Chapter 6

Standard Query Language (SQL) Features

By

Dr. Mona A. F. Ismail

Chapter Outline

1. **Basic SQL**
2. **Data Manipulation Language for Relational DBs**
3. **Data Definition Language for Relational DBs**
4. **Views**
5. **Summary**

SQL "Structured Query Language"

- SQL language is considered one of the major reasons for the commercial success of relational databases
- SQL has the Data definition language (DDL) and Data Manipulation Language (DML)
- DDL used to CREATE, DROP, and ALTER the descriptions of the tables of a database
- SQL provides four DML statements SELECT, UPDATE, DELETE, and INSERT
- **Table**, **row**, and **column** are used for relational model terms relation, tuple, and attribute
- Each statement in SQL ends with a semicolon

SQL "Structured Query Language"

- **SQL schema** is identified by a **schema name**
  - It includes an **authorization identifier** (owner)
  - Components are **descriptors** for each element
  - Tables, constraints, views, domains, and other constructs

CREATE SCHEMA COMPANY **AUTHORIZATION** JOHN;

- Schema **elements** include Tables, constraints, views, domains, and other constructs
- **Catalog** named collection of schemas in an SQL environment

CREATE TABLE

The general format for this statement is:

CREATE TABLE base-table-name
(column-definition [,column-definition]….
[, primary-key-definition]
[, foreign-key-definition[, foreign-key-definition]….);

Where a “column-definition” has the form:

Column-name data-type [NOT NULL]

Data Types

Numeric data

- **INTEGER** Signed full word binary integer.
- **SMALLINT** Signed half word binary integer.
- **DECIMAL(p,q)** or **NUMERIC(p,q)** Signed packed decimal number.

p digits number

decimal point q digits

Example: decimal(5,2) is a number that has 3 digits before the decimal and 2 digits after the decimal

- **FLOAT** Signed floating point number.

String data

CHARACTER(n) or **CHAR(n)** Fixed length string of exactly n 8-bit characters.

VARCHAR(n) Varying length string of up to n 8-bit characters.

GRAPHIC(n) Fixed length string of exactly n 16-bit characters.

VARGRAPHIC(n) Varying length string of up to n 16-bit characters.

Data Types

- **Bit-string** data types
  - Fixed length: BIT(*n*)
  - Varying length: BIT VARYING(*n*)
- **Boolean** data type
  - Values of TRUE or FALSE
- **Date and Time**
  - **DATE** date (yyyy-mm-dd)
  - **TIME** time (hh:mm:ss)
  - **TIMESTAMP** combination of date and time
  - Optional WITH TIME ZONE qualifier

CREATE TABLE : Example

- A constraint NOT NULL may be specified on an attribute

```
CREATE TABLE DEPARTMENT (
        DNAME                   VARCHAR(10)  NOT NULL,
        DNUMBER                 INTEGER      NOT NULL,
        MGRSSN                  CHAR(9),
        MGRSTARTDATE            DATE  );
```

- Key attributes can be specified via the PRIMARY KEY and UNIQUE phrases

```
 CREATE TABLE DEPT (
        DNAME                   VARCHAR(10)  NOT NULL,
        DNUMBER                 INTEGER      NOT NULL,
        MGRSSN                  CHAR(9),
        MGRSTARTDATE            DATE,
        PRIMARY KEY (DNUMBER),
        UNIQUE (DNAME),
        FOREIGN KEY (MGRSSN) REFERENCES EMP  );
```

REFERENTIAL INTEGRITY OPTIONS

- We can specify RESTRICT, CASCADE, SET NULL or SET DEFAULT on referential integrity constraints (foreign keys)

```
CREATE TABLE DEPT (
    DNAME                VARCHAR(10)  NOT NULL,
    DNUMBE               INTEGER      NOT NULL,
    MGRSSN               CHAR(9),
    MGRSTARTDATE         DATE,
    PRIMARY KEY (DNUMBER),
    UNIQUE (DNAME),
    FOREIGN KEY (MGRSSN) REFERENCES EMP  ON DELETE
    SET DEFAULT
    ON UPDATE CASCADE);
```

REFERENTIAL INTEGRITY OPTIONS (cont.)

```
CREATE TABLE EMP(
    ENAME       VARCHAR(30)  NOT NULL,
    ESSN        CHAR(9),
    BDATE       DATE,
    DNO         INTEGER  DEFAULT 1,
    SUPERSSN    CHAR(9),
    PRIMARY KEY (ESSN),
    FOREIGN KEY (DNO) REFERENCES DEPT ON DELETE SET
    DEFAULT ON UPDATE  CASCADE,
    FOREIGN KEY (SUPERSSN) REFERENCES EMP ON DELETE
    SET NULL ON UPDATE CASCADE);
```

