HOA SEN UNIVERSITY

Lecture 3
SQL 2 – DML & Select Basic

Objectives

- INSERT, UPDATE and DELETE statements
- Assertions and Triggers concept
- Select Basic
- Comparison operators
- Between … And, In
- Like
- Not – And – Or
- More than one table query

- Ref.: Chapter 6

Specifying Updates in SQL

- There are three SQL commands to modify the database:
  - **INSERT**
  - **DELETE**
  - **UPDATE**

INSERT

- In its simplest form, it is used to add one or more tuples to a relation

- Syntax:

```
Insert Into Table(F1, F2, …, Fm)
     Values (V1, V2, …, Vm);
GO
```

- Attribute values should be listed in the same order as the attributes were specified in the **CREATE TABLE** command

INSERT (2)

- Example:

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

```
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.

```
INSERT INTO EMPLOYEE (Fname, Lname, Ssn)
      VALUES ('Richard', 'Marini','653298653');
```

INSERT (3)

- 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

INSERT (4)

- Example: Suppose we want to create a temporary table that has the name, number of employees, and total salaries for each department.

```
CREATE TABLE  DEPTS_INFO
            (Dept_name VARCHAR(10),
             No_of_emps     INTEGER,
             Total_sal INTEGER);
GO


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 ;
GO
```

Note: The DEPTS_INFO table may ***not be up-to-date*** *if we change the tuples in either the DEPARTMENT or the EMPLOYEE relations*

DELETE

- Syntax:

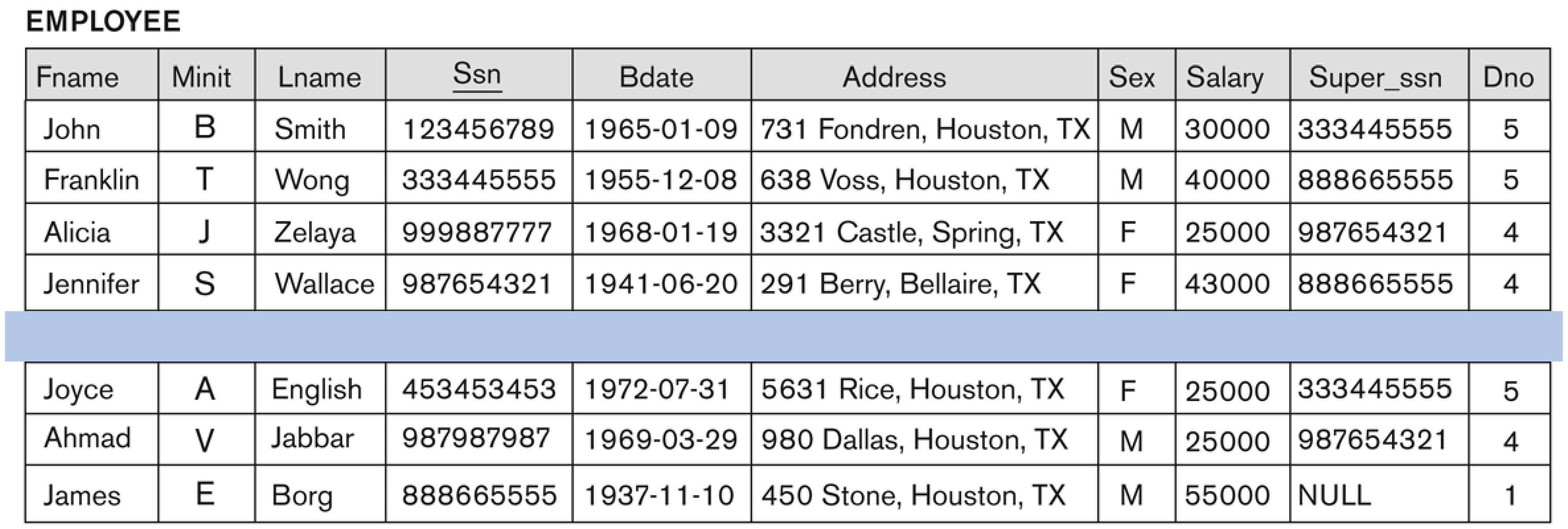
```
Delete From Table
        [Where <cond.>];
GO
```

- Removes tuples from a relation/table
  - Includes a WHERE-clause to select the tuples to be deleted
  - Referential integrity should be enforced
  - 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

DELETE (2)

- Examples:

