

Real Alternative DBMS ALTIBASE, Since 1999

MS SQL SERVER to ALTIBASE 변환 가이드

ALTIBASE 6

2013.02

Document Control

Change Reason

Date	Author	Change Reference
2012-02-28	JHJEONG	Created

Reviews

Date	Name (Position)

Distribution

Name	Location

목차

개요	5
OBJECT 변환	6
DATATYPE	6
FUNCTION & EXPRESSION	7
OBJECT	11
CREATE TABLESPACE	11
CREATE TABLE	12
CREATE USER	15
CREATE INDEX	15
SQL Conversion	17
1. JOIN	17
2. Execution Query	17
3. Temporary Table	17
4. Control Statement	17
5. Identity attribute	18
6. Procedure Conversion	18
1. Parameter Declaration	18
2. Variable Declaration	19
3. ASSIGNMENT	20
4. Control_flow – IF 문	21
5. SELECT 절	21
6. Exception Handler	22
7. SP Call	23
8. Functions	24

9. Etc.....	25
7. Using Cursor.....	26

개요

본 문서는 Microsoft사의 SQL Server에서 ALTIBASE로 변환할 때 고려할 사항과 변환 방법에 대해 설명한다.

SQL Server 2012와 ALTIBASE 6.1 버전을 대상으로 한다.

아래 문서를 사전에 참고할 것을 권장한다.

- ALTIBASE, MS SQL Server 비교 자료

OBJECT 변환

MS SQL SERVER의 OBJECT를 ALTIBASE로 변환할 때 고려할 사항에 대해 기술한다.

DATATYPE

SQL SERVER의 TABLE을 ALTIBASE로 변환할 때 각각의 DATATYPE을 어떻게 변환하는지에 대해 설명한다.

분류	SQL Server	ALTIBASE	비고
문자타입	CHAR	CHAR	
	VARCHAR	VARCHAR,	varchar(max)는 CLOB으로 변환
	TEXT	CLOB사용	
	NCHAR	NCHAR	Max length: 16000(UTF16), Max length: 10666(UTF8)
	NVARCHAR	NVARCHAR	Max length: 16000(UTF16), Max length: 10666(UTF8) NVARCHAR(MAX)는 CLOB으로 변환
	NTEXT	NVARCHAR 사용	Max length: 16000(UTF16), Max length: 10666(UTF8)
	BINARY	BYTE사용	
	VARBINARY	BLOB사용	
	IMAGE	BLOB사용	
숫자타입	BIGINT	BIGINT	
	NUMERIC	NUMERIC	
	BIT	BIT	
	SMALLINT	SMALLINT	
	REAL	REAL	
날짜타입	DATE	DATE	
	DATETIMEOFFSET	DATE 사용	
	DATETIME2	DATE 사용	알티베이스의 Scale은 마이크로초(6자리)까지 지원.
	SAMMLLDATETIME	DATE 사용	
	DATETIME	DATE 사용	
	TIME	DATE 사용	

FUNCTION & EXPRESSION

분류	SQL Server	ALTIBASE	비고
Aggregate	AVG	AVG	
	CHECKSUM_AGG	N/A	
	COUNT	COUNT	
	COUNT_BIG	COUNT	
	GROUPING	N/A	
	GROUPING_ID	N/A	
	MAX	MAX	
	MIN	MIN	
	STDEV	STDDEV 사용	
	STDEVP	N/A	
	SUM	SUM	
	VAR	VARIANCE	
Ranking	DENSE_RANK	DENSE_RANK	
	NTILE	N/A	
	RANK	RANK	
	ROW_NUMBER	ROW_NUMBER	
Conversion	CAST, CONVERT	CAST 사용	ALTIBASE의 CONVERT 함수는 다른 기능제공
	PARSE	CAST 사용	CONVERT 함수를 이용하여 캐릭터셋 변경가능
	TRY_CAST	CAST 사용	캐스팅 실패시 오류 반환
	TRY_CONVERT	N/A	
	TRY_PARSE	N/A	
Date	CURRENT_TIMESTAMP	SYSDATE 사용	
	DATENAME	DATENAME	
	DATEPART	DATEPART	
	DAY	DATEPART 사용	
	DATEADD	DATEADD	구문 변경 DATEADD (date, number, date_field_name)