DROP and ALTER TABLE

DROP TABLE

- Used to remove a relation (base table) and its definition
- The relation can no longer be used in queries, updates, or any other commands since its description no longer exists
- Example: **DROP TABLE DEPENDENT;**

ALTER TABLE

- Used to add an attribute to one of the base relations
  - The new attribute will have NULLs in all the tuples of the relation right after the command is executed; hence, the NOT NULL constraint is not allowed for such an attribute
- Example: **ALTER TABLE EMPLOYEE ADD JOB VARCHAR(12);**
- The database users must still enter a value for the new attribute JOB for each EMPLOYEE tuple.
  - This can be done using the UPDATE command.

Retrieval Queries in SQL

- SQL has one basic statement for retrieving information from a database; the **SELECT** statement
- Important distinction between SQL and the formal relational model:
  - SQL allows a table to have 2 or more tuples that are identical in all their attribute values
  - Hence, an SQL relation (table) is a **multi-set** (sometimes called a **bag**) of tuples; it is *not* a set of tuples
- SQL relations can be constrained to be sets by specifying PRIMARY KEY or UNIQUE attributes, or by using the DISTINCT option in a query

Chapter Outline

1. **Basic SQL**
2. **Data Manipulation Language for Relational DBs**
3. **Data Definition Language for Relational DBs**
4. **View**
5. **Summary**

Retrieval Queries in SQL (contd.)

- Basic form of the SQL SELECT statement is called a *mapping* or a SELECT-FROM-WHERE *block*

SELECT <attribute list>
FROM <table list>
WHERE <condition>

is a list of attribute names whose values are to be retrieved by the query

is a list of the relation names required to process the query

is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query

All subsequent examples use the COMPANY database

Relational Database Schema

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE

DEPT_LOCATIONS

DNUMBER	DLOCATION

PROJECT

PNAME	PNUMBER	PLOCATION	DNUM

WORKS_ON

ESSN	PNO	HOURS

DEPENDENT

ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP

Populated Database

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
	John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
	Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
	Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
	Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
	Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
	Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
	Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
	James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
	Research	5	333445555	1988-05-22
	Administration	4	987654321	1995-01-01
	Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS	DNUMBER	DLOCATION
	1	Houston
	4	Stafford
	5	Bellaire
	5	Sugarland
	5	Houston

WORKS_ON	ESSN	PNO	HOURS
	123456789	1	32.5
	123456789	2	7.5
	666884444	3	40.0
	453453453	1	20.0
	453453453	2	20.0
	333445555	2	10.0
	333445555	3	10.0
	333445555	10	10.0
	333445555	20	10.0
	999887777	30	30.0
	999887777	10	10.0
	987987987	10	35.0
	987987987	30	5.0
	987654321	30	20.0
	987654321	20	15.0
	888665555	20	null

PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
	ProductX	1	Bellaire	5
	ProductY	2	Sugarland	5
	ProductZ	3	Houston	5
	Computerization	10	Stafford	4
	Reorganization	20	Houston	1
	Newbenefits	30	Stafford	4

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
	333445555	Alice	F	1986-04-05	DAUGHTER
	333445555	Theodore	M	1983-10-25	SON
	333445555	Joy	F	1958-05-03	SPOUSE
	987654321	Abner	M	1942-02-28	SPOUSE
	123456789	Michael	M	1988-01-04	SON
	123456789	Alice	F	1988-12-30	DAUGHTER
	123456789	Elizabeth	F	1967-05-05	SPOUSE

Example of Simple SQL Queries

Query 1: Retrieve the birthdate and address of the employee whose name is 'John B. Smith'.

Q1:
```
SELECT    BDATE, ADDRESS
FROM      EMPLOYEE
WHERE     FNAME='John' AND MINIT='B'
AND       LNAME='Smith'
```

- The SELECT-clause specifies the projection attributes and the WHERE-clause specifies the selection condition
  - The result of the query may contain duplicate tuples

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

Example of Simple SQL Queries (contd.)

Query 2: Retrieve the name and address of all employees who work for the 'Research' department.

Q2:
```
SELECT    FNAME, LNAME, ADDRESS
FROM      EMPLOYEE, DEPARTMENT
WHERE     DNAME='Research' AND DNUMBER=DNO
```

- (DNAME='Research') is a selection condition
- (DNUMBER=DNO) is a join condition

EMPLOYEE

FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-07-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	null	1

DEPARTMENT

DNAME	DNUMBER	MGRSSN	MGRSTARTDATE
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

Example of Simple SQL Queries (contd.)