```
DELETE FROM EMPLOYEE
   WHERE Lname='Narayan';
GO
```



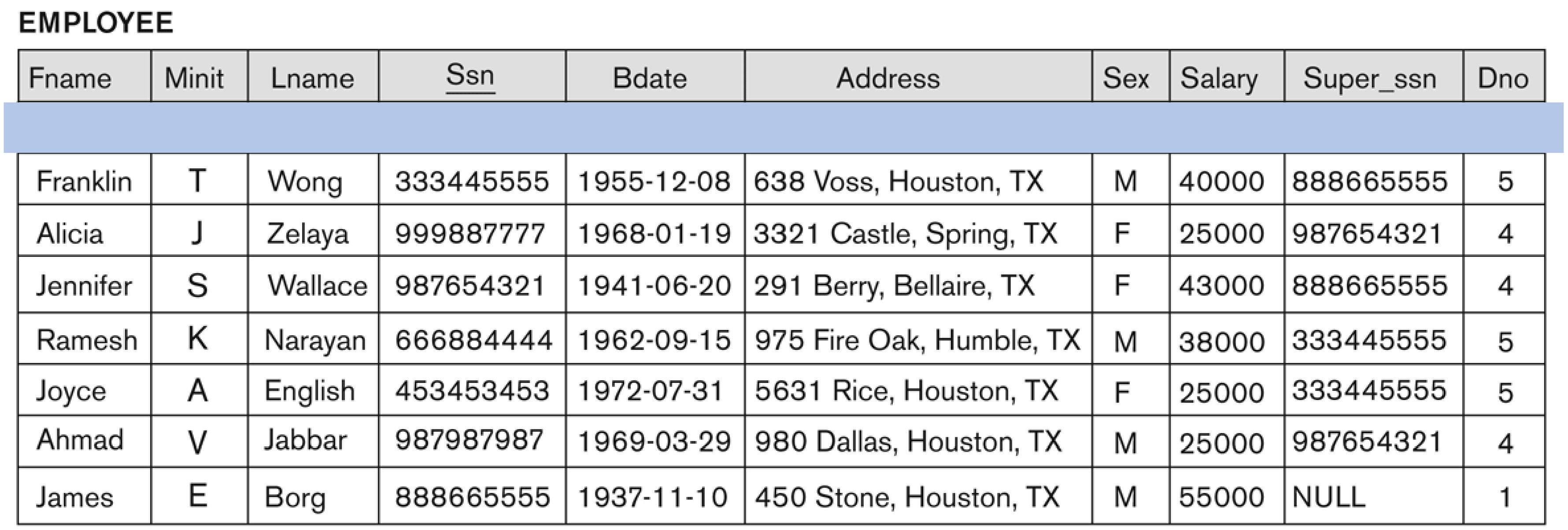
EMPLOYEE

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
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

DELETE (2)

- Examples:

```
DELETE FROM EMPLOYEE
   WHERE Ssn='123456789';
GO
```



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
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

DELETE (2)

- Examples:

```
DELETE FROM EMPLOYEE
   WHERE DNO IN
         (SELECT Dnumber
         FROM DEPARTMENT
         WHERE Dname='Research');
GO
```

DEPARTMENT

Dname	Dnumber
Research	5
Administration	4
Headquarters	1

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
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
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

DELETE (2)

- Examples:

```
DELETE FROM EMPLOYEE;
GO
```

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

UPDATE

- Syntax:

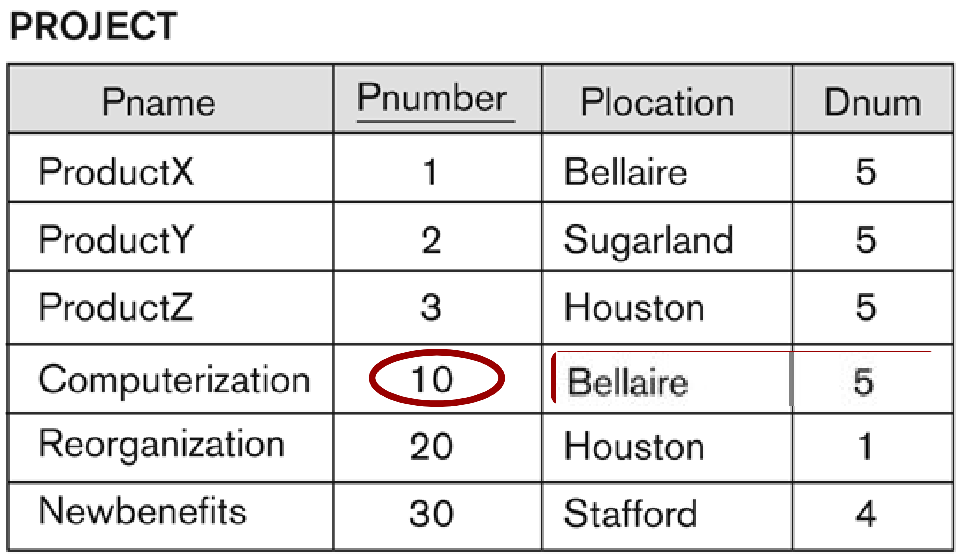
```
Update Table
    Set Field = <Value>
    [Where <cond.>];
GO
```

- 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

UPDATE (2)

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

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



PROJECT

Pname	Pnumber	Plocation	Dnum
ProductX	1	Bellaire	5
ProductY	2	Sugarland	5
ProductZ	3	Houston	5
Computerization	10	Bellaire	5
Reorganization	20	Houston	1
Newbenefits	30	Stafford	4

UPDATE (3)

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

```
UPDATE EMPLOYEE
   SET   Salary = Salary * 1.1
   WHERE Dno IN (SELECT Dnumber
                 FROM   DEPARTMENT
                 WHERE Dname='Research');
GO
```

- In this request, 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

Constraints as Assertions

- General constraints: constraints that do not fit in the basic SQL categories
- Defines a new rule that will constrain the set of valid values for one or more Base tables
- Mechanism: **`CREAT ASSERTION`**
  - Components include:
    - a constraint name,
    - followed by `CHECK`,
    - followed by a condition

Assertions: An Example

- “The salary of an employee must not be greater than the salary of the manager of the department that the employee works for”

```
CREAT ASSERTION SALARY_CONSTRAINT
    CHECK (NOT EXISTS (SELECT *
         FROM EMPLOYEE E, DEPARTMENT D, EMPLOYEE M,
         WHERE E.Dno=D.Number AND
              D.Mgr_ssn=M.Ssn AND
              E.Salary > M.salary));
```

Using General Assertions

- Specify a query that violates the condition; include inside a `NOT EXISTS` clause
- Query result must be empty
  - if the query result is not empty, the assertion has been violated

SQL Triggers

- Objective: to monitor a database and take initiate action when a condition occurs
- Triggers are expressed in a syntax similar to assertions and include the following:
  - Event
    - Such as an insert, deleted, or update operation
  - Condition
  - Action
    - To be taken when the condition is satisfied

SQL Triggers: An Example

- DML trigger with a reminder message:

