

Fundamentals of Database Systems

Chapter 4. Basic SQL

Myungsun Kim

Information_Security@The_University_of_Suwon

2017, Fall

Contents

- CHAP. 4 Basic SQL
 - * §4.1 SQL Data Definition and Data Types
 - * §4.2 Specifying Constraints in SQL
 - * §4.3 Basic Retrieval Queries in SQL
 - * §4.4 INSERT, DELETE, and UPDATE Statements in SQL
 - * §4.5 Additional Features of SQL
 - * §4.6 Summary
- Wrap-up & Questions

§4.1 SQL Data Definition and Data Types

- SQL uses **table**, **row**, and **column** instead of relation, tuple, and attribute
- Main SQL DDL command: The **CREATE** statement.

§4.1.1 Schema and Catalog Concepts in SQL

- ① Create an RDB schema

Syntax:

```
CREATE SCHEMA db_name AUTHORIZATION auth_id;
```

Example:

```
CREATE SCHEMA COMPANY AUTHORIZATION "mscho";
```

- ② Catalog: INFORMATION_SCHEMA

A special schema provides information on all the schema in the catalog and all element descriptors in these schemas

§4.1 SQL Data Definition and Data Types

§4.1.2 The `CREATE TABLE` Command in SQL

The `CREATE TABLE` command: Define a new relation by (1) giving the name, (2) specifying its attributes, and (3) initial constraints

- Syntax: `CREATE TABLE table_name (column_list);`

- * To define a column for the table `table_name`

- * Syntax:

```
column_name data_type(size) [NOT NULL|NULL]
           [DEFAULT value] [CHECK(condition)]
```

- * Examples:

①	Ssn	CHAR(9)	NOT NULL
②	Address	VARCHAR(3)	
③	Dno	INT	NOT NULL
④	Salary	DECIMAL(10,2)	
⑤	Bdate	DATE	

§4.1 SQL Data Definition and Data Types

Example: EMPLOYEE

```
CREATE TABLE EMPLOYEE(  
    Fname      VARCHAR(15)    NOT NULL,  
    Minit      CHAR,  
    Lname      VARCHAR(15)    NOT NULL,  
    Ssn        CHAR(9)        NOT NULL,  
    Bdate      DATE,  
    Address    VARCHAR(30),  
    Sex        CHAR,  
    Salary     DECIMAL(10,2),  
    Super_ssn CHAR(9),  
    Dno        INT            NOT NULL);
```

§4.1 SQL Data Definition and Data Types

§4.1.3 Attribute Data Types and Domains in SQL

- Basic data types
 - * Numeric data types
 - `INTEGER`, `INT`, `SMALLINT`, `FLOAT`, `REAL`
 - `DECIMAL(i,j)` where *i*: precision, *j*: scale
 - * Character-string data types
 - `CHAR(n)`, `VARCHAR(n)`
 - `CLOB(n)`
 - * Bit-string data type
 - `BIT(n)` or `BIT VARYING(n)`
 - `BLOB(n)`
 - * Boolean data type has three values: `TRUE`, `FALSE`, `UNKNOWN`
 - * DATE/TIME data types
 - `DATE` type: In the form of YYYY-MM-DD
 - `TIME` type: In the form of HH:MM:SS

§4.2 Specifying Constraints in SQL

§4.2.1 Specifying Attribute Constraints and Attribute Defaults

- Default value: **DEFAULT**

```
CREATE TABLE EMPLOYEE(  
    ...  
    Ssn CHAR(9) NOT NULL DEFAULT "123456789",  
    ...  
    Dno INT NOT NULL DEFAULT 1);
```

- Restrict attributes: **CHECK**

```
CREATE TABLE EMPLOYEE(  
    ...  
    Sex CHAR CHECK(Sex="M" OR Sex="F"),  
    ...  
    Dno INT NOT NULL DEFAULT 1);
```

§4.2 Specifying Constraints in SQL

§4.2.2 Specifying Key and Referential Integrity Constraints

- Key constraint: **PRIMARY KEY** or **UNIQUE**

- * The **PRIMARY KEY** clause
- * The **UNIQUE** clause
- * Example

```
CREATE TABLE EMPLOYEE(  
    Fname VARCHAR(15) NOT NULL,  
    ...  
    Dno     INT           NOT NULL DEFAULT 1,  
    PRIMARY KEY (Ssn));
```

```
CREATE TABLE DEP_LOCATIONS(  
    ...  
    PRIMARY KEY (Dnumber, Dlocation));
```


