

Basic SQL

Lecture 6 [Chapter 6]

Mohamed Elshaikh

Chapter 6 Outline

- SQL Data Definition and Data Types
- Specifying Constraints in SQL
- Basic Retrieval Queries in SQL
- `INSERT`, `DELETE`, and `UPDATE` Statements in SQL
- Additional Features of SQL

Basic SQL

- SQL language
 - Considered one of the major reasons for the commercial success of relational databases
- SQL
 - The origin of SQL is relational predicate calculus called tuple calculus (see Ch.8) which was proposed initially as the language SQUARE.
 - SQL Actually comes from the word “SEQUEL” which was the original term used in the paper: “SEQUEL TO SQUARE” by Chamberlin and Boyce. IBM could not copyright that term, so they abbreviated to SQL and copyrighted the term SQL.
 - Now popularly known as “Structured Query language”.
 - SQL is an informal or practical rendering of the relational data model with syntax

SQL Data Definition, Data Types, Standards

- Terminology:
 - **Table**, **row**, and **column** used for relational model terms relation, tuple, and attribute
- CREATE statement
 - Main SQL command for data definition
- The language has features for : Data definition, Data Manipulation, Transaction control (Transact-SQL, Ch. 20), Indexing (Ch.17), Security specification (Grant and Revoke- see Ch.30), Active databases (Ch.26), Multi-media (Ch.26), Distributed databases (Ch.23) etc.

SQL Standards

- SQL has gone through many standards: starting with SQL-86 or SQL 1.A. SQL-92 is referred to as SQL-2.
- Later standards (from SQL-1999) are divided into **core** specification and specialized **extensions**. The extensions are implemented for different applications – such as data mining, data warehousing, multimedia etc.
- SQL-2006 added XML features (Ch. 13); In 2008 they added Object-oriented features (Ch. 12).
- SQL-3 is the current standard which started with SQL-1999. It is not fully implemented in any RDBMS.

Schema and Catalog Concepts in SQL

- We cover the basic standard SQL syntax – there are variations in existing RDBMS systems
- **SQL schema**
 - Identified by a **schema name**
 - Includes an **authorization identifier** and **descriptors** for each element
- **Schema elements** include
 - Tables, constraints, views, domains, and other constructs
- Each statement in SQL ends with a **semicolon**

Schema and Catalog Concepts in SQL (cont'd.)

- `CREATE SCHEMA` statement
 - `CREATE SCHEMA COMPANY AUTHORIZATION 'Jsmith' ;`
- **Catalog**
 - Named collection of schemas in an SQL environment
- SQL also has the concept of a cluster of catalogs.

The CREATE TABLE Command in SQL

- Specifying a new relation
 - Provide name of table
 - Specify attributes, their types and initial constraints
- Can optionally specify schema:
 - `CREATE TABLE COMPANY.EMPLOYEE ...`
 - or
 - `CREATE TABLE EMPLOYEE ...`

The CREATE TABLE Command in SQL (cont'd.)