```
CREATE TRIGGER reminder1
ON Sales.Customer
AFTER INSERT, UPDATE
AS RAISERROR ('Notify Customer Relations', 16, 10);
GO
```

- DML trigger with a reminder e-mail message

```
CREATE TRIGGER reminder2
ON Sales.Customer
AFTER INSERT, UPDATE, DELETE
AS
   EXEC msdb.dbo.sp_send_dbmail
        @profile_name = 'AdventureWorks2012 Administrator',
        @recipients = 'danw@Adventure-Works.com',
        @body = 'Don''t forget to print a report for the sales force.',
        @subject = 'Reminder';
GO
```

Retrieval Queries in SQL

- SQL has one basic statement for retrieving information from a database; the **SELECT** statement
  - This is *not the same as* the SELECT operation of the relational algebra
- Important distinction between SQL and the formal relational model:
  - SQL allows a table (relation) to have two 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

Retrieval Queries in SQL (2)

- A **bag** or **multi-set** is like a set, but an element may appear more than once.
  - Example:
    - {A, B, C, A} is a bag.
    - {A, B, C} is also a bag that also is a set.
  - Bags also resemble lists, but the order is irrelevant in a bag.
- Example:
  - {A, B, A} = {B, A, A} as bags
  - However, [A, B, A] is not equal to [B, A, A] as lists

Retrieval Queries in SQL (3)

- 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>;
```

  - <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

Retrieval Queries in SQL (4)

- Example of a simple query on one table

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno

List all Employee.

```
SELECT   *
  FROM   EMPLOYEE;
```

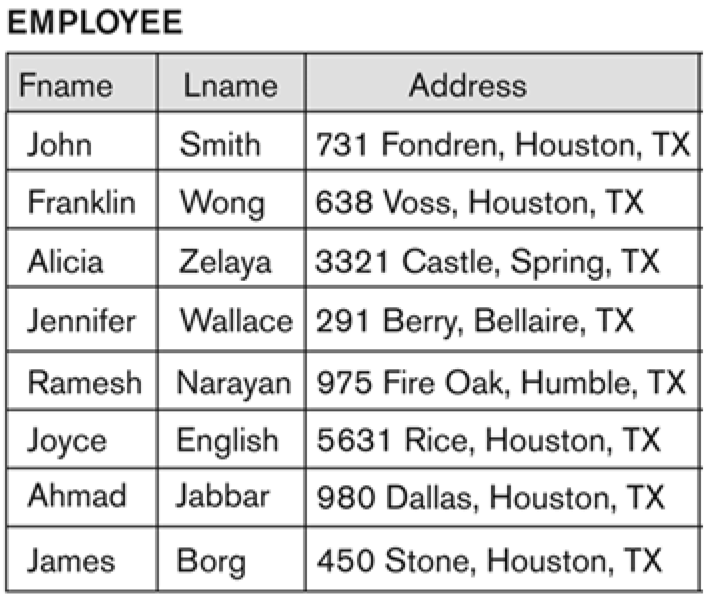
EMPLOYEE

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

Retrieval Queries in SQL (5)

- Retrieve the First name, Last name and address of the employees

```
Select Fname, Lname, Address
   From Employee;
```



EMPLOYEE

Fname	Lname	Address
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, Houston, TX
Alicia	Zelaya	3321 Castle, Spring, TX
Jennifer	Wallace	291 Berry, Bellaire, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX
Joyce	English	5631 Rice, Houston, TX
Ahmad	Jabbar	980 Dallas, Houston, TX
James	Borg	450 Stone, Houston, TX

ĐẠI HỌC HOA SEN

Comparison operators

- **Comparison operators**: =,!=,<>,<,<=,>,>=

Retrieve the birthday and address of the employee whose first name is 'John‘ (string).

```
SELECT Bdate, Address
 FROM EMPLOYEE
 WHERE Fname='John‘;
```

EMPLOYEE

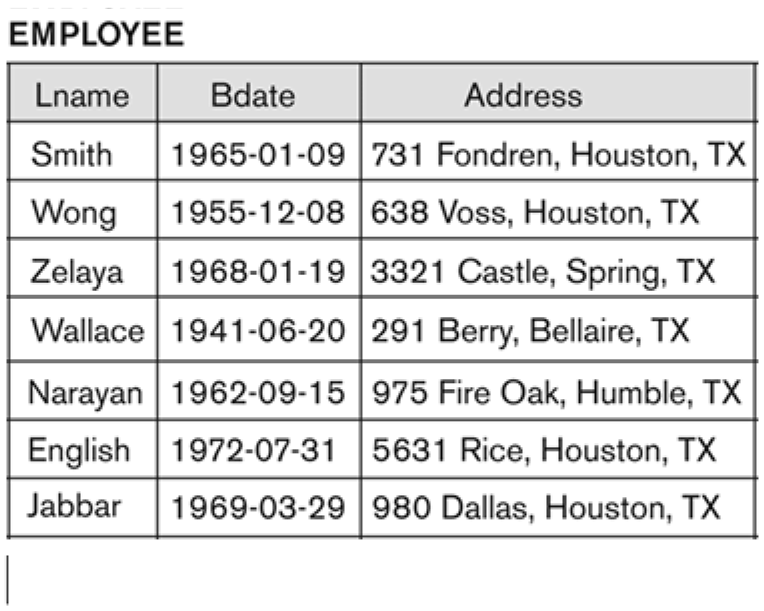
Fname
John

Bdate	Address
1965-01-09	731 Fondren, Houston, TX

Comparison operators

- Retrieve the last name, birth date and address of the employees whose last name is not ‘Borg‘ (string).

```
SELECT Lname, Bdate, Address
  FROM EMPLOYEE
  WHERE Lname <>‘Borg’;
```



