INFORMATION ANYWHERE

iAnywhere SOLUTIONS A SYBASE COMPANY

Technical Guide

IBM DB2から Adaptive Server Anywhere (現SQL Anywhere)への 移行

目次

はじめに

ソフトウェアバージョン
本書の内容は、SQL Anywhere Studioバージョン9.0.2以降、およびIBM DB2 UDBバージョン8.2以降を対象にしています。

Adaptive Server Anywhereには、DB2 UDB（およびその他のRDBMS）からAdaptive Server Anywhereへのスムーズな移行を実現するビルトインツールが含まれています。したがって、移行対象のデータベースあるいはアプリケーションにDB2拡張が多用されていなければ、DB2からAdaptive Server Anywhereへのデータの移行は容易です。

本書では、Adaptive Server AnywhereとDB2の違いについて、データ型、機能、構文などを含めてくわしく説明します。移行に際しては、Adaptive Server AnywhereとDB2の間に存在する意味（セマンティック）上の違いが問題となる可能性があります。本書では基本的にこの種の問題は扱いませんが、対処法がわかっている場合は記載しています。

さらに本書では、Sybase Centralの［データベース移行］ウィザードを使ってDB2データベースのデータをAdaptive Server Anywhereのデータベースに移行する手順を紹介し、また各種のアプリケーションについて、移行の際の具体的な注意事項を提供しています。

アプリケーションの移行に関する一般的な問題

下記に示すのは、Adaptive Server Anywhereがサポートしないか、またはAdaptive Server AnywhereとDS2間に大きな差分が存在する拡張です。移行対象のアプリケーションがこれらの拡張に大幅に依存しているのであれば、移行に際して相当な作業量が必要となります。

- XAトランザクション（非MS DTC）
- アプリケーション上でのSQLとCLIの混用
- data change table
- 構造データ型
- マテリアライズド・ビュー（AST）
- LOB（Adaptive Server Anywhereはインディケータをサポートしない）
- DATALINKデータ型
- C/C++コードによるDB2ユーティリティの記述

DB2スクリプトのデフォルトのコマンド終了区切り文字（デリミタ）は@ですが、Adaptive Server Anywhereでは;または**go**です。Interactive SQLの場合、起動時に**-d**@オプションを指定するか、またはcommand_delimiterオプションの値を変更することで、区切り文字を@に変えることも可能です。

データ型

本書では基本的なSQLデータ型について説明します。

NOTE DB2 UDBとAdaptive Server Anywhereのデータ型の対応について、くわしくはASA SQL ユーザーズ・ガイド データ型変換：DB2をご覧ください。

なおDB2のDATALINK型と構造データ型は、Adaptive Server Anywhereでサポートされないので注意が必要です。

DB2のデフォルトのタイムスタンプフォーマットはYYYY-MM-DD-HH-MM-SS.nnnnnですが、Adaptive Server AnywhereではYYYY-MM-DD-HH-MM-SS.nnnとなります。タイムスタンプフォーマットを変更するには、Adaptive Server Anywhereで次のSQL文を実行してください。

```
SET OPTION PUBLIC.TIMESTAMP_FORMAT="YYYY-MM-DD-HH-MM-SS.nnnnn"
```

DB2のデフォルト日付フォーマットはMM-DD-YYYYですが、Adaptive Server AnywhereではYYYY-MM-DDです。日付フォーマットを変えるには、Adaptive Server Anywhereで次のSQL文を実行してください。

```
SET OPTION PUBLIC.DATE_FORMAT="MM-DD-YYYY"
```

日付情報を挿入する方法は、デフォルトではIBM DB2とAdaptive Server Anywhereで同じです。またAdaptive Server Anywhereでは、DATE_ORDERオプションにより、日付情報の順番を変更できます。デフォルトでは、日付は年ー月ー日（YMD）の順で並んでいます。たとえばこの順番を月ー日ー年に変えるには、次のコマンドを実行します。

```
SET OPTION PUBLIC.DATE_ORDER='MDY'
```

特殊レジスタ

Adaptive Server Anywhereのレジスタはアップデートできません。

表1. 特殊レジスタの対応

DB2	Adaptive Server Anywhere	備考
CURRENT CLIENT_ACCTNG	AppInfo	
CURRENT CLIENT_APPLNAME	Current user	
CURRENT CLIENT_USERID	AppInfo	
CURRENT CLIENT_WRKSTNNAME	SELECT nodeaddr FROM sa_conn_info( ) WHERE number= @@spid	IP アドレスは取得できるが名称は取得不可
CURRENT DATE	CURRENT DATE	
CURRENT DBPARTITIONNUM	N/A	
CURRENT DEFAULT TRANSFORM GROUP	N/A	
CURRENT DEGREE	N/A	
CURRENT EXPLAIN MODE	N/A	
CURRENT EXPLAIN SNAPSHOT	N/A	
CURRENT ISOLATION	SELECT CONNECTION_PROPERTY('isolation_level')	
CURRENT LOCK TIMEOUT	N/A	
CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION	N/A	
CURRENT PACKAGE PATH	N/A	
CURRENT PATH	N/A	
CURRENT QUERY OPTIMIZATION	SELECT CONNECTION_PROPERTY('optimization_level')	
CURRENT REFRESH AGE	N/A	
CURRENT SCHEMA	CURRENT USER	
CURRENT SERVER	CURRENT DATABASE	
CURRENT TIME	CURRENT TIME	
CURRENT TIMESTAMP	CURRENT TIMESTAMP	
CURRENT TIMEZONE	SELECT CONNECTION_PROPERTY('TimeZoneAdjustment')	
CURRENT USER	CURRENT USER	
SESSION_USER	CURRENT USER	
SYSTEM_USER	CURRENT USER	
USER	CURRENT USER	

定数

文字列定数と数値定数は、DB2とAdaptive Server Anywhereで同一です。

DB2で指定する必要がある16進定数のフォーマットはx'FFFF'ですが、これはAdaptive Server Anywhereでは0xFFFFに相当します。この違いはスクリプトに影響を及ぼす可能性がありますが、アプリケーションに影響することはないはずです。

演算子と述部

DB2では、算術エラーの処理はDFT_SQLMATHWARNで制御されます。Adaptive Server Anywhereでは、divide_by_zeroオプションを使うことで、同様のエラー処理を行うように設定できます。

DB2とは異なり、Adaptive Server Anywhereではユーザ定義のSQLオペランドをサポートしません。

表2. 演算子および述部の対応

DB2	Adaptive Server Anywhere	備考
ALL、ANY、SOME 述部	ALL、ANY、SOME 述部	
BETWEEN 述部	BETWEEN 述部	
CASE	CASE	
CAST	CAST	SCOPE 句は構造型に関連するためサポートされない
CONCAT	||	
DATE TIME 演算	サポート、ただし相違点あり(備考参照)	期間を加算あるいは減算するとき、DB2 は integer へのキャストを解釈し、値(期間)が DATE、TIME、TIMESTAMP のどれかを判断します。Adaptive Server Anywhere は、TIMESTAMP と DATE への日付けの加算と減算をサポートします。 特定の機能については DATEADD 関数を使うことを推奨します。 日時の減算については、Adaptive Server Anywhere は DATE – DATE と TIMESTAMP – TIMESTAMP をサポートしますが、TIME – TIME はサポートしません。 この種の関数としては DATEDIFF を推奨します。
EXISTS 述部	EXISTS 述部	
IN 述部	IN 述部	下記の Quantified 述部参照
IS [NOT] NULL	IS [NOT] NULL	
LIKE 述部	LIKE	Adaptive Server Anywhere は ESCAPE 句をサポートしません。検索パターン(文字列)の終端にブランクを含む場合、Adaptive Server Anywhere はブランクを無視せず、ブランクを含む値とのパターン・マッチングのみを実行します。

Quantified 述部	行値コンストラクタはサポートせず(ANSI feature F641)	本機能を使うクエリの大半は書き換え可能です。
		〈クエリ例〉
		`SELECT * FROM TA`
		`WHERE ( TA.x, TA.y )`
		`IN ( SELECT TB.x, TB.y`
		`FROM TB`
		`WHERE z = 'abc' )`
		上述のクエリは同様の意味を持つ以下の文に書き換えられます。
		`SELECT * FROM TA`
		`WHERE EXISTS (`
		`SELECT * FROM TB`
		`WHERE TB.x = TA.x`
		`AND TB.y = TA.y AND TB.z`
		`= 'abc' )`

= イコール >大なり <小なり >= ～以上 <= ～以下 != 等しくない <> 等しくない !> 未満 !< ～より大きい	= イコール > 大なり < 小なり >= ～以上 <= ～以下 != 等しくない <> 等しくない !> 未満 !< ～より大きい	
参照解除演算子		Adaptive Server Anywhere は構造データ型をサポートしませ

DB2	Adaptive Server Anywhere	備考
		ん。
スカラー全選択	サポート	
SELECTIVITY	(CONDITION *selectivitypercentage*)	Adaptive Server Anywhere では SELECTIVITY ではなく、ユーザ推定を使用します。 詳細については、ASA SQL リファレンス・マニュアル 明示的な選択性推定を参照してください。
シーケンス参照		Adaptive Server Anywhere はシーケンスをサポートしません。シーケンスの代わりに GLOBAL AUTOINCREMENT または AUTOINCREMENT と@@identity を使用してください。
TREAT		Adaptive Server Anywhere は構造データ型をサポートしません。
TYPE 述部		Adaptive Server Anywhere は構造データ型をサポートしません。
VALUES(X)		SELECT X の使用を推奨します。

データ操作言語

アプリケーションは、データベースがサポートするSQL言語に大幅に依存します。Adaptive Server AnywhereとDB2のデータベースがそれぞれサポートする構文は、大きく重複しています。data change table、構造データ型、LOBインジケータ、LOBファイル、DATALINKS、管理句（administrative clause）、データ連携句（data federation clause）などのDB2拡張は、移行の妨げとなる可能性があるので注意が必要です。アプリケーションがこれらの拡張を一切利用していなければ、移行は簡単です。

ALLOCATE CURSOR

Adaptive Server Anywhereでは必要ありません。

ASSOCIATE LOCATORS

Adaptive Server Anywhereでは、LOCATORS変数はサポートされません。

CALL

Adaptive Server AnywhereはSQLおよびJavaプロシージャの呼び出しをサポートします。

CLOSE

DB2はCLOSE文をインタラクティブにサポートしますが、Adaptive Server AnywhereはEmbedded SQLと手続き型言語でサポートします。ロックはWITH RELEASE句を必要とすることなく、暗黙的に解除されます。

COMMIT

標準の構文とSAVEPOINTの構文をサポートします。

CONNECT

USING *password*句は、Adaptive Server AnywhereではIDENTIFIED BY *password*句に変更する必要があります。lock-block句は、他のユーザによるシステム使用の可否を制御するものですが、Adaptive Server Anywhereではサポートされないので、削除しなくてはなりません。Adaptive Server Anywhereでシングルユーザモードを取得するには、**-b**オプションを設定してからデータベースを起動するか、あるいは**sa_server_option**ストアド・プロシージャを使用してください。

Adaptive Server Anywhereは、authorization blockのNEW句とCONFIRM句によるパスワード変更をサポートしません。パスワード管理ロジックを変更してGRANT CONNECT文を使うようにしてください。

DECLARE CURSOR

Adaptive Server Anywhereは、WITH HOLDをDECLARE CURSOR文中ではサポートしませんが、OPEN文中ではサポートします。

```
WITH RETURN TO CALLER
WITH RETURN TO CLIENT
```

リターン値は、結果セットを受信するデータベースサーバによって動的に決定されます。Adaptive Server Anywhereでは、これらの句は必要とされません。詳細については本書のBIGIN 3.eを参照してください。

DECLARE GLOBAL TEMPORARY TABLE

Adaptive Server AnywhereはDECLARE GLOBAL TEMPORARY TABLEをサポートします。Adaptive Server Anywhereのテンポラリ・テーブルは、IN TABLESPACEのような格納オプションの設定をサポートしません。