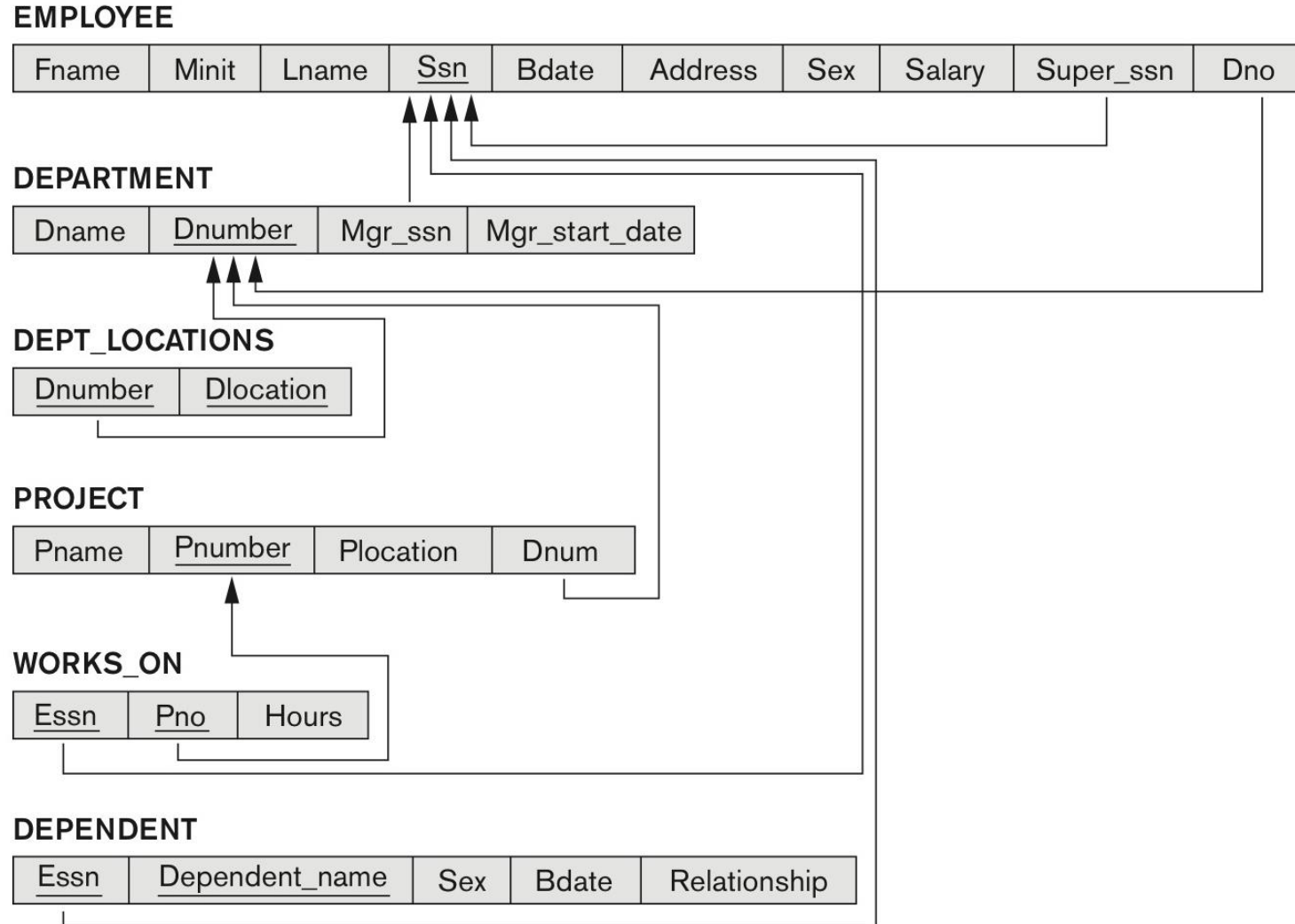
- **Base tables (base relations)**

- Relation and its tuples are actually created and stored as a file by the DBMS

- **Virtual relations (views)**

- Created through the `CREATE VIEW` statement. Do not correspond to any physical file.

COMPANY relational database schema (Fig. 5.7)



One possible database state for the COMPANY relational database schema (Fig. 5.6)

EMPLOYEE

Fname	Minit	Lname	<u>Ssn</u>	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	<u>Dnumber</u>	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

<u>Dnumber</u>	<u>Dlocation</u>
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

One possible database state for the COMPANY relational database schema – continued (Fig. 5.6)

WORKS_ON

<u>Essn</u>	<u>Pno</u>	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

<u>Pname</u>	<u>Pnumber</u>	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

<u>Essn</u>	<u>Dependent_name</u>	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

SQL CREATE TABLE data definition statements for defining the COMPANY schema from Figure 5.7 (Fig. 6.1)

```
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,
    PRIMARY KEY (Ssn),
CREATE TABLE DEPARTMENT
  ( Dname          VARCHAR(15)          NOT NULL,
    Dnumber        INT                  NOT NULL,
    Mgr_ssn        CHAR(9)              NOT NULL,
    Mgr_start_date DATE,
    PRIMARY KEY (Dnumber),
    UNIQUE (Dname),
    FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn) );
CREATE TABLE DEPT_LOCATIONS
  ( Dnumber        INT                  NOT NULL,
    Dlocation      VARCHAR(15)          NOT NULL,
    PRIMARY KEY (Dnumber, Dlocation),
    FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber) );
```

continued on next slide

SQL CREATE TABLE data definition statements for defining the COMPANY schema from Figure 5.7 (Fig. 6.1)-continued

```
CREATE TABLE PROJECT
( Pname          VARCHAR(15)          NOT NULL,
  Pnumber        INT                 NOT NULL,
  Plocation      VARCHAR(15),
  Dnum           INT                 NOT NULL,
  PRIMARY KEY (Pnumber),
  UNIQUE (Pname),
  FOREIGN KEY (Dnum) REFERENCES DEPARTMENT(Dnumber) );

CREATE TABLE WORKS_ON
( Essn           CHAR(9)             NOT NULL,
  Pno            INT                 NOT NULL,
  Hours          DECIMAL(3,1)        NOT NULL,
  PRIMARY KEY (Essn, Pno),
  FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn),
  FOREIGN KEY (Pno) REFERENCES PROJECT(Pnumber) );

CREATE TABLE DEPENDENT
( Essn           CHAR(9)             NOT NULL,
  Dependent_name VARCHAR(15)         NOT NULL,
  Sex            CHAR,
  Bdate         DATE,
  Relationship   VARCHAR(8),
  PRIMARY KEY (Essn, Dependent_name),
  FOREIGN KEY (Essn) REFERENCES EMPLOYEE(Ssn) );
```

Attribute Data Types and Domains in SQL

- **Basic data types**
 - **Numeric data types**
 - Integer numbers: `INTEGER`, `INT`, and `SMALLINT`
 - Floating-point (real) numbers: `FLOAT` or `REAL`, and `DOUBLE PRECISION`
 - **Character-string data types**
 - Fixed length: `CHAR (n)`, `CHARACTER (n)`
 - Varying length: `VARCHAR (n)`, `CHAR VARYING (n)`, `CHARACTER VARYING (n)`

Attribute Data Types and Domains in SQL (cont'd.)

