Fundamentals of Database Systems (6th Edition)

2018-2019 Academic Year

Department of Computer Engineering and Information Technology

Relational Data Model and SQL

Chapter-4: Basic SQL

Lectured by

Daw Honey Htun

Assistant Lecturer (Ph.D, Thesis)

Department of Computer Engineering and Information Technology

Data Definition, Constraints, and Schema Changes

- Used to CREATE, DROP, and ALTER the descriptions of the tables (relations) of a database

CREATE DATABASE

- Specifies a new database by giving it a name

  CREATE DATABASE STUDENT;

- **OR**

CREATE DATABASE STUDENT CHARACTER SET UTF8

COLLATE UTF8_UNICODE_CI;

CREATE TABLE

- Specifies a new base relation by giving it a name, and specifying each of its attributes and their data types **(INTEGER, FLOAT, DECIMAL(i, j), CHAR(n), VARCHAR(n))**
- 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   CHAR(9)  );
```

CREATE TABLE

- Use the CREATE TABLE command for specifying the primary key attributes, secondary keys, and referential integrity constraints (foreign keys).
- 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        CHAR(9),
       PRIMARY KEY       (DNUMBER),
       UNIQUE            (DNAME),
       FOREIGN KEY       (MGRSSN) REFERENCES EMP  );
```

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 DEPARTMENT;

ALTER TABLE

- Used to add an attribute to one of the base relations
- 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.

Additional Data Types

Has DATE, TIME, and TIMESTAMP data types

- **DATE:**
  - Made up of year-month-day in the format yyyy-mm-dd
- **TIME:**
  - Made up of hour:minute:second in the format hh:mm:ss
- **TIME(i):**
  - Made up of hour:minute:second plus i additional digits specifying fractions of a second
  - format is hh:mm:ss:ii...i
- **TIMESTAMP:**
  - Has both DATE and TIME components

Additional Data Types (cont.)

- **INTERVAL:**
  - Specifies a relative value rather than an absolute value
  - Can be DAY/TIME intervals or YEAR/MONTH intervals
  - Can be positive or negative when added to or subtracted from an absolute value, the result is an absolute value

