

# SQL-The Relational Database Standard-II

政治大學

資訊科學系

沈錦坤

**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

**WORKS\_ON**

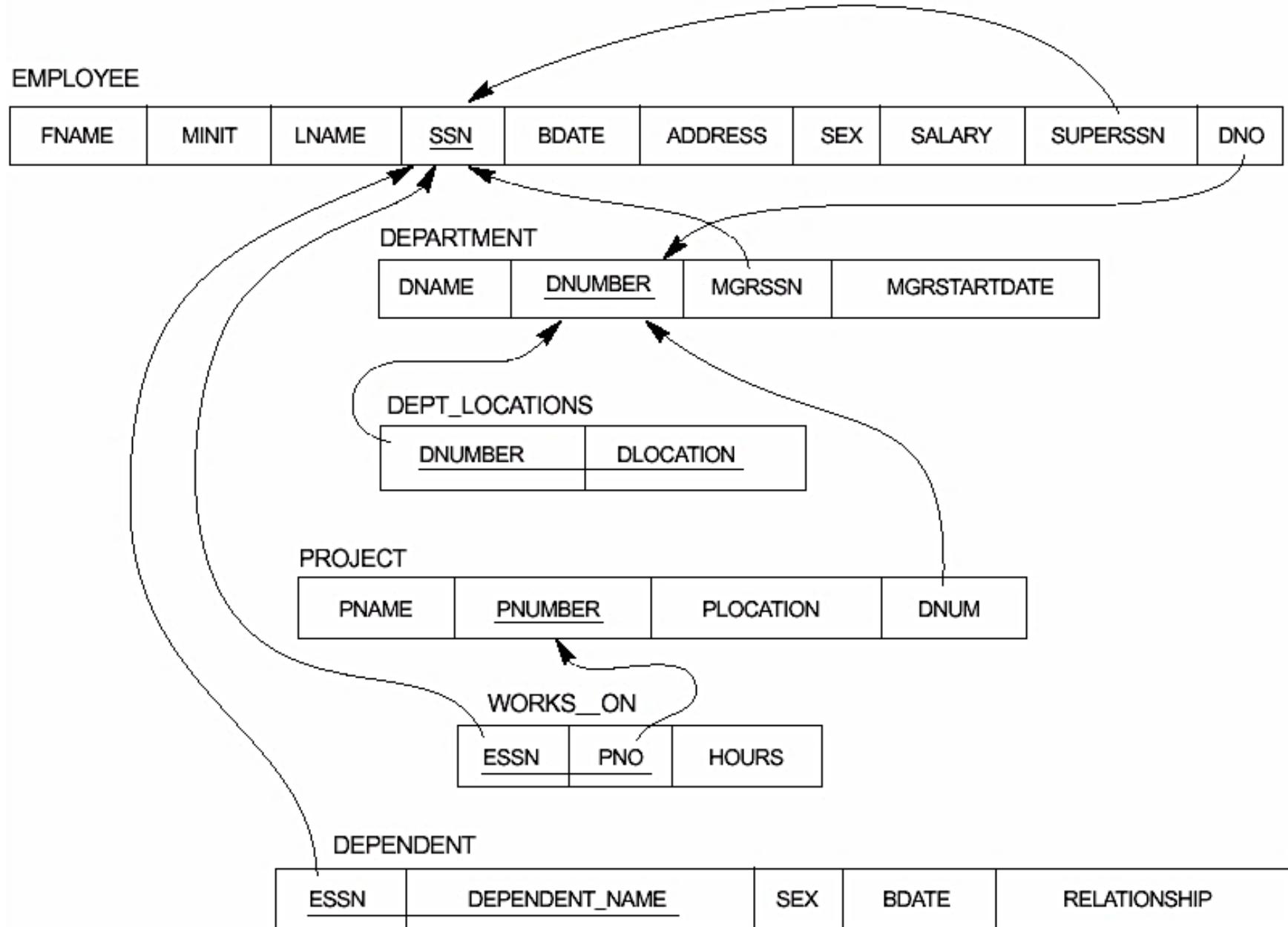
<u>Essn</u>	Pno	Hours
123456789	1	32.5
123456789	2	7.5
666884444	3	40.0
453453453	1	20.0
453453453	2	20.0
333445555	2	10.0
333445555	3	10.0
333445555	10	10.0
333445555	20	10.0
999887777	30	30.0
999887777	10	10.0
987987987	10	35.0
987987987	30	5.0
987654321	30	20.0
987654321	20	15.0
888665555	20	NULL

**PROJECT**

Pname	<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



**EMPLOYEE**

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

**DEPARTMENT**

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

**DEPT\_LOCATIONS**

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

**PROJECT**

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

**WORKS\_ON**

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

**DEPENDENT**

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

**Figure 3.5**  
Schema diagram for the COMPANY relational database schema.

# Nested Query

# Nested Queries

- ♦ Q4A: 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

**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

**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

**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

# Nested Queries (cont.)

- ◆ 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

**EMPLOYEE**

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	-----	-------	---------	-----	--------	-----------	-----

**DEPARTMENT**

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

**DEPT\_LOCATIONS**

Dnumber	Dlocation
---------	-----------

**PROJECT**

Pname	Pnumber	Plocation	Dnum
-------	---------	-----------	------

**WORKS\_ON**

Essn	Pno	Hours
------	-----	-------

**DEPENDENT**

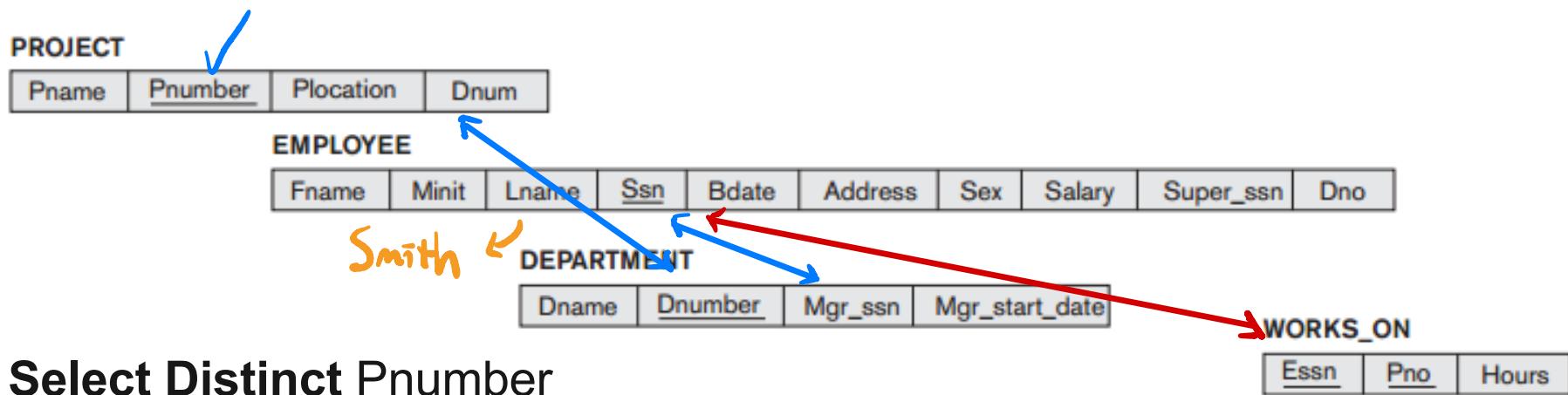
Essn	Dependent_name	Sex	Bdate	Relationship
------	----------------	-----	-------	--------------

**Figure 3.5**

Schema diagram for the COMPANY relational database schema.

# Nested Queries (cont.)

- ◆ 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 Pnumber

From Project

manages

Where Pnumber In (

Select Pnumber

From Project, Department, Employee

Where Dnum=Dnumber and MgrSSN=SSN and  
Lname='Smith')

OR

Pnumber In (

Select Pno

From Works\_On, Employee

Where ESSN=SSN and Lname='Smith');

subqueries

works

logical  
operator

```
Select Distinct Pnumber
From Project
Where Pnumber In (
    Select Pnumber
    From Project, Department, Employee
    Where Dnum=Dnumber and MgrSSN=SSN
        and Lname='Smith')

OR    Pnumber In (
    Select Pno
    From Works_On, Employee
    Where ESSN=SSN and Lname='Smith');
```

```
(Select Distinct PNumber
From Project, Department, Employee
Where Dnum = DNumber and
    MgrSSN = SSN and
    Lname = 'Smith')
```

Union

```
(Select Distinct Pnumber
From Project, Works_On, Employee
Where PNUmber = Pno and
    ESSN = SSN and
    Lname = 'Smith');
```

set operation

# All, In, Any Operation

- ◆ Select the social security numbers of all employee who work the same (project hours) combination **on some project** that employee ‘John Smith’ (whose SSN = ‘123456789’) works on

**EMPLOYEE**

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

**DEPARTMENT**

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

**DEPT\_LOCATIONS**

<u>Dnumber</u>	Dlocation
----------------	-----------

**PROJECT**

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

**WORKS\_ON**

<u>Essn</u>	Pno	Hours
-------------	-----	-------

**DEPENDENT**

<u>Essn</u>	Dependent_name	Sex	Bdate	Relationship
-------------	----------------	-----	-------	--------------

**Figure 3.5**

Schema diagram for the COMPANY relational database schema.