분류	SQL Server	ALTIBASE	비고
Date	DATEDIFF	DATEDIFF	구문 변경 DATEDIFF (startdate, enddate, date_field_name)
	DATEFROMPARTS	TO_DATE 사용	
	DATETIME2FROMPARTS	TO_DATE 사용	
	DATETIMEFROMPARTS	TO_DATE 사용	
	DATETIMEOFFSETFROMPARTS	TO_DATE 사용	
	EOMONTH	LAST_DAY 사용	
	GETDATE	SYSDATE 사용	
	GETUTCDATE	N/A	
	ISDATE	N/A	
	SMALLDATETIMEFROMPARTS	TO_DATE 사용	
	SWITCHOFFSET	N/A	
	SYSDATETIME	SYSDATE 사용	
	SYSDATETIMEOFFSET	N/A	
	SYSUTCDATETIME	N/A	
	TIMEFROMPARTS	TO_DATE 사용	
	TODATETIMEOFFSET	N/A	
	YEAR	TO_CHAR 사용	
Logical	CHOOSE	N/A	
	IIF	NVL2 사용	
Mathematical	ABS	ABS	
	ACOS	ACOS	
	ASIN	ASIN	
	ATAN	ATAN	
	ATN2	ATN2	
	CEILING	CEIL 사용	

분류	SQL Server	ALTIBASE	비고
Mathematical	COS	COS	
	COT	N/A	
	DEGREES	N/A	
	EXP	EXP	
	FLOOR	FLOOR	
	LOG	LN 사용	ALTIBASE의 LOG 함수는 다른 기능제공
	LOG10	N/A	
	PI	N/A	
	POWER	POWER	
	RADIANS	N/A	
	RAND	RANDOM 사용	ALTIBASE는 Integer형식으로 반환
	ROUND	ROUND	
	SIGN	SIGN	
	SIN	SIN	
	SQRT	SQRT	
	SQUARE	N/A	
	TAN	TAN	
	TAN	TAN	
String	ASCII	ASCII	
	CHAR	CHAR	
	CHARINDEX	INSTR, POSITION 사용	
	CONCAT	CONCAT	
	DIFFERENCE	N/A	
	FORMAT	N/A	
	LEFT	SUBSTR, SUBSTRING 사용	SUBSTR (expr, start [, length]) Start에 양수 지정
	LEN	LENGTH 사용	
	LOWER	LOWER	
	LTRIM	LTRIM	ALTIBASE는 LTRIM (expr1 [,expr2])
	NCHAR	NCHAR	

분류	SQL Server	ALTIBASE	비고
String	PATINDEX	INSTR, POSITION 사용	
	QUOTENAME	N/A	
	REPLACE	REPLACE2 사용	
	REPLICATE	REPLICATE	
	REVERSE	REVERSE_STR 사용	
	RIGHT	SUBSTR, SUBSTRING 사 용	SUBSTR (expr, start [, length]) Start에 음수 지정
	RTRIM	RTRIM	ALTIBASE는 RTRIM (expr1 [,expr2])
	PATINDEX	INSTR, POSITION 사용	
	QUOTENAME		
	REPLACE	REPLACE2 사용	
	REPLICATE	REPLICATE	
	REVERSE	REVERSE_STR 사용	
	RIGHT	SUBSTR, SUBSTRING 사 용	SUBSTR (expr, start [, length]) Start에 음수 지정
	RTRIM	RTRIM	ALTIBASE는 RTRIM (expr1 [,expr2])
	SOUNDEX	N/A	
	SPACE	LPAD, RPAD 사 용	
	STR	TO_CHAR 사용	
	STUFF	STUFF	
	SUBSTRING	SUBSTRING	
	UNICODE	N/A	
	UPPER	UPPER	