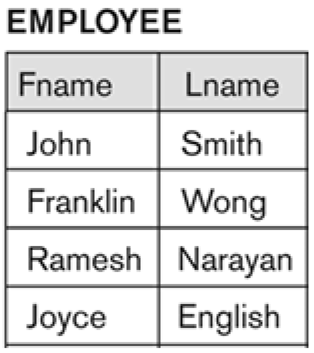
EMPLOYEE

Lname	Bdate	Address
Smith	1965-01-09	731 Fondren, Houston, TX
Wong	1955-12-08	638 Voss, Houston, TX
Zelaya	1968-01-19	3321 Castle, Spring, TX
Wallace	1941-06-20	291 Berry, Bellaire, TX
Narayan	1962-09-15	975 Fire Oak, Humble, TX
English	1972-07-31	5631 Rice, Houston, TX
Jabbar	1969-03-29	980 Dallas, Houston, TX

Comparison operators

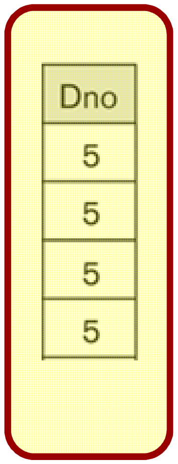
- Retrieve the first name and last name of the employee whose DNo is 5 (number).

```
SELECT Fname, Lname
   FROM EMPLOYEE
   WHERE DNo = 5;
```



EMPLOYEE

Fname	Lname
John	Smith
Franklin	Wong
Ramesh	Narayan
Joyce	English

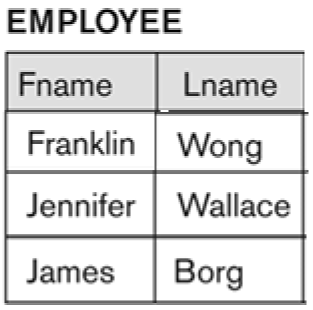


Dno
5
5
5
5

Comparison operators

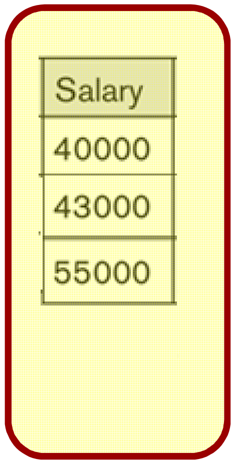
- Retrieve the first name and last name of the employee whose salary is greater or equal 40000 (number).

```
SELECT Fname, Lname
  FROM EMPLOYEE
  WHERE Salary >= 40000;
```



EMPLOYEE

Fname	Lname
Franklin	Wong
Jennifer	Wallace
James	Borg

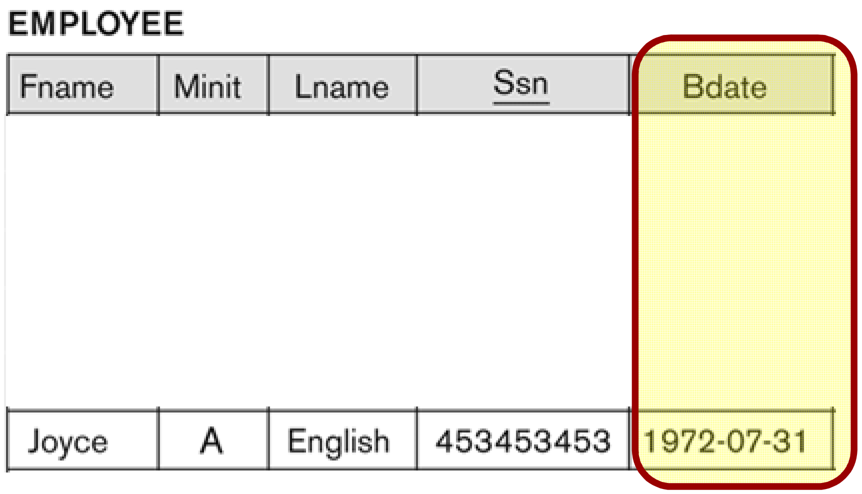


Salary
40000
43000
55000

Comparison operators

- Retrieve the name, SSN and address of the employee whose birthdates ‘Jul-31-1972’ (date).

```
SELECT Lname, Minit, LName, SSn
  FROM EMPLOYEE
  WHERE Bdate = ‘1972-07-31’;
```



EMPLOYEE

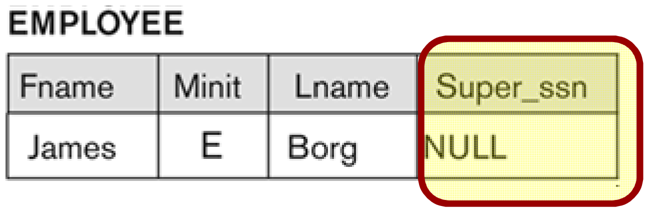
Fname	Minit	Lname	Ssn	Bdate
Joyce	A	English	453453453	1972-07-31