# All, In, Any Operation (cont.)

- ◆ Select the social security numbers of all employee who work the same (project hours) combination **on some project** that employee ‘John Smith’ (whose SSN = ‘123456789’) works on

**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

# All, In, Any Operation (cont.)

- ◆ Select the social security numbers of all employee who work the same (project hours) combination **on some project** that employee ‘John Smith’ (whose SSN = ‘123456789’) works on

WORKS_ON		
Essn	Pno	Hours

**Select Distinct ESSN**

**From** Works\_On

**Where** (PNo, Hours) **In**

(Select PNo, Hours

From Works\_On

Where SSN='123456789');

# All, In, Any Operation (cont.)

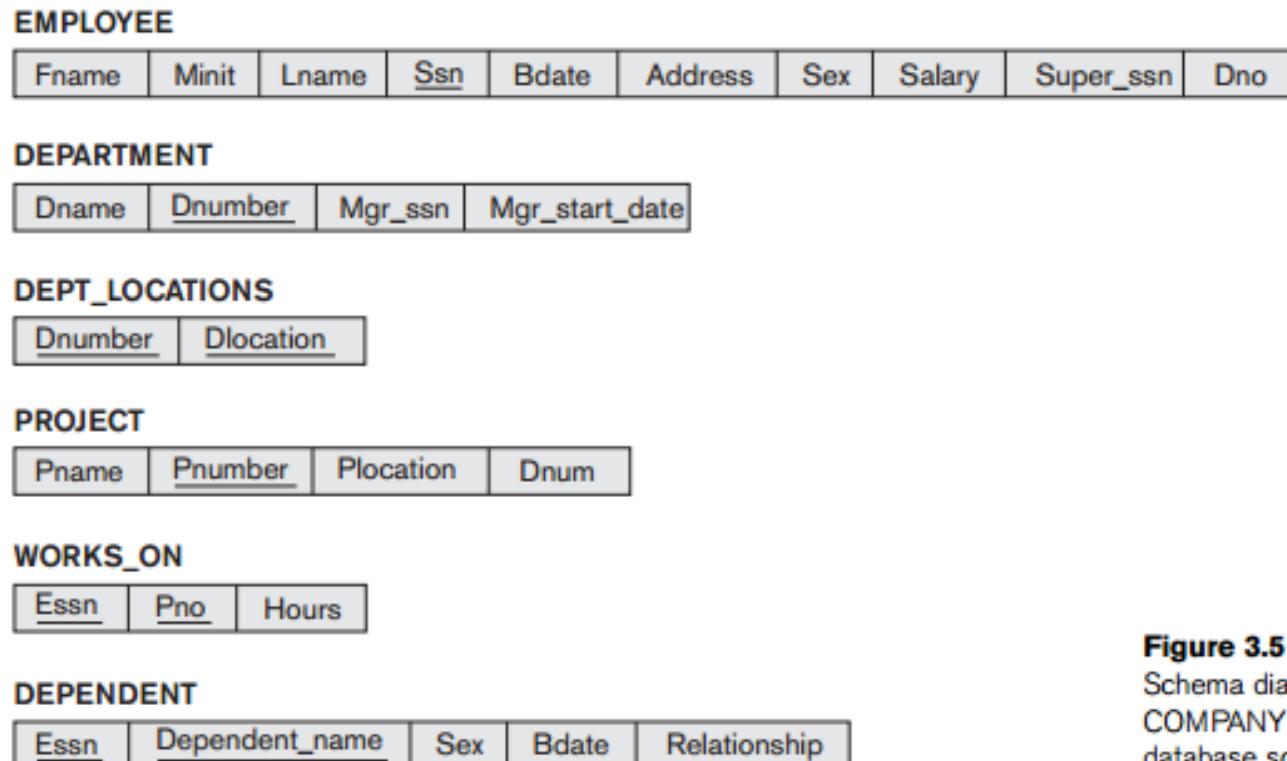
- ◆ List the names of employees whose salary is greater than the salary of all the employees in department 5

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

# All, In, Any Operation (cont.)

- ◆ List the names of employees whose salary is greater than the salary of all the employees in department 5



**Figure 3.5**  
Schema diagram for the COMPANY relational database schema.

# All, In, Any Operation (cont.)

- ◆ List the names of employees whose salary is greater than the salary of all the employees in department 5

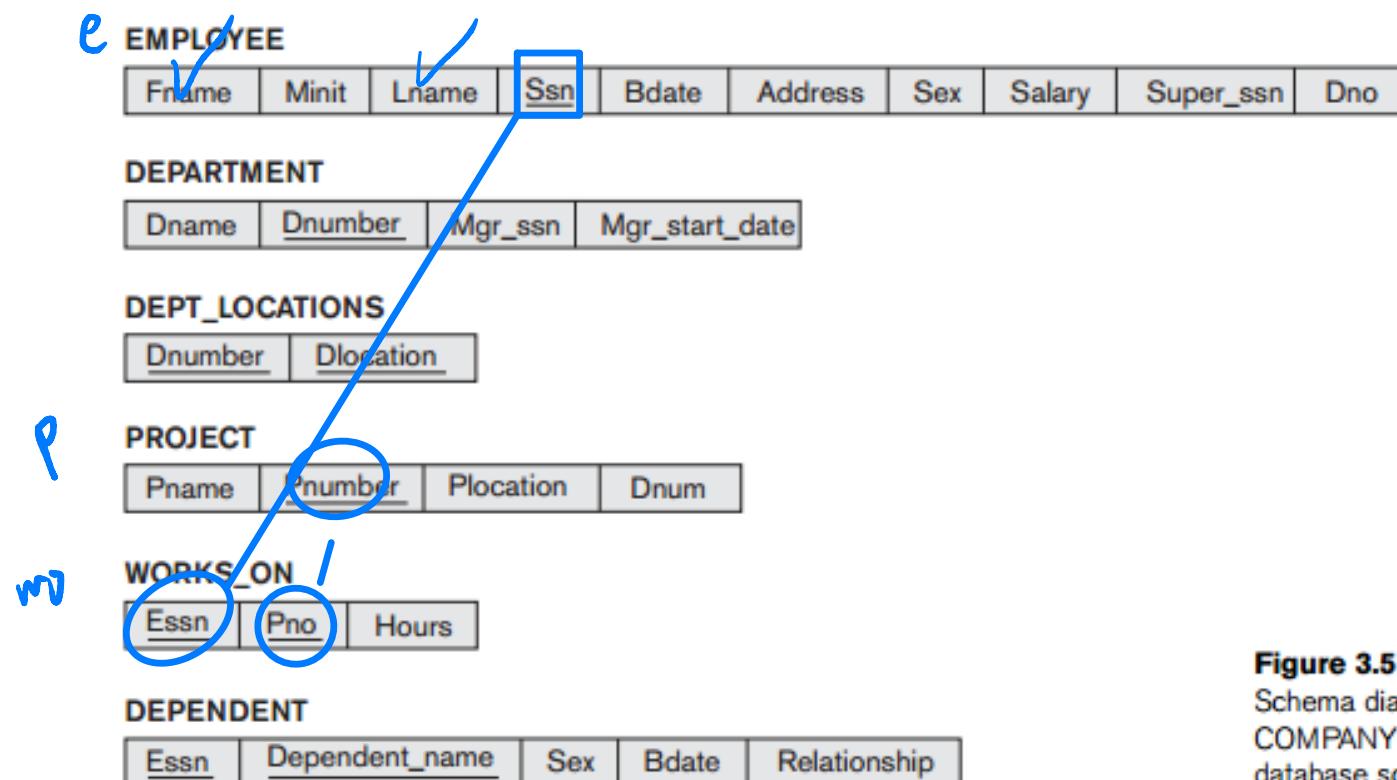
EMPLOYEE										
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno	

**Select Lname, Fname  
From Employee  
Where Salary > all (**  
**Select Salary  
    From Employee  
    Where Dno=5);**

# Correlated Nested Query

# Potential Ambiguity of Attribute Names in Nested Queries

- ♦ Q16: Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee



**Figure 3.5**  
Schema diagram for the COMPANY relational database schema.

# Potential Ambiguity of Attribute Names in Nested Queries (cont.)

- ♦ Q16: Retrieve the name of each employee who has a dependent with the same first name and same sex as the 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

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	John	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	John	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

**Select E.Fname, E.Lname  
From Employee As E  
Where E.SSN In (  
    Select ESSN  
    From Dependent  
    Where E.Fname = Dependent\_Name  
        and E.Sex=Sex);**

# Potential Ambiguity of Attribute Names in Nested Queries (cont.)

- ♦ Q16: Retrieve the name of each employee who has a dependent with the same first name and same sex as the 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