OBJECT

분류	SQL Server	ALTIBASE	비고
CONSTRAINT	지원	지원	
TRIGGER	지원	지원	
Multi Key-Index	지원	지원	
VIEW	지원	지원	
UPDATABLE VIEW	지원	지원하지 않음	
SEQUENCE	지원	지원	
STORED FUNCTION/PROCEDURE	지원	지원	
SYNONYM	지원	지원	
TABLE Temporary Table	지원	지원	Temporary Table은 Volatile Table Space 이용
USER	지원	지원	
REPLICATION	지원	지원	

CREATE TABLESPACE

SQL Server에서는 데이터 저장소로 Database를 관리하나 ALTIBASE HDB에서는 테이블스페이스를 관리한다. ALTIBASE HDB에서의 테이블스페이스는 저장 공간에 따라 메모리 테이블스페이스, 디스크 테이블스페이스로 나뉘며 생성 주체에 따라 시스템 테이블스페이스, 사용자 정의 테이블 스페이스, 저장 내용에 따라 딕셔너리 테이블스페이스, 언두 테이블스페이스, 임시 테이블스페이스, 데이터 테이블 스페이스 등으로 나뉘어 진다.

따라서 ALTIBASE HDB로 변환할 때는 데이터 저장 공간에 따라 CREATE MEMORY DATA TABLESPACE, CREATE DISK DATA TABLESPACE 구문을 이용하여 생성한다.

다음은 SQL Server의 DATABASE를 ALTIBASE HDB의 테이블스페이스로 변환할 때 CREATE TABLESPACE 구문에서 지정하는 여러 옵션들의 변환 가이드를 설명한 것이다.

생성 예)	
CREATE TABLESPACE user_data DATAFILE '/tmp/tbs.user' SIZE 10M	
AUTOEXTEND ON NEXT 128M;	

SQL Server	ALTIBASE	비고
FILENAME	FILENAME	
SIZE	SIZE	기본 100MB
MAXSIZE	MAXSIZE	
FILEGROWTH	AUTOEXTEND ON NEXT	
FILESTREAM	N/A	
DEFAULT_FULLTEXT_LANGUAGE	N/A	
DEFAULT_LANGUAGE	N/A	
NESTED_TRIGGERS	N/A	
TRANSFORM_NOISE_WORDS	N/A	
TWO_DIGIT_YEAR_CUTOFF	N/A	
DB_CHAINING	N/A	
TRUSTWORTHY	N/A	

CREATE TABLE

SQL Server의 TABLE을 ALTIBASE로 변환할 때 CREATE TABLE 시 사용한 여러 옵션을 알맞게 변경해야 한다. ALTIBASE는 TEMPORARY TABLE, OBJECT TABLE, XMLType TABLE을 제공하지 않는다.

ALTIBASE는 메모리 TABLE을 제공한다. 따라서 변환할 TABLE의 특성을 잘 파악하여 메모리 TABLE을 생성한다면 메모리 TABLESPACE를 지정하여 생성해야 한다. 만약 메모리 TABLE을 생성한다면 SQL Server에서 사용했던 CREATE TABLE 구문에 사용한 옵션들을 사용할 수 없다. 메모리 TABLE 생성 구문은 ALTIBASE의 SQL 관련 매뉴얼을 참조하면 된다.

ORACLE의 TABLE을 DISK TABLE로 변환하고자 한다면, CREATE TABLE 시 설정할 수 있는 여러 옵션을 다음과 같이 ALTIBASE에 맞게 변환해야 한다.

1. Column_definition

SQL Server	ALTIBASE	비고
FILESTREAM	N/A	
COLLATE	N/A	
CONSTRAINT	CONSTRAINT	ALTIBASE는 CHECK를 제외한 Constraint를 지원한다. 또한 Constraint 지정시 references 절의 ON DELETE SET NULL(ON DELETE CASCADE는 지원) 옵션을 제공하지 않으므로 삭제. ALTIBASE는 PRIMARY KEY, UNIQUE 지정 시 index 이름은 제공하지 않는다.
IDENTITY	N/A	
ROWGUIDCOL	N/A	

2. data type

SQL Server	ALTIBASE	비고
Precision, scale	Precision, scale	
max	N/A	
CONTENT	N/A	
DOCUMENT	N/A	
xml_schema_collection	N/A	

3. column_constraint

SQL Server	ALTIBASE	비고
PRIMARY KEY	PRIMARY KEY	
NULL, NOT NULL	NULL, NOT NULL	
UNIQUE	UNIQUE	
CLUSTERED, NONCLUSTERED	N/A	

SQL Server	ALTIBASE	비고
FOREIGN KEY REFERENCES	FOREIGN KEY REFERENCES	
partition_scheme_name	PARTITION BY RANGE HASH LIST	Table 생성시 파티션테이블 종류 지정.