Null

- Retrieve the first name and last name of the employees who does not have his supper.

```
SELECT  Fname, Minit, Lname
   FROM  EMPLOYEE
   WHERE Supper_SSN is null;
```

- *Note that NULL indicates a value which is missing, not known, inappropriate, etc. NULL is not a blank or zero. NULL cannot be tested for equality with other NULL values.*



EMPLOYEE

Fname	Minit	Lname	Super_ssn
James	E	Borg	NULL

String Operations

- Pattern matching
- Simple pattern matching is carried out using LIKE:

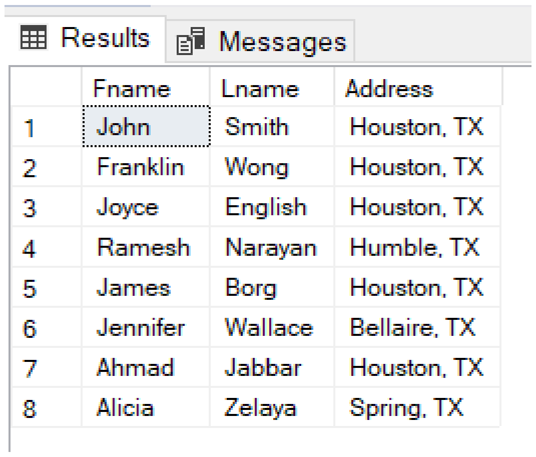
```
LIKE ’pattern-to-match’
```

  - Where the pattern can include special wildcard characters:
    - % (percent) 0 or more arbitrary characters
    - _ (underscore) any one character

String Operations

- Retrieve all employees whose address is in Houston, Texas. Here, the value of the ADDRESS attribute must contain the substring 'Houston,TX‘ in it.

```
SELECT Fname, Lname
    FROM EMPLOYEE
    WHERE Address LIKE ‘Houston,TX%‘;
```

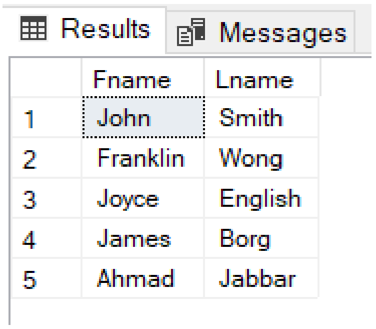


Results | Messages

	Fname	Lname	Address
1	John	Smith	Houston, TX
2	Franklin	Wong	Houston, TX
3	Joyce	English	Houston, TX
4	Ramesh	Narayan	Humble, TX
5	James	Borg	Houston, TX
6	Jennifer	Wallace	Bellaire, TX
7	Ahmad	Jabbar	Houston, TX
8	Alicia	Zelaya	Spring, TX

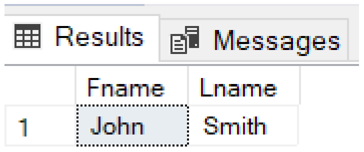
- Find the names of all employees whose first name starts with ‘J’ character and are at least 4 characters in length.

```
Select FName, Lname
    From employee
    Where FName like 'J___';
```



Results | Messages

	Fname	Lname
1	John	Smith
2	Franklin	Wong
3	Joyce	English
4	James	Borg
5	Ahmad	Jabbar



Results | Messages

	Fname	Lname
1	John	Smith

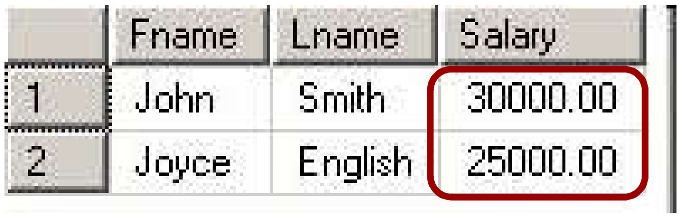
ĐẠI HỌC HOA SEN

Arithmetic Operations

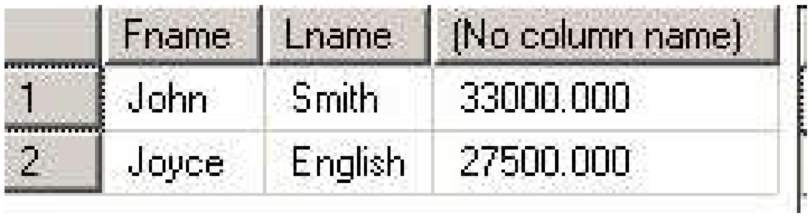
- The standard arithmetic operators **'+', '-'. '*', and '/'** (for addition, subtraction, multiplication, and division, respectively) can be applied to numeric values in an SQL query result

- Show the effect of giving all employees who work on the 'ProductX' project a 10% raise.

```
SELECT Fname, Lname, 1.1*Salary
   FROM EMPLOYEE, WORKS_ON, PROJECT
   WHERE Ssn=Essn AND Pno=Pnumber
         AND Pname='ProductX';
```



	Fname	Lname	Salary
1	John	Smith	30000.00
2	Joyce	English	25000.00



	Fname	Lname	(No column name)
1	John	Smith	33000.000
2	Joyce	English	27500.000

Distinct – Order clause

```
SELECT [DISTINCT] column_list
 FROM table_list
    [WHERE condition]
    [ORDER BY attribute[DESC/ASC]
            [,attribute [DESC,ASC]]...];
```

Distinct

- 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

Salary
30000
40000
25000
43000
38000
25000
25000
55000

```
SELECT Salary
    FROM EMPLOYEE;
```