Select E.Fname, E.Lname

From Employee As E

Where E.SSN In (

Select ESSN

From Dependent as D

Where E.Fname = D.Dependent\_Name

and E.Sex = D.Sex);

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	John	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	John	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

# Correlated Nested Queries

♦ **Select E.Fname, E.Lname  
From Employee as E  
Where E.SSN in (  
    Select ESSN  
    From Dependent as D  
    Where E.Fname = D.Dependent\_Name  
        and E.Sex = D.Sex);**

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	John	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	John	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

♦ **Select E.Fname, E.Lname  
From Employee As E, Dependent As D  
Where E.SSN=D.ESSN and  
    E.Sex=D.Sex and  
    E.Fname=D.Dependent\_Name;**

# Correlated Nested Queries (cont.)

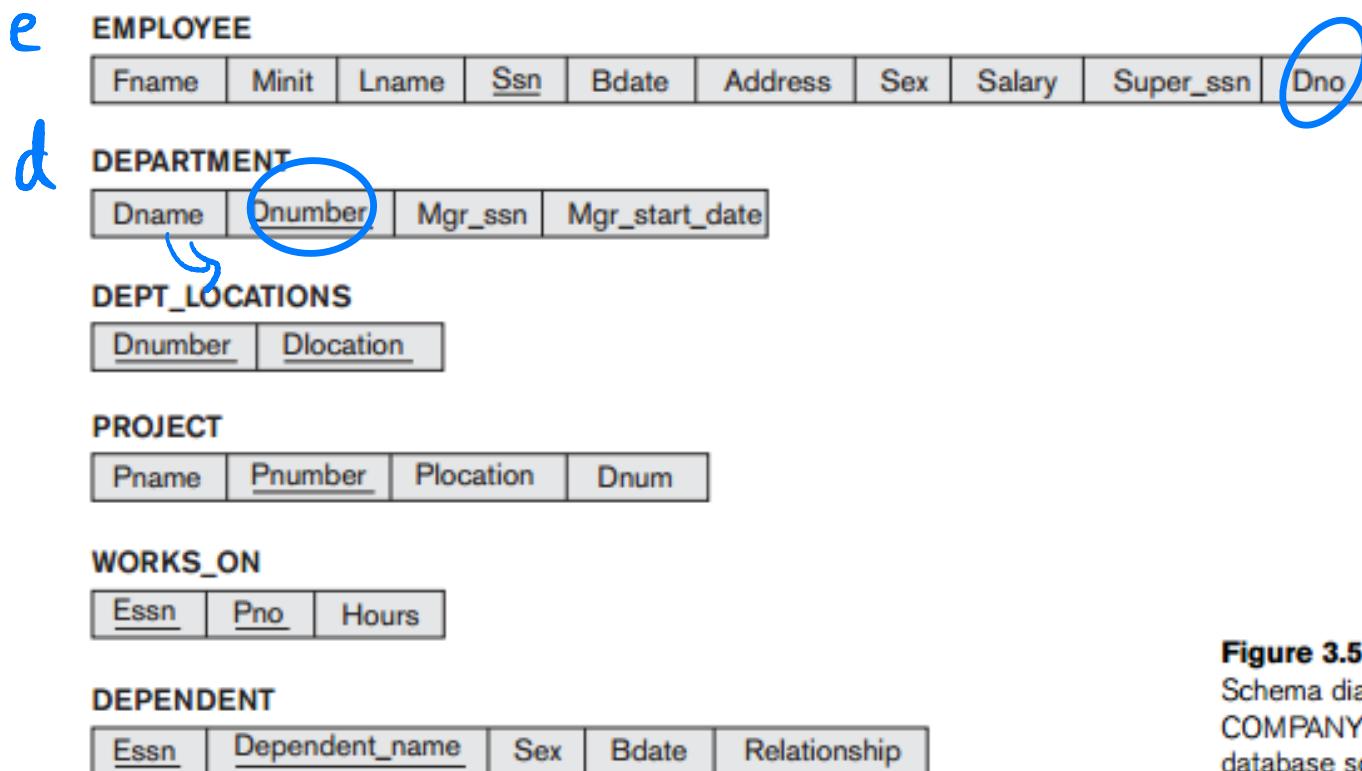
- ◆ Correlated nested query: whenever a condition in the WHERE clause of a nested inner query references some attribute of a relation declared in the outer query

```
Select E.Fname, E.Lname  
From Employee As E  
Where E.SSN in (  
    Select ESSN  
    From Dependent As D  
    Where E.Fname = D.Dependent_Name  
        and E.Sex = D.Sex);
```

- ◆ In general, a query written with nested select-from-where blocks and using the = or IN comparison operators can always be expressed as a single block query

# Exists Functions

- Q16B: Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee



**Figure 3.5**  
Schema diagram for the  
COMPANY relational  
database schema.

# Exists Functions (cont.)

- Q16B: Retrieve the name of each employee who has a dependent with the same first name and same sex as the employee

EMPLOYEE									
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
DEPENDENT									
Essn	Dependent_name	Sex	Bdate	Relationship					

**Select E.Fname, E.LName**

**From Employee As E**

**Where Exists (Select \***

**From Dependent**

**Where      E.SSN=ESSN and**

**E.Sex=Sex and**

**E.Fname=Dependent\_Name);**

- ◆ **Select E.Fname, E.Lname  
From Employee As E  
Where E.SSN in (**

**Select ESSN  
From Dependent as D  
Where E.Sex = D.Sex and  
E.Fname = D.Dependent\_Name);**

*equiv.*

EMPLOYEE

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

- ◆ **Select E.Fname, E.LName  
From Employee As E  
Where Exists (Select \***

**From Dependent  
Where E.SSN = ESSN and  
E.Sex = Sex and  
E.Fname = Dependent\_Name);**

- ◆ **Select E.Fname, E.LName  
From Employee As E  
Where Exists (Select \***

**From Dependent  
Where E.Sex = Sex and  
E.Fname = Dependent\_Name);**

DEPENDENT

Essn	Dependent_name	Sex	Bdate	Relationship
333445555	Alice	F	1986-04-05	Daughter
333445555	John	M	1983-10-25	Son
333445555	Joy	F	1958-05-03	Spouse
987654321	Abner	M	1942-02-28	Spouse
123456789	John	M	1988-01-04	Son
123456789	Alice	F	1988-12-30	Daughter
123456789	Elizabeth	F	1967-05-05	Spouse

*does not  
check SSN !*

# Exists Functions (cont.)

- ♦ Q6: Retrieve the names of employees who have no dependents

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

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

# Exists Functions (cont.)

- ♦ Q6: Retrieve the names of employees who have no dependents

**EMPLOYEE**

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	-----	-------	---------	-----	--------	-----------	-----

**DEPARTMENT**

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

**DEPT\_LOCATIONS**

Dnumber	Dlocation
---------	-----------

**PROJECT**

Pname	Pnumber	Plocation	Dnum
-------	---------	-----------	------

**WORKS\_ON**

Essn	Pno	Hours
------	-----	-------

**DEPENDENT**

Essn	Dependent_name	Sex	Bdate	Relationship
------	----------------	-----	-------	--------------

**Figure 3.5**

Schema diagram for the COMPANY relational database schema.

# Exists Functions (cont.)

- ♦ Q6: Retrieve the names of employees who have no dependents

EMPLOYEE									
Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
DEPENDENT									
Essn	Dependent_name	Sex	Bdate	Relationship					

**Select Fname, Lname**

**From Employee**

**Where Not Exists ( Select \***

**From Dependent**

**Where Employee.SSN=ESSN);**

# Exists Functions (cont.)

- ♦ Q7: List the names of managers who have at least one dependent

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

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

# Exists Functions (cont.)

- ♦ Q7: List the names of managers who have at least one dependent

**EMPLOYEE**

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

**DEPARTMENT**

Dname	<u>Dnumber</u>	Mgr_ssn	Mgr_start_date
-------	----------------	---------	----------------

**DEPT\_LOCATIONS**

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

**PROJECT**

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

**WORKS\_ON**

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

**DEPENDENT**

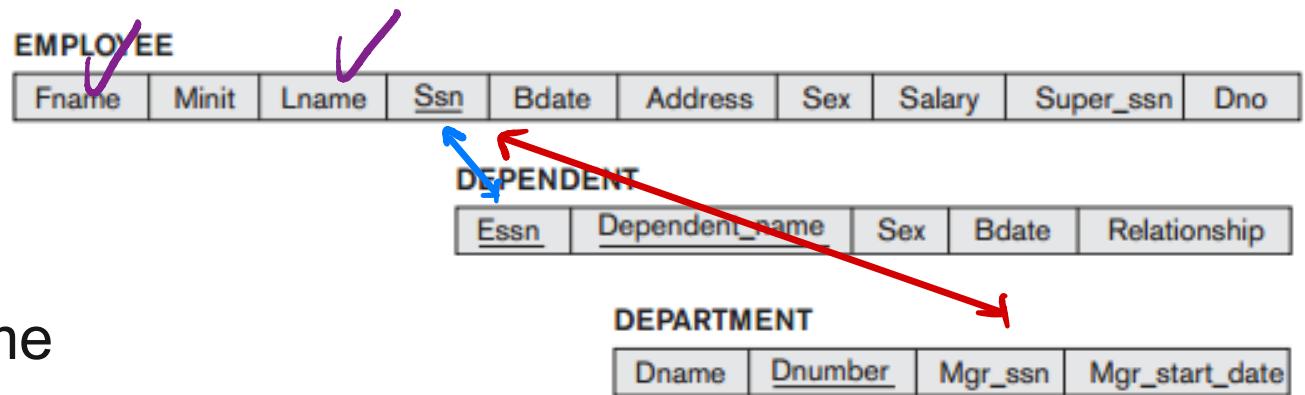
<u>Essn</u>	Dependent_name	Sex	Bdate	Relationship
-------------	----------------	-----	-------	--------------

**Figure 3.5**

Schema diagram for the COMPANY relational database schema.