4. computed_column_definition

지원하지 않음

5. table_constraint

SQL Server	ALTIBASE	비고
PRIMARY KEY	PRIMARY KEY	
CLUSTERED, NONCLUSTERED	N/A	
FOREIGN KEY REFERENCES	FOREIGN KEY REFERENCES	
partition_scheme_name	PARTITION BY RANGE HASH LIST	Table 생성시 파티션테이블 종류 지정.

6. table_option

SQL Server	ALTIBASE	비고
DATA_COMPRESSION	N/A	

7. index_option

SQL Server	ALTIBASE	비고
PAD_INDEX	N/A	
FILLFACTOR	N/A	
IGNORE_DUP_KEY	N/A	ALTIBASE는 ON 상태임
STATISTICS_NORECOMPUTE	N/A	ALTIBASE는 OFF 상태임
ALLOW_ROW_LOCKS	N/A	Durability Level 설정에 따라 정의
ALLOW_PAGE_LOCKS	N/A	Durability Level 설정에 따라 정의
DATA_COMPRESSION	N/A	

CREATE USER

ALTIBASE는 데이터베이스 로그인과 유저를 분리하여 생성하지 않고 하나의 유저로 생성하여 관리한다.

SQL Server
CREATE LOGIN <login_name> WITH PASSWORD = '<password>'; Users based on logins in master CREATE USER user_name [{ FOR FROM } LOGIN login_name] [WITH DEFAULT_SCHEMA = schema_name] [;]
ALTIBASE
CREATE USER <user_name> IDENTIFIED BY '<password>' DEFAULT TABLESPACE = tablespace_name

CREATE INDEX

ALTIBASE는 BTREE와 RTREE INDEX만 제공하고 BITMAP, CLUSTER, Function based, REVERSE, Global partitioned INDEX를 제공하지 않는다.

다음은 ALTIBASE로 변환할 때 CREATE INDEX 구문에 사용하는 옵션들에 대한 변환 방법이다.

SQL Server	ALTIBASE	비고
UNIQUE	UNIQUE	
CLUSTERED NONCLUSTERED	N/A	
ASC DESC	ASC DESC	
INCLUDE	N/A	
filter_predicate	N/A	
partition_scheme_name	PARTITION ON	Local Index만 지원
ON filegroup_name	N/A	
table_or_view_name	Table_name	View는 지원하지 않음
PAD_INDEX	N/A	
FILLFACTOR	N/A	

SQL Server	ALTIBASE	비고
SORT_IN_TEMPDB	N/A	
IGNORE_DUP_KEY	N/A	ALTIBASE는 ON 상태임
STATISTICS_NORECOMPUTE	N/A	Altibase는 OFF 상태임
DROP_EXISTING	N/A	
ONLINE		Altibase는 OFF 상태임
ALLOW_ROW_LOCKS	N/A	Durability Level 설정에 따라 정의
ALLOW_PAGE_LOCKS	N/A	Durability Level 설정에 따라 정의
MAXDOP	PARALLEL	생성시에만 적용
DATA_COMPRESSION	N/A	

SQL Conversion

SQL Server의 SQL 문들을 ALTIBASE로 변환하는 방법을 설명에 대하여 설명한다.

1. JOIN

ALTIBASE는 ANSI 표준의 JOIN 연산만 제공한다.

