

SQL-The Relational Database Standard

政治大學
資訊科學系
沈錕坤

SQL

- ◆ SQL (Structure Query language)
 - Standard language for commercial relational DBMS
 - Declarative language
 - users specify **what** the result is to be
 - Based on **relational calculus**
 - **Relational algebra**: users specify **how**, in what order, to execute the query operations
 - Originally, SEQUEL (Structure English QUery Language) & was designed and implemented for IBM System R

SQL (cont.)

- ◆ SQL (Structure Query language)
 - Versions
 - SQL1: SQL-86 (ANSI 1986)
 - SQL2: SQL-92
 - SQL3:
 - SQL 2003, 2006: added XML
 - SQL 2008: incorporate more object database features
 - SQL 2011: add temporal data
 - SQL 2016: row pattern matching, polymorphic table functions, JSON
 - SQL 2019: multidimensional array
 - Support DDL, DML, VIEW, Security, Authorization, Integrity Constraints, Transaction Control

An Example Database Application

- ◆ keeps track of a company's **employees, departments & projects**
 - The company is organized into departments.
 - Each department has a unique name, a unique number & a particular employee who manages the department.
 - We keep track of the start date when that employee began managing the department.
 - A department may have several locations
 - A department controls a number of projects, each of which has a unique name, a unique number & a single location
 - We store each employee's name, social security number, address, salary, sex & birth date

An Example Database Application (cont.)

- An employee is assigned to one department but may work on several projects, which are not necessarily controlled by the same department.
- We keep track of the number of hours per week that an employee works on each project.
- We keep track of the direct supervisor of each employee.
- We keep track of the dependents of each employee for insurance purpose.
- We keep each dependent's first name, sex, birth date & relationship to 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

DEPARTMENT

Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19

DEPT_LOCATIONS

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

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

EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

DEPT_LOCATIONS

<u>DNUMBER</u>	<u>DLOCATION</u>
----------------	------------------

PROJECT

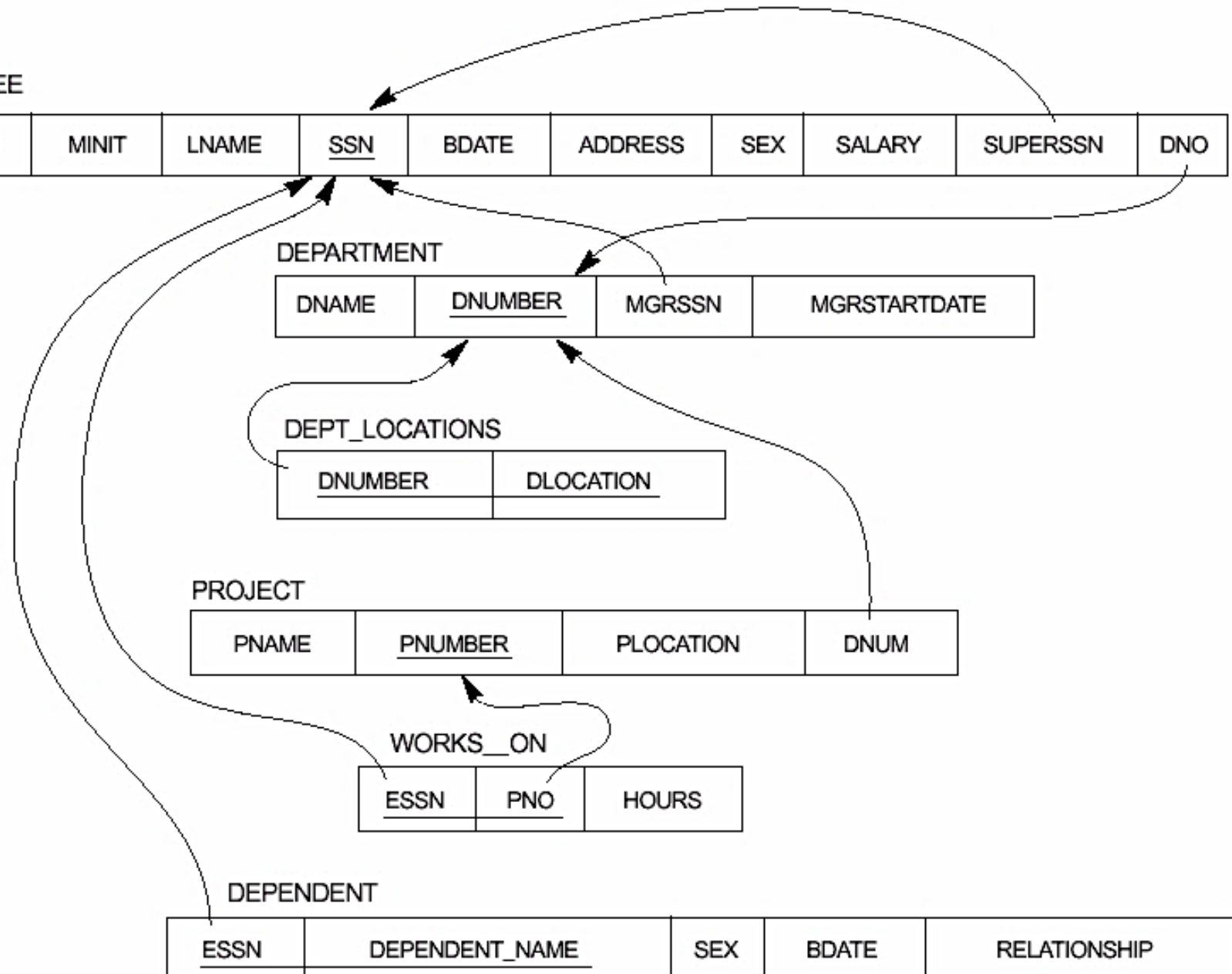
PNAME	<u>PNUMBER</u>	PLOCATION	DNUM
-------	----------------	-----------	------

