Database Management Systems

Part II: The Relational Model

Lecture 8

The SQL Language

Contents

- Introduction
- The SQL Language
- Data Definition
- Data Manipulation
- Retrieval Operations
- Update Operations

Introduction

- The standard language for interacting with relational databases is Structured Query Language (SQL).
- SQL was originally developed in IBM Research in early 1970s.
- SQL is not the perfect relational language but it is extremely important from a commercial point of view.
- It is supported by every product on the market today.
- Every database professional needs to know something about it.

The SQL Language

- SQL is used to formulate relational operations, that is, operations that define and manipulate data in relational form.
- SQL includes a data definition language (DDL) component and a data manipulation language (DML) component.
- The SQL DDL can be used to define objects at the external level, the conceptual level and even in most systems, though not in the standard – the internal level.
- The SQL DML can operate at both the external and conceptual level.

Data Definition

- The SQL data-definition language (DDL) provides commands for defining relation schemas, deleting relations, and modifying relation schemas.
- The principal DDL statements are as follows:

CREATE DOMAIN	CREATE TABLE
ALTER DOMAIN	ALTER TABLE
DROP DOMAIN	DROP TABLE

Data Definition (Cont.)

CREATE DOMAIN

- An SQL domain must be defined in terms of one of the built-in, system-defined data types, not another user-defined domain.
- Here is the syntax for creating an SQL domain.

CREATE DOMAIN domain data-type

[default-definition]

[domain-constraint-definition-list] ;

Data Definition (Cont.)

CREATE DOMAIN

- SQL supports the scalar data types.
- The optional default-definition specifies a default value that applies to every column that is defined on the domain and does not have an explicit default value of its own.
- If the user inserts a row into table and does not provide a value for one column within that row, then the value "???" will be placed in that position by default.

Data Definition (Cont.)

CREATE DOMAIN

- The optional list of domain constraint definitions specifies a set of integrity constraints that apply to every column defined on the domain.
- If the user does provide a value but it is not one of the legal set, the operation will fail and the system will produce a diagnostic that mentions the VALID constraint.

Data Definition (Cont.)

CREATE DOMAIN

- Here is an example:

```
CREATE DOMAIN S#      CHAR(5) ;
CREATE DOMAIN NAME    CHAR(20) ;
CREATE DOMAIN STATUS  NUMERIC(5) ;
CREATE DOMAIN CITY    CHAR(15) ;
CREATE DOMAIN P#      CHAR(6) ;
CREATE DOMAIN COLOR   CHAR(6) ;
CREATE DOMAIN WEIGHT  NUMERIC(5) ;
CREATE DOMAIN QTY     NUMERIC(9) ;
```

Data Definition (Cont.)

ALTER DOMAIN

- Next, an existing domain can be altered at any time in a variety of ways by means of the statement ALTER DOMAIN.
- Specifically, ALTER DOMAIN allows a new default definition to be specified for an existing domain or an existing one to be deleted.
- It also allows a new integrity constraint to be specified for an existing domain or an existing one to be deleted.

Data Definition (Cont.)

DROP DOMAIN

- Finally, an existing domain can be destroyed by means of the statement DROP DOMAIN – syntax;

  DROP DOMAIN domain option;

  where option is either RESTRICT or CASCADE.

- If RESTRICT is specified, the DROP will fail if the domain is referenced anywhere.

- If CASCADE is specified, the DROP will succeed and will cascade in various ways.

Data Definition (Cont.)

Base Tables

- SQL tables are allowed to include duplicate rows; they therefore do not necessarily have any candidate keys.
- SQL tables are considered to have a left-to-right column ordering; for example, in the supplier table, column S# might be the first column, column SNAME might be the second column, and so on.
- Base tables are defined by means of the CREATE TABLE statement.

Data Definition (Cont.)

CREATE TABLE

- Figure shows how the suppliers-and-parts database might be defined, using SQL data definition operations.

```
CREATE BASE RELATION S
      ( S#       DOMAIN ( S# ),
        SNAME    DOMAIN ( NAME ),
        STATUS   DOMAIN ( STATUS ),
        CITY     DOMAIN ( CITY ) )
      PRIMARY KEY ( S# ) ;
```

```
CREATE BASE RELATION P
      ( P#       DOMAIN ( P# ),
        PNAME    DOMAIN ( NAME ),
        COLOR    DOMAIN ( COLOR ),
        WEIGHT   DOMAIN ( WEIGHT ),
        CITY     DOMAIN ( CITY ) )
      PRIMARY KEY ( P# ) ;
```

```
CREATE BASE RELATION SP
      ( S#       DOMAIN ( S# ),
        P#       DOMAIN ( P# ),
        QTY      DOMAIN ( QTY ) )
      PRIMARY KEY ( S#, P# )
      FOREIGN KEY ( S# ) REFERENCES S
      FOREIGN KEY ( P# ) REFERENCES P ;
```

Data Definition (Cont.)

CREATE TABLE

- The resulting base tables with sample values are as follows.

S

S#	SNAME	STATUS	CITY
S1	Smith	20	London
S2	Jones	10	Paris
S3	Blake	30	Paris
S4	Clark	20	London
S5	Adams	30	Athens

P

P#	PNAME	COLOR	WEIGHT	CITY
P1	Nut	Red	12	London
P2	Bolt	Green	17	Paris
P3	Screw	Blue	17	Rome
P4	Screw	Red	14	London
P5	Cam	Blue	12	Paris
P6	Cog	Red	19	London

SP

S#	P#	QTY
S1	P1	300
S1	P2	200
S1	P3	400
S1	P4	200
S1	P5	100
S1	P6	100
S2	P1	300
S2	P2	400
S3	P2	200
S4	P2	200
S4	P4	300
S4	P5	400

Data Manipulation

- The SQL data manipulation language (DML) provides the ability to query information from the database and to insert tuples into, delete tuples from, and modify tuples in the database.
- The basic structure of an SQL query consists of three clauses: SELECT, FROM, and WHERE.
- The query takes as its input the relations listed in the from clause, operates on them as specified in the where and select clauses, and then produces a relation as the result.