SQL Server	ALTIBASE	비고
INNER JOIN	INNER JOIN	
LEFT OUTER JOIN	LEFT OUTER JOIN	
RIGHT OUTER JOIN	RIGHT OUTER JOIN	
FULL OUTER JOIN	FULL OUTER JOIN	
CROSS JOIN	N/A	

2. Execution Query

ALTIBASE의 쿼리 종결 문자는 ";" 이다. 따라서 여러 쿼리를 한번에 실행하려 한다면 쿼리마다 종결문자 ";"로 구분해 주어야 한다.

SQL Server	ALTIBASE	비고
GO	; Commit;	기본값은 Autocommit

3. Temporary Table

ALTIBASE는 Temporary Table을 생성하는 명령문은 존재하지 않으며 Temporary Table의 경우 Volatile Tablespace에 Table을 생성하여 사용한다.

SQL Server	ALTIBASE
CREATE TABLE #TempProcess	CREATE TABLE TEMP1 (...) TABLESPACE volatile_tablespace_name

4. Control Statement

	SQL Server	ALTIBASE
IF	If (조건) Else if (조건) Else End	If 조건 then Elseif 조건 then Else End
While	While (조건)	While 조건 loop

	BEING END BREAK =>while문 탈출	End loop Exit when 조건 =>조건 만족할 때 반복 loop탈출
--	-----------------------------------	--

5. Identity attribute

ALTIBASE에는 Identity 속성은 존재하지 않는다. 따라서 필요시 SEQUENCE를 만들어 사용한다.

6. Procedure Conversion

ALTIBASE의 PROCEDURE 생성 및 실행구문은 SQL 서버와 다른 부분이 있다. 예를 통하여 PROCEDURE 변경 방법을 확인한다.

1. Parameter Declaration

SQL Server의 파라미터 선언에서 '@' 기호를 없앤다.

파라미터의 성격에 따라 in , out , in out 을 명시해준다.

Function일 경우 실행 후 하나의 값을 반환해 주는 것으로 반드시 **RETURN** 다음의 데이터 타입을 명시해야 한다.

SQL Server
<pre> CREATE Procedure [dbo].[sp1] @nTop INT --무시 , @nGroupCode INT = -2 , @nObjectCode INT = -2 , @nRCLS INT=1 -- 1:Set , 0:Unset , @nLCRS INT=1 -- 1:Set , 0:Unset , @nLCLS INT=1 -- 1:Set , 0:Unset , @sStartDate VARCHAR(19) = " -- 무시 , @sEndDate VARCHAR(19) = " -- 무시 , @nSort INT = 0 -- 0:탐지회수, 1:시도횟수 2:DataSize , @sSignatureName VARCHAR(100) = " As ... </pre>
ALTIBASE

```

create or replace procedure sp1
(
    i_nTop          IN INT --무시
    , i_nGroupCode  IN INT := -2
    , i_nObjectCode IN INT := -2
    , i_nRCLS       IN INT:=1 -- 1:Set , 0:Unset
    , i_nLCRS       IN INT:=1 -- 1:Set , 0:Unset
    , i_nLCLS       IN INT:=1 -- 1:Set , 0:Unset
    , i_sStartDate  IN VARCHAR(19) := " -- 무시
    , i_sEndDate    IN VARCHAR(19) := " -- 무시
    , i_nSort       IN INT := 0 -- 0:탐지회수, 1:시도횟수 2:DataSize
    , i_sSignatureName IN VARCHAR(100) := "
)
as ...

```

2. Variable Declaration

Altibase의 변수 선언은 AS와 BEGIN 사이에서 정의된다.

SQL Server의 변수 선언에서 '@' 기호를 없앤다.

변수 선언 할 때 구분자인 ','를 ';'로 바꾼다.

마지막 변수 선언에도 항상 ';'로 마지막임을 선언 한다.