- **Bit-string** data types
 - Fixed length: `BIT (n)`
 - Varying length: `BIT VARYING (n)`
- **Boolean** data type
 - Values of `TRUE` or `FALSE` or `NULL`
- **DATE** data type
 - Ten positions
 - Components are `YEAR`, `MONTH`, and `DAY` in the form `YYYY-MM-DD`
 - Multiple mapping functions available in RDBMSs to change date formats

Attribute Data Types and Domains in SQL (cont'd.)

- Additional data types

- **Timestamp** data type

Includes the `DATE` and `TIME` fields

- Plus a minimum of six positions for decimal fractions of seconds
 - Optional `WITH TIME ZONE` qualifier

- **INTERVAL** data type

- Specifies a relative value that can be used to increment or decrement an absolute value of a date, time, or timestamp

- **DATE, TIME, Timestamp, INTERVAL** data types can be **cast** or converted to string formats for comparison.

Attribute Data Types and Domains in SQL (cont'd.)

- **Domain**

- Name used with the attribute specification
- Makes it easier to change the data type for a domain that is used by numerous attributes
- Improves schema readability
- Example:

- `CREATE DOMAIN SSN_TYPE AS CHAR(9);`

- **TYPE**

- User Defined Types (UDTs) are supported for object-oriented applications. (See Ch.12) Uses the command:

- `CREATE TYPE`

Specifying Constraints in SQL

Basic constraints:

- Relational Model has 3 basic constraint types that are supported in SQL:
 - **Key** constraint: A primary key value cannot be duplicated
 - **Entity Integrity** Constraint: A primary key value cannot be null
 - **Referential integrity** constraints : The “foreign key “ must have a value that is already present as a primary key, or may be null.

Specifying Attribute Constraints

Other Restrictions on attribute domains:

- Default value of an attribute
 - **DEFAULT** <value>
 - NULL is not permitted for a particular attribute (NOT NULL)
- **CHECK** clause
 - Dnumber INT NOT NULL CHECK (Dnumber > 0 AND Dnumber < 21);

Specifying Key and Referential Integrity Constraints

- **PRIMARY KEY** clause
 - Specifies one or more attributes that make up the primary key of a relation
 - `Dnumber INT PRIMARY KEY;`
- **UNIQUE** clause
 - Specifies alternate (secondary) keys (called CANDIDATE keys in the relational model).
 - `Dname VARCHAR(15) UNIQUE;`

Specifying Key and Referential Integrity Constraints (cont'd.)

- **FOREIGN KEY** clause
 - Default operation: reject update on violation
 - Attach **referential triggered action** clause
 - Options include `SET NULL`, `CASCADE`, and `SET DEFAULT`
 - Action taken by the DBMS for `SET NULL` or `SET DEFAULT` is the same for both `ON DELETE` and `ON UPDATE`
 - `CASCADE` option suitable for “relationship” relations

Giving Names to Constraints

- Using the Keyword **CONSTRAINT**
 - Name a constraint
 - Useful for later altering

Default attribute values and referential integrity triggered action specification (Fig. 6.2)

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

CREATE TABLE DEPARTMENT
(
  ... ,
  Mgr_ssn CHAR(9)          NOT NULL      DEFAULT '888665555',
  ... ,
  CONSTRAINT DEPTPK
  PRIMARY KEY(Dnumber),
  CONSTRAINT DEPTSK
  UNIQUE (Dname),
  CONSTRAINT DEPTMGRFK
  FOREIGN KEY (Mgr_ssn) REFERENCES EMPLOYEE(Ssn)
  ON DELETE SET DEFAULT   ON UPDATE CASCADE);

CREATE TABLE DEPT_LOCATIONS
(
  ... ,
  PRIMARY KEY (Dnumber, Dlocation),
  FOREIGN KEY (Dnumber) REFERENCES DEPARTMENT(Dnumber)
  ON DELETE CASCADE      ON UPDATE CASCADE);
```


Specifying Constraints on Tuples Using CHECK

- Additional Constraints on individual tuples within a relation are also possible using CHECK
- CHECK clauses at the end of a CREATE TABLE statement
 - Apply to each tuple individually
 - CHECK (Dept_create_date <= Mgr_start_date);

Summary

- SQL
 - A Comprehensive language for relational database management
 - Data definition, queries, updates, constraint specification, and view definition
- Covered :
 - Data definition commands for creating tables
 - Commands for constraint specification