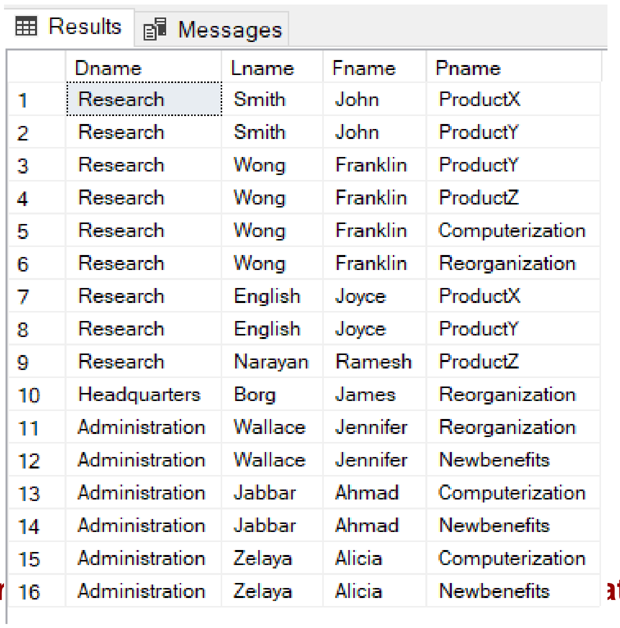
```
SELECT DISTINCT Salary
    FROM EMPLOYEE;
```

Salary
30000
40000
25000
43000
38000
55000

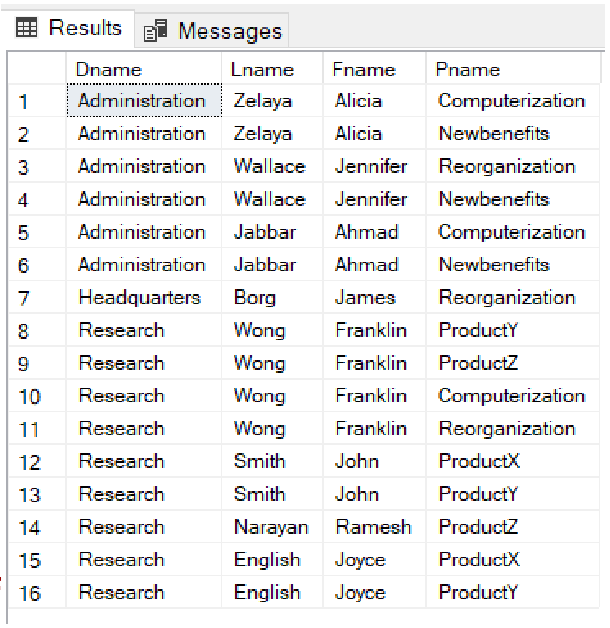
Order By

- We can 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**
- Example:

```
SELECT    Dname, Lname, Fname, Pname
     FROM  DEPARTMENT, EMPLOYEE,
     WORKS_ON, PROJECT
     WHERE Dnumber=Dno AND Ssn=Essn AND Pno=Pnumber
     ORDER BY   Dname, Lname DESC;
```



	Dname	Lname	Fname	Pname
1	Research	Smith	John	ProductX
2	Research	Smith	John	ProductY
3	Research	Wong	Franklin	ProductY
4	Research	Wong	Franklin	ProductZ
5	Research	Wong	Franklin	Computerization
6	Research	Wong	Franklin	Reorganization
7	Research	English	Joyce	ProductX
8	Research	English	Joyce	ProductY
9	Research	Narayan	Ramesh	ProductZ
10	Headquarters	Borg	James	Reorganization
11	Administration	Wallace	Jennifer	Reorganization
12	Administration	Wallace	Jennifer	Newbenefits
13	Administration	Jabbar	Ahmad	Computerization
14	Administration	Jabbar	Ahmad	Newbenefits
15	Administration	Zelaya	Alicia	Computerization
16	Administration	Zelaya	Alicia	Newbenefits