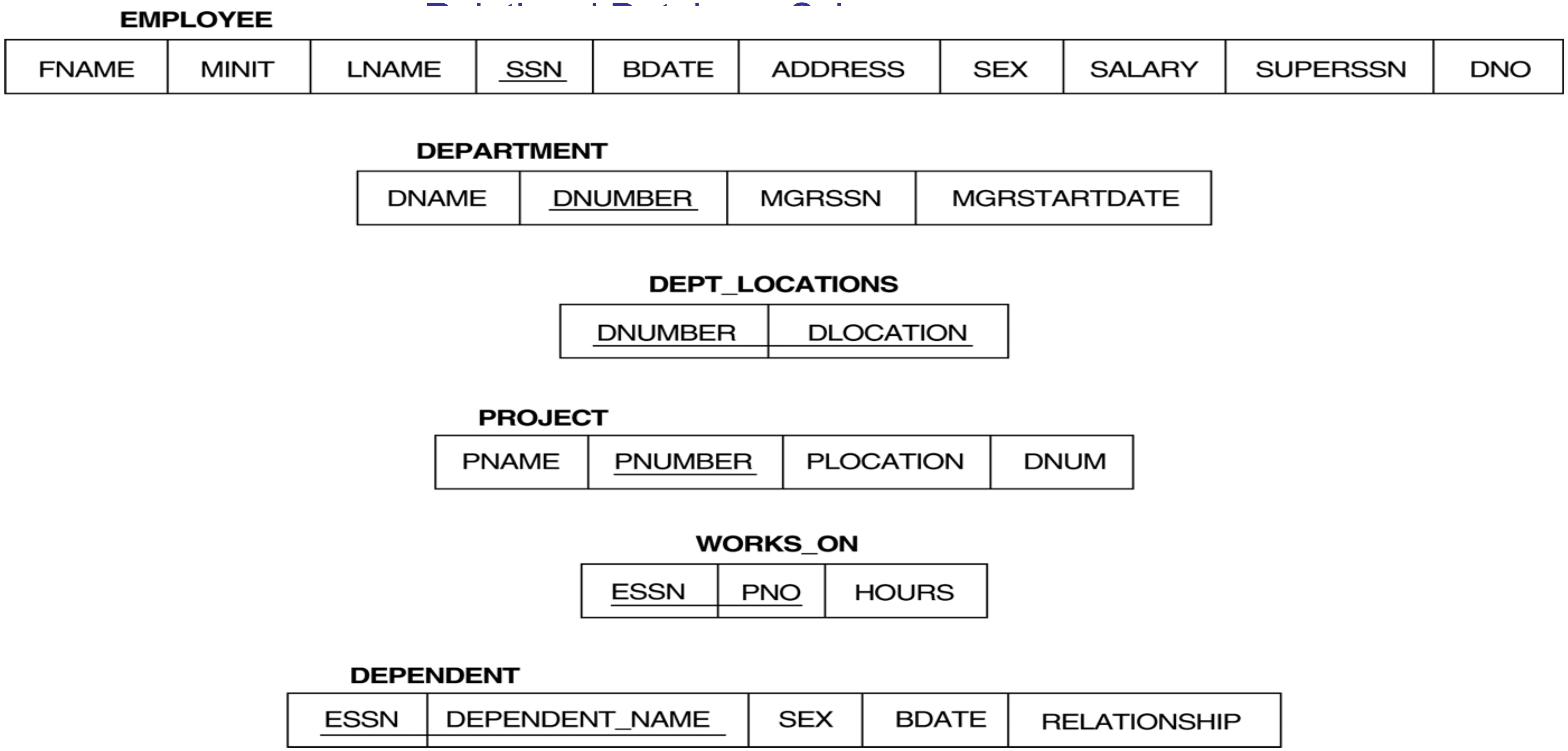
Retrieval Queries in SQL

- SQL has one basic statement for retrieving information from a database; the SELECT statement
- SQL relations can be constrained to be sets by specifying PRIMARY KEY or UNIQUE attributes, or by using the DISTINCT option in a query
- Basic form of the SQL SELECT statement is called a *mapping* or a *SELECT-FROM-WHERE block*

Retrieval Queries in SQL (cont.)

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

- <attribute list> is a list of attribute names whose values are to be retrieved by the query
- <table list> is a list of the relation names required to process the query
- <condition> is a conditional (Boolean) expression that identifies the tuples to be retrieved by the query