§4.2 Specifying Constraints in SQL

§4.2.2 Specifying Key and Referential Integrity Constraints

- Referential integrity constraint: **FOREIGN KEY REFERENCES**

- * The **FOREIGN KEY** clause

- * Example

```
CREATE TABLE EMPLOYEE(  
    Fname  VARCHAR(15) NOT NULL,  
    ...  
    Dno    INT          NOT NULL DEFAULT 1,  
    PRIMARY KEY (Ssn),  
    FOREIGN KEY (Super_ssn) REFERENCES  
        EMPLOYEE(Ssn),  
    FOREIGN KEY (Dno) REFERENCES  
        DEPARTMENT(Dnumber));
```

§4.2 Specifying Constraints in SQL

§4.2.2 Specifying Key and Referential Integrity Constraints

- Referential triggered actions
 - * Combine options
{**RESTRICT**, **SET NULL**, **SET DEFAULT**, **CASCADE**} and
quantifiers {**ON DELETE**, **ON UPDATE**}
 - * Example

```
CREATE TABLE EMPLOYEE(  
    ...  
    Dno      INT          NOT NULL DEFAULT 1,  
    PRIMARY KEY (Ssn),  
    FOREIGN KEY (Super_ssn) REFERENCES  
        EMPLOYEE(Ssn),  
    FOREIGN KEY (Dno) REFERENCES  
        DEPARTMENT(Dnumber)  
    ON DELETE SET NULL ON UPDATE CASCADE);
```

§4.3 Basic Retrieval Queries in SQL

§4.3.1 The SELECT-FROM-WHERE Structure of Basic SQL Queries

- The basic form of the **SELECT** statement

```
SELECT attribute_list
```

```
FROM table_list
```

```
WHERE condition;
```

where

- * `attribute_list`: a list of attribute names whose values are to be retrieved by the query
- * `table_list`: a list of the relation names required to process the query
- * `condition`: a Boolean expression that identifies the tuples to be retrieved by the query. The basic logical comparison operators: =, <, <=, >, >=, <>

§4.3 Basic Retrieval Queries in SQL

§4.3.1 The SELECT-FROM-WHERE Structure of Basic SQL Queries

- Query #0. Retrieve the birth date and address of the employee whose name is “John B. Smith”.

```
⇒ SELECT Bdate, Address
   FROM EMPLOYEE
   WHERE Fname="John" AND Lname="Smith"
      AND Minit="B";
```

- Query #1. Retrieve the name and address of all employees who work for the “Research” department

```
⇒ SELECT Fname, Lname, Address
   FROM EMPLOYEE, DEPARTMENT
   WHERE Dname="Research" AND Dnumber=Dno;
```

§4.3 Basic Retrieval Queries in SQL

§4.3.2 Ambiguous Attribute Names, Aliasing, Renaming, and Tuple Variables

- Two or more attributes with the **same** name are used in a multitable query → How to avoid ambiguity?
- Prefixing + Aliasing (or Renaming is also possible)
- Recall Query #1

```
⇒ SELECT E.Fname, E.Lname, E.Address  
FROM EMPLOYEE E, DEPARTMENT D  
WHERE D.Dname="Research"  
AND D.Dnumber=E.Dno;
```

§4.3 Basic Retrieval Queries in SQL

§4.3.3 Unspecified WHERE Clause and Use of the Asterisk

- No **WHERE** clause
- Example

```
SELECT E.Fname  
FROM EMPLOYEE E;
```

- To retrieve all the attribute values
- Example

```
SELECT *  
FROM EMPLOYEE E, DEPARTMENT D  
WHERE D.Dname="Research" AND D.Dnumber=E.Dno;
```

§4.3 Basic Retrieval Queries in SQL

§4.3.4 Tables as Sets in SQL

- In SQL, Tables \neq Sets but Tables \approx Multisets
- The keyword **DISTINCT**
- Query #11A. Retrieve all distinct salary values

```
⇒ SELECT DISTINCT Salary  
FROM EMPLOYEE;
```