Data Manipulation (Cont.)

- The principal data manipulation language (DML) statements are SELECT, INSERT, UPDATE, and DELETE.
- SELECT is a retrieval operation.
- INSERT, UPDATE, and DELETE are update operations.
- Here, a set of examples will serve to highlight some of the most important points by using the three base tables S, P and SP (described in slide 14).

Retrieval Operations (Cont.)

Example 1

- Get color and city for “nonParis” parts with weight greater than ten.

```
SELECT   P.COLOR, P.CITY
FROM     P
WHERE    P.CITY <> ‘Paris’
AND      P.WEIGHT > 10;
```

- Since SQL does not eliminate redundant duplicate rows from the result of a SELECT, the query will return four rows, even though three of those four rows are identical, all being of the form (Red, London).

Retrieval Operations (Cont.)

Example 2

- To force the elimination of duplicates, the keyword DISTINCT is inserted after SELECT.
- This query will return two rows only.

```
SELECT   DISTINCT P.COLOR, P.CITY
FROM     P
WHERE    P.CITY <> 'Paris'
AND      P.WEIGHT > 10;
```

Retrieval Operations (Cont.)

Example 3

- The ORDER BY clause causes the tuples in the result of a query to appear in ascending order by default.
- To specify the sort order, we may specify DESC for descending order or ASC for ascending order.

```
SELECT   DISTINCT P.COLOR, P.CITY
FROM     P
WHERE    P.CITY <> 'Paris'
AND      P.WEIGHT > 10;
ORDER BY CITY DESC;
```

Retrieval Operations (Cont.)

Example 4

- Consider to get the part number and the weight of that part in grams for all parts.

```
SELECT   P.P#, P.WEIGHT * 454 AS GMWT

FROM     P ;
```

- The specification AS GMWT introduces an appropriate result column name for the computed column.
- The two columns of the result table are thus called P# and GMWT, respectively.

Retrieval Operations (Cont.)

Example 5

- Consider the query to get full details of all suppliers.

```
SELECT  *                    SELECT  S.*
                    or
FROM    S ;                  FROM    S ;
```

- The star or asterisk is shorthand for a list of all column names in the table(s) referenced in the FROM clause, in the left-to-right order in which those column(s) are defined within those table(s).
- The result is a copy of the entire S table.

Retrieval Operations (Cont.)

Example 6

- Consider the query to get the total number of suppliers.

```
SELECT  COUNT (*) AS N
FROM    S
```

- SQL offers five built-in aggregate functions: COUNT, SUM, AVG, MAX, and MIN.
- We use the aggregate function COUNT to count the number of tuples in a relation.
- The result here is a table with one column called N, and one row, containing the value 5.

Update Operations

- The SQL DML includes three update operations: INSERT, UPDATE, and DELETE.
- The INSERT statement is a request to insert one tuple or many tuples..

```
INSERT INTO     P ( P#, PNAME, COLOR, WEIGHT, CITY )
VALUES          ('P8', 'Sprocket', 'Pink', 14, 'Nice' )

INSERT INTO     TEMP ( S#, CITY )
SELECT          S.S#, S.CITY
FROM            S
WHERE           S.STATUS > 15 ;
```

Update Operations (Cont.)

- In certain situations, we may wish to change a value in a tuple without changing all values in the tuple.
- For example, suppose that all instructors with salary over $100,000 receive a 3 percent raise, whereas all others receive a 5 percent raise.
- We could write two UPDATE statements:

```
UPDATE instructor
SET salary = salary * 1.03
WHERE salary > 100000;
```

```
UPDATE instructor
SET salary = salary * 1.05
WHERE salary <= 100000;
```

Update Operations (Cont.)

- For using DELETE statement, note that only whole tuples can be deleted; cannot delete values on only particular attributes.
- Consider the query to delete all tuples in the instructor relation pertaining to instructors in the Finance department.

```
DELETE FROM instructor
WHERE dept name= 'Finance';
```

- Consider the query to delete all instructors with a salary between $13,000 and $15,000.

```
DELETE FROM instructor
WHERE salary between 13000 and 15000;
```

Summary

- SQL base tables permit duplicate rows, and they have a left-to-right ordering to their columns.
- DDL provides CREATE command for defining, DROP for deleting and ALTER for modifying relations.
- DML provides the SELECT statement for retrieving information from a database and INSERT, UPDATE and DELETE statements for modifying information in the database.

Next Lecture

Part III: Database Design

Functional Dependencies

- Basic Definitions
- Trivial and Nontrivial Dependencies
- Closure of a Set of Dependencies
- Closure of a Set of Attributes
- Irreducible Sets of Dependencies

Textbook and References

Textbook

- C. J. Date, “An Introduction to Database Systems”, 6th Edition, 1994.

Additional References

- Abraham Silberschatz, Henry F. Korth, S. Sudarshan, “Database System Concepts”, 6th Edition, 2011.
- Ramez Elmasri, Shamkant B. Navathe, “Fundamentals of Database Systems”, 6th Edition, 2010.