# Exists Functions (cont.)

- ♦ Q7: List the names of managers who have at least one dependent



Select Fname, Lname

From Employee

Where Exists ( Select \*

From Dependent

Where SSN=ESSN)

has dependents?

And

Exists ( Select \*

From Department

Where SSN=MgrSSN);

is manager?

# Except Functions

- ♦ Retrieve the name of each employee who works on all the projects controlled by department number 5.

**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

**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

**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

# Set Operations

## Set Union

STUDENT	FN	LN
I	Susan	Yao
I	Ramesh	Shah
I	Johnny	Kohler
I	Barbara	Jones
I	Amy	Ford
I	Jimmy	Wang
I	Ernest	Gilbert

INSTRUCTOR	FNAME	LNAME
I	John	Smith
I	Ricardo	Browne
I	Susan	Yao
I	Francis	Johnson
I	Ramesh	Shah

	FN	LN
I	Susan	Yao
I	Ramesh	Shah
I	Johnny	Kohler
I	Barbara	Jones
I	Amy	Ford
I	Jimmy	Wang
I	Ernest	Gilbert
U	John	Smith
U	Ricardo	Browne
U	Francis	Johnson

# Set Intersection

STUDENT	FN	LN
Susan	Yao	
Ramesh	Shah	
Johnny	Kohler	
Barbara	Jones	
Amy	Ford	
Jimmy	Wang	
Ernest	Gilbert	

INSTRUCTOR	FNAME	LNAME
John	Smith	
Ricardo	Browne	
Susan	Yao	
Francis	Johnson	
Ramesh	Shah	

$$7 + 5 = 10 + 2$$

2

FN	LN
Susan	Yao
Ramesh	Shah

# Set Difference (Student-Instructor)

STUDENT	FN	LN
Susan	Yao	
Ramesh	Shah	
Johnny	Kohler	
Barbara	Jones	
Amy	Ford	
Jimmy	Wang	
Ernest	Gilbert	

INSTRUCTOR	FNAME	LNAME
John	Smith	
Ricardo	Browne	
Susan	Yao	
Francis	Johnson	
Ramesh	Shah	

FN	LN
Johnny	Kohler
Barbara	Jones
Amy	Ford
Jimmy	Wang
Ernest	Gilbert

5

# Set Difference (Instructor-Student)

STUDENT	FN	LN
Susan	Yao	
Ramesh	Shah	
Johnny	Kohler	
Barbara	Jones	
Amy	Ford	
Jimmy	Wang	
Ernest	Gilbert	

INSTRUCTOR	FNAME	LNAME
John	Smith	
Ricardo	Browne	
Susan	Yao	
Francis	Johnson	
Ramesh	Shah	

	FNAME	LNAME
3	John	Smith
	Ricardo	Browne
	Francis	Johnson

Set

Difference

# Except Functions

- ♦ Retrieve the name of each employee who works on all the projects controlled by department number 5.

**EMPLOYEE**

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

**DEPARTMENT**

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

**DEPT\_LOCATIONS**

Dnumber	Dlocation
---------	-----------

**PROJECT**

Pname	Pnumber	Plocation	Dnum
-------	---------	-----------	------

**WORKS\_ON**

Essn	Pno	Hours
------	-----	-------

**DEPENDENT**

Essn	Dependent_name	Sex	Bdate	Relationship
------	----------------	-----	-------	--------------

**Figure 3.5**

Schema diagram for the COMPANY relational database schema.

# Except Functions

- ♦ Retrieve the name of each employee who works on all the projects controlled by department number 5.

*(Solution 1) : Except*

Select Fname, Lname

From Employee

Where Not Exists

( ( Select PNumber  
From Project  
Where DNum=5)

controlled by  
department # 5

Except

( Select PNo  
From Works\_On  
Where Employee.SSN=ESSN) );

finds projects  
in department  
#5 that the  
employee does  
not work on

PROJECT

Pname	Pnumber	Plocation	Dnum
-------	---------	-----------	------

WORKS\_ON

Essn	Pno	Hours
------	-----	-------

# Q3B

- ♦ Retrieve the name of each employee who works on all the projects controlled by department number 5.

**EMPLOYEE**

Fname	Minit	Lname	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

**DEPARTMENT**

Dname	Dnumber	Mgr_ssn	Mgr_start_date
-------	---------	---------	----------------

**DEPT\_LOCATIONS**

<u>Dnumber</u>	<u>Dlocation</u>
----------------	------------------

**PROJECT**

Pname	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

**WORKS\_ON**

<u>Essn</u>	<u>Pno</u>	Hours
-------------	------------	-------

**DEPENDENT**

<u>Essn</u>	<u>Dependent_name</u>	Sex	Bdate	Relationship
-------------	-----------------------	-----	-------	--------------

**Figure 3.5**

Schema diagram for the COMPANY relational database schema.



- ♦ Retrieve the name of each employee who works on all the projects controlled by department number 5.  
( select each employee such that  
there doesn't exist a project controlled by dept. 5 that  
the employee doesn't work on)

<Solution 2> Double Negative : Not Exists

EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	-----	-------	---------	-----	--------	-----------	-----

WORKS\_ON

Essn	Pno	Hours
------	-----	-------

Select Lname, Fname  
From Employee  
Where Not Exists

( Select \*

From Works\_On B

Where (B.PNo in ( Select Pnumber  
From Project  
Where Dnum=5))

And

Not Exists (Select \*

From Works\_On C

Where C.ESSN=SSN AND C.Pno=B.Pno ));

finds projects  
in department  
#5 that the  
employee does  
not work on

the project is controlled  
by dept. 5

the employee works on all the projects  
controlled by dept. 5

correlated subquery

PROJECT

Pname	Pnumber	Plocation	Dnum
-------	---------	-----------	------

```
1 •  SELECT e.fname, e.lname
2   FROM employee e
3   WHERE NOT EXISTS (
4       -- projects controlled by dept.5 but the employee does not work on
5       SELECT *
6       FROM works_on b
7       WHERE b.pno IN ( -- projects controlled by dept.5
8           SELECT Pnumber
9           FROM PROJECT
10          WHERE Dnum = 5
11      ) AND NOT EXISTS ( -- exclude any projects controlled by dept.5 and the employee works on
12          SELECT *
13          FROM works_on c
14          -- correlated subquery
15          WHERE c.essn = e.ssn AND c.pno = b.pno
16      )
17 );
```

## <Solution 3> GROUP BY + HAVING

- ```
SELECT e.fname, e.lname
  FROM employee e
  JOIN works_on w ON e.ssn = w.essn
  JOIN project p ON w.pno = p.pnumber
 WHERE p.dnum = 5
 GROUP BY e.fname, e.lname
 HAVING COUNT(DISTINCT p.pnumber) = (SELECT COUNT(*) FROM project WHERE dnum = 5);
```

# Explicit Sets: Q17

- ♦ Retrieve the social security numbers of all employees who work on project 1, 2 or 3.

| 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  |

# Explicit Sets: Q17

- ♦ Retrieve the social security numbers of all employees who work on project 1, 2 or 3.