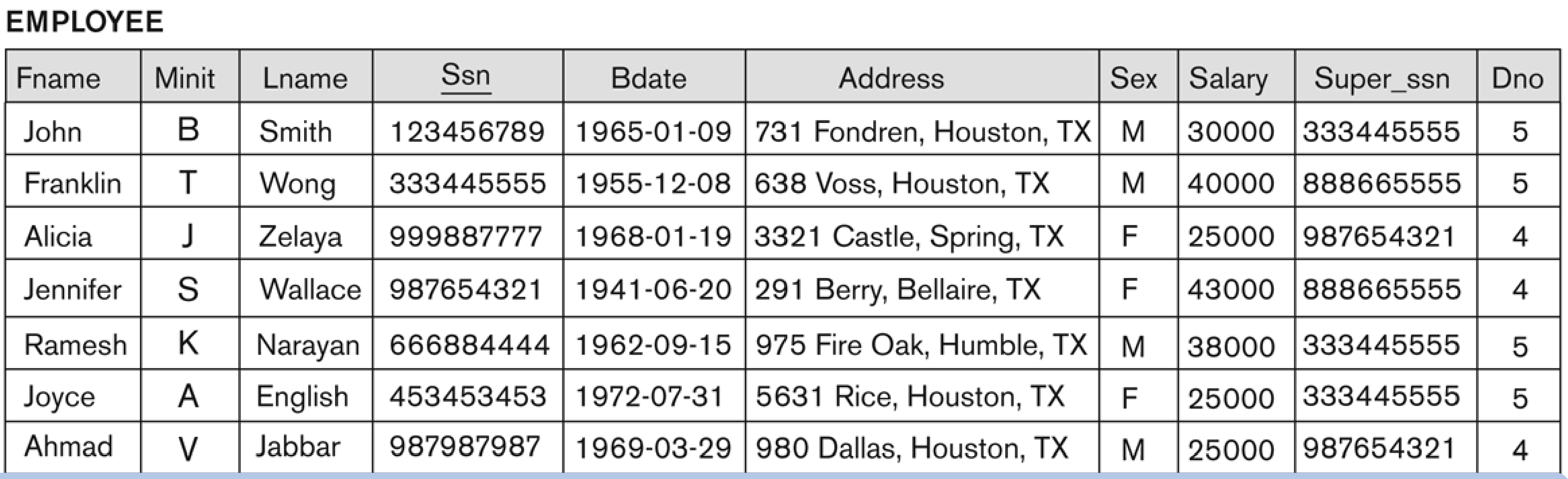


	Dname	Lname	Fname	Pname
1	Administration	Zelaya	Alicia	Computerization
2	Administration	Zelaya	Alicia	Newbenefits
3	Administration	Wallace	Jennifer	Reorganization
4	Administration	Wallace	Jennifer	Newbenefits
5	Administration	Jabbar	Ahmad	Computerization
6	Administration	Jabbar	Ahmad	Newbenefits
7	Headquarters	Borg	James	Reorganization
8	Research	Wong	Franklin	ProductY
9	Research	Wong	Franklin	ProductZ
10	Research	Wong	Franklin	Computerization
11	Research	Wong	Franklin	Reorganization
12	Research	Smith	John	ProductX
13	Research	Smith	John	ProductY
14	Research	Narayan	Ramesh	ProductZ
15	Research	English	Joyce	ProductX
16	Research	English	Joyce	ProductY

Operator: Not – And – Or

- **Not**

```
Select *
     From Employee
     Where Supper_SSn is Not Null;
```



EMPLOYEE

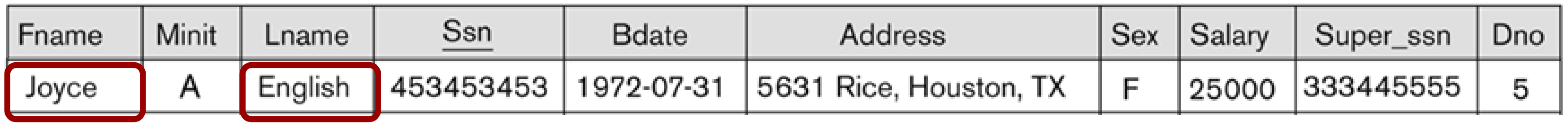
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	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

Operator: Not – And – Or

- **And**

```
Select *
     From Employee
     Where Fname = ‘Joyce’ And Lname = ‘English’;
```

EMPLOYEE



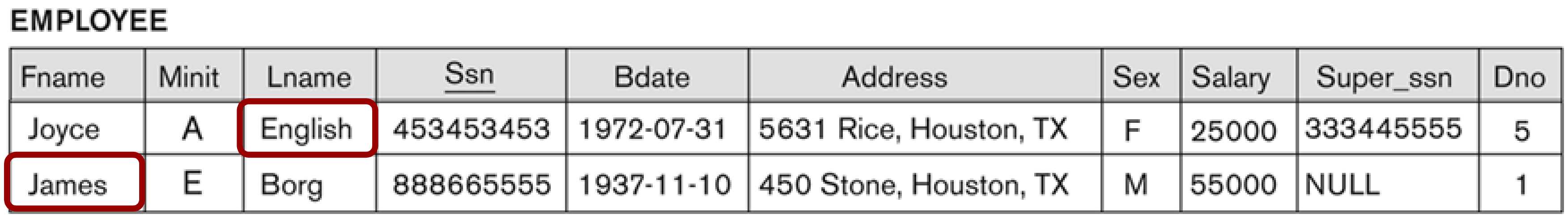
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5

ĐẠI HỌC HOA SEN

Operator: Not – And – Or

- **Or**

```
Select *
    From Employee
    Where Lname = ‘English’ Or Fname = ‘James’;
```



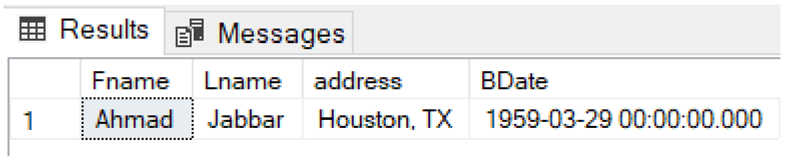
EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

Operator: Between … And …

- **Between … And …**
- Retrieve the first name, last name and address of the employee whose birthdates from June-01-1959 to Dec-31-1959.

```
SELECT   Fname, Lname, address
     FROM  EMPLOYEE
     WHERE Bdate Between ‘1959-01-01’ And ‘1959-06-31’;
```



	Fname	Lname	address	BDate
1	Ahmad	Jabbar	Houston, TX	1959-03-29 00:00:00.000

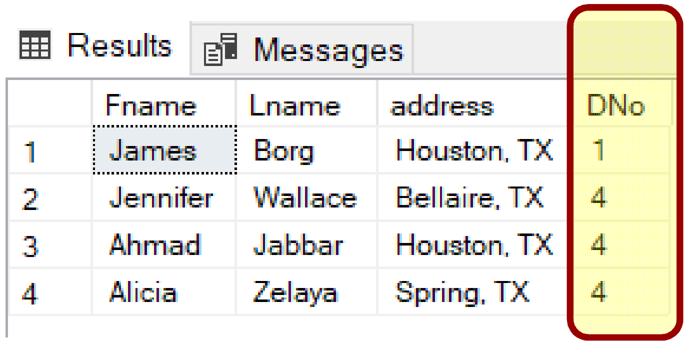
- Note that the BETWEEN predicate is inclusive. The above condition is equivalent to :