WORKS_ON

<u>ESSN</u>	<u>PNO</u>	HOURS
-------------	------------	-------

DEPENDENT

<u>ESSN</u>	<u>DEPENDENT_NAME</u>	SEX	BDATE	RELATIONSHIP
-------------	-----------------------	-----	-------	--------------



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.

DDL in SQL

(Data Definition Language)

Data Definition

- ◆ SQL terms
 - Table: relation
 - Row: tuple
 - Column: attribute
- ◆ DDL support
 - CREATE
 - ALTER
 - DROP

Schema & Catalog Concepts in SQL

◆ SQL schema

- Group together tables & other constructs that belong to the same database application
- Identified by
 - Schema name
 - Authorization identifier
 - Descriptors for each elements (tables, constraints, views, domains & other construct) in schema
- e.g.

CREATE SCHEMA company **AUTHORIZATION** 'Jsmith';

◆ Catalog (Data Dictionary)

- Collection of schemas
- Contains information of schema, referential constraints, domain definition sharing

EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

DEPT_LOCATIONS

<u>DNUMBER</u>	<u>DLOCATION</u>
----------------	------------------

PROJECT

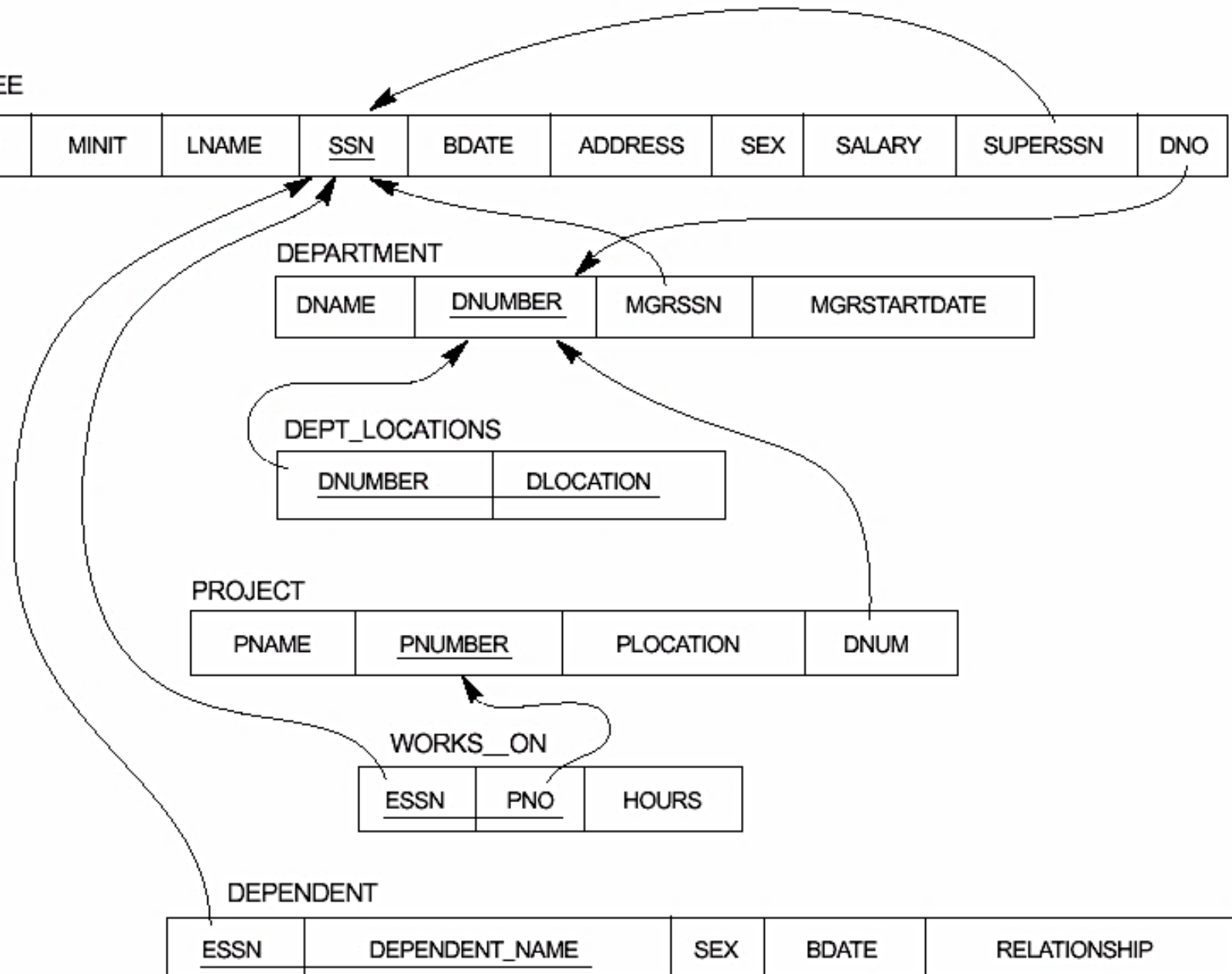
PNAME	<u>PNUMBER</u>	PLOCATION	DNUM
-------	----------------	-----------	------

WORKS_ON

<u>ESSN</u>	<u>PNO</u>	HOURS
-------------	------------	-------

DEPENDENT

<u>ESSN</u>	<u>DEPENDENT_NAME</u>	SEX	BDATE	RELATIONSHIP
-------------	-----------------------	-----	-------	--------------



CREATE TABLE

◆ **CREATE TABLE** Department

(Dname	Varchar(15)	Not Null,
	Dnumber	Int	Not Null,
	MgrSSN	Char(9)	Not Null,
	MgrStartDate	Date	Not Null,

Primary Key(Dnumber),
Unique(Dname),
Foreign Key(MgrSSN) **References** Employee (SSN));

Data Types & Domains

- ◆ Numeric
 - Integer: INTEGER (INT), SMALLINT
 - Real numbers (FLOAT, REAL, DOUBLE PRECISION)
 - Formatted numbers: DECIMAL (i, j) or DEC(i, j)
- ◆ Character string
 - Fixed length: CHAR(n) or CHARACTER (n)
 - Variable length
 - VARCHAR(n) or CHAR VARYING(n) or CHARACTER VARYING(n)
- ◆ Bit string
 - Fixed length: BIT(n)
 - Variable length: BIT VARYING(n)

Data Types (cont.)

- ◆ Boolean: 3-value logic, TRUE, FALSE, UNKNOWN
- ◆ Data and time
 - DATE: year-month-day format yyyy-mm-dd
 - TIME: hour:minute:second format hh:mm:ss
 - TIME(i): hour:minute:second plus i additional digits
specifying fractions of a second format is hh:mm:ss:ii...i
 - TIMESTAMP: DATE and TIME components
- ◆ **CREATE DOMAIN SSN_TYPE AS CHAR(9);**

EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

DEPT_LOCATIONS

<u>DNUMBER</u>	<u>DLOCATION</u>
----------------	------------------

PROJECT

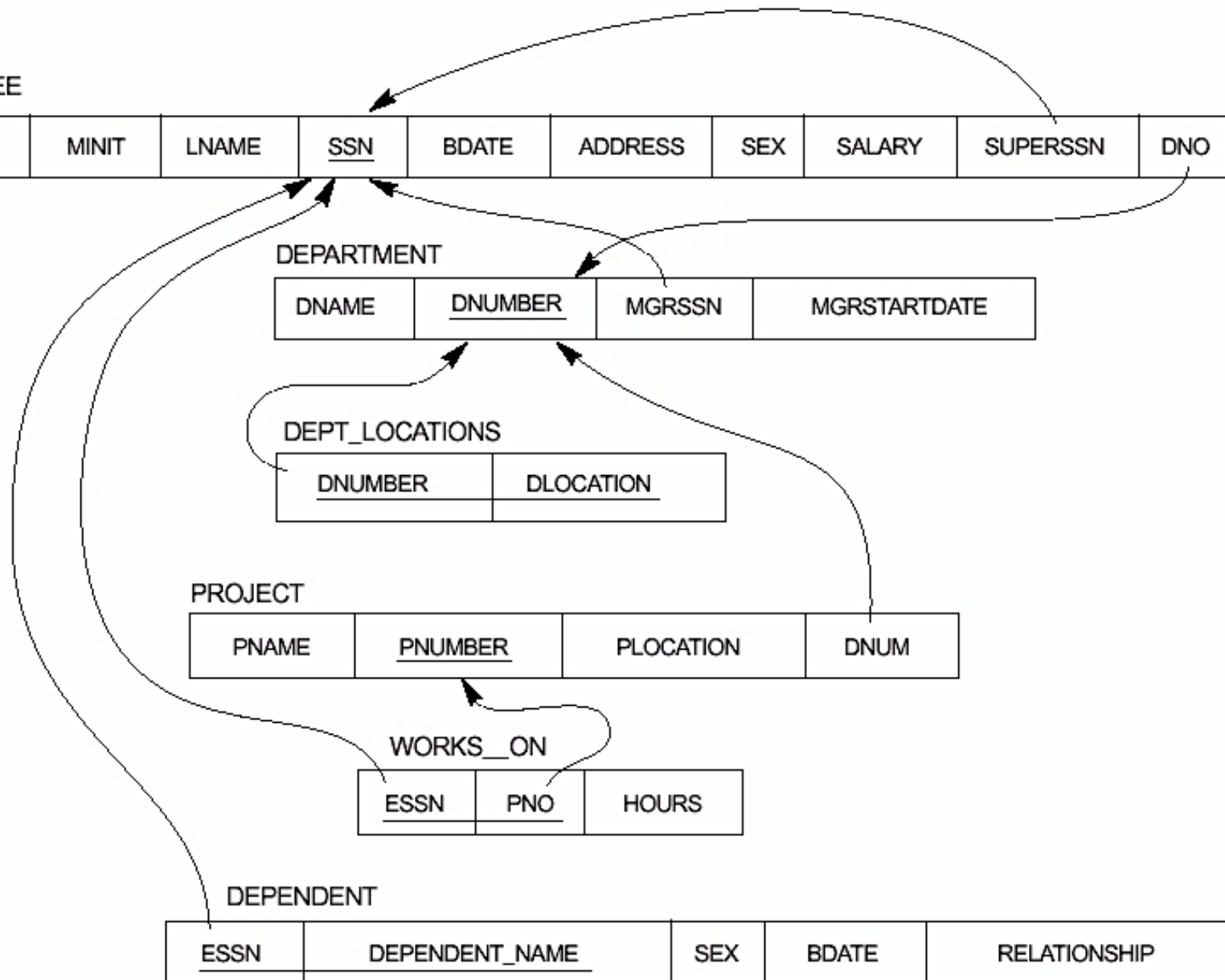
PNAME	<u>PNUMBER</u>	PLOCATION	DNUM
-------	----------------	-----------	------

WORKS_ON

<u>ESSN</u>	<u>PNO</u>	HOURS
-------------	------------	-------

DEPENDENT

<u>ESSN</u>	<u>DEPENDENT_NAME</u>	SEX	BDATE	RELATIONSHIP
-------------	-----------------------	-----	-------	--------------



Constraints

◆ **CREATE TABLE** Department

(Dname **VARCHAR(15)** **NOT NULL**,
 Dnumber **INT** **NOT NULL CHECK** (Dnumber <20)
 MgrSSN **CHAR(9)** **NOT NULL DEFAULT** '888665555',
 MgrStartDate **DATE** **NOT NULL**,

CONSTRAINT Deptpk

PRIMARY KEY (Dnumber),

CONSTRAINT Deptsk

UNIQUE (Dname),

CONSTRAINT Deptmgrfk

FOREIGN KEY (MgrSSN) **REFERENCES** Employee (SSN)

ON DELETE Set Default

ON UPDATE Cascade

);

Drop Schema & Drop Table

- ◆ Drop Schema Company Cascade
 - Cascade: remove all tables, domains & other elements
 - Restrict: schema is dropped only if it has no elements in it
- ◆ Drop table Dependent Cascade
 - Cascade: all constraints & views that reference the table are dropped automatically
 - Restrict: a table is dropped only if it is not referenced in any constraints

Alter Table

- ◆ Alter table command: schema evolution
 - Adding or dropping a column
 - Changing a column definition
 - Adding or dropping table constraints
- ◆ e.g.
 - **Alter Table** Company.Employee **Add** Job varchar(12);
 - **Alter Table** Company.Employee **Drop** Address **Cascade**;
 - **Alter Table** Company.Department **Alter** MgrSSN **Drop** **Default**;
 - **Alter Table** Company.Department **Alter** MgrSSN **Set** **Default** “333445555”;
 - **Alter Table** Company.Department **Drop** **Constraint** EmpSuperFK **Cascade**;

Basic SQL DML Queries (Data Manipulation Language)

SQL DML Queries

- ◆ Basic form of Select statement

SELECT <attribute list>

FROM <table list>

WHERE <condition>;

- Attribute list: list of attribute names whose value are to be retrieved by the query
- Table list: a list of relation names required to process the query
- Condition: a conditional (Boolean) expression that identifies the tuples to be retrieved by the query

EMPLOYEE

FNAME	MINIT	LNAME	<u>SSN</u>	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
-------	-------	-------	------------	-------	---------	-----	--------	----------	-----

DEPARTMENT

DNAME	<u>DNUMBER</u>	MGRSSN	MGRSTARTDATE
-------	----------------	--------	--------------

DEPT_LOCATIONS

<u>DNUMBER</u>	<u>DLOCATION</u>
----------------	------------------

PROJECT

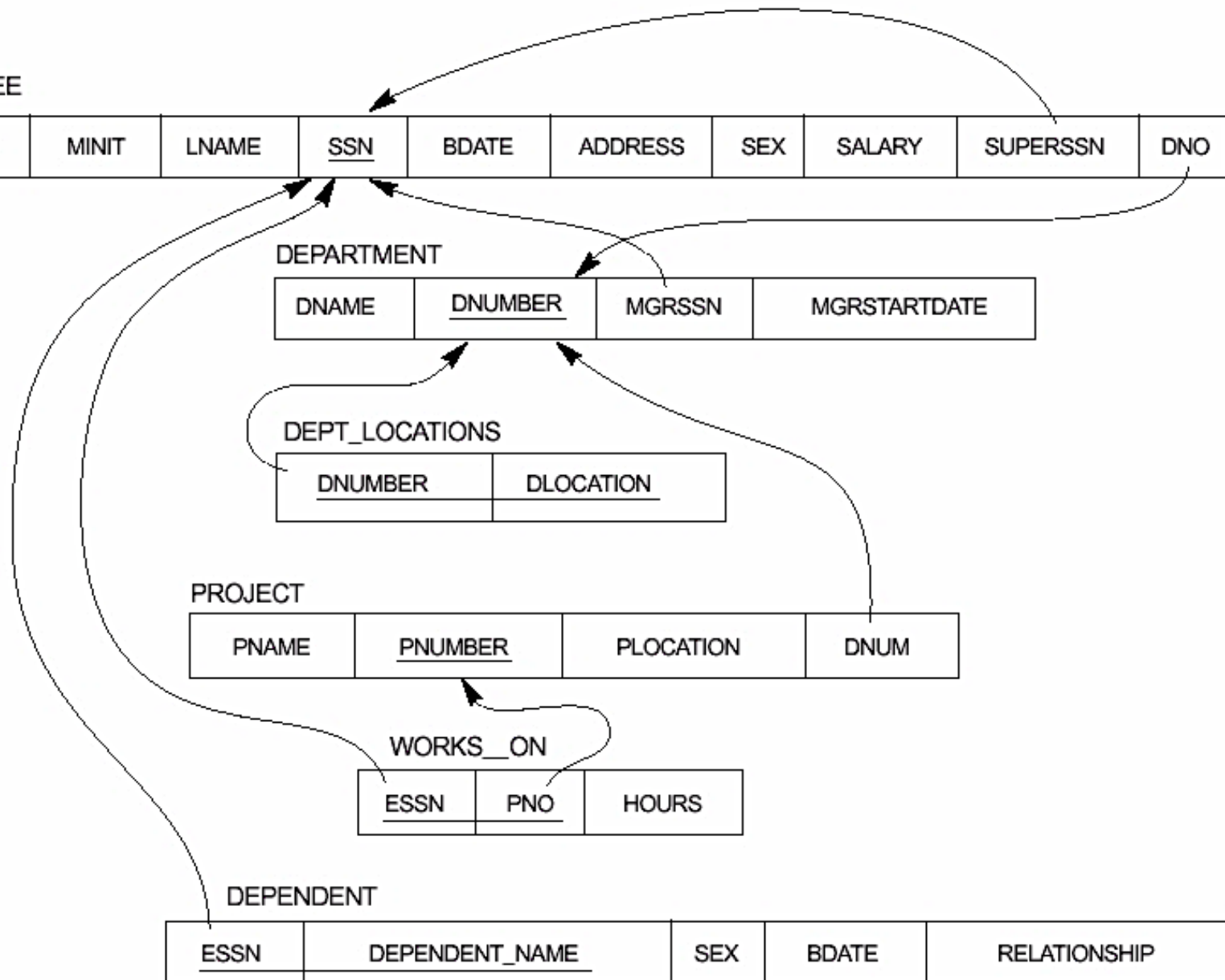
PNAME	<u>PNUMBER</u>	PLOCATION	DNUM
-------	----------------	-----------	------

WORKS_ON

<u>ESSN</u>	<u>PNO</u>	HOURS
-------------	------------	-------

DEPENDENT

<u>ESSN</u>	<u>DEPENDENT_NAME</u>	SEX	BDATE	RELATIONSHIP
-------------	-----------------------	-----	-------	--------------



Basic SQL Queries

- ◆ Q0: Retrieve the birthday and address of the employee(s) whose name is 'John B. Smith'

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

Basic SQL Queries

- ◆ Q0: Retrieve the birthday and address of the employee(s) whose name is 'John B. Smith'

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.

Basic SQL Queries

- ◆ Q0: Retrieve the birthday and address of the employee(s) whose name is 'John B. Smith'

Select Bdate, Address

From Employee

Where Fname='John' **And** Minit='B' **And** Lname='Smith';

Basic SQL Queries (cont.)

- ♦ Q1: Retrieve the name and address of all employees who work for the 'Research' department.

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

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

Basic SQL Queries (cont.)

- ◆ Q1: Retrieve the name and address of all employees who work for the 'Research' department.

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.

Basic SQL Queries (cont.)

- ◆ Q1: Retrieve the name and address of all employees who work for the 'Research' department.

EMPLOYEE

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

DEPARTMENT

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

Select Fname, Lname, Address

From Employee