§4.3 Basic Retrieval Queries in SQL

§4.3.4 Tables as Sets in SQL

- SQL set operations: **UNION**, **INTERSECT**, **EXCEPT**
- Query #4. Make a list of all project numbers for projects that involve an employee whose last name is “Smith”, either as a worker or as a manager of the department that controls the project

```
⇒ (SELECT DISTINCT P.Pnumber
    FROM EMPLOYEE E, DEPARTMENT D, PROJECT P
    WHERE P.Dnum=D.Dnumber AND D.Mgr_ssn=E.Ssn
        AND E.Lname="Smith") UNION
(SELECT DISTINCT P.Pnumber
    FROM EMPLOYEE E, WORKS_ON W, PROJECT P
    WHERE P.Pnumber=W.Pno AND W.Essn=E.Ssn
        AND E.Lname="Smith");
```


§4.3 Basic Retrieval Queries in SQL

§4.3.5 Substring Pattern Matching and Arithmetic Operators

- String pattern matching
- The **LIKE** operator with wildcards: % and _
- Query #12A. Find all employees who were born during the 1950s

```
⇒ SELECT Fname, Lname  
   FROM   EMPLOYEE  
   WHERE  Bdate LIKE "__5_____";
```

```
or SELECT Fname, Lname  
   FROM   EMPLOYEE  
   WHERE  Bdate LIKE "__5%";
```

§4.3 Basic Retrieval Queries in SQL

§4.3.5 Substring Pattern Matching and Arithmetic Operators

- Arithmetics in queries
- Query #13. Show the resulting salaries if every employee working on the “ProjectX” project is given a 10 percent raise.

```
⇒ SELECT E.Fname, E.Lname, 1.1*E.Salary
FROM   EMPLOYEE E, WORKS_ON W, PROJECT P
WHERE  E.Ssn=W.Essn AND W.Pno=P.Pnumber
      AND P.Pname="ProjectX";
```

§4.3 Basic Retrieval Queries in SQL

§4.3.6 Ordering of Query Results

- The **ORDER BY** clause outputs the ordered result of a query by some attributes
- Query #15. Retrieve a list of employees and project they are working on, ordered by department and, within each department, ordered alphabetically by the last name, then first name.

```
⇒ SELECT   D.Dname, E.Fname, E.Lname, P.Pname
FROM      EMPLOYEE E, DEPARTMENT D,
            PROJECT P, WORKS_ON W
WHERE     D.Dnumber=E.Dno AND E.Ssn=W.Essn
            AND      W.Pno=P.Pnumber
ORDER BY D.Dname, E.Lname, E.Fname;
```

§4.4 INSERT, DELETE, and UPDATE Statements in SQL

§4.4.1 The INSERT Command

- Add a single tuple to a relation
- Syntax.

```
INSERT INTO table_name(attribute1, attribute2, ...)
VALUES      (value1, value2, ...);
```

- Example.

```
INSERT INTO EMPLOYEE(Fname, Lname, Dno, Ssn)
VALUES      ("Richard", "Marini", 4, "653298653");
```



Other attributes not specified in the Example?

§4.4 INSERT, DELETE, and UPDATE Statements in SQL

§4.4.2 The DELETE Command

- Remove tuples from a relation
- Syntax.

```
DELETE FROM table_name  
WHERE          condition;
```

- Example.

```
DELETE FROM EMPLOYEE  
WHERE      Lname="Brown";
```

§4.4 INSERT, DELETE, and UPDATE Statements in SQL

§4.4.3 The UPDATE Command

- Modify attribute values of one or more selected tuples
- Syntax.

```
UPDATE table_name
SET     attribute1=value1, attribute2=value2, ...
WHERE  condition;
```

- Example.

```
UPDATE PROJECT
SET     Plocation="Bellaire", Dnum=5
WHERE  Pnumber=10;
```

Wrap-up & Questions

정리

- Relational DB Schema definition and data types
- Basics in retrieval queries
- Simple update queries

***** Thanks & Question? *****

A state of RDB schema: COMPANY

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	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	1968-01-19	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	Mgr_ssn	Mgr_start_date
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

← back §4.3.1

WORKS_ON

Esn	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

← back §4.3.4

DEPENDENT

Esn	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