP; END; /</pre>
	SQL Anywhere	未対応

## 1. 8. 例外処理

Oracle と SQL Anywhere における例外処理の相違点について、以下に説明します。

### 1. 8. 1. 例外ハンドラ

Oracle シンタックス	例外ハンドラ	
変更内容	Oracle では幾つかの一般的な内部例外 (NO_DATA_FOUND 例外など) については例外名が事前定義されているため、この例外名を使用してハンドラを作成できますが、SQL Anywhere では作成したいハンドラ全てに例外名を定義	
使用例 1	SQL Anywhere では例外ハンドラを個別に定義	
	Oracle	<pre> DECLARE   e_empno NUMBER; BEGIN   SELECT empno INTO e_empno FROM emp; EXCEPTION   WHEN too_many_rows THEN     DBMS_OUTPUT.PUT_LINE('TOO_MANY_ROWS!');   WHEN no_data_found THEN     DBMS_OUTPUT.PUT_LINE('NO_DATA_FOUND!'); END; / </pre>
	SQL Anywhere	<pre> BEGIN   DECLARE NO_DATA_FOUND EXCEPTION FOR     SQLSTATE '02000';   DECLARE TOO_MANY_ROWS EXCEPTION FOR     SQLSTATE '21000';   DECLARE e_empno INT;   DECLARE msg VARCHAR(50);   SELECT empno INTO e_empno FROM emp; EXCEPTION   WHEN NO_DATA_FOUND THEN     MESSAGE 'NO_DATA_FOUND!' TO CLIENT;   WHEN TOO_MANY_ROWS THEN     MESSAGE 'TOO_MANY_ROWS!' TO CLIENT;   WHEN OTHERS THEN     MESSAGE 'その他の例外発生' TO CLIENT; END; </pre>

## 1. 8. 2. ユーザ定義例外

Oracle シンタックス	ユーザ定義例外	
変更内容	Oracle のユーザ定義例外の宣言および呼び出しを SQL Anywhere の宣言および SIGNAL 文に変換する	
使用例 1	Oracle のユーザ定義例外を SQL Anywhere のシンタックスに変換	
	Oracle	<pre>DECLARE   no NUMBER;   err EXCEPTION; BEGIN   no := 90;   IF no &lt; 100 THEN     RAISE err;   END IF; EXCEPTION   WHEN err THEN     DBMS_OUTPUT.PUT_LINE('ユーザ定義例外！'); END; /</pre>
	SQL Anywhere	<pre>BEGIN   DECLARE no INT;   DECLARE err EXCEPTION FOR SQLSTATE '99999';   SET no = 90;   IF no &lt; 100 THEN     SIGNAL err;   END IF; EXCEPTION   WHEN err THEN     MESSAGE 'ユーザ定義例外！' TO CLIENT; END;</pre>

無断複製、譲渡、一部または全部の転載を禁じます。

Copyright (C) 2011 iAnywhere Solutions, Inc. All rights reserved. SQL Anywhere、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。