TABLE定義の違いについては本書の「CREATE TABLE」を参照してください。

DELETE（POSITIONEDまたはSEARCHED）

DELETE文は、アプリケーションがTYPED TABLESまたはdata change tableを使っていない限り、Adaptive Server Anywhereに移行できます。Adaptive Server Anywhereでは、TYPED TABLESに適用されるFROM ONLY句はサポートされません。またdata change table（または中間結果セット）に適用されるInclude句とassignment句もサポートされません。data change tableを使うクエリは、トリガとテンポラリ・テーブルを使ってアップデート／削除を行い、所望のフロー制御を実行することで代替できます。

WITH句は変更する必要があります。WITH句の対応は下記をご覧ください。

表3. DELETE文中のWITH句の対応

DB2	Adaptive Server Anywhere
WITH RR	WITH SERIALIZABLE
WITH RS	WITH REPEATABLEREAD
WITH CS	WITH READCOMMITED
WITH UR	WITH READUNCOMMITED

DESCRIBE

Adaptive Server AnywhereではEmbedded SQLとしてサポートします。詳細については本書の「Embedded SQLアプリケーションの移行」を参照してください。

DISCONNECT

Adaptive Server Anywherの構文では、サーバ名ではなく接続名を指定します。接続名を指定しない場合、カレントの接続は切断されます。またAdaptive Server Anywhereでは、キーワード**ALL**はサポートされません。

EXPLAIN

Adaptive Server Anywhereは、インタラクティブな環境ではEXPLAINをサポートしませんが、Embedded SQL中のEXPLAINはサポートします。Adaptive Server Anywhereのパフォーマンスチューニングは、IBM DB2とは異なるアプローチを取っているため、EXPLAIN文をアプリケーション中で使用することはほとんどないはずです。

Adaptive Server Anywhereのパフォーマンスのチューニングについては、ASA SQL ユーザーズ・ガイド クエリの最適化と実行- オプティマイザの仕組みおよびASA SQL リファレンス・マニュアル PLAN関数を参照してください。

FETCH

両製品に構文上の違いはありませんが、セマンティックは異なります。またAdaptive Server Anywhereは、意図的ロック、FETCH FOR UPDATEクエリ、書き込みロックをサポートしません。

FREE LOCATOR

Adaptive Server AnywhereはLOCATORS変数をサポートしません。

INSERT

Adaptive Server AnywhereのINSERT文は、次の二つの点でDB2のINSERT文と異なります。

- Adaptive Server AnywhereのINSERT文では、フルSELECT文へのインサートはサポートされません。ただしSELECT文からテーブル／ビューへのインサートはサポートします。

- Adaptive Server Anywhereは中間結果テーブルをサポートしないため、そのINSERT文はINCLUDE（カラム）構文をサポートしません。以下の例をごらんください。

```
SELECT inorder.ordernum
FROM ( INSERT INTO orders ( custno )INCLUDE ( insertnum
          integer )
VALUES( :cnum1,1 ),( :cnum2,2 ) )InsertedOrders
ORDER BY insertnum;
```

LOCK TABLE

Adaptive Server AnywhereはSYNONYMをサポートしないので、Adaptive Server AnywhereのLOCK TABLE名は、ベース・テーブル名でなければなりません。DB2の構文は、Adaptive Server Anywhereでサポートされます。DB2とは異なり、Adaptive Server Anywhereでは、EXCLUSIVEモードでISOLATION 0 （UNCOMMITED READS）リードオンリークエリを実行できません。

MERGE

Adaptive Server AnywhereはMERGEをサポートしませんが、INSERTとDELETEを使って同様の機能を実現できます。ただしこの方法では、MERGE文使用時のようなエラーメッセージを提供できず、またカラムレベルの柔軟なアップデートもサポートしません。たとえば次のMERGE文をご覧ください。

```
MERGE INTO archive ar
USING ( SELECT activity,description,date,last_modified
FROM activities_groupA )ac
ON ( ar.activity =ac.activity )AND ar.group =’A ’
WHEN MATCHED AND ac.date <CURRENT DATE THEN
DELETE
)
WHEN MATCHED AND ar.last_modified <ac.last_modified THEN
UPDATE SET
( description,date,last_modified )=(
          ac.description,ac.date,DEFAULT )
WHEN NOT MATCHED
INSERT ( description,date,last_modified )
VALUES ( ac.activity,ac.date,DEFAULT )
```

上記と同様の機能を提供するAdaptive Server Anywhereの構文は下記のとおりです。

```
INSERT INTO archive ( activity,description,date ) SELECT
        activity,description,date,last_modified
FROM activities_group ON EXISTING UPDATE;
DELETE FROM archive where archive.group = ’A’ AND
        archive.activity=
( SELECT activity
FROM activities_group WHERE date<CURRENT DATE )
```

INSERT ON EXISTING UPDATEがプライマリ･キーを持つために宛先テーブルを必要とする点に注意してください。

RELEASE SAVEPOINT

Adaptive Server Anywhereでは**TO**キーワードオプションをサポートしないので、削除する必要があります。

ROLLBACK

Adaptive Server Anywhereの構文は、ROLLBACK WORK TO SAVEPOINT *savepoint - name*が使われている場合に限り、DB2の構文とは異なるものとなります。この構文の**WORK**キーワードオプションは削除する必要があります。

SAVEPOINT

Adaptive Server AnywhereはSAVEPOINT *savepoint - name*をサポートしますが、UNIQUE句、ON ROLLBACK RETAIN CURSORS句、ON ROLLBACK RETAIN LOCKS句はサポートしません。close_on_endtransデータベースオプションを使い、Adaptive Server Anywhereがロールバックとコミットでカーソルを保持するように設定できます。

SELECT

SELECT文についての説明はやや複雑です。SELECT文は、複数のキーコンポーネントに分割できます。

WITH共通表式（common-table-expression）

DB2の共通表式はWITH句を認識します。Adaptive Server AnywhereはWITH句をサポートします。

select-list notation

構文上の違いはありません。特殊レジスタと関数を選択できます。

table-reference（表参照）

Adaptive Server Anywhereは、結果テーブルに適用される相関ネーミング用のカラムリストをサポートします。DB2とAdaptive Server Anywhereはどちらも次の構文をサポートします。

```
SELECT * FROM ( SELECT * FROM department ) AS e ( i, j , k )
```

Adaptive Server Anywhereでは、DB2のtablesamplingあるいはdata-change-table-references句はサポートされません。タイプされたテーブルに適用されるONLY句とOUTER句は、Adaptive Server Anywhereではサポートされません。なおAdaptive Server Anywhereはニックネームまたはシノニムをサポートしませんが、エリイアスはサポートします。

Adaptive Server Anywhereでは、TABLE（*function-name*）コンストラクタはサポートされません。しかしAdaptive Server Anywhereではストアド・プロシージャからの選択をサポートするので、結果セットを戻すストアド・プロシージャを使って同様の結果を得ることができます。以下にその例を示します。

```
SELECT t.id, t.quantity_ordered AS q, p.name
FROM sp_customer_products( 149 ) t JOIN product p
ON t.id = p.id
```

joined table構文

製品間に構文上の違いは存在しません。

GROUP BY句

Adaptive Server Anywhereは、基本的なグルーピング式であるGROUP BY ROLLUP、GROUP BY CUBE、GROUP BY GROUPING SETSをサポートします。一方DB2は、これらの長大なグルーピングセット仕様のネスティングをサポートすることで、構文の速記を実現しています。Adaptive Server Anywhereは、たとえばDB2でネストされたグルーピングセット仕様GROUP BY ROLLUP（ Province, （ County,

City ））を直接サポートしませんが、同じ意味のGROUP BY GROUPING SETS （（ Province, County, City ）, （Province ）, （ ） ）をサポートします。

ORDER BY句

Adaptive Server Anywhereでは、INPUT SEQUENCE句とORDER OF句はサポートされません。またDB2はUNION、INTERCEPT、EXCEPTを伴うORDER BY句をサポートしますが、Adaptive Server Anywhereはこれらをサポートしません。ただしORDER BY句中のintegerやcolumn nameといった単純な式はサポートされます。

fetch-first文節

Adaptive Server Anywhereは、DB2より簡潔にこのコンストラクトをサポートします。Adaptive Server Anywhereは、xがintegerの場合にSELECT FIRST xをサポートします。この句はDB2では文の終りに位置しますが、Adaptive Server Anywhereではselect-list notation句の手前に位置します。

update文節

製品間に構文上の違いは存在しません。

NOTE

キーコンポーネントの組み合わせは、Adaptive Server AnywhereとDB2で意味が異なることがあるので、同様のクエリがAdaptive Server AnywhereとDB2で異なる結果をもたらす可能性があります。
Adaptive Server Anywhereはサブクエリをサポートします。しかしDB2の比較述部（行値コンストラクタ）が複数のカラムをサポートするのに対し、Adaptive Server Anywhereがサポートするカラムは一つだけです。詳細については「演算子と述部」の比較述部の項を参照してください。

read-only文節

Adaptive Server Anywhereは、FOR READ ONLYはサポートしますが、FOR FETCH ONLYはサポートしません。詳細についてはASA SQL リファレンス・マニュアル- SQL 文- SELECT 文をご覧ください。

optimize-for文節

DB2と同様に、Adaptive Server Anywhereは、デフォルトで結果セット中の全行を迅速に取得するよう最適化されています。ただし結果セットのごく一部を必要とするときは、別の最適化手段を用いることも可能です。DB2はOPTIMIZE FOR x ROWS（xはinteger）、Adaptive Server AnywhereはWITH（FASTFIRSTROW）句でこの最適化を実現します。

isolation文節

WITH句は下記のように対応します。

表4. isolation文節中のWITH句の対応

DB2	Adaptive Server Anywhere
WITH RR	WITH SERIALIZABLE
WITH RS	WITH REPEATABLEREAD
WITH CS	WITH READCOMMITED
WITH UR	WITH READUNCOMMITED

lock-request文節

Adaptive Server AnywhereにはWITH句で指定するテーブルヒントがあり、クエリと同様のロッキングを行う際に使うことができます。

表5. Lock-request文節中のWITH句の対応

DB2	Adaptive Server Anywhere
USE AND KEEP SHARE LOCKS	WITH (NOLOCK)
USE AND KEEP UPDATE LOCKS	WITH (HOLDLOCK)
USE AND KEEP EXCLUSIVE LOCKS	WITH (XLOCK)

SET CURRENT DEFAULT TRANSFORM GROUP

Adaptive Server Anywhereは構造データ型をサポートしません。

SET CURRENT DEGREE

Adaptive Server Anywhereは自動的にクエリ内の並列処理を実行するので、この文はサポートしません。

SET CURRENT EXPLAIN MODE

Adaptive Server AnywhereはEXPLAIN MODEのサポートを必要としません。EXPLAIN MODEは主としてパフォーマンスとチューニングのために使われ、大半のアプリケーションには使われません。

SET CURRENT EXPLAIN SNAPSHOT

Adaptive Server Anywhereは、EXPLAIN SNAPSHOTを必要としません。EXPLAIN SNAPSHOTは主としてパフォーマンスとチューニングのために使われ、大半のアプリケーションには使われません。

SET CURRENT ISOLATION

Adaptive Server Anywhereは、SET OPTION ISOLATION_LEVELによる隔離レベルの設定をサポートします。Adaptive Server AnywhereのISOLATION_LEVELは下記のDB2設定に対応します。

表6. 隔離レベルの対応

DB2	Adaptive Server Anywhere
RR | SERIALIZABLE | REPEATABLE READ	3
RS	2
CS | CURSOR STABILITY | READ COMMITTED	1
UR | DIRTY READ | READ UNCOMMITTED	0
RESET	0

SET CURRENT LOCK TIMEOUT

Adaptive Server Anywhereはロックタイムアウトをサポートしません。BLOCKINGオプションをOFFに設定し、アプリケーションからこの設定を制御することで、同じ機能を実現できます。

SET CURRENT MAINTAINED TABLE TYPES FOR OPTIMIZATION

Adaptive Server Anywhereは構造化されたテーブル型をサポートしません。

SET CURRENT PACKAGE PATH

Adaptive Server AnywhereはPACKAGE PATHを必要としません。このコマンドは、アプリケーションが使用するスキーマ／ユーザ・オブジェクト／パッケージを示すものです。Adaptive Server Anywhereでは、データベースオブジェクトにアクセスするためのスキーマ（およびパーミッション）を指定するために、グループとグループメンバーシップを使います。