SQL Server

```

create or replace procedure sp1
(
    .....
)
as
DECLARE @sTotalQry  VARCHAR(3000)
        , @sQry      VARCHAR(3000)
        , @sFilterQry VARCHAR(1000)
        , @dStartDate DATETIME
        , @dEndDate  DATETIME

```

```
, @sSort VARCHAR(20)
, @sSortSub VARCHAR(90)
, @sTop VARCHAR(10)
, @sTable VARCHAR(50)
, @nSec INT
, @sBaseDate VARCHAR(23)
```

ALTIBASE

```
create or replace procedure sp1
(
.....
)
as

    i_sTotalQry VARCHAR(3000);
    i_sQry VARCHAR(3000);
    i_sFilterQry VARCHAR(1000);
    i_dStartDate DATE;
    i_dEndDate DATE;
    i_sSort VARCHAR(20);
    i_sSortSub VARCHAR(90);
    i_sTop VARCHAR(10);
    i_sTable VARCHAR(50);
    i_nSec INT;
    i_sBaseDate VARCHAR(23);
```

3. ASSIGNMENT

SQL Server의 변수에 값 할당 시 SET 를 이용한다. 여기서 SET, '@' 문자를 없앤다.

그리고 '=' 를 ':='로 바꾼다.

구분자인 ','를 ';'로 바꾸고 마지막에 항상 ';'로 마지막임을 선언 한다.

위와 같이 SET 을 명시하여 변수에 값을 할당 할 수도 있다

SQL Server
<pre>SET @sTop = '1000' SET @sBaseDate = CONVERT(VARCHAR, DATEADD(dd, -1, GETDATE()), 121)</pre>
ALTIBASE
<pre>i_sTop := '1000' ; i_sBaseDate := to_char(DATEADD (SYSDATE, -1, 'DAY'), 'YYYY-MM-DD HH:MI:SS') ; OR Set i_sTop = '1000' ; Set i_sBaseDate = to_char(DATEADD (SYSDATE, -1, 'DAY'), 'YYYY-MM-DD HH:MI:SS') ;</pre>

4. Control_flow – IF 문

ALTIBASE의 문장 끝에는 항상 ';'로 마지막임을 표시한다.

IF문 처음에는 IF (condition) **THEN**으로 처음을 표시한다.

IF문 끝에는 **END IF**로 IF문이 끝임을 표시한다

SQL Server
<pre>IF (@nRCLS = 1) AND (@nLCRS = 1) AND (@nLCLS = 1) SET @sFilterQry = " ELSE IF (@nRCLS = 0) AND (@nLCRS = 0) AND (@nLCLS = 0) SET @sFilterQry = "</pre>
ALTIBASE
<pre>IF (i_nRCLS = 1) AND (i_nLCRS = 1) AND (i_nLCLS = 1) THEN i_sFilterQry := " ; ELIF (i_nRCLS = 0) AND (i_nLCRS = 0) AND (i_nLCLS = 0) THEN i_sFilterQry := " ; END IF;</pre>

5. SELECT 절

ALTIBASE의 문장 끝에는 항상 ';'로 마지막임을 표시한다.

ALTIBASE에서 select의 resultset을 client에 보내기 위해서는 DB 객체에 REF CURSOR 를 먼저 작성해주어야 한다. 그리고 sp의 인자에 ref cursor를 인자로 작성하여야 한다. 그리고 해당 쿼리를 실행할 때 Open이라는 명령어로 실행 한다.

SQL Server의 '@' 문자를 없앤다

SQL Server
<pre>..... SET @sQry = 'select * from test_tbl' EXEC(@sQry)</pre>
ALTIBASE
<pre>CREATE TYPESET MY_TYPE AS TYPE MY_CUR IS REF CURSOR; END; / create or replace procedure spTMSGetEventSignatureRankVariation (..... , P1 OUT MY_TYPE.MY_CUR) As i_sQry := 'select * from test_tbl' ; OPEN P1 FOR i_sQry; END;</pre>

6. Exception Handler

ALTIBASE의 문장 끝에는 항상 ';'로 마지막임을 표시한다.

ALTIBASE의 Exception Handler는 해당 Exception이 발생했을 때 처리하는 기술이다.

Exception 처리는 항상 BEGIN END; 구문 안에서 처리 해야 한다.

시스템 정의 EXCEPTION이나 사용자 정의 EXCEPTION의 이름을 기술하고

해당 Exception을 만날 때 해당 statement를 처리하고, 현재 발생한 Exception을 처리하지 못할 경우 최종적으로 OTHERS 루틴으로 처리 된다. 적용 받은 레코드수가 0인지 체크하는 구문은 SQL%ROWCOUNT상수로 확인 할 수 있다.

