CHAPTER 12

Differences from Other SQL Dialects

About this chapter

Adaptive Server Anywhere conforms to the ANSI SQL89 standard but has many additional features defined in IBM's DB2 and SAA specification, and in ANSI SQL/92.

This chapter describes those features of Adaptive Server Anywhere that are not commonly found in other SQL implementations.

Contents

Adaptive Server Anywhere SQL features

The following features of the SQL supported by Adaptive Server Anywhere are not found in many other SQL implementations.

Type conversions

Full type conversion is implemented. Any data type can be compared with or used in any expression with any other data type.

Dates

Adaptive Server Anywhere has date, time and timestamp types that includes a year, month and day, hour, minutes, seconds and fraction of a second. For insertions or updates to date fields, or comparisons with date fields, a free format date is supported.

In addition, the following operations are allowed on dates:

- **date + integer** Add the specified number of days to a date.
- **date - integer** Subtract the specified number of days from a date.
- **date - date** Compute the number of days between two dates.
- **date + time** Make a timestamp out of a date and time.

Also, many functions are provided for manipulating dates and times. See "SQL Functions" on page 267 for a description of these.

Integrity

Adaptive Server Anywhere supports both entity and referential integrity. This has been implemented via the following two extensions to the CREATE TABLE and ALTER TABLE commands.

```
PRIMARY KEY ( column-name, ... )

[NOT NULL] FOREIGN KEY [role-name]
              [(column-name, ...)]
           REFERENCES table-name [(column-name, ...)]
              [ CHECK ON COMMIT ]
```

The PRIMARY KEY clause declares the primary key for the relation. Adaptive Server Anywhere will then enforce the uniqueness of the primary key, and ensure that no column in the primary key contains the NULL value.

The FOREIGN KEY clause defines a relationship between this table and another table. This relationship is represented by a column (or columns) in this table which must contain values in the primary key of another table. The system will then ensure referential integrity for these columns - whenever these columns are modified or a row is inserted into this table, these columns will be checked to ensure that either one or more is NULL or the values match the corresponding columns for some row in the primary key of the other table. For more information, see "CREATE TABLE statement" on page 415.

Joins

Adaptive Server Anywhere allows **automatic joins** between tables. In addition to the NATURAL and OUTER join operators supported in other implementations, Adaptive Server Anywhere allows KEY joins between tables based on foreign key relationships. This reduces the complexity of the WHERE clause when performing joins.

Updates

Adaptive Server Anywhere allows more than one table to be referenced by the UPDATE command. Views defined on more than one table can also be updated. Many SQL implementations will not allow updates on joined tables.

Altering tables

The ALTER TABLE command has been extended. In addition to changes for entity and referential integrity, the following types of alterations are allowed:

```
ADD column data-type

MODIFY column data-type

DELETE column

RENAME new-table-name

RENAME old-column TO new-column
```

The MODIFY can be used to change the maximum length of a character column as well as converting from one data type to another. For more information, see "ALTER TABLE statement" on page 351.

Subqueries where expressions are allowed

Adaptive Server Anywhere allows subqueries to appear wherever expressions are allowed. Many SQL implementations only allow subqueries on the right side of a comparison operator. For example, the following command is valid in Adaptive Server Anywhere but not valid in most other SQL implementations.

```
SELECT    emp_lname,
          emp_birthdate,
          (  SELECT skill
             FROM department
             WHERE emp_id = employee.emp_ID
             AND dept_id = 200 )
FROM employee
```

Additional functions

Adaptive Server Anywhere supports several functions not in the ANSI SQL definition. See "SQL Functions" on page 267 for a full list of available functions.

Cursors

When using Embedded SQL, cursor positions can be moved arbitrarily on the FETCH statement. Cursors can be moved forward or backward relative to the current position or a given number of records from the beginning or end of the cursor.