SET CURRENT PACKAGESET

Adaptive Server AnywhereはPACKAGESETを必要としません。このコマンドは、アプリケーションが使用するスキーマ／ユーザ･オブジェクト／パッケージを示すものです。Adaptive Server Anywhereでは、データベースオブジェクトにアクセスするためのスキーマ（およびパーミッション）を指定するために、グループとグループメンバーシップを使います。

SET CURRENT QUERY OPTIMIZATION

Adaptive Server AnywhereではOPTIMIZATION_LEVELオプションを設定できます。このオプションの値は次のクエリで取得できます。

```
SELECT CONNECTION_PROPERTY ( 'OPTIMIZATION_LEVEL' )
```

SET CURRENT REFRESH AGE

Adaptive Server Anywhereはマテリアライズド・ビューをサポートしません。

SET ENCRYPTION PASSWORD

Adaptive Server Anywhereはこの暗号化手段をサポートしません。

SET INTEGRITY

Adaptive Server Anywhereにはテーブルステートの概念はないので、INTEGRITYの設定は必要ありません。またこの文は通常は配置されるアプリケーションで使われることはありません。

SET PASSTHRU

Adaptive Server Anywhereは、FORWARD TO文を使うことで、SET PASSTHRUと同様の機能をサポートします。FORWARD TO文はまたSET PASSTHRU RESETの役割も果たします。

SET PATH

Adaptive Server AnywhereはPACKAGE PATHを必要としません。

SET SCHEMA

Adaptive Server Anywhereは接続のスキーマの設定をサポートしません。Adaptive Server Anywhereでは、グループメンバーシップを介してスキーマ修飾を制御できます。

SET SERVER OPTION

Adaptive Server Anywhereは、SERVER OPTIONSの代わりに複数のサーバ機能を提供します。これらの機能はALTER SERVER *server-name* CAPABILITY *capability-name* { ON | OFF }により設定可能です。これらのリモートデータ（協調化）機能の対応付けはまだ行われていません。

SET SESSION AUTHORIZATION | SET SESSION USER

Adaptive Server Anywhereは、SET SESSION AUTHORIZATIONまたはSET USERによるパーミッションの設定をサポートしますが、特殊レジスタUSER、CURRENT USER、SESSION USERはサポートしません。またALLOW ADMINISTRATION句オプションもサポートしません。

SET VARIABLE

DB2とAdaptive Server AnywhereのどちらもSET文をサポートしますが、Adaptive Server Anywhereでは一つの変数を割り当てるのに対し、DB2では一つの変数、変数のローセットまたはタプルの割り当てをサポートする点で異なります。Adaptive Server Anywhereによる割り当てでは、クエリが一つの値を返す場合、full selectを含めることも可能です。また複数のパラメータや変数を用いる一つのSETによる割り当てを行っている場合は、単純に複数のSET文を使うことで対応できます。

SIGNAL

Adaptive Server Anywhereは業界標準のSQLSTATE構文をサポートしますが、DB2は定義済みのエラーに加えて、ユーザ定義のエラーの発信もサポートします。アプリケーションがユーザ定義のエラーの発信を必要とする場合は、RAISERRORを使用してください。

UPDATE

UPDATE文は、アプリケーションにTYPED TABLESまたはdata change tableを使っていない限り、Adaptive Server Anywhereに移行できます。Adaptive Server Anywhereは、full-selectのアップデートをサポートします。またテーブルリストへの複数テーブルの取り込みも同じくサポートするので、テーブル結合と複数テーブルのアップデートを実行できます。ただしDB2のTYPED TABLESに適用されるFROM ONLY句、またdata change table（あるいは中間結果セット）に適用されるInclude句はサポートしません。

行依存の構文を持つUPDATE（例: UPDATE tblname SET （ col1,col2 ） = （ 1,2 ） WHERE col3 =3）はAdaptive Server Anywhereではサポートされないので、修正する必要があります。

Adaptive Server Anywhereでは、中間結果セットへのassignment句の適用をサポートしません。data change tableを使うクエリの場合、data change tableの代わりとしてトリガとテンポラリ・テーブルを使い、アップデートと削除によって所定のフロー制御を実現することが考えられます。Adaptive Server Anywhereはローセットのアップデートをサポートしません。SET文は、値の列ではなく個々の値に適用しなくてはなりません。WITH句も変更する必要があります。WITH句の対応は以下のとおりです。

表6. UPDATE文中のWITH句の対応

DB2	Adaptive Server Anywhere
WITH RR	WITH SERIALIZABLE
WITH RS	WITH REPEATABLEREAD
WITH CS	WITH READCOMMITED
WITH UR	WITH READUNCOMMITED

VALUES [INTO]

VALUESが一つのレコードを返すクエリの場合、Adaptive Server Anywhere ではSELECTで置換できます。以下の例をご覧ください。

```
SELECT 'A' INTO :hostvar
```

複合文

アプリケーションには、ストアド・プロシージャ、トリガ、バッチを使用するものがあります。Adaptive Server AnywhereとDB2のデータベースがそれぞれサポートする構文には、かなりの重複があります。ロジックの種類に応じたSQLベースのエラー処理については、移行に際して改めて検討する必要があります。

DB2にSQLベースの手続き言語が導入されたのは最近のことなので、古いスタイルのDB2プロシージャの移行は、特にアプリケーションがホスト言語に大きく依存している場合には困難かもしれません。またAdaptive Server AnywhereはC/C++ベースの外部ストアド・プロシージャをサポートしないので、注意が必要です。

BEGIN

Adaptive Server AnywhereとDB2のいずれも動的な複合文をサポートします。また両者の構文の大部分は同じです。現時点までに発見されている構文上の違いは下記のとおりです。

1. DB2では、SQLSTATEに対するCONDITIONSのDECLAREが認められています。一方Adaptive Server Anywhereでは、SQLSTATESに対するDECLARE EXCEPTIONが認められています。これらのSQLSTATE値は製品間で異なる可能性があります。

2. DB2ではデフォルト値の変数を認めています。Adaptive Server Anywhereは、変数宣言を伴うデフォルト値をサポートしません。ただし代わりにSET assignment文を用いることは可能です。

3. 手続き的複合文について、Adaptive Server Anywhereは以下の点でDB2と異なります。

    a. Adaptive Server Anywhereでは、RESULT_SET_LOCATOR VARYINGは正当な変数宣言ではありません。

    b. Adaptive Server Anywhereは、SQLCODEまたはSQLSTATEの宣言を必要としません。これらの値は宣言なしで常にアクセス可能です。

    c. Adaptive Server Anywhereは、手続き型言語のDECLARE STATEMENTをサポートしません。これらはおそらくEmbedded SQLで書かれたプロシージャで使われるはずで、SQLプロシージャで必要とされることはないでしょう。

    d. Adaptive Server Anywhereのexception-case（例外ハンドラ）はIBM DB2のハンドラの宣言におおむね対応します。例外ハンドラの方がより高度な制御を提供します。

    e. DB2ではDECLARE CURSORがWITH HOLDを実現します。Adaptive Server Anywhereでは、OPEN文中でWITH HOLDをサポートします。DB2は、RETURNのための句をCLIENTまたはCALLERに提供します。Adaptive Server Anywhereは、プロシージャ定義中でこれらのカーソルをオープンしないことを前提として、クライアントに対するリターンカーソルをプロシージャ中でサポートします。またAdaptive Server Anywhereはリターンカーソルを自動的に処理します。

CASE

製品間に構文上の違いは存在しません。

FOR

DB2とAdaptive Server Anywhereの構文の唯一の違いは、WITH HOLD句です。WITH HOLD句は、Adaptive Server AnywhereのFOR文中のデフォルト動作です。*cursor-name* CURSOR FOR句は、DB2ではオプションですが、Adaptive Server Anywhereでは必須です。

FETCH

製品間に構文上の違いは存在しません。

GET DIAGNOSTICS

DB2のGET DIAGNOSTICS ROW_COUNTの意味は、Adaptive Server Anywhereの文SET *variable-name* = @@rowcountに似ています。DB2のGET DIAGNOSTICS EXCEPTION . . . の意味は、収集対象のエラー情報にもよりますが、SET *variable-name* = SQLCODE | SQLSTATEまたはERRORMSGとほぼ同じです。Adaptive Server Anywhereでは、以下の文でDB2のGET DIAGNOSTICS DB2_RETURN_STATUSと同様の意味を表現できます。

```
BEGIN
DECLARE i INT;
i= call xp_cmdshell( 'dirt' );
MESSAGE 'i'||i;
END
```

GOTO

Adaptive Server Anywhereは、Watcom-SQLとTransact-SQLの二つのSQLダイアレクトをサポートしています。DB2ダイアレクトに近いWatcom-SQL言語ではGOTO文をサポートしませんが、Transact-SQ形式の言語ではGOTO文をサポートします。GOTO文はTransact-SQLに移行できます。あるいはDB2とAdaptive Server Anywhereの両方と互換性があるWatcom-SQLでサポートされるコンストラクトで、ロジックを再コード化することも可能です。

IF

製品間に構文上の違いは存在しません。

ITERATE

Adaptive Server AnywhereではITERATE文はサポートされませんが、Transact-SQLのCONTINUE文はサポートされます。ストアド・プロシージャはTransact-SQL構文を使うように再修正する必要があります。

ロジックをWatcom-SQLのANSI形式に修正することも可能です。

LEAVE

製品間に構文上の違いは存在しません。

LOOP

製品間に構文上の違いは存在しません。

OPEN

製品間に構文上の違いは存在しません。

REPEAT/UNTIL

Adaptive Server AnywhereではREPEAT文はサポートされません。WHILE/LOOPまたはFOR LOOPを使うには、ストアド・プロシージャ、バッチ、トリガを修正する必要があります。

RESIGNAL/SIGNAL

Adaptive Server Anywhereは業界標準のSQLSTATE構文をサポートしますが、DB2は事前定義のエラーに加えて、ユーザ定義のエラーの発信をもサポートします。アプリケーションがユーザ定義のエラーの発信を必要とする場合は、RAISERRORを使用してください。

RETURN

Adaptive Server Anywhereの構文は、関数またはプロシージャにユーザが定義したスカラー値からの一つの戻り値をサポートします。スカラー値構文は、Adaptive Server AnywhereとDB2で同一です。またDB2では、関数から結果セットを返すことが可能です。これらの関数は一般に**テーブル関数**と呼ばれます。Adaptive Server Anywhereでテーブル関数と同様の機能を実現するには、結果セットを返すプロシージャを作成する必要があります。

詳細については本書の「テーブル関数」の項をご覧ください。

WHILE

Adaptive Server AnywhereのLOOPのコンストラクトは、DB2のWHILE構文と非常によく似ています。LOOPについては、本書の「LOOP」の項をご覧ください。

関数、プロシージャ、トリガ

アプリケーションには、ストアド・プロシージャ、トリガ、バッチを使うものがあります。Adaptive Server AnywhereとDB2のデータベースがそれぞれサポートする構文は、大きく重複しています。エラー処理は、SQLベースのロジックのタイプによっては、移行に際して問題になる可能性があります。DB2ではSQLベースの手続き型言語を最近初めて導入したので、アプリケーションがホスト言語に大きく依存している場合、古いスタイルのDB2プロシージャの移行は困難が予想されます。

Adaptive Server AnywhereはC/C++外部ストアドプロシージャと関数をサポートしますが、これはスカラー値のリターンに限られ、またSQLプロシージャのように接続を継承することはできません。一方のDB2は、接続環境を継承して結果セットを返すことができる外部プロシージャを提供します。

Java関数とプロシージャ

EXTERNAL NAME句は、Adaptive Server AnywhereとDB2で一貫しています。Adaptive Server Anywhereは、Javaメソッドのロードにネームおよびパラメータシグネチャを必要とするため、関数シグネチャを必要としますが、DB2で必要となるのはJAR、クラス、関数仕様だけです。