시스템 정의 Exception은 파일 마지막에 첨부합니다

SQL Server
<pre> IF @@ERROR <> 0 BEGIN IF @@ROWCOUNT = 0 BEGIN ROLLBACK END END END </pre>
ALTIBASE
<pre> BEGIN EXCEPTION WHEN NO_DATA_FOUND THEN rollback; END; OR IF SQL%ROWCOUNT = 0 then ROLLBACK END if; </pre>

7. SP Call

ALTIBASE의 문장 끝에는 항상 ';'로 마지막임을 표시한다.

SQL Server의 SP 호출은 exec sp_name 로 호출 하고 sp 인자를 괄호 없이 지정해주지만, ALTIBASE에서 SP_name 을 실행하고 괄호안에 sp의 인자를 명시해준다.

SQL Server의 '@' 문자를 없앤다.

SQL Server
<pre> IF @아이디 IS NOT NULL BEGIN EXEC SP_상품_예약조회_회원체크 @주민번호,@구분,@날짜 1,@날짜 2 </pre>

ALTIBASE
IF i_아이디 IS NOT NULL THEN SP_상품_예약조회_회원체크(i_주민번호,i_구분,i_날짜 1,i_날짜 2);

8. Functions

ALTIBASE의 문장 끝에는 항상 ';'로 마지막임을 표시한다.

함수별로 사용법은 상이할 수 있고, 함수 명이 다를 수도 있다.

Mssql의 '@' 문자를 없앤다.

SQL Server
<ol style="list-style-type: none"> LEFT, RIGHT 함수 - LEFT(@주민번호,6) - RIGHT(@주민번호,7) ISNULL(V1, 0) CASE 연합 WHEN '0' THEN B.대외모객+B.패키지 ELSE ISNULL(D.연합모객,0) END Convert CONVERT(VARCHAR(10),,120) <= B.출발일 TO_CHAR(CASE2(i_출발일1=NULL,SYSDATE),'YYYY-MM-DD'); LEN RTRIM(LTRIM()) DATEADD(mi, -20, GETDATE()) CEILING 나머지 연산자 % (a % b) charIndex('aaa', 'aaabbbcccddd') 문자열 연결 + : ('alti' + 'base')
ALTIBASE
<ol style="list-style-type: none"> SubStr로 치환 - SUBSTR(주민번호,1,6) - SUBSTR(주민번호,8,7) NVL(V1, 0)

3. CASE2(연합 = '0', B.대외모객 || B.패키지,D.연합모객=NULL,0,)
4. Convert
 - CONVERT(VARCHAR(23), dStartDate, 121) => TO_CHAR(dStartDate, 'YYYY-MM-DD HH:MI:SS.FF3');
 - CONVERT(DATETIME, sEndDate) => TO_DATE(sEndDate, 'YYYY-MM-DD HH:MI:SS');
5. LENGTH
6. TRIM()
7. DATEADD(sysdate, -20, 'MINUTUE');
8. CEIL
9. MOD(a, b)
10. INSTR('aaabbbccdd', 'aaa')
11. || 사용 ('alti' || 'base')

9. Etc

ALTIBASE의 문장 끝에는 항상 ';'로 마지막임을 표시한다.
SQL Server의 '@' 문자를 없앤다.

SQL Server
<ol style="list-style-type: none"> 1. UPDATE STATISTICS index_name 2. with (nolock) 3. order, level 등의 keyword 사용
ALTIBASE
<ol style="list-style-type: none"> 1. 필요 없음. 2. 삭제 3. order, level 등의 keyword 사용 금지(알티베이스 예약어)

10. DB Link

Altibase에서 DB Link를 사용시에는
SELECT의 경우 REMOTE_TABLE(dblink명, Query);

DML의 경우 REMOTE_EXECUTE_IMMEDIATE(dblink명, Query);
를 사용한다.

11. JOIN UPDATE

Altibase에서 Join Update를 처리하기 위해서는 두가지의 방법이 있다.