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

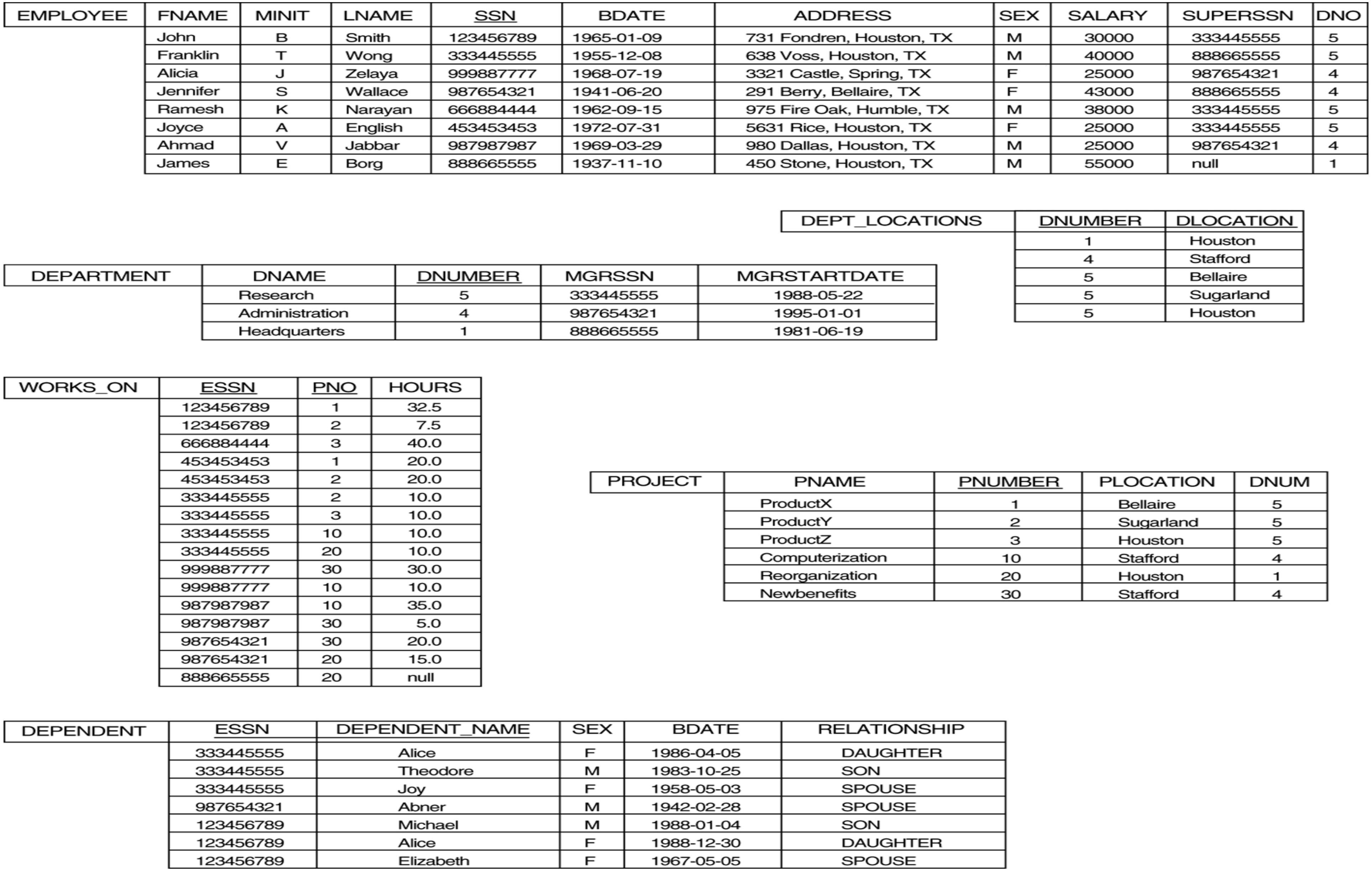
WORKS_ON

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

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

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

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

Simple SQL Queries

- Basic SQL queries correspond to using the SELECT, PROJECT, and JOIN operations of the relational algebra
- All subsequent examples use the COMPANY database
- Example of a simple query on *one* relation

- **Query 0:** Retrieve the birthdate and address of the employee whose name is 'John B. Smith'.

```
Q0:  SELECT  BDATE, ADDRESS
     FROM    EMPLOYEE
     WHERE   FNAME='John' AND MINIT='B’
     AND     LNAME='Smith’;
```

❑ The result of the query *may contain* duplicate tuples

- **Query 1:** Retrieve the name and address of all employees who work for the 'Research' department.

```
Q1:  SELECT    FNAME, LNAME, ADDRESS
     FROM   EMPLOYEE, DEPARTMENT
     WHEREDNAME='Research' AND DNUMBER=DNO;
```

- ❑ (DNAME='Research') is a *selection condition* (corresponds to a SELECT operation in relational algebra)
- ❑ (DNUMBER=DNO) is a *join condition* (corresponds to a JOIN operation in relational algebra)

- **Query 2:** For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthdate.

```
Q2:  SELECT     PNUMBER, DNUM, LNAME, BDATE,
   ADDRESS
             FROM   PROJECT, DEPARTMENT, EMPLOYEE
             WHERE     DNUM=DNUMBER
             AND       MGRSSN=SSN
             AND       PLOCATION='Stafford‘;
```

Aliases

- Can use the same name for two (or more) attributes as long as the attributes are in *different relations*.