パラメータ

Adaptive Server Anywhereが持つパラメータタイプのスタイルは一つ（SQL）だけなので、パラメータスタイルを指定する必要はありません。DB2は、外部ストアド・プロシージャと関数向けに各種のパラメータスタイル用の句をサポートします。Adaptive Server Anywhereはこれらを SQLパラメータに単純化して、パラメータスタイルに構文を提供します。またAdaptive Server AnywhereはCSSID句をサポートします。

NOTE DB2 UDBとAdaptive Server Anywhereのデータ型の対応について、くわしくはASA SQL ユーザーズ・ガイド データ型変換：DB2をご覧ください。

CREATE FUNCTION

Adaptive Server Anywhereは、SQL、Java、ユーザ定義の外部C/C++スカラー関数をサポートします。ただしOLE DB外部テーブル関数、SOURCED関数、テンプレート関数をサポートしません。DB2は、関数の生成と変更に関わる数多くの変換修飾子をサポートしています。Adaptive Server Anywhereでは、（NOT） DETERMINISTIC句とEXTERNAL句で関数の変換修飾子をサポートしますが、locking、SQL contents、thread-safety等の他の修飾子は不要であるか、またはサポートされません。Adaptive Server AnywhereとDB2のスカラー関数は、どちらもRETURNS句を提供します。さらにAdaptive Server AnywhereではSQL関数についてBEGIN. . .END文が必要となります。

テーブル関数

Adaptive Server Anywhereでは、DB2のテーブル関数を、結果セットを返すストアド・プロシージャに移行できます。以下のDB2のテーブル関数を例に説明します。

```
CREATE FUNCTION DEPTEMPLOYEES ( DEPTNO CHAR( 3 ) )
RETURNS TABLE ( EMPNO CHAR( 6 ),
```

```
LASTNAME VARCHAR( 15 ),
FIRSTNAME VARCHAR( 12 ) )
LANGUAGE SQL
READS SQL DATA
NO EXTERNAL ACTION
DETERMINISTIC
RETURN
SELECT EMPNO,LASTNAME,FIRSTNME
FROM EMPLOYEE
```

DB2では、これを次のように呼び出します。

```
SELECT * FROM TABLE ( DEPTEMPLOYEES( 'ABC' ) )
AS functioncall
```

この関数は、Adaptive Server Anywhereでは次のように実装できます。

```
CREATE PROCEDURE DEPTEMPLOYEES ( DEPTNO CHAR( 3 ) )
RESULT(EMPNO CHAR( 6 ),LASTNAME VARCHAR( 15 ), FIRSTNAME
VARCHAR( 12 ) )
BEGIN
SELECT EMPNO,LASTNAME,FIRSTNME FROM EMPLOYEE
END
```

Adaptive Server Anywhereでは、この関数を以下のいずれかの方法で呼び出すことができます。

```
//Option 1
SELECT * FROM DEPTEMPLOYEES( 'ABC' )
AS functioncall

//Option 2
CALL DEPTEMPLOYEES( 'ABC' )
```

CREATE PROCEDURE

Adaptive Server AnywhereとDB2は、いずれもSQLベースのプロシージャを提供します。DB2は、バージョン8.2からSQLベースのロジックを導入しました。この項では、両製品のSQLベースのプロシージャダイアレクト間の違いを説明します。
なお非SQLベースのプロシージャが存在する場合、SQLはネイティブの言語に埋め込まれることになりますが、これはSQLベースのストアド・プロシージャと非常によく似ています。本項の情報はこうした状況で有用なはずです。

Adaptive Server AnywhereとDB2の構文と動作の大部分は大半の製品でほぼ同じですが、下記に示す違いがあります。

1. Adaptive Server AnywhereとDB2は、どちらも呼出しプログラムへの結果セットの返信をサポートしますが、Adaptive Server AnywhereはFOR RETURN TO CLIENT | CALLER句を必要としません。Adaptive Server Anywhereのプロシージャ言語は、データ型をRESULT句で指定します。

2. どちらの製品もローカル変数をサポートしています。ただしAdaptive Server Anywhereはデフォルト設定をサポートしないので、代入文が必要です。

3. 例外ハンドラの違いについては本書の「一般的なエラー処理」を参照してください。

4. ALLOCATE CURSOR

5. Adaptive Server AnywhereではDECLARE STATEMENTはサポートされません。

6. Adaptive Server AnywhereではSQLCODEとSQLSTATEの宣言は必要ないので、移行時には削除する必要があります。

7. Adaptive Server AnywhereとDB2の複合文の違いについては、本書の「複合文」の項を参照してください。

8. Adaptive Server Anywhereでは、LANGUAGE SQLとLANGUAGE JAVAに対する変換修飾子がサポートされています。

9. Adaptive Server Anywhereは下記のコンテキストをサポートしません。

   a . SPECIFIC name

   b. [NOT] FENCED

   c. [NOT] DETERMINISTIC

   d. MODIFIES SQL DATA | CONTAINS SQL | READS SQL DATA

   e. CALLED NULL ON INPUT

   f. PARAMETER CCSID

   g. DYNAMIC RESULT SETS

   h. NO EXTERNAL ACTION

   i. OLD | NEW SAVEPOINT LEVEL

   j. INHERIT SPECIAL REGISTERS

   k. PARAMETER STYLE any

   l. PROGRAM TYPE

   m. [NO] DBINFO

Adaptive Server Anywhereのサンプル中のストアド・プロシージャには、大抵はBEGIN EXCEPTIONとENDが一つづつあります。例外発生時に処理を継続する必要があるときは、Adaptive Server Anywhereのプロシージャ中に複数の複合文を記述することも可能です。

CREATE TRIGGER

Adaptive Server AnywhereとDB2は、いずれも文レベルとロー・レベルのトリガをサポートします。両

データベースはSQLベースのトリガを提供します。トリガの構文と動作の大半は両製品で同じですが、下記に示す違いがあります。

1. DB2のCASCADE BEFORE句ではbefore triggerが認められていますが、Adaptive Server Anywhereではこれを BEFORE句でサポートします。ただしAdaptive Server Anywhereはcascadingトリガの抑制をサポートしません。

2. Adaptive Server Anywhereは、ビューに関するトリガおよびINSTEAD OFトリガをサポートしません。INSTEAD OFトリガがある場合は、ベーステーブルトリガを使うように修正する必要があるでしょう。

3. Adaptive Server AnywhereとDB2の複合文の違いについては、前述の「複合文」を参照してください。

4. 文レベルのトリガを使っている場合、OLD_TABLEをOLDへ、またNEW_TABLEをNEWへ変更する必要があります。

5. Adaptive Server Anywhereではトリガが複合文を使うことを想定しているため、BEGINとENDを必要とします。これらの文中ではラベルが生じる可能性があります。

6. MODE DB2SQL句を削除する必要があります。

一般的なエラー処理

Adaptive Server Anywhere、DB2手続き型言語、ODBCはすべてSQLSTATESをサポートします。ODBCの仕様では、SQLSTATEはMicrosoft社の仕様に従って定義されています。Adaptive Server AnywhereとDB2（DB2 CLI）のODBCドライバはこの仕様に準拠します。ODBCアプリケーションの移行については、後述する「CLI/ODBCアプリケーションの移行」を参照してください。

DB2は手続き型言語とEXCEPTIONをサポートします。Adaptive Server AnywhereとDB2のいずれも例外を宣言する必要があります。DB2は、ハンドラとしてCONTINUE、ERROR、UNDOを提供します。Adaptive Server Anywhereは、ハンドラとしてCONTINUEまたはERRORに相当するEXCEPTIONを提供します。同じくAdaptive Server Anywhereは、例外発生時の処理のために、プロシージャ定義中のON EXCEPTION RESUME句を提供します。DB2のUNDOハンドラは、Adaptive Server Anywhereではsavepointロジックを追加する必要があります。両製品の例外構文は、下記に示すように若干異なります。

DB2

```
DECLARE EXIT HANDLER FOR SQLEXCEPTION
SIGNAL SQLSTATE value SQLSTATE SET MESSAGE_TEXT = errorLabel;
```

Adaptive Server Anywhere

```
DECLARE EXCEPTION SIGNAL SQLSTATE value;
```

両製品は共にユーザ定義のエラーをサポートしますが、その構文は異なります。Adaptive Server Anywhereでユーザ定義のエラーに使われるのは、SIGNAL構文ではなくRAISERROR文です。Adaptive Server AnywhereとDB2はどちらも発信例外と再発信例外をサポートします。Adaptive Server Anywhereは、警告（たとえばROW_NOT_FOUND）をEXCEPTIONとみなさないので、これらのCONDITIONSやHANDLERは削除する必要があります。たとえばSQLCODE 100（SQLSTATE 02000）は、ROW_NOT_FOUNDを指定するものですが、Adaptive Server Anywhereでは、SQLSTATEやSQLCODEのような暗黙に定義された復帰コードハンドラの宣言は必要ありません。

ODBCエラーコード（SQLSTATES）はデータベースの世界ではよく知られていますが、多くのアプリケーションは、自身のネイティブのデータベースエラーを利用します。これらのエラーはベンダー間で異なることがあるので注意が必要です。ネイティブエラーを利用するアプリケーションの場合は、DB2のネイティブエラーをAdaptive Server Anywhereにマッピングする必要があります。なおIBMのCLIとPHPはODBCエラーメッセージとSQLSTATEに準拠します。

組み込み関数

アプリケーションの修正を望まない場合、通常はユーザ定義関数（UDF）を使ってDB2関数に相当するAdaptive Server Anywhere関数を生成できます。

以下に組み込み関数の対応を示します。以下の表の中央カラムに「CAST」とある関数については、インタフェースライブラリ（例: ODBC）を暗黙に変換することができます。

表7. 組み込み関数の対応

DB2 の関数	対応する Adaptive Server Anywhere の関数	備考
AVG (ALL | DISTINCT)	AVG (DISTINCT)	デフォルトで ALL。
CORRELATION/CORR	CORR	
COUNT (ALL | DISTINCT)	AVG (DISTINCT)	デフォルトで ALL。
COUNT_BIG (ALL | DISTINCT)		count 以外の対応物なし、count は integer を返す。
COVARIANCE/COVAR	COVAR_POP	Adaptive Server Anywhere は COVAR も提供。
GROUPING	GROUPING	複数に関係する句、移行に影響あり。
MAX (ALL | DISTINCT)	MAX (DISTINCT).	デフォルトで ALL。
MIN (ALL | DISTINCT)	MIN (DISTINCT)	デフォルトで ALL。
REGRESSION関数	REGRESSION関数	REGR_ICPT は REGR_INTERCEPT へ変更要。演算は類似。
STDDEV (ALL | DISTINCT)	STDDEV_POP	デフォルトで ALL。
VARIANCE/VAR (ALL | DISTINCT)	VAR_POP	
ABS ABSVAL	ABS	ABSVAL は ABS を参照する UDF 使用可。
ACOS	ACOS	
ASCII		
ASIN	ASIN	
ATAN/ATAN2/ATANH	ATAN/ATAN2	ATANH はサポートせず。
BIGIN	CAST	
BLOB	CAST	
CEILING またはCHAR	CEIL CEILING	CEIL の場合 UDF。
CHR	CHAR	
CLOB	CAST	
COALESCE	COALESCE	
CONCAT	STRING	UDF
COS とCOSH	COS	COSH なし。
COT	COT	
DATE	DATE	
DAY	DAY	
DAYNAME	DAYNAME	
DAYOFWEEK	DOW	UDF
DAYOFWEEK_ISO	DOW	モジュロ、追加、オプションを使うUDF。
DAYOFYEAR		UDF
DAYS	DAYS	結果に差分あり。