1. JOIN UPDATE를 사용
 - 각 Table에 Primary Key 또는 Unique Key가 존재하여야 한다.
2. Merge Join의 update를 사용.
 - Merge into A using (SELECT ...)
WHEN matched then
Update ...

7. Using Cursor

SQL Server의 커서를 ALTIBASE로 변환하는 방법을 예제를 통하여 확인한다.

SQL Server

```
declare security_cursor cursor for
select fldID from tblDept where fldParentID=@fldID
open security_cursor
fetch next from security_cursor into @fldID
while @@fetch_status = 0
begin
    exec sr_GetSubDeptID_Str @fldID, @DeptID OUTPUT
    declare @sql as varchar(8000)
    begin
        exec(@sql)
    end
    fetch next from security_cursor into @fldID
end
close security_cursor
deallocate security_cursor
```

ALTIBASE

```
부서별 인원수와 급여 합계를 구하는 프로시저입니다
CREATE OR REPLACE PROCEDURE ForCursor_Test
AS
BEGIN
    DECLARE CURSOR dept_sum IS
    SELECT b.dname, COUNT(a.empno) cnt, SUM(a.sal) salary
    FROM emp a, dept b
    WHERE a.deptno = b.deptno
```

```
GROUP BY b.dname;
-- Cursor를 FOR문에서 실행시킨다
BEGIN
  FOR emp_list IN dept_sum LOOP
    println('부서명 : ' || emp_list.dname);
    println ('사원수 : ' || emp_list.cnt);
    println('급여합계 : ' || emp_list.salary);
  END LOOP;
END;
END;

/
```

※ EXCEPTION CODE

Exception Name	Error Code (integer)	Error Code (hexadecimal)	Error Section
"CURSOR_ALREADY_OPEN"	201062	31166	qpERR_ABORT_QSX_CURSOR_ALREADY_OPEN
"DUP_VAL_ON_INDEX"	201063	31167	qpERR_ABORT_QSX_DUP_VAL_ON_INDEX
"INVALID_CURSOR"	201064	31168	qpERR_ABORT_QSX_INVALID_CURSOR
"INVALID_NUMBER"	201065	31169	qpERR_ABORT_QSX_INVALID_NUMBER
"NO_DATA_FOUND"	201066	3116A	qpERR_ABORT_QSX_NO_DATA_FOUND
"PROGRAM_ERROR"	201067	3116B	qpERR_ABORT_QSX_PROGRAM_ERROR
"STORAGE_ERROR"	201068	3116C	qpERR_ABORT_QSX_STORAGE_ERROR
"TIMEOUT_ON_RESOURCE"	201069	3116D	qpERR_ABORT_QSX_TIMEOUT_ON_RESOURCE
"TOO_MANY_ROWS"	201070	3116E	qpERR_ABORT_QSX_TOO_MANY_ROWS
"VALUE_ERROR"	201071	3116F	qpERR_ABORT_QSX_VALUE_ERROR
"ZERO_DIVIDE"	201072	31170	qpERR_ABORT_QSX_ZERO_DIVIDE
"INVALID_PATH"	201237	31215	qpERR_ABORT_QSX_FILE_INVALID_PATH
"INVALID_MODE"	201235	31213	qpERR_ABORT_QSX_INVALID_FILEOPEN_MODE
"INVALID_FILEHANDLE"	201238	31216	qpERR_ABORT_QSX_FILE_INVALID_FILEHANDLE
"INVALID_OPERATION"	201239	31217	qpERR_ABORT_QSX_FILE_INVALID_OPERATION
"READ_ERROR"	201242	3121A	qpERR_ABORT_QSX_FILE_READ_ERROR
"WRITE_ERROR"	201243	3121B	qpERR_ABORT_QSX_FILE_WRITE_ERROR
"ACCESS_DENIED"	201236	31214	qpERR_ABORT_QSX_DIRECTORY_ACCESS_DENIED
"DELETE_FAILED"	201240	31218	qpERR_ABORT_QSX_FILE_DELETE_FAILED
"RENAME_FAILED"	201241	31219	qpERR_ABORT_QSX_FILE_RENAME_FAILED