**Select Distinct ESSN**

**From Works\_On**

**Where PNo in (1, 2, 3)**

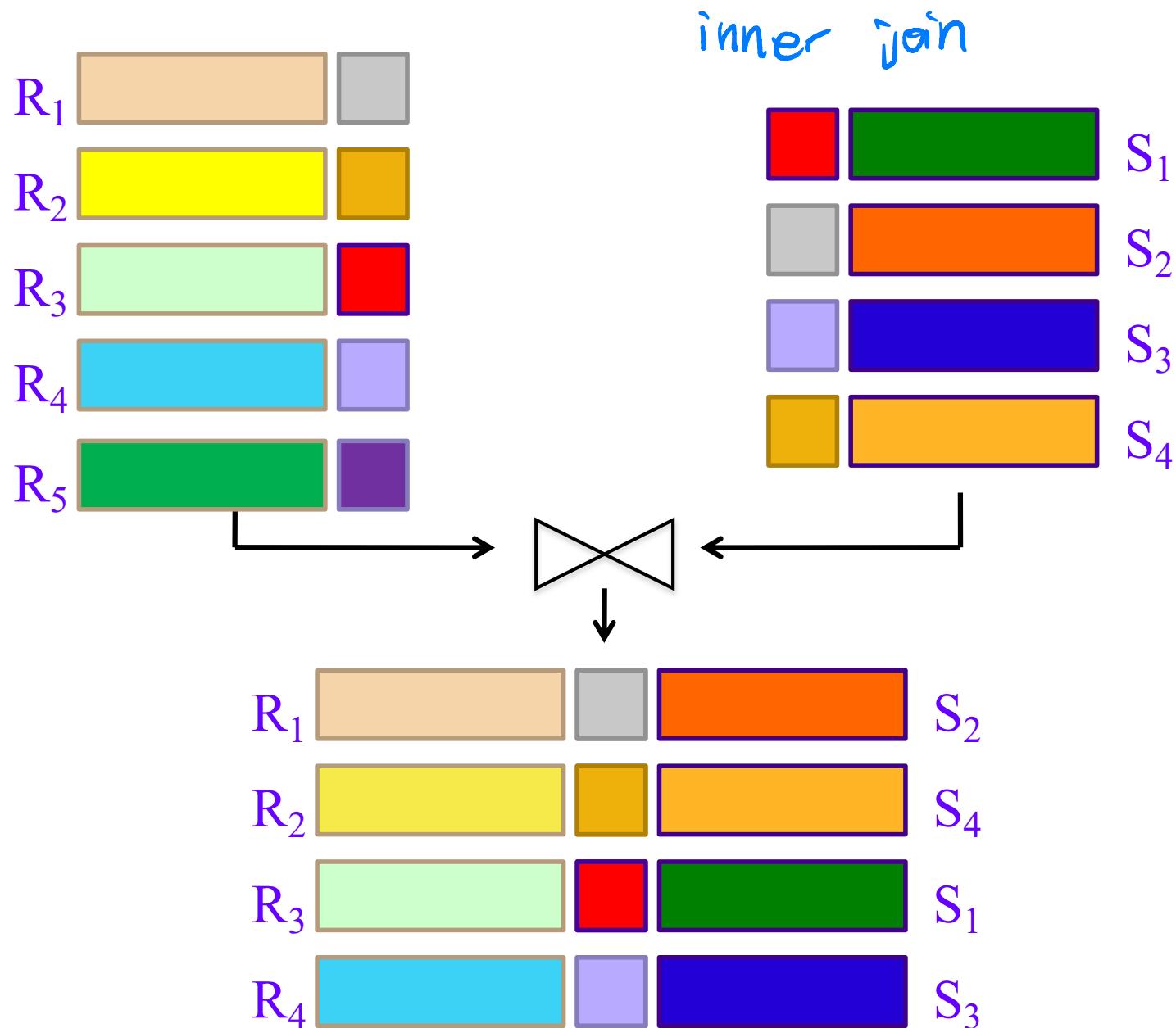
# Renaming of Attributes

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

```
Select E.LName As Employee_Name, S.LName As Supervisor_Name  
From Employee As E, Employee As S  
Where E.SuperSSN=S.SSN;
```

# Join Operation in SQL

# Relational Join



# Joined Tables: Q1A

- ♦ Retrieve the name and address of every employee who works for the ‘Research’ department

**EMPLOYEE**

| Fname | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|-------|-------|-------|-----|-------|---------|-----|--------|-----------|-----|
|-------|-------|-------|-----|-------|---------|-----|--------|-----------|-----|

**DEPARTMENT**

| Dname | Dnumber | Mgr_ssn | Mgr_start_date |
|-------|---------|---------|----------------|
|-------|---------|---------|----------------|

**Select Fname, Lname, Address**

**From (Employee Join Department on Dno=Dnumber)**

**Where DName='Research';**

# Joined Tables (cont.)

- ◆ Q1: **SELECT** FNAME, LNAME, ADDRESS  
**FROM** EMPLOYEE, DEPARTMENT  
**WHERE** DNAME = 'Research' **AND** DNUMBER = DNO

could be written as:

- ◆ Q1: **SELECT** FNAME, LNAME, ADDRESS  
**FROM** (EMPLOYEE **JOIN** DEPARTMENT  
          **ON** DNUMBER = DNO)  
**WHERE** DNAME = 'Research'

join operations are generally faster

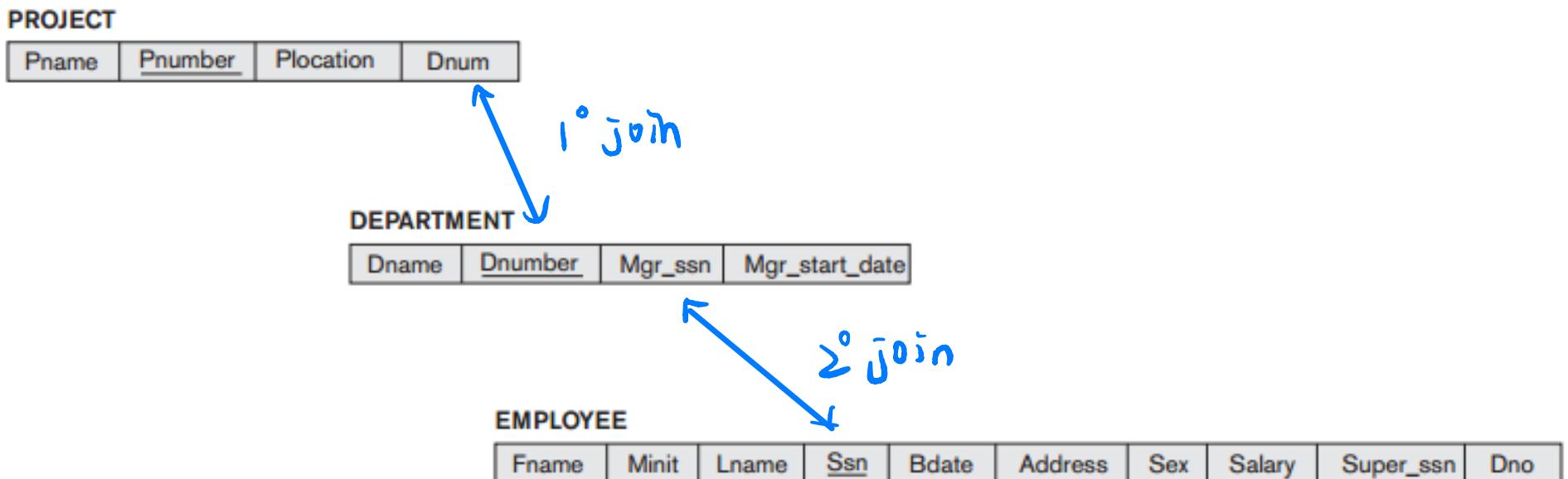
or as:

- ◆ Q1: **SELECT** FNAME, LNAME, ADDRESS  
**FROM** (EMPLOYEE **NATURAL JOIN**( DEPARTMENT  
          **AS** DEPT(DNAME, **DNO**, MSSN, MSDATE)))  
**WHERE** DNAME = 'Research"

\* rename DNUMBER as DNO, JOIN is performed on attributes of the same name

# Nested Join

```
SELECT PNUMBER, DNUM, LNAME, BDATE, ADDRESS  
FROM ((PROJECT JOIN DEPARTMENT ON DNUM = DNUMBER)  
      JOIN  
      EMPLOYEE ON MGRSSN = SSN)  
WHERE PLOCATION = 'Stafford'
```



$$AB = BA$$

JOIN 運算是否具有交換律？ Yes

是否具有結合律？ Yes

$$(AB)C = A(BC)$$



# Outer-Join

- ◆ INNER JOIN: only where both tables match  
(INNER JOIN aka JOIN)
- ◆ OUTER JOIN: where either one or both tables match
  - LEFT OUTER JOIN aka LEFT JOIN
  - RIGHT OUTER JOIN aka RIGHT JOIN
  - FULL OUTER JOIN aka FULL JOIN
- ◆ CROSS JOIN: Cartesian product (all combinations)

| 學號   | 姓名  |
|------|-----|
| 1101 | 陳綺貞 |
| 1102 | 陳信宏 |
| 1103 | 曹雅雯 |
| 1201 | 蔡依林 |
| 1301 | 吳青峰 |
| 2301 | 魏如萱 |
| 2302 | 林依晨 |

## Left Outer Join

| 學號   | 姓名  | 代碼    | 分數  |
|------|-----|-------|-----|
| 1101 | 陳綺貞 | C3001 | 90  |
| 1102 | 陳信宏 | C3001 | 70  |
| 1102 | 陳信宏 | J2010 | 80  |
| 1103 | 曹雅雯 | C3001 | 100 |
| 1201 | 蔡依林 |       |     |
| 1301 | 吳青峰 | C3001 | 80  |
| 2301 | 魏如萱 | C3001 | 90  |
| 2301 | 魏如萱 | J3020 | 85  |
| 2302 | 林依晨 | J2010 | 70  |
| 2302 | 林依晨 | J3020 | 80  |