DB2 の関数	対応する Adaptive Server Anywhere の関数	備考
DBCLOB	CAST	暗黙の変換を実行。
DBPARTITIONNUM		非サポート。
DEC | DECIMAL	CAST	dec DECIMAL の UDF はキーワードであり、CAST が必要となる。暗黙の変換を実行。
DECRYPT_BIN | DECRYPT_CHAR	DECRYPT	二つのパラメータが必要。データベース接続変数の使用により DECRYPT_CHARまたは DECRYPT_BINを実装可能。暗号化アルゴリズムは異なるので、異機種環境では動作しない可能性あり。 DECRYPT_CHAR 上の UDF では CASTING が必要。
DEGREES	DEGREES	
DEREF		非サポート。
DIFFERENCE	DIFFERENCE	
DIGITS		非サポート。
各種DATALINK関数 (DLCOMMENT、DLVALUE等)		非サポート。
DOUBLE	CAST	
ENCRYPT	ENCRYPT	上述 DECRYPT も参照。HINTS は非サポート。
EVT_MON_STATE		非サポート。
EXP	EXP	
FLOAT	CAST	または暗黙の変換。
GETHINT		非サポート。
GENERATE_UNIQUE	NEWID	実装とアルゴリズムは異なるが、目的は同様。異なるデータ型が生成される。
GRAPHIC	CAST	暗黙の変換を実行。
HASHEDVALUE		非サポート。
HEX		
HOUR	HOUR	
IDENTITY_LOCAL_VAL	@@identity	@@identity を含む UDF。
INSERT	第三引数が0のときINSERT	第三引数が 0 以外の場合クエリの修正を要す。
INTEGER	CAST	
JULIAN_DAY		以下のものを使用するUDF。 DAYS('01-01-0001', NOW(*))+2451911.
LCASE/LOWER	LCASE/LOWER	
LEFT	LEFT	Adaptive Server Anywhereはキャラクタベース、DB2はバイトベース。
LENGTH	LENGTH	Adaptive Server Anywhereはキャラクタベースでデータ型は考慮せず。expr-typeの使用が望ましい。
LN/LOG	LOG	
LOG10	LOG10	
LOCATE	LOCATE	最初の二つの引数を交換する要あり。
LONG VARCHAR	CAST	暗黙の CAST も使用可能。
LONG VARGRAPHIC	CAST	暗黙の CAST も使用可能。

DB2 の関数	対応する Adaptive Server Anywhere の関数	備考
LTRIM	LTRIM	
MICROSECOND.		UDFにより実装可能。 例: SELECT right(NOW(*)||'000',6)
MIDNIGHT_SECONDS		下記の使用が望ましい。 SELECT seconds('10:10:11')- seconds(TODAY(*))
MINUTE	MINUTE	
MOD	MOD	
MONTH	MONTH	
MONTHNAME	MONTHNAME	
MULTIPLY_ALT		
NULLIF	NULLIF	
POSSTR	LOCATE	引数の交換を要す。
POWER	POWER	
QUARTER	QUARTE	
RADIANS	RADIANS	
RAISE_ERROR	RAISERROR	RAISERROR は関数ではなく文。
RAND	RAND	
REAL	CAST	代替的に暗黙の CAST 使用。
REC2XML		クエリは XMLELEMENTS 句または FOR XML 句で修正を要す。
REPEAT	REPEAT	
REPLACE	第三引数が0のときREPLACE	第三引数が 0 以外のときクエリの修正を要す。
RIGHT	RIGHT	
ROUND	ROUND	
RTRIM	RTRIM	
SECOND	SECOND	
SIGN	SIGN	
SIN	SIN	
SINH		外部ストアド・プロシージャの実装。
SMALLINT	CAST	代替的に暗黙の CAST 使用。
SOUNDEX	SOUNDEX	
SPACE	SPACE	
SQRT	SQRT	
SUBSTR	SUBSTR || SUBSTRING	
TABLE_NAME		
TABLE_SCHEMA		
TAN	TAN	
TANH		外部関数を介して実装。
TIME	CAST	代替的に暗黙の CAST 使用。
TIMESTAMP	CAST	代替的に暗黙の CAST 使用。
TIMESTAMP_FORMAT	CAST || CONVERT	UDF の実装も可能。CAST の結果は timestamp_format オプション設定に依存。
TIMESTAMP_ISO	CONVERT	代替的に timestamp_format オプション

DB2 の関数	対応する Adaptive Server Anywhere の関数	備考
		を使用。
TIMESTAMPDIFF	DATEDIFF	UDF と同時に実装可能。
TO_CHAR	CAST	CAST を使う UDF も使用可能。
TO_DATE	CAST || CONVERT	CAST を使う UDF も使用可能。date_format データベースオプション設定が dates のフォーマットに影響する可能性あり。
TRANSLATE		外部関数の実装が必要。
TRUNC || TRUNCATE	TRUNCATE || TRUNCNUM	
TYPE_ID		提供不可。
TYPE_NAME		提供不可。
TYPE_SCHEMA		提供不可。
UCASE || UPPER	UCASE || UPPER	
TO_VARCHAR	CAST	CAST を使う UDF も使用可能。timestamp_format データベースオプション設定が times のフォーマットに影響する可能性あり。
VALUE	COALESCE	
VARCHAR	CAST	暗黙の cast も可能。
VARCHAR_FORMAT	CONVERT || CAST	CAST を使用した UDF も使用可能。date_format データベースオプション設定が dates のフォーマットに影響する可能性あり。
VARGRAPHIC	CAST	暗黙の CAST も使用可。
WEEK	DATEPART	
WEEK_ISO		UDF も実装可能。
YEAR	YEAR	
TABLE関数	ストアド・プロシージャ	結果セットを返すストアド・プロシージャを介して実装可能。
OLAP関数	OLAP 構文の多くは Adaptive Server Anywhere と IBM で共通。. 例外についてはコメント参照。	本書の組み込み関数の項の「GROUPING」参照。
XML関数	XMLAGG、XMLCONCAT、XMLELEMENT、XMLATTRIBUTES、XMLFORESTをサポート	XML2CLOB、XMLSERIALIZE、XMLNAMESPACE は非サポート。Adaptive Server Anywhere では XML2CLOB と XMLSERIALIZE は不要。また暗黙の CAST もサポート。

データ定義言語

ここでは、データ定義レベルでのDB2とAdaptive Server Anywhereの機能の対応について説明します。IBMが提供するコンセプトの一部は、Adaptive Server Anywhereの管理思想に合致しないため、若干のDDLに変更を加える必要があるはずです。これは本書で述べる[データベース移行] ウィザードを利用することで容易に実行できます。

この項の内容は、既存のスクリプトの移行を行う際にお役に立つことでしょう。

ALTER TYPE ADD METHOD

Adaptive Server Anywhereではサポートされません。

COMMENT

Adaptive Server Anywhereでサポートを提供するオブジェクトタイプに限りサポートされます。

CREATE DISTINCT TYPE

Adaptive Server AnywhereではDISTINCT TYPEはサポートされません。ただし同様の機能を実現するためのドメインを生成することはできます。

Adaptive Server Anywhereのユーザ定義のデータ型については、ASA SQL リファレンス・マニュアル CREATE DOMAIN文を参照してください。

CREATE FUNCTION

本書の「関数、プロシージャ、トリガ」の項を参照してください。

CREATE INDEX

Adaptive Server AnywhereとDB2の構文と動作は大部分同じですが、以下に示す違いが存在します。

- Adaptive Server AnywhereはINCLUDE（column-list）句をサポートしません。
- DB2のCLUSTER句は、Adaptive Server AnywhereのCLUSTERED句に相当し、インデックス名の前に配置されます。
- SPECIFICATION ONLYは協調データベース（federated database）に適用されるものですが、Adaptive Server Anywhereのリモートデータ・アクセスではこのコンセプトをサポートしません。
- Adaptive Server Anywhereは構造データ型をサポートしないので、EXTEND USINGはサポートされません。

- Adaptive Server Anywhere ではREVERSE INDEX SCANS句は不要です。Adaptive Server Anywhereのすべてのインデックスは、後方スキャンの対象です。
- Adaptive Server Anywhereのインデックス定義では、PCTFREE句とMINPCTUSED句は使用できません。
- Adaptive Server AnywhereではPAGE SPLITTING句は使用できません。
- Adaptive Server Anywhereでは、インデックス統計の収集は自動的に実行されます。

CREATE NICKNAME | ALIAS

ニックネームは、Adaptive Server AnywhereのRemote Data Access機能に適用される協調システムで使われます。ニックネームの代わりにプロキシテーブル（CREATE EXISTING TABLE）を使うこともできます。

Adaptive Server AnywhereのビューにはDB2のビューにあるオーバーヘッドがないため、CREATE ALIASは不要です。エイリアスの代わりにビューを生成することも可能です。

CREATE PROCEDURE

本書の「関数、プロシージャ、トリガ」の項を参照してください。

CREATE SCHEMA

SCHEMA関係の構文は、Adaptive Server AnywhereとDB2で異なります。DB2は暗黙的にユーザを生成し、新しいユーザの下でオブジェクトを生成します。Adaptive Server Anywhereは、オブジェクト生成のコンテクストとしてカレント・ユーザを使います。混乱を回避するには、スキーマ名として接続するか、またはテーブルを十分に制限してください。SCHEMA名は、Adaptive Server Anywhereの既存のデータベースユーザかまたはグループでなくてはなりません。Adaptive Server Anywhereの構文は、SCHEMAとAUTHORIZATIONに別の名称を使用せず、かつトークンAUTHORIZATIONが必要とされる点でDB2と異なります。以下の例をご覧ください。

```
CREATE SCHEMA AUTHORIZATION sample_user
CREATE TABLE sample_user.t1 ( id1 INT PRIMARY KEY )
CREATE TABLE sample_user.t2 ( id2 INT PRIMARY KEY );
```

CREATE SEQUENCE

Adaptive Server AnywhereはSEQUENCEをサポートしません。SEQUENCEの代わりにDEFAULT AUTOINCREMENT、DEFAULT GLOBAL AUTOINCREMENT、またはget_identityを使ってください。コードの書き換えが必要となるはずです。

CREATE SERVER

Adaptive Server AnywhereのCREATE SERVER文はDB2とは構文が異なりますが、同様の機能を提供します。

CREATE TRANSFORM

Adaptive Server Anywhereは構造データ型をサポートしません。

CREATE TRIGGER

本書の「関数、プロシージャ、トリガ」の項を参照してください。

CREATE TABLE

CREATE TABLE文は複雑ですが、Adaptive Server AnywhereとDB2は、数多くのANSI標準SQLコンストラクトを共通で使用しています。本書の「データ型」の項も参照してください。

以下の表は、Adaptive Server AnywhereとDB2で共通の構文と機能について、両製品の相違点をまとめたものです。

表9. CREATE TABLE文の構文の対応

DB2 の構文	対応する Adaptive Server Anywhere の構文	備考
CREATE TABLE LIKE ...	SELECT INTO *new-tablename* FROM *existing-tablename* WHERE 1=0	SELECT INTO *table-name* を CREATE TABLE 文の代わりに使用可能。
CAPTURE DATA CHANGES log extra information	ALTER TABLE table-name REPLICATE ON	SQL Anywhere Studio は三種類の複製デクノロジーを提供。
IN tablespace-name	IN dbspace-name	
CONSTRAINT constraintname CHECK(columnname DETERMINED BY column-name)		非サポート
CONSTRAINT constraintname CHECK(searchcondition)	CONSTRAINT constraintname CHECK(searchcondition)	検索条件として ASA は特殊レジスタ、システム関数、直定数、DB2 はさらに多くの条件をサポート。CHECK 制約の代わりにトリガを実装可能。
CONSTRAINTS . . . . ENABLE QUERY OPTIMIZATION		非サポート
CONSTRAINTS . . . . [NOT] ENFORCED		データベースオプション wait_for_commit または check_on_commit も参照。
GENERATE ALWAYS AS	COMPUTE	
DEFAULT GENERATED AS IDENTITY	DEFAULT AUTOINCREMENT	[NO] CYCLE、[NO] CACHE、[NO] ORDERはサポートせず、また Adaptive Server Anywhere の GLOBAL AUTOINCREMENT は以下の句が必要。 [NO] MINVALUE [NO] MAXVALUE

DB2 の構文	対応する Adaptive Server Anywhere の構文	備考
		START WITH INCREMENT BY かわりにget_identityも使用可能。
OPTIONS (協調サーバオプション)		ASA SQL ユーザーズ・ガイド プロキシ・テーブルの編集を参照。

Adaptive Server AnywhereでサポートされないCREATE TABLEまたはALTER TABLE中の機能カテゴリと、それらに関係する句を以下に示します。