Query 3: For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate.

Q3: **SELECT PNUMBER, DNUM, LNAME, BDATE, ADDRESS**
FROM PROJECT, DEPARTMENT, EMPLOYEE
WHERE DNUM=DNUMBER AND MGRSSN=SSN
AND PLOCATION='Stafford‘

In Q3, there are 2 join conditions

- The join condition DNUM=DNUMBER relates a project to its controlling department
- The join condition MGRSSN=SSN relates the controlling department to the employee who manages that department

Aliases Clause

- In SQL, we can use the same name for 2 (or more) attributes as long as the attributes are in *different relations*
- A query that refers to 2 or more attributes with the same name must *qualify* the attribute name with the relation name by *prefixing* the relation name to the attribute name

Example: **EMPLOYEE.**LNAME, **DEPARTMENT.**DNAME

- Some queries need to refer to the same relation twice
  - In this case, *aliases* are given to the relation name

- Aliasing can also be used in any SQL query for convenience

ALIASES (contd.)

Query 4: For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.

Q4:
```
SELECT    E.FNAME, E.LNAME, S.FNAME, S.LNAME
FROM      EMPLOYEE E S
WHERE     E.SUPERSSN=S.SSN
```

In Q4, the alternate relation names E and S are called *aliases* or *tuple variables* for the EMPLOYEE relation

- We can think of E and S as 2 different *copies* of EMPLOYEE; E represents employees in role of *supervisees* and S represents employees in role of *supervisors*

- Can also use the AS keyword to specify aliases

Q4:
```
SELECT   E.FNAME, E.LNAME, S.FNAME, S.LNAME
FROM     EMPLOYEE AS E,    EMPLOYEE AS S
WHERE    E.SUPERSSN=S.SSN
```

UNSPECIFIED WHERE-clause

- A *missing WHERE-clause* indicates no condition; hence, all tuples of the relations in the FROM-clause are selected
  - This is equivalent to the condition WHERE TRUE

Query 5: Retrieve the SSN values for all employees.

Q5:
```
SELECT    SSN
FROM      EMPLOYEE
```

- If more than one relation is specified in the FROM-clause *and* there is no join condition, then the *CARTESIAN PRODUCT* of tuples is selected
  - It is extremely important not to overlook specifying any selection and join conditions in the WHERE-clause; otherwise, incorrect and very large relations may result

Q6:
```
SELECT    SSN, DNAME
FROM      EMPLOYEE, DEPARTMENT
```

*USE OF **

- To retrieve all the attribute values of the selected tuples, a * is used, which stands for *all the attributes*

Examples:

Q7:
```
SELECT    *
FROM      EMPLOYEE
WHERE     DNO=5
```

Q8:
```
SELECT    *
FROM      EMPLOYEE, DEPARTMENT
WHERE     DNAME='Research' AND DNO=DNUMBER
```

USE OF DISTINCT

- SQL does not treat a relation as a set; duplicate tuples can appear
- To eliminate duplicate tuples in a query result, the keyword **DISTINCT** is used
- For example, the result of Q9 may have duplicate SALARY values whereas Q10 does not have any duplicate value

Q9:
```
SELECT    SALARY
FROM      EMPLOYEE
```

Q10:
```
SELECT    DISTINCT SALARY
FROM      EMPLOYEE
```

SET OPERATIONS

- SQL has directly incorporated some set operations
- There is a union operation (UNION), and in *some versions* of SQL there are set difference (MINUS) and intersection (INTERSECT) operations
- The resulting relations of these set operations are sets of tuples; *duplicate tuples are eliminated from the result*
- The set operations apply only to *union compatible relations*; the two relations must have the same attributes and the attributes must appear in the same order

SET OPERATIONS (contd.)

Query 11: List of all project names for projects that involve an employee whose last name is 'Smith' as a worker or as a manager of the department that controls the project.

Q11:
```
(SELECT  PNAME
 FROM    PROJECT, DEPARTMENT, EMPLOYEE
 WHERE   DNUM=DNUMBER AND MGRSSN=SSN AND
         LNAME='Smith')
 UNION
(SELECT  PNAME
 FROM    PROJECT, WORKS_ON, EMPLOYEE
 WHERE   PNUMBER=PNO AND ESSN=SSN AND NAME='Smith')
```

NESTING OF QUERIES

- A complete SELECT query, called a *nested query*, can be specified within the WHERE-clause of another query, called the *outer query*
  - Many of the previous queries can be specified in an alternative form using nesting
- In general, we can have several levels of nested queries
- A reference to an *unqualified attribute* refers to the relation declared in the *innermost nested query*