*WHERE Bdate >= ‘1959-01-01’ **And** Bdate <= ‘1959-06-31’;*

Operator: In

- **In**
- Retrieve the first name, last name and address of the employee whose Dno is 1 or 4.

```
SELECT Fname, Lname, address
    FROM EMPLOYEE
    WHERE Dno In (1,4);
```



	Fname	Lname	address	DNo
1	James	Borg	Houston, TX	1
2	Jennifer	Wallace	Bellaire, TX	4
3	Ahmad	Jabbar	Houston, TX	4
4	Alicia	Zelaya	Spring, TX	4

- The above condition is equivalent to :

```
WHERE Dno = 1 Or Dno = 4;
```

Display Data from multiple tables

- Obtaining Data from Multiple Tables

EMPLOYEE

Fname	Minit	Lname	Ssn	...	Dno
John	B	Smith	123456789	...	5
Franklin	T	Wong	333445555	...	5
Alicia	J	Zelaya	999887777	...	4
Jennifer	S	Wallace	987654321	...	4
Ramesh	K	Narayan	666884444	...	5
Joyce	A	English	453453453	...	5
Ahmad	V	Jabbar	987987987	...	4
James	E	Borg	888665555	...	1

DEPARTMENT

Dnumber	Dname
5	Research
4	Administration
1	Headquarters

?

Fname	LName	SSN	Dno	Dnumber	Dname
Join	Smith	123456789	5	5	Research
…	…	…	…	…	…
James	Borg	888665555	1	1	Headquarters

- Syntax

```
SELECT  table1.column, table2.column
  FROM   table1, table2
  WHERE  table1.column1 = table2.column2;
```

- *Use a join to query data from more than one table.*
- *Write the join condition in the WHERE clause.*
- *Prefix the column name with the table name when the same column name appears in more than one table.*

- Ex:

```
Select Fname, Lname, Dname
  From Employee, Department
  Where Employee.Dno = Department.Dnumber;
```

Aliases

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

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

```
SELECT  E.Fname, E.Lname, S.Fname, S.Lname
    FROM     EMPLOYEE as E, EMPLOYEE as S
    WHERE    E.Super_ssn = S.Ssn;
```

  - 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 two different *copies* of EMPLOYEE; E represents employees in role of *supervisees* and S represents employees in role of *supervisors*

Aliases (2)

- Aliasing can also be used in any SQL query for convenience
- Can also use the AS keyword to specify aliases

```
SELECT  E.Fname, E.Lname, S.Fname, S.Lname
   FROM EMPLOYEE AS E, EMPLOYEE AS S
   WHERE E.Super_ssn=S.Ssn;
```

EMPLOYEE E

Fname	Minit	Lname	Ssn	Super_ssn
John	B	Smith	123456789	333445555
Franklin	T	Wong	333445555	888665555
Alicia	J	Zelaya	999887777	987654321
Jennifer	S	Wallace	987654321	888665555
Ramesh	K	Narayan	666884444	333445555
Joyce	A	English	453453453	333445555
Ahmad	V	Jabbar	987987987	987654321
James	E	Borg	888665555	NULL

EMPLOYEE S

Ssn	Fname	Minit	Lname
123456789	John	B	Smith
333445555	Franklin	T	Wong
999887777	Alicia	J	Zelaya
987654321	Jennifer	S	Wallace
666884444	Ramesh	K	Narayan
453453453	Joyce	A	English
987987987	Ahmad	V	Jabbar
888665555	James	E	Borg

Joining More than Two Tables

EMPLOYEE

Fname	Minit	Lname	Ssn	Dno
John	B	Smith	123456789	5
Franklin	T	Wong	333445555	5
Alicia	J	Zelaya	999887777	4
Jennifer	S	Wallace	987654321	4
Ramesh	K	Narayan	666884444	5
Joyce	A	English	453453453	5
Ahmad	V	Jabbar	987987987	4
James	E	Borg	888665555	1

DEPARTMENT

Dname	Dnumber
Research	5
Administration	4
Headquarters	1

DEPT_LOCATIONS

Dnumber	Dlocation
1	Houston
4	Stafford
5	Bellaire
5	Sugarland
5	Houston

Fname	LName	SSN	Dno	Dnumber	Dname	Dnumber	Dlocation
Join	Smith	123456789	5	5	Research	5	Bellaire
Join	Smith	123456789	5	5	Research	5	Sugarland
Join	Smith	123456789	5	5	Research	5	Houston
…	…	…	…	…	…	…	…
James	Borg	888665555	1	1	Headquarter	1	Houston

UNSPECIFIED WHERE-clause

- Example:

```
SELECT  Ssn, Dname
   FROM EMPLOYEE, DEPARTMENT;
```

- 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

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

How many rows?

ĐẠI HỌC HOA SEN

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

```
SELECT Pnumber, Dnum, Lname, Bdate, Address
    FROM PROJECT, DEPARTMENT, EMPLOYEE
    WHERE Dnum=Dnumber AND Mgr_ssn=Ssn
         AND Plocation='Stafford‘;
```

There are two join conditions

- The join condition Dnum=Dnumber relates a project to its controlling department
- The join condition Mgr_ssn=Ssn relates the controlling department to the employee who manages that department

ĐẠI HỌC HOA SEN

Q & A