=

| 學號   | 課程    | 分數  |
|------|-------|-----|
| 1101 | C3001 | 90  |
| 1102 | C3001 | 70  |
| 1102 | J2010 | 80  |
| 1103 | C3001 | 100 |
| 1301 | C3001 | 80  |
| 1302 | J2010 | 85  |
| 2301 | C3001 | 90  |
| 2301 | J3020 | 85  |
| 2302 | J2010 | 70  |
| 2302 | J3020 | 80  |

| 學號   | 姓名  |
|------|-----|
| 1101 | 陳綺貞 |
| 1102 | 陳信宏 |
| 1103 | 曹雅雯 |
| 1201 | 蔡依林 |
| 1301 | 吳青峰 |
| 2301 | 魏如萱 |
| 2302 | 林依晨 |

## Right Outer Join

| 學號   | 課程    | 分數  |
|------|-------|-----|
| 1101 | C3001 | 90  |
| 1102 | C3001 | 70  |
| 1102 | J2010 | 80  |
| 1103 | C3001 | 100 |
| 1301 | C3001 | 80  |
| 1302 | J2010 | 85  |
| 2301 | C3001 | 90  |
| 2301 | J3020 | 85  |
| 2302 | J2010 | 70  |
| 2302 | J3020 | 80  |

| 學號   | 姓名  | 代碼    | 分數  |
|------|-----|-------|-----|
| 1101 | 陳綺貞 | C3001 | 90  |
| 1102 | 陳信宏 | C3001 | 70  |
| 1102 | 陳信宏 | J2010 | 80  |
| 1103 | 曹雅雯 | C3001 | 100 |
| 1301 | 吳青峰 | C3001 | 80  |
| 1302 |     | J2010 | 85  |
| 2301 | 魏如萱 | C3001 | 90  |
| 2301 | 魏如萱 | J3020 | 85  |
| 2302 | 林依晨 | J2010 | 70  |
| 2302 | 林依晨 | J3020 | 80  |

=

| 學號   | 姓名  |
|------|-----|
| 1101 | 陳綺貞 |
| 1102 | 陳信宏 |
| 1103 | 曹雅雯 |
| 1201 | 蔡依林 |
| 1301 | 吳青峰 |
| 2301 | 魏如萱 |
| 2302 | 林依晨 |

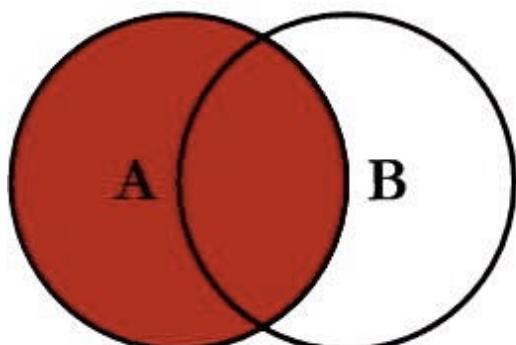
## Full Outer Join

| 學號   | 姓名  | 代碼    | 分數  |
|------|-----|-------|-----|
| 1101 | 陳綺貞 | C3001 | 90  |
| 1102 | 陳信宏 | C3001 | 70  |
| 1102 | 陳信宏 | J2010 | 80  |
| 1103 | 曹雅雯 | C3001 | 100 |
| 1201 | 蔡依林 |       |     |
| 1301 | 吳青峰 | C3001 | 80  |
| 1302 |     | J2010 | 85  |
| 2301 | 魏如萱 | C3001 | 90  |
| 2301 | 魏如萱 | J3020 | 85  |
| 2302 | 林依晨 | J2010 | 70  |
| 2302 | 林依晨 | J3020 | 80  |

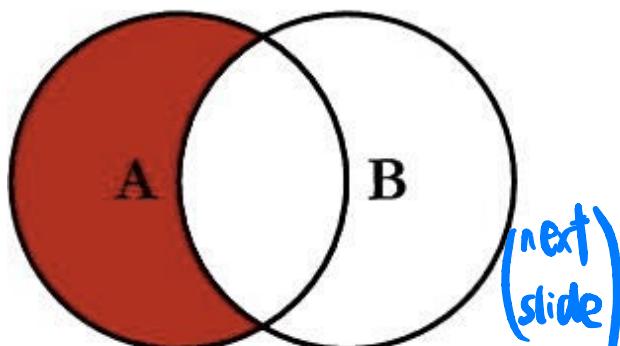
=

| 學號   | 課程    | 分數  |
|------|-------|-----|
| 1101 | C3001 | 90  |
| 1102 | C3001 | 70  |
| 1102 | J2010 | 80  |
| 1103 | C3001 | 100 |
| 1301 | C3001 | 80  |
| 1302 | J2010 | 85  |
| 2301 | C3001 | 90  |
| 2301 | J3020 | 85  |
| 2302 | J2010 | 70  |
| 2302 | J3020 | 80  |

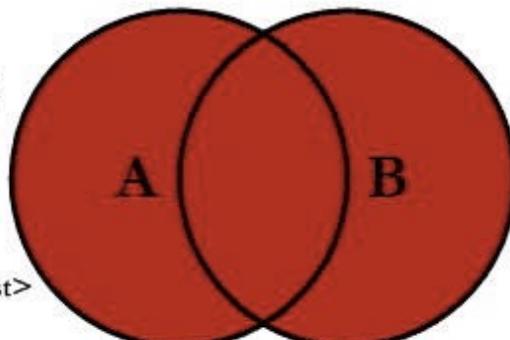
# SQL JOINS



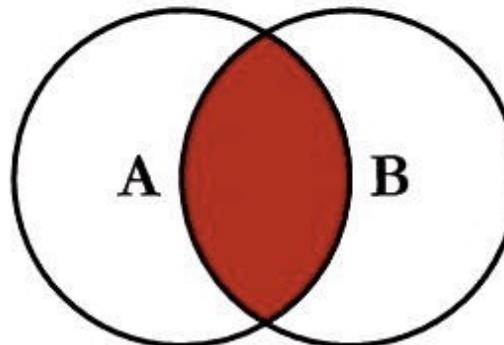
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



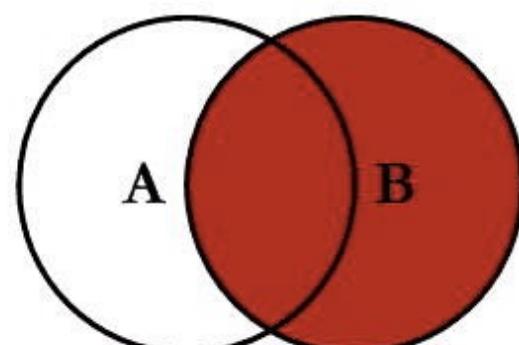
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
WHERE B.Key IS NULL
```



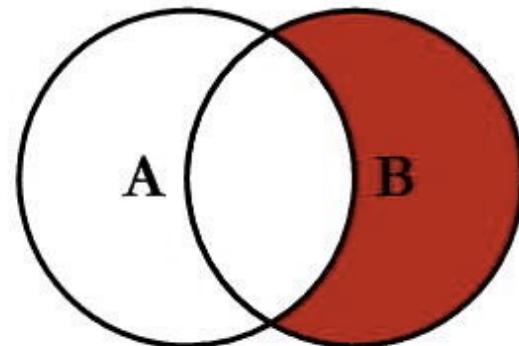
```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
```



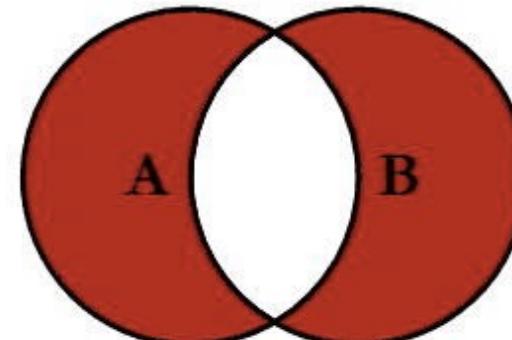
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



```
SELECT <select_list>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
OR B.Key IS NULL
```

# Replace Set Difference by Left Outer Join

| STUDENT | FN      | LN      |
|---------|---------|---------|
|         | Susan   | Yao     |
|         | Ramesh  | Shah    |
|         | Johnny  | Kohler  |
|         | Barbara | Jones   |
|         | Amy     | Ford    |
|         | Jimmy   | Wang    |
|         | Ernest  | Gilbert |

→

| INSTRUCTOR | FNAME   | LNAME   |
|------------|---------|---------|
|            | John    | Smith   |
|            | Ricardo | Browne  |
|            | Susan   | Yao     |
|            | Francis | Johnson |
|            | Ramesh  | Shah    |

=

| FN      | LN      |
|---------|---------|
| Johnny  | Kohler  |
| Barbara | Jones   |
| Amy     | Ford    |
| Jimmy   | Wang    |
| Ernest  | Gilbert |

| FN      | LN      | FNAME  | LNAME |
|---------|---------|--------|-------|
| Susan   | Yao     | Susan  | Yao   |
| Ramesh  | Shan    | Ramesh | Shan  |
| Johnny  | Kohler  |        |       |
| Barbara | Jones   |        |       |
| Amy     | Ford    |        |       |
| Jimmy   | Wang    |        |       |
| Emest   | Gilbert |        |       |

SELECT STUDENT.FN STUDENT.LN  
FROM STUDENT A  
LEFT JOIN INSTRUCTOR B  
ON STUDENT.FN = INSTRUCTOR.FNAME AND  
STUDENT.LN = INSTRUCTOR.LNAME  
WHERE INSTRUCTOR.FNAME IS NULL AND  
INSTRUCTOR.LNAME IS NULL;  
≡ B.Key is NULL

SELECT FN LN  
FROM STUDENT  
EXCEPT  
SELECT FNAME LNAME  
FROM INSTRUCTOR;

# Aggregate Function & Grouping in SQL

# Aggregate Functions

- ◆ Include **COUNT**, **SUM**, **MAX**, **MIN**, and **AVG**
- ◆ Q19: Find the sum, maximum salary, the minimum salary, and the average salary among all employees.