1. マテリアライズド・ビュー
    a. WITH NO DATA
    b. INCLUDE | EXCLUDE COLUMN DEFAULTS
    c. INCLUDE | EXCLUDE COLUMN ATTRIBUTES
    d. ENABLE QUERY OPTIMIZATION
    e. MAINTAINED BY SYSTEM | USER | FEDERATED TOOL
    f. DATA INITIALLY DEFERRED
    g. REFRESH INITIALLY DEFERRED
2. 構造化テーブルおよび構造データ型の句
    a. OF
    b. INLINE LENGTH
    c. HIERARCHY
    d. SCOPE
    e. UNDER
    f. INHERIT SELECT PRIVILEGES
    g. REF
    h. REF IS .. USER GENERATED
3. テーブルの物理レイアウト
    a. NOT LOGGED INITIALLY

b. 分割KEY

ｉ）REPLICATED

c. COMPRESS SYSTEM DEFAULT

d. LONG IN

e. INDEX IN

f. DATALINK格納オプション

ｉ）LINK TYPE URL、LINK CONTROL、FILE LINK CONTROL、INTEGRITY、ALL、READ PERMISSION DB | FS、WRITE PERMISSION BLOCKED | FS | ADMIN [NOT] REQUIRING TOKEN UPDATE、RECOVERY YES | NO、ON UNLINK RESTORE | DELETE、MODE DB2OPTIONS

g. LOB格納オプション

ｉ）[NOT] COMPACT、[NOT] LOGGED

h. CCSID

ｉ）dbinitによる制御

i. ORGANIZE BY DIMENSION | KEY SEQUENCE DB2のkey-sequence-clause、DISALLOW | ALLOW OVERFLOW

4. WITH RESTRICT ON DROP

CREATE TABLESPACE

Adaptive Server Anywhereのdbspaceは、DB2のtablespaceにきわめて近いコンストラクトです。これによって管理者は、デバイスにまたがってデータベースを設置することができます。dbspaceは、未加工のパーティションではなく、オペレーティングシステムの管理するファイルをサポートします。dbspaceページのサイズは、データベースの初期設定に依存します。サイズパラメータEXTENTSIZEとPREFETCHは、Adaptive Server Anywhereのデータベースサーバが自動的に決定します。

dbspace上での転送レートは、、ALTER DATABASE文を使って決めることができます。Adaptive Server Anywhereのデータベースは自動的に復旧するよう設計されており、設定を行うことはできません。またAdaptive Server Anywhereでは、テンポラリdbspaceの生成は必要ありません。

詳細についてはASA SQL ユーザーズ・ガイド CREATE DBSPACE文を参照してください。

CREATE TYPE

Adaptive Server Anywhereは構造データ型をサポートしません。

CREATE USER MAPPING

同様のコンセプトについて、ASA SQL リファレンス・マニュアル CREATE EXTERNLOGIN文を参照してください。

CREATE VIEW

Adaptive Server AnywhereとDB2のCREATE VIEW文は、ビューのリスティングカラムとWITH CHECK OPTION句が共通です。CREATE VIEW構文の大部分はSELECT文に依拠します。

詳細については「SELECT」の項を参照してください。

CREATE WRAPPER

Adaptive Server Anywhereは、リモートデータ・アクセス用にあらかじめ定義されたCLASSを提供します。したがってこの構文はAdaptive Server Anywhereには適用できません。

DROP

Adaptive Server Anywhereのサポート対象のオブジェクトタイプに限り、サポートされます。

RENAME

Adaptive Server Anywhereでは、ALTER TABLEを使ってテーブルの名称を変更できます。TOキーワードの削除とALTER TABLEキーワードの追加という構文変更が必要となるはずです。インデックスを改名するには、ALTER INDEX構文にALTERキーワードを加えなければなりません。

RENAME TABLESPACE

Adaptive Server Anywhereでは、DBSPACE用ファイルの保存場所はALTER DBSPACEで変更できます。dbspaceの名称を正しく変更するには、dbspaceをいったん削除して再度生成する必要があります。オリジナルのファイルの名称は変更できません。

管理言語

この項では、両製品の下位レベルの管理機能について説明します。SQL Anywhereでは、管理言語の複雑さを減じる方針を採用しており、ここで言及する句をサポートしないことをむしろ利点であると考えています。以下では若干の管理SQLについて、それがAdaptive Server Anywhereにはどう当てはまるかを示します。

CREATE BUFFERPOOL

Adaptive Server Anywhereの管理では、BUFFERPOOLによる管理要求は必要ありません。Adaptive Server Anywhereは、WindowsおよびUNIX上で動的にキャッシュをリサイズできます。

CREATE DATABASE PARTITION GROUP

Adaptive Server Anywhereはデータベースの分割をサポートしません。

CREATE EVENT MONITOR | SET EVENT MONITOR STATE | FLUSH EVENT MONITOR

DB2のイベントモニタは、パフォーマンスチューニング機能とシステム保守／監視機能を提供します。またDB2のCREATE EVENT MONITOR文は、主として特定のイベントに関する統計の収集を実現します。なおDB2には、他にも保守とモニタリングのためのシステム監視機能があります。

Adaptive Server AnywhereとDB2では、パフォーマンスチューニングの設計思想が根本的に異なります。Adaptive Server Anywhereは設定不要で動作するように設計されていますが、一方のDB2は、CREATE EVENTなど、DBAからパフォーマンスの監視を実現するツールを提供しています。一方のAdaptive Server Anywhereは、PROPERTY関数を使うSQLプロパティ（Sybase Centralでは統計と呼ばれる）、sa_conn_properties、sa_db_properties、sa_eng_propertiesストアドプロシージャにより、データベースの動作を監視する機能を提供しています。

Adaptive Server Anywhereでは、データベース保守の一部をCREATE EVENT文によって自動化します。ディスクスペースの監視、バックアップの実行、テーブルの再構成といったデータベース保守動作は、Adaptive Server AnywhereとDB2で共通です。

FLUSH PACKAGE CACHE

Adaptive Server Anywhereは全キャッシュのフラッシングをサポートします。これはsa_flush_cache（）プロシージャを使う試験で有用です。

REFRESH TABLE

Adaptive Server Anywhereはマテリアライズド・ビューをサポートしません。

GRANT/REVOKE

Adaptive Server Anywhereは、データベース・オーソリティ、ルーチン、テーブル／ビュー特権のサブセットをサポートします。しかしAdaptive Server Anywhereはインデックス、パッケージ、スキーマ、シーケンス、サーバ、tablespace等については特権を必要とせず、またサポートも提供しません。

Adaptive Server AnywhereとDB2のデータベースオーソリティ特権の対応を以下に示します。

表9. データベースオーソリティ特権の対応

DB2	Adaptive Server Anywhere	備考
CONNECT	CONNECT	
CREATETAB	RESOURCE	
CREATE_EXTERNAL_ROUTINE	RESOURCE	
CREATE_NOT_FENCED_ROUTINE	RESOURCE	Adaptive Server Anywhereは、FENCED ルーチンのコンセプトをサポートしない。
DBADM	DBA	
LOAD		許可はサーバオプションで設定。
QUIECSE_CONNECT		

Adaptive Server Anywhereへの移行に際しては、ON DATABASE句を削除する必要があります。またAdaptive Server Anywhereは、許可がユーザとグループのどちらに該当するかを自動的に判断するので、構文には、USERおよびGROUPキーワードは含まれません。

ルーティン特権は、Adaptive Server Anywhereのストアド・プロシージャと関数に適用できます。メソッドはサポートされません。データベースが自動的にタイプを決定するので、この構文ではキーワードPROCEDUREとFUNCTIONは使用できません。

Adaptive Server AnywhereとDB2の間のテーブル／ビュー特権の大まかな対応を以下に示します。USERおよびGROUPキーワードは、Adaptive Server Anywhereの構文に含まれません。両データベースでWITH GRANT OPTIONがサポートされます。

表10. テーブル／ビュー特権の対応

DB2	Adaptive Server Anywhere	備考
ALL	ALL	
CREATETAB	ALTER	
CONTROL		GRANT ALTER、DELETE、INSERT、REFERENCES、SELECT、UPDATE に近い。主な違いはユーティリティ関数の用法（特に LOAD、REORG、RUNSTATS、SET INTEGRITY）。一般に ASA はこれらの関数を自動的に制御するが、そうでない場合はデータベース・ワイドのオプションで管理する。ASA の REORGANIZE テーブルは DBA またはテーブルオーナのパーミッションを要する。
INDEX	REFERENCE	
INSERT	INSERT	
REFERENCES	REFERENCES	
SELECT	SELECT	
UPDATE	UPDATE	

なおBY ALL構文はAdaptive Server Anywhereではサポートされないので、REVOKE文から削除する必要があります。

[データベース移行]ウィザードの使用

構造データ型、リファレンス型、XML型、空間データ型、DATALINKデータ型といった対応するSQLが存在しないDB2固有のデータ型を使っていなければ､わずかな作業でDB2からAdaptive Server Anywhereへデータを移行できます。上述のデータ型を使っている場合、データベースの移行にあたって若干の検討が必要です。

データベースの移行は、Sybase Centralの[データベース移行]ウィザードで実行できます。またAdaptive Server Anywhereのストアド・プロシージャのsa_migrateセットを使い、より高度な移行を行うことも可能です。さらにAdaptive Server AnywhereのLOAD TABLE文とDB2のEXPORTコマンドを組み合わせてデータを移行することもできます。DATALINKデータ型を使っているアプリケーションの場合は、DB2からAdaptive Server Anywhereへのデータベース移行に際して特に注意が必要です。

NOTE DB2 UDBとAdaptive Server Anywhereのデータ型の対応の詳細については、ASA SQLユーザーズ・ガイド データ型変換：DB2を参照してください。

前提条件

本項の記述は、下記の条件が満たされていることを前提としています。

- サポート対象のプラットフォーム上で稼動するDB2データベースが存在し、またサポート対象WindowsプラットフォームにAdaptive Server Anywhere 9.0.1以上がインストール済みであること。
- DB2のサンプルデータベースを作成していない場合は、db2sampleを稼動して移行手順を進めることができること。
- DB2 ODBC 3.5.2以上のドライバが、Adaptive Server Anywhereデータベースを実行するコンピュータにインストール済みであること。

Adaptive Server Anywhereデータベースの作成手順

まずDB2のデータベースの移行先となるAdaptive Server Anywhereデータベースを作成する必要があります。以下にSybase Centralで新しいデータベースを作成する手順を示します。

1. Sybase Centralを起動する。
   スタートメニューから[すべてのプログラム]→[SQL Anywhere 9]→[Sybase Central]を選択する。
2. 新しいAdaptive Server Anywhere 9データベースを作成する。
   A Sybase Centralの左のペインでAdaptive Server Anywhere 9を選択する。
   B 右のペインで［ユーティリティ］タブをクリックする。
   C ［データベースの作成］をダブルクリックする。

［データベースの作成］ウィザードが表示されます。

D　　ウィザードの指示に従い、新しいデータベースを作成する。

DB2データベースのデータソースの作成

移行では、ソースデータベースへのODBC接続が必要となります。したがってまずDB2データベース用のODBCデータソース（DSN）を作成する必要があります。

1. ODBCアドミニストレータを起動する。
   スタートメニューから［すべてのプログラム］→［SQL Anywhere 9］→［Adaptive Server Anywhere］→［ODBCアドミニストレータ］を選択する。
   ［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。

2. ［追加］をクリックする。
   ［データソースの新規作成］ウィザードが表示されます。

3. 利用可能なドライバのリストから、ドライバとして［iAnywhere Solutions 9 - DB2 Wire Protocol］を選択し、［完了］をクリックする。
   ［DB2 Wire Protocol Driver Setup］ダイアログが表示されます。

4. ［Data Source Name］フィールドでデータソースの名称を入力する。
   この例では、データソースを**DB2Migrate**と命名します。

5. 他のフィールドに、DB2データベースに必要な適切な値を入力する。

6. ［ODBC］タブ上で［Test Data Source］をクリックし、データソースが正しく作成されたことを確認する。

7. ［OK］をクリックする。

DB2データベースのAdaptive Server Anywhereへの移行

新しいAdaptive Server Anywhereデータベースに移行するには、まずAdaptive Server Anywhereのデータベースに接続しなければなりません。以下にデータベースファイルによる接続方法を説明します。