- A query that refers to two 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
- **Query 8:** For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.

```
Q8:   SELECT    E.FNAME, E.LNAME, S.FNAME,  S.LNAME
      FROM   EMPLOYEE E S
      WHEREE.SUPERSSN=S.SSN
```

- For convenience, can also use the ***AS*** keyword to specify aliases

- Q8:

```
Q8:    SELECT    E.FNAME, E.LNAME, S.FNAME,
S.LNAME
       FROM   EMPLOYEE AS E, EMPLOYEE AS S
       WHEREE.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
- **<u>Query 9:</u>** Retrieve the SSN values for all employees.

```
Q9:    SELECT    SSN
       FROM   EMPLOYEE
```

UNSPECIFIED WHERE-clause (cont.)

- **Example:**

```
Q10: SELECT    SSN, DNAME
     FROM   EMPLOYEE, DEPARTMENT
```

  - 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

USE OF *

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

Examples:

```
Q1C: SELECT    *
     FROM   EMPLOYEE
     WHEREDNO=5;

Q1D: SELECT    *
     FROM   EMPLOYEE, DEPARTMENT
     WHEREDNAME='Research' AND
DNO=DNUMBER;
```

USE OF DISTINCT

QL does not treat a relation as a set, ***duplicate tuples can appear***

eliminate duplicate tuples in a query result, the keyword **STINCT** is used

r example, the result of Q11 may have duplicate SALARY values hereas Q11A does not have any duplicate values

```
11:  SELECT    SALARY
     FROM   EMPLOYEE;

11A:    SELECT    DISTINCT SALARY
     FROM   EMPLOYEE;
```

SUBSTRING COMPARISON

- The **LIKE** comparison operator is used to compare partial strings.
- Two reserved characters are used:
- **'%'** replaces an arbitrary number of characters
- **'_'** replaces a single arbitrary character

- **Query 25:** Retrieve all employees whose address is in Houston, Texas. Here, the value of the ADDRESS attribute must contain the substring 'Houston,TX'.

```
Q25: SELECT    FNAME, LNAME
     FROM   EMPLOYEE
     WHEREADDRESS LIKE '%Houston,TX%’;
```

- **Query 26:** Retrieve all employees who were born during the 1950s. Here, '5' must be the 8th character of the string (according to our format for date), so the BDATE value is '_______5_', with each underscore as a place holder for a single arbitrary character.

```
Q26:  SELECT     FNAME, LNAME
      FROM   EMPLOYEE
      WHEREBDATE LIKE '_______5_’;
```

ORDER BY

- Used to sort the tuples in a query result based on the values of some attribute(s).

- <u>**Query 28:**</u> Retrieve a list of employees and the projects each works in, ordered by the employee's department, and within each department ordered alphabetically by employee last name.

```
Q28:  SELECT     DNAME, LNAME, FNAME, PNAME
         FROM    DEPARTMENT, EMPLOYEE, WORKS_ON,
         PROJECT
      WHEREDNUMBER=DNO AND SSN=ESSN             AND
   PNO=PNUMBER
      ORDER BY    DNAME, LNAME;
```

- The default order is in **ascending order** of values.

- Specify the keyword **DESC** if we want a descending order; the keyword **ASC** can be used to explicitly specify ascending order, even though it is the default.

Summary of SQL Queries

- A query in SQL can consist of **up to six clauses**, but only the first two, SELECT and FROM, are mandatory.

```
SELECT          <attribute list>
FROM        <table list>
[WHERE          <condition>]
[GROUP BY   <grouping attribute(s)>]
[HAVING     <group condition>]
[ORDER BY    <attribute list>]
```

- The **SELECT**-clause lists the attributes or functions to be retrieved
- The **FROM**-clause specifies all relations (or aliases) needed in the query but not those needed in nested queries
- The **WHERE**-clause specifies the conditions for selection and join of tuples from the relations specified in the FROM-clause
- **GROUP BY** specifies grouping attributes
- **HAVING** specifies a condition for selection of groups
- **ORDER BY** specifies an order for displaying the result of a query
- A query is evaluated by first applying the WHERE-clause, then GROUP BY and HAVING, and finally the SELECT-clause

Specifying Updates in SQL

- There are three SQL commands to modify the database;
- INSERT,
- DELETE, and
- UPDATE

INSERT

- In its simplest form, it is used to add one or more tuples to a relation.
- Attribute values should be listed in the same order as the attributes were specified in the CREATE TABLE command

- Example:

```
U1:    INSERT INTO  EMPLOYEE
    VALUES ('Richard','K','Marini', '653298653', '30-DEC-52',
    '98 Oak Forest,Katy,TX', 'M', 37000,'987654321', 4 );
```

- An alternate form of INSERT specifies explicitly the attribute names that correspond to the values in the new tuple
- Attributes with NULL values can be left out

- Example: Insert a tuple for a new EMPLOYEE for whom we only know the FNAME, LNAME, and SSN attributes.
- **U1A: INSERT INTO EMPLOYEE (FNAME, LNAME, SSN)**

  VALUES ('Richard', 'Marini', '653298653');
- ***Important Note***: Only the constraints specified in the DDL commands are automatically enforced by the DBMS when updates are applied to the database.
- Another variation of INSERT allows insertion of *multiple tuples* resulting from a query into a relation.

- **Example:** Suppose we want to create a temporary table that has the name, number of employees, and total salaries for each department. A table DEPTS_INFO is created by U3A, and is loaded with the summary information retrieved from the database by the query in U3B.

```
U3A:  CREATE TABLE  DEPTS_INFO
          (DEPT_NAME    VARCHAR(10),
           NO_OF_EMPS   INTEGER,
           TOTAL_SAL    INTEGER);

U3B:  INSERT INTO    DEPTS_INFO (DEPT_NAME,
      NO_OF_EMPS, TOTAL_SAL)
      SELECT      DNAME, COUNT (*), SUM (SALARY)
      FROM        DEPARTMENT, EMPLOYEE
      WHERE       DNUMBER = DNO
      GROUP BY         DNAME ;
```

INSERT (cont.)

- Note: The DEPTS_INFO table may not be up-to-date if we change the tuples in either the DEPARTMENT or the EMPLOYEE relations *after* issuing U3B.
- We have to create a view (see later) to keep such a table up to date.

DELETE

- Removes tuples from a relation
- Includes a WHERE-clause to select the tuples to be deleted
- Tuples are deleted from only *one table* at a time (unless CASCADE is specified on a referential integrity constraint)
- A missing WHERE-clause specifies that *all tuples* in the relation are to be deleted; the table then becomes an empty table
- The number of tuples deleted depends on the number of tuples in the relation that satisfy the WHERE-clause
- Referential integrity should be enforced

- <u>Examples:</u>

```
U4A: DELETE       FROM  EMPLOYEE
     WHERE    LNAME='Brown’;

U4B: DELETE       FROM  EMPLOYEE
     WHERE    SSN='123456789’;

U4C: DELETE       FROM  EMPLOYEE
     WHERE    DNO  IN                (SELECT
DNUMBER
      FROM    DEPARTMENT
     WHERE    DNAME='Research');

U4D: DELETE FROM  EMPLOYEE;
```

UPDATE

- Used to modify attribute values of one or more selected tuples
- A WHERE-clause selects the tuples to be modified
- An additional SET-clause specifies the attributes to be modified and their new values
- Each command modifies tuples *in the same relation*
- Referential integrity should be enforced

- Example: Change the location and controlling department number of project number 10 to 'Bellaire' and 5, respectively.

```
U5:   UPDATE    PROJECT
    SET       PLOCATION = 'Bellaire', DNUM = 5
    WHEREPNUMBER=10;
```

- Example: Give all employees in the 'Research' department a 10% raise in salary.

```
U6:  UPDATE  EMPLOYEE
  SET     SALARY = SALARY *0.1
  WHEREDNO  IN (  SELECT   DNUMBER
          FROM    DEPARTMENT
          WHEREDNAME='Research');
```

- The modified SALARY value depends on the original SALARY value in each tuple.
- The reference to the SALARY attribute on the right of = refers to the old SALARY value before modification
- The reference to the SALARY attribute on the left of = refers to the new SALARY value after modification

End of Chapter-4