```
SELECT SUM(SALARY), MAX(SALARY),
       MIN(SALARY), AVG(SALARY)
FROM   EMPLOYEE
```

- ◆ Some SQL implementations *may not allow more than one function* in the SELECT-clause

# Aggregate Functions (cont.)

- ◆ Q20: Find the sum of salary, maximum salary, the minimum salary, and the average salary among employees who work for the 'Research' department.

```
SELECT SUM(SALARY),MAX(SALARY),  
       MIN(SALARY), AVG(SALARY)  
FROM   EMPLOYEE, DEPARTMENT  
WHERE  DNO=DNUMBER AND DNAME='Research'
```

**EMPLOYEE**

| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|

**DEPARTMENT**

| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|-------|----------------|---------|----------------|
|-------|----------------|---------|----------------|

# Aggregate Functions (cont.)

- ♦ Retrieve the total number of employees in the company (Q21), and the number of employees in the 'Research' department (Q22).

**Q21:** `SELECT COUNT (*)  
FROM EMPLOYEE`

**Q22:** `SELECT COUNT (*)  
FROM EMPLOYEE, DEPARTMENT  
WHERE DNO=DNUMBER AND  
DNAME='Research'`

Note: COUNT (\*) returns number of tuples/rows

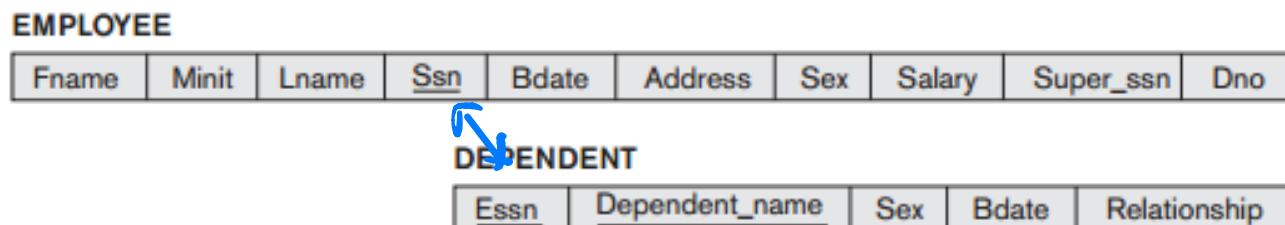
# Aggregate Functions (cont.)

- ◆ count the number of distinct salary values

```
SELECT COUNT(DISTINCT SALARY)  
FROM EMPLOYEE
```

- ◆ retrieve names of all employees who have two or more dependents

```
SELECT      LNAME,FNAME  
FROM        EMPLOYEE  
WHERE       (SELECT COUNT(*)  
                FROM DEPENDENT  
                WHERE EXPLOYEE.SSN=DEPEDENT.ESSN) >= 2;
```



# GROUPING

- ◆ In many cases, we want to apply the aggregate functions *to subgroups of tuples in a relation*
- ◆ Each *subgroup of tuples* consists of the set of tuples that have *the same value for the grouping attribute(s)*
- ◆ The function is applied to each subgroup *independently*
- ◆ SQL has a **GROUP BY**-clause for specifying the grouping attributes, which must also appear in the **SELECT**-clause

# Grouping (cont.)

- ◆ Q24: For each department, retrieve the department number, the number of employees in the department, and their average salary

```
SELECT      DNo, COUNT(*), AVG(Salary)  
FROM        Employee  
GROUP BY   DNo
```

| EMPLOYEE |       |       |     |       |         |     |        |           |     |  |
|----------|-------|-------|-----|-------|---------|-----|--------|-----------|-----|--|
| Fname    | Minit | Lname | Ssn | Bdate | Address | Sex | Salary | Super_ssn | Dno |  |

- EMPLOYEE tuples are divided into **groups**--each group having the **same value** for the **grouping attribute** DNO
- COUNT and AVG **functions** are applied to each such **group** of tuples separately
- SELECT-clause includes only the grouping attribute and the functions to be applied on each group of tuples

The diagram illustrates the grouping of EMPLOYEE tuples by the value of Dno. On the left, a partial view of the EMPLOYEE table is shown, containing tuples for employees John, Franklin, Ramesh, Joyce, Alicia, Jennifer, Ahmad, and James. The Dno column values for these tuples are 5, 5, 5, 5, 4, 4, 4, and 1 respectively. To the right, a summary table titled "Result of Q24" provides the count of tuples and the average salary for each group. Arrows point from the Dno values in the partial EMPLOYEE table to the corresponding rows in the summary table.

| Fname    | Minit | Lname   | Ssn       | ... | Salary | Super_ssn | Dno | Dno | Count (*) | Avg (Salary) |       |
|----------|-------|---------|-----------|-----|--------|-----------|-----|-----|-----------|--------------|-------|
| John     | B     | Smith   | 123456789 | ... | 30000  | 333445555 | 5   |     | 5         | 4            | 33250 |
| Franklin | T     | Wong    | 333445555 |     | 40000  | 888665555 | 5   |     | 4         | 3            | 31000 |
| Ramesh   | K     | Narayan | 666884444 |     | 38000  | 333445555 | 5   |     | 1         | 1            | 55000 |
| Joyce    | A     | English | 453453453 |     | 25000  | 333445555 | 5   |     |           |              |       |
| Alicia   | J     | Zelaya  | 999887777 |     | 25000  | 987654321 | 4   |     |           |              |       |
| Jennifer | S     | Wallace | 987654321 |     | 43000  | 888665555 | 4   |     |           |              |       |
| Ahmad    | V     | Jabbar  | 987987987 |     | 25000  | 987654321 | 4   |     |           |              |       |
| James    | E     | Bong    | 888665555 |     | 55000  | NULL      | 1   |     |           |              |       |

Result of Q24

Grouping EMPLOYEE tuples by the value of Dno

# Grouping (cont.)

- ◆ Q25: For each project, retrieve the project number, project name, and the number of employees who work on that project.

| PROJECT |         |           |      | WORKS_ON |     |       |
|---------|---------|-----------|------|----------|-----|-------|
| Pname   | Pnumber | Plocation | Dnum | Essn     | Pno | Hours |

```
SELECT      PNumber, PName, COUNT (*)  
FROM        Project, Works_On  
WHERE       Pnumber = PNo  
GROUP BY    PNumber, PName
```

- ◆ the grouping functions are applied **after** the joining of the two relations

# THE HAVING-CLAUSE

- ◆ Sometimes we want to retrieve the values of these functions for only those **groups that satisfy certain conditions**
- ◆ The HAVING-clause is used for specifying a **selection condition on groups** (rather than on individual tuples)

# THE HAVING-CLAUSE (cont.)

- ♦ Q26: For each project *on which more than two employees work* , retrieve the project number, project name, and the number of employees who work on that project.

| PROJECT |         |           |      |
|---------|---------|-----------|------|
| Pname   | Pnumber | Plocation | Dnum |

| WORKS_ON |     |       |
|----------|-----|-------|
| Essn     | Pno | Hours |