Adaptive Server Anywhereデータベースとの接続

1. Sybase Centralの左のペインで［Adaptive Server Anywhere 9］を、また［ファイル］メニューから［接続］を選択する。
   ［接続］ダイアログが表示されます。

2. ［ID］タブ上で、データベースの正規のユーザIDとパスワードを入力する。
   すべてのAdaptive Server Anywhereデータベースには、デフォルトでユーザID「DBA」、パスワード「SQL」が与えられています。

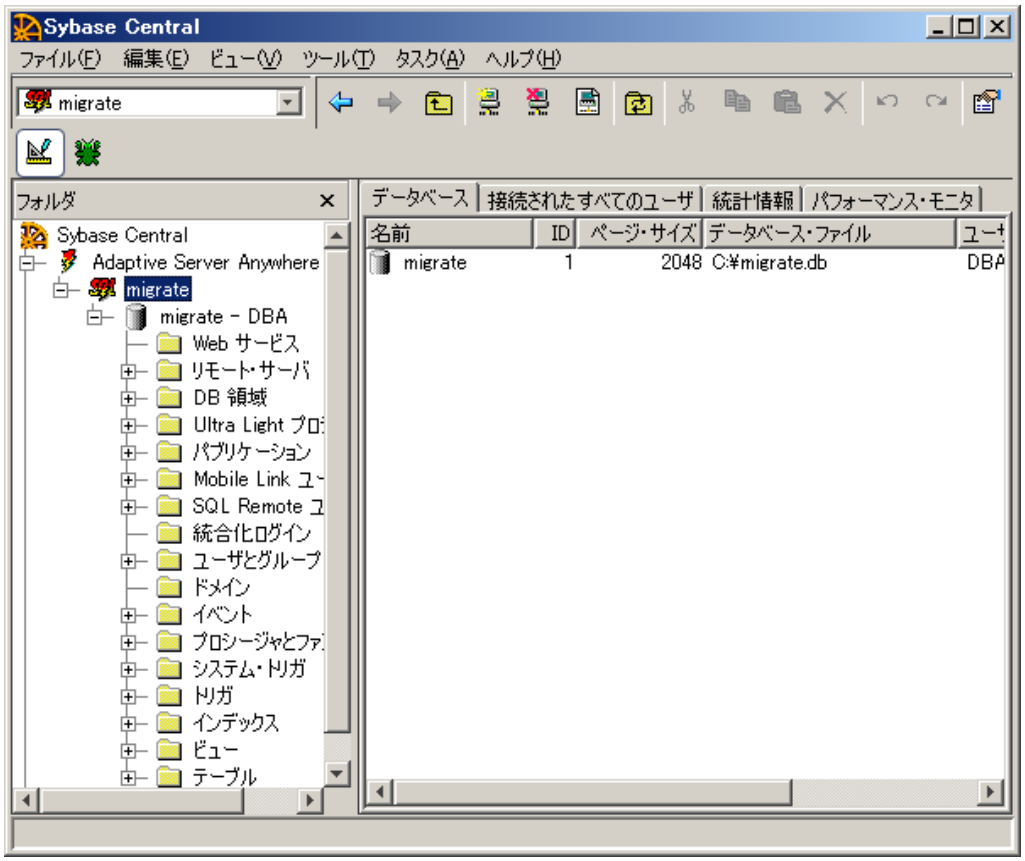
3. ［Database］タブ上で［検索］ボタンをクリックし、作成済みのAdaptive Server Anywhereデータベースファイルを選択する。

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

リモート・サーバの作成

次にリモート・サーバを作成し、DB2データベースのロケーションをSybase Centralに設定します。

1. Sybase Centralの左ペインで、データベースサーバとデータベースアイコンを展開する。
   下記の例では、「migrate」というデータベースサーバ上でデータベース「migrate」が稼動しています。



2. Sybase Centralの左ペインで、［リモート・サーバ］フォルダを選択する。

3. ［ファイル］メニューから［新規］→［リモート・サーバ］を選択する。
   ［新しいリモート・サーバの作成］ウィザードが表示されます。

4. ウィザードの指示に従い、DB2データベースに接続するリモートサーバを作成する。

   A　ウィザードの最初のページでリモート・サーバの名前（例: DB2Migrate）を入力し、［次へ］をクリックする。

B　リモート・サーバのタイプとして［一般］を選択し、［次へ］をクリックする。

C　［オープン・データベース・コネクティビティ（ODBC）］オプションを選択し、接続情報フィールドにDB2データベースのODBCデータソース名を入力する。
この例では、ODBCデータソースの作成時に指定した名称「DB2Migrate」と入力します。

5. ［完了］をクリックする。
新しいリモート・サーバがSybase Central上に表示されます。

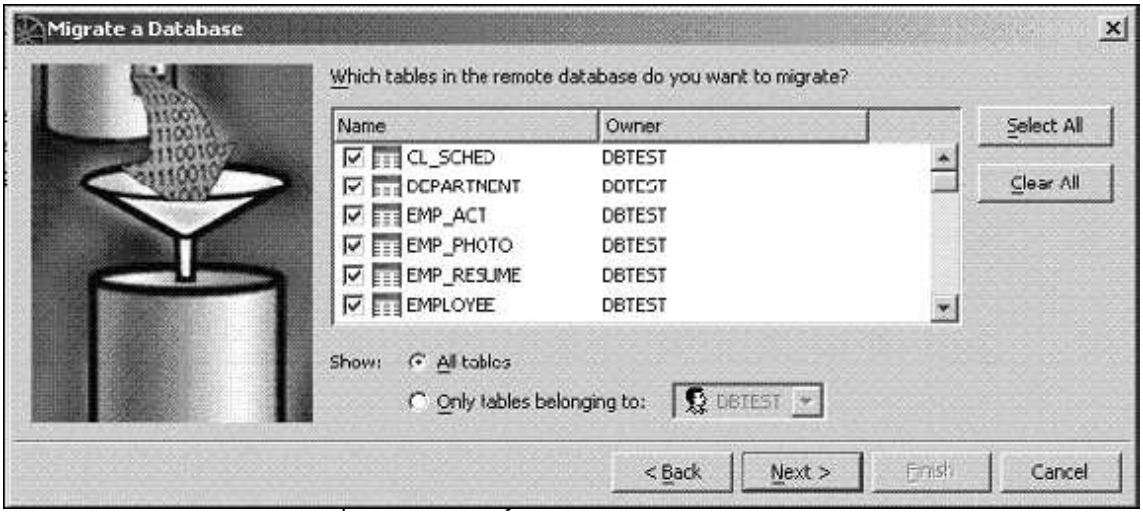
外部ログインの作成

Adaptive Server Anywhereデータベースへの接続用ユーザIDを持つユーザをリモート・サーバに定義していないときは、カレント・ユーザ用の外部ログインを作成する必要があります。
つまりAdaptive Server Anywhereデータベースにユーザ ID「DBA」で接続できる一方、DB2データベースにユーザID「DBA」が設定されていない場合は、外部ログインを作成しなければなりません。

1. Sybase Centralの左ペインで［リモート・サーバ］フォルダを開き、リモート・サーバを選択する。
2. 右のペインで［外部ログイン］タブをクリックする。
3. ［ファイル］メニューから［新規］→［外部ログイン］を選択する。
［新しい外部ログインの作成］ウィザードが表示されます。
4. ユーザのリストから現在接続しているユーザを選択し、［次へ］をクリックする。
5. ［ログイン名］フィールドにDB2データベースのユーザ名の一つを入力し、［パスワード］および［パスワードの確認］フィールドにこのユーザのパスワードを入力し、［完了］をクリックする。

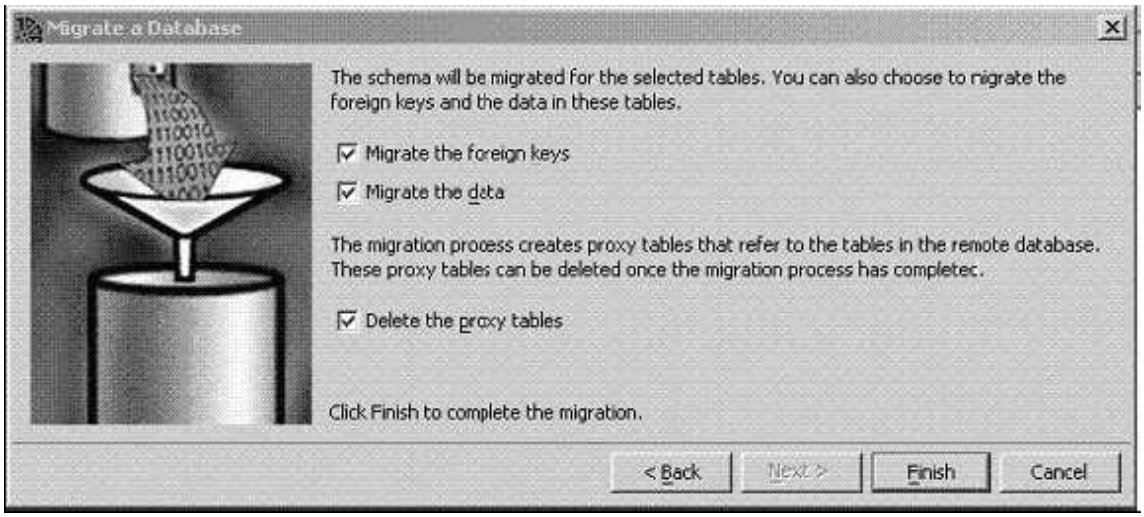
DB2データベースの移行

1. Sybase Centralの左ペインで、Adaptive Server Anywhereのデータベースを選択する。
2. ［ファイル］メニューから、［データベースの移行］を選択する。
［データベースの移行］ウィザードが表示されます。
3. 最初のページで［次へ］をクリックする。
4. 現在のデータベースを選択し、［次へ］をクリックする。
5. 作成したDB2リモート・サーバ（この例ではDB2Migrate）を選択し、［次へ］をクリックする。
6. すべてのDB2テーブルをAdaptive Server Anywhereデータベースに移行させる場合、［全て選択］を選択して［次へ］をクリックする。



クする。

8. 移行のオプションを選択する。



9. ［完了］をクリックして移行を開始する。
［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。

DB2からAdaptive Server Anywhereへの アプリケーションの移行

DB2からAdaptive Server Anywhereへのアプリケーションの移行方法は、DB2アプリケーションへのアクセスに使われているインタフェースに応じて異なります。この項では、最小限の作業で移行できる一般的なインタフェースの例を説明します。

接続について

DB2のデータベース・エイリアスのコンセプトは、ODBCデータソースに非常によく似ています。Adaptive Server Anywhereは、サポート対象のすべてのAPIについて、ODBCデータソースを介する接続をサポートします。DB2データベース・エイリアスを使うアプリケーションであれば、接続コードをDB2エイリアスからAdaptive Server AnywhereおよびODBCデータソースへと移行するのはごく簡単です。またAdaptive Server AnywhereのAPIの大半はODBCデータソース情報を保持するため、アプリケーション間で接続を移行するのは非常に容易です。

CLI/ODBCアプリケーションの移行

Adaptive Server AnywhereとDB2は、どちらもODBC API仕様をサポートします。一般に、両者間でのアプリケーションの移行に際しては、DB2ではなくAdaptive Server Anywhereをポイントするよう、ODBCデータソースを変更する必要があります。一部のAPI関数の実装が多少異なる可能性はありますが、ODBC仕様は十分成熟しているので、それが大きな問題になることは考えられません。

DB2とAdaptive Server AnywhereはいずれもC/C++でODBC APIをサポートします。DB2の仕様では、このAPIをCLIと呼んでいます。DB2からのアプリケーションを移行に際しては「データ操作言語」の項もご覧ください。

なお以下のCLI関数はAdaptive Server Anywhereではサポートされません。

- SQLBindFiletoCol
- SQLBindFiletoParam
- SQLExtendedBind
- SQLExtendedPrepare
- SQLGetDataLinkAttr
- SQLGetLength
- SQLGetPosition
- SQLGetSQLCA

- SQLGetSubstring
- SQLNextResult
- SQLSetCollAttributes
- SQLSetConnection

Javaアプリケーションの移行

DB2はJDBCバージョン3.0をサポートし、タイプ2およびタイプ4 JDBCドライバ（100% Java実装）として動作するユニバーサルドライバを持っています。Adaptive Server Anywhereへの移行にあたり、100% Java実装が必要とされる場合は、Sybase jConnectドライバを使用する必要があります。ただしAdaptive Server Anywhereの性能を最大限に発揮したいのであれば、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2 JDBCドライバです。このドライバは、JDBC 2.0仕様のすべてのコア仕様と若干のオプション仕様をサポートします。

現時点ではAdaptive Server AnywhereはJDBC 3.0ドライバを持っていません。SAVEPOINTあるいはHOLDABLEカーソル（バージョン3.0固有）を使うアプリケーションのために、Adaptive Server Anywhereには、SQL savepoint構文と、トランザクション完了時にカーソルをオープン状態に保つオプション（CLOSE_ON_ENDTRANS）が用意されています。

Adaptive Server AnywhereはSQLJをサポートしません。DB2と同様、Adaptive Server Anywhereはクラス名「com.sybase.jdbc2.jdbc.SybDriver」のタイプ4ドライバ（jConnect）と、クラス名「ianywhere.ml.jdbcodbc.Idriver」のタイプ2 JDBCドライバ（iAnywhere JDBCドライバ）を提供します。これらのドライバのURLは異なります。

PHPアプリケーションの移行

DB2からAdaptive Server AnywhereへのPHPアプリケーションの移行は簡単です。ODBCを使ってAdaptive Server Anywhereに接続するオプションを使ってもよいし、米国iAnywhere Solutions,Incの[iAnywhere Developerウェブサイト]からダウンロード可能なSQL Anywhere PHPモジュールを使うこともできます。

IBMのサンプルはODBC PHPを利用しています。Adaptive Server AnywhereとDB2はどちらもODBCドライバを提供するので、PHPアプリケーションがすでにDB2データベースへの接続にODBCを使っているのであれば、関数呼び出しを変更する必要はありません。

以下の表に、PHPアプリケーションに関係する関数（管理関連の関数除く）について、DB2とAdaptive Server AnywhereのPHPドライバの対応を示します。

表11. PHPドライバの対応

DB2	Adaptive Server Anywhere
db2_autocommit	sqlanywhere_set_option($conn, “ auto_commit ” , “off”);
db2_bind_param	
db2_close	sqlanywhere_disconnect
db2_commit	sqlanywhere_commit
db2_conn_error	sqlanywhere_errorcode
db2_conn_errormsg	sqlanywhere_error
db2_connect	sqlanywhere_connect
db2_cursor_type	
db2_exec	sqlanywhere_query
db2_execute	sqlanywhere_query または sqlanywhere_execute カーソルに対するパラメータオプションは、SELECT または SET OPTION 文の句で指定可能
db2_fetch_array	sqlanywhere_fetch_array relative fetch については、sqlanywhere_data_seek を参照
db2_fetch_assoc	sqlanywhere_fetch_array
db2_fetch_both	sqlanywhere_fetch_array
db2_fetch_object	sqlanywhere_fetch_object、sqlanywhere_fetch_field
db2_fetch_row	sqlanywhere_fetch_row db2_result はアレイとして管理
db2_field_displaysize (設計関数かどうか不明)	
db2_free_result	sqlanywhere_free_result
db2_free_stmt	sqlanywhere_free_result
db2_next_result	
db2_num_fields	sqlanywhere_num_fields
db2_num_rows	sqlanywhere_num_rows
db2_pconnect	sqlanywhere_pconnect
db2_prepare	
db2_result	上述の db2_fetch_row を参照
db2_rollback	sqlanywhere_rollback
db2_special_columns	
db2_stmt_error	sqlanywhere_error sqlanywhere_errorcode
db2_stmt_errormsg	sqlanywhere_error sqlanywhere_errorcode

db2_primary_key、db2_foreign_key、db2_column、db2_tableといった設計関数、管理関数は、sp_column等のシステムテーブルやシステムプロシージャに対してsqlanywhere_queryでクエリを行うことで移行できます。

Perlアプリケーションの移行

DB2からAdaptive Server AnywhereへのPerlアプリケーションの移行は非常に簡単です。DBD::ODBCドライバとDBD:DB2ドライバのどちらも使うことができます。ただしいずれにしてもDBD::ODBCドライバあるいはAdaptive Server Anywhereと共に提供されるネイティブドライバ（DBD::ASAnyと呼ばれる）により、接続ストリングを変更する必要があるはずです。

接続ストリングの変更した場合であっても、本書の「データ操作言語」の項で述べたAdaptive Server AnywhereとDB2の違いに基づく多少の問題が生じる可能性がありますが、移行に要する作業量はわずかで済みます。

.NET、ADO、RDOアプリケーションの移行

これらのインタフェースは、作成可能な関数呼び出しに関してよく定義されているので、移行作業の大半は共通のデータ操作言語がベースとなります。必要となる作業はSQLとエラー処理に関するものとなるはずです。DB2 ADO、ADO.NET、RDOからAdaptive Server Anywhereへのアプリケーションの移行では、本書の「データ操作言語」の項もご参照ください。

ADOとOLE DBの場合は、おそらくdatasource名をAdaptive Server Anywhereに適したものに変更する必要があります。ただしMSDASQLをご使用であれば、OLE DBプロバイダを変更する必要はありません。IBMDADB2を使っている場合は、Adaptive Server AnywhereのOLE DBプロバイダの名前としてASAPROVを使ってください。

ADO.NETアプリケーションを移行する場合も、おそらくdatasource名をAdaptive Server Anywhereに適したものに変更する必要があります。ADO.NETアプリケーションの変更では、若干の検索・置換操作が必要になります。

表12. 置換が必要な名称

DB2 .NET プロバイダ	Adaptive Server Anywhere .NET データプロバイダ
Imports IBM.Data.DB2	Imports iAnywhere.Data.AsaClient
DB2Type.Clob	AsaDbType.LongVarchar
DB2Type	AsaDbType
DB2Type.Blob	AsaDbType.LongBinary
DB2	Asa
DB2Connection	ASAConnection
DB2Transaction	ASATransaction
DB2DataReader	ASADataReader
DB2Command	ASACommand
.GetDate	GetDateTime
Server= （接続用）	EngineName=
Database=	DataSourceName

RDOおよびADOアプリケーションは単純にODBCを使うので、ODBCのdatasource名を変更するだけで済むはずです。

Embedded SQLアプリケーションの移行

Adaptive Server AnywhereとDB2は、いずれも動的SQLと静的（Embedded）SQLにC/C++のAPIを提供します。Embedded SQLのAPIはANSI標準に準拠するので、Adaptive Server AnywhereとDB2の静的な構文の大半は共通です。双方の製品で、動的SQL向けにSQL Descriptor Area（sqlda）がサポートされています。ただしCタイプのSQLに対するC/C++の定義と動的SQLのエラー定義が異なるので、移行はやや複雑で困難を伴います。

本項と本書の「データ操作言語」の項は、DB2からAdaptive Server AnywhereへのEmbedded SQLアプリケーションの移行の参考となるはずです。

Adaptive Server AnywhereのEmbedded SQLプリプロセッサ（sqlppユーティリティ）は、prepコマンドと同じ機能を提供します。sqlppは前処理として、Embedded SQLファイルをデータベース接続なしでC/C++へと変換します。

Adaptive Server AnywhereはグローバルSQLCAを想定しているので、SQL関数を伴うsqlca宣言をSQLで個々に記述する必要があるかも知れません。またSQLCAの正当性と最終形を次の方法でチェックすることを推奨します。

```
If (!db_init(&sqlca) ) {
Return FALSE;
}
...
dbfini(&sqlca);
```

両製品は共にマルチスレッドのアプリケーションをサポートします。Adaptive Server Anywhereは構文EXEC SQL SET SQLCA sqlcaを使い、アクティブ・リクエストに使用するSQLCAを制御します。

以下にアプリケーションの移動に伴うEmbedded SQL構文への影響につき説明します。

BEGIN DECLARE SECTION/END DECLARE SECTION

Adaptive Server Anywhereはこの構文をサポートします。Adaptive Server AnywhereはC/C++のプリミティブデータ型を許容するので、注意が必要です。「“SQL TYPE is . . .」であれば、「BLOB」や「CLOB」ではなく「DECL_LONGVARCHAR」や「DECL_LONGBINARY」への移行が必要です。Adaptive Server Anywhereは、現時点でBLOBファイル、CLOBファイル、LOBインジケータ値をサポートしていません。

COMPOUND SQL

Adaptive Server AnywhereのEmbedded SQLでは複合文をサポートしません。

この種のSQLは、Embedded SQL中でストアド・プロシージャを作成するために使われている可能性があります。アプリケーションが複合文を使っている場合は、本書の「関数、プロシージャ、トリガ」の項を参照してください。

EXPLAIN

EXPLAIN文は句の性質に応じてまったく異なるものとなります。Adaptive Server AnywhereとDB2は管理要件に関して異なる設計思想を採用しているため、クエリチューニングも異なります。

EXECUTE

DB2では、INPUTおよびOUTPUT記述子を一つのEXECUTE文中に記述できます。Adaptive Server Anywhereで同じオペレーションを実行しようとすると、二つのEXECUTE文が必要となります。

EXECUTE IMMEDIATE

製品間に構文上の違いは存在しません。

INCLUDE

Adaptive Server AnywhereとDB2は、共にEXEC SQL INCLUDE SQLCA | SQLDAをサポートします。ただしDB2のサンプルは、一般に‘include“sqlca.h”’をC/C++の形で提供します。Adaptive Server Anywhereは、Embedded SQL型のINCLUDEのみをサポートします。Adaptive Server AnywhereのSQLCAはglobal externalですが、DB2のSQLCAはexternalではありません。Adaptive Server Anywhereで使用するために、SQLCAを移転する必要があるかもしれません。スレッドセーフなアプリケーションについては別途検討する必要がありますのでご注意ください。

詳細についてはASA SQL リファレンス・マニュアル INCLUDE文を参照してください。

OPEN

製品間に構文上の違いは存在しません。

PREPARE

Adaptive Server Anywhereでは、この文のFROM句の位置が異なる上、DESCRIBEキーワードが必要となります。DB2では、INPUTおよびOUTPUT記述子に一つのPREPAREを使うことができますが、Adaptive Server Anywhereで同じオペレーションを実行するためには、二つのPREPARE文が必要となります。

DB2構文の一例を下記に示します。

```
EXEC SQL PREPARE statement-name
OUTPUT INTO result-descriptor-name
FROM host-variable;
```

Adaptive Server Anywhereで上記に対応するのは以下の構文となります。

```
EXEC SQL PREPARE statement-name FROM host-variable
DESCRIBE OUTPUT INTO result-descriptor-name;
```

RELEASE

Adaptive Server Anywhereにはrelease pending stateは存在しませんが、大半の移行では、DISCONNECT構文で十分代行できます。

SELECT

詳細については「SELECT」の項を参照してください。

SELECT INTO

Adaptive Server Anywhereでサポートされます。

詳細については「SELECT」の項を参照してください。

SET CONNECTION

DB2がサーバ名を使うところでAdaptive Server Anywhereは接続名を使いますが、両製品の構文は同一です。

VALUES INTO

本書の「VALUES [INTO]」の項を参照してください。

WHENEVER

Adaptive Server Anywhere のWHENEVER文は、FOUNDとGO TOをそれぞれ独立したトークン（NOTFOUND、GOTO）としてサポートします。

Embedded SQLの構築

上位レベルでは、サンプル用のDB2 bldappは、次のテキストボックスに表示されます。

```
Db2 prep myfile.sqc bindfile
```

Adaptive Server Anywhereのプリプロセッサコマンドは次のとおりです。

```
Sqlpp myfile.sqc myfile.cpp
```

コンパイルして連結する必要があるライブラリdblib9は、win32¥libフォルダまたはlibフォルダ（UNIX）にあります。ビルド作業時には、SQL Anywhere Studioで実装されたsamplesディレクトリをご覧ください。

法的注意

アイエニウェア・ソリューションズ株式会社
http://www.ianywhere.jp/