```
SELECT      PNUMBER, PNAME, COUNT (*)  
FROM        PROJECT, WORKS_ON  
WHERE       PNUMBER = PNO  
GROUP BY    PNUMBER, PNAME  
HAVING     COUNT (*) > 2
```

| Pname           | Pnumber | ... | Essn      | Pno | Hours |
|-----------------|---------|-----|-----------|-----|-------|
| ProductX        | 1       |     | 123456789 | 1   | 32.5  |
| ProductX        | 1       |     | 453453453 | 1   | 20.0  |
| ProductY        | 2       |     | 123456789 | 2   | 7.5   |
| ProductY        | 2       |     | 453453453 | 2   | 20.0  |
| ProductY        | 2       |     | 333445555 | 2   | 10.0  |
| ProductZ        | 3       |     | 666884444 | 3   | 40.0  |
| ProductZ        | 3       |     | 333445555 | 3   | 10.0  |
| Computerization | 10      | ... | 333445555 | 10  | 10.0  |
| Computerization | 10      |     | 999887777 | 10  | 10.0  |
| Computerization | 10      |     | 987987987 | 10  | 35.0  |
| Reorganization  | 20      |     | 333445555 | 20  | 10.0  |
| Reorganization  | 20      |     | 987654321 | 20  | 15.0  |
| Reorganization  | 20      |     | 888665555 | 20  | NULL  |
| Newbenefits     | 30      |     | 987987987 | 30  | 5.0   |
| Newbenefits     | 30      |     | 987654321 | 30  | 20.0  |
| Newbenefits     | 30      |     | 999887777 | 30  | 30.0  |

These groups are not selected by the HAVING condition of Q26.

After applying the WHERE clause but before applying HAVING

The diagram illustrates the execution flow of a query. It consists of three tables and their relationships:

- Main Table:** Contains columns Pname, Pnumber, ..., Essn, Pno, and Hours.
- Subquery:** Contains columns Pname and Count (\*).
- Result Table:** Contains columns Pname and Count (\*) with specific values.

Relationships are indicated by arrows:

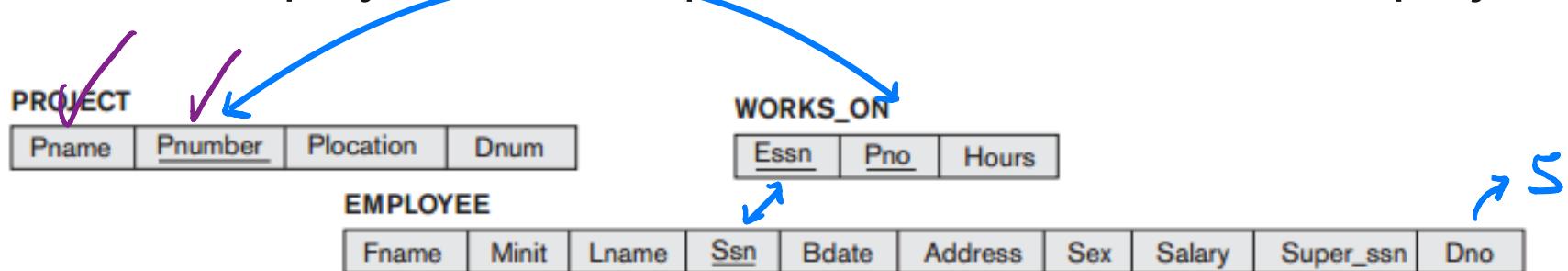
- A vertical arrow points from the Main Table's Pname column to the Subquery's Pname column.
- A vertical arrow points from the Main Table's Pnumber column to the Subquery's Count (\*) column.
- A horizontal arrow points from the Main Table's Essn, Pno, and Hours columns to the Subquery's Pname column.
- A horizontal arrow points from the Subquery's Pname and Count (\*) columns to the Result Table's Pname and Count (\*) columns.

| Pname           | Pnumber | ... | Essn      | Pno | Hours | Pname                                | Count (*) |
|-----------------|---------|-----|-----------|-----|-------|--------------------------------------|-----------|
| ProductY        | 2       | ... | 123456789 | 2   | 7.5   | ProductY                             | 3         |
| ProductY        | 2       |     | 453453453 | 2   | 20.0  | Computerization                      | 3         |
| ProductY        | 2       |     | 333445555 | 2   | 10.0  | Reorganization                       | 3         |
| Computerization | 10      | ... | 333445555 | 10  | 10.0  | Newbenefits                          | 3         |
| Computerization | 10      |     | 999887777 | 10  | 10.0  | Result of Q26<br>(Pnumber not shown) |           |
| Computerization | 10      |     | 987987987 | 10  | 35.0  |                                      |           |
| Reorganization  | 20      |     | 333445555 | 20  | 10.0  |                                      |           |
| Reorganization  | 20      |     | 987654321 | 20  | 15.0  |                                      |           |
| Reorganization  | 20      |     | 888665555 | 20  | NULL  |                                      |           |
| Newbenefits     | 30      |     | 987987987 | 30  | 5.0   |                                      |           |
| Newbenefits     | 30      |     | 987654321 | 30  | 20.0  |                                      |           |
| Newbenefits     | 30      |     | 999887777 | 30  | 30.0  |                                      |           |

After applying the HAVING clause condition

# THE HAVING-CLAUSE (cont.)

- ♦ Q27: For each project, retrieve project number, project name, the number of employees from department 5 who work on the project



```
SELECT PNUMBER, PNAME, COUNT (*)  
FROM PROJECT, WORKS_ON, EMPLOYEE  
WHERE PNUMBER=PNO AND SSN=ESSN  
      AND DNO=5  
GROUP BY PNUMBER, PNAME
```

# Q28

- ♦ for each department that has more than five employees, retrieve the department number, number of its employees who are making more than \$40K

DEPARTMENT

| Dname | <u>Dnumber</u> | Mgr_ssn | Mgr_start_date |
|-------|----------------|---------|----------------|
|-------|----------------|---------|----------------|



EMPLOYEE

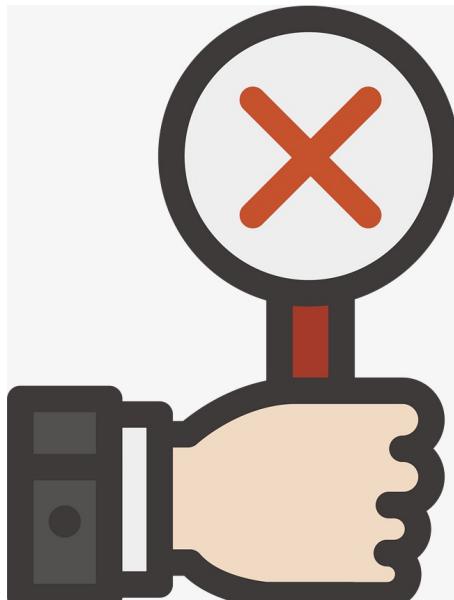
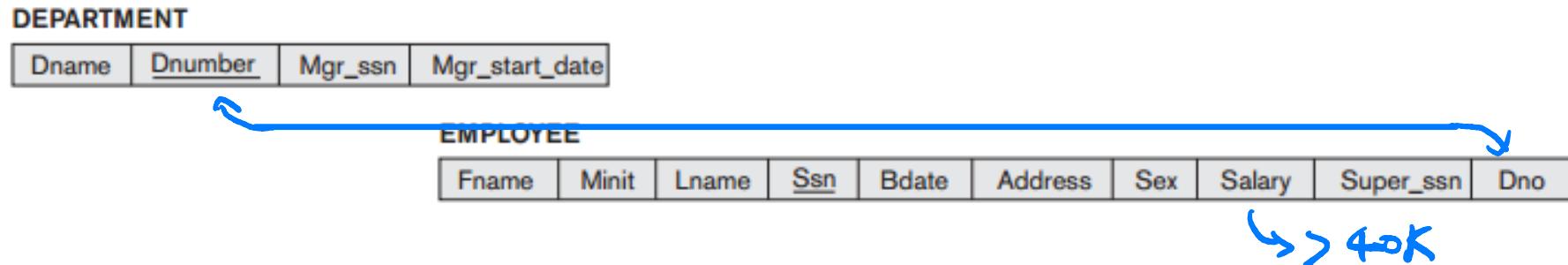
| Fname | Minit | Lname | <u>Ssn</u> | Bdate | Address | Sex | Salary | Super_ssn | Dno |
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|
|-------|-------|-------|------------|-------|---------|-----|--------|-----------|-----|

→ >40K

```
SELECT DNUMBER, COUNT (*)
FROM DEPARTMENT, EMPLOYEE
WHERE DNUMBER=DNO AND SALARY>40000 AND
DNO IN (SELECT DNO
          FROM EMPLOYEE
          GROUP BY DNO
          HAVING COUNT(*) >5)
GROUP BY DNUMBER
```

# Q28

Q28: for each department that has more than five employees, retrieve the department number, number of its employees who are making more than \$40K



```
SELECT DNAME, COUNT (*)  
FROM DEPARTMENT, EMPLOYEE  
WHERE DNUMBER=DNO AND SALARY>40000  
GROUP BY DNAME  
HAVING COUNT(*) > 5 < incorrect
```

Output: *Count(Salary > 40k)*

# SQL queries run in this order

FROM + JOIN



WHERE



GROUP BY



HAVING



SELECT (window functions)  
happen here !



ORDER BY



LIMIT

# Summary of SQL Queries

SELECT <attribute list>

FROM <table list>

[WHERE <condition>]

[GROUP BY <grouping attribute(s)>]

[HAVING <group condition>]

[ORDER BY <attribute list>]

**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          |

**WORKS\_ON**

| <u>Essn</u> | 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           | <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       |

# Summary of SQL Queries (cont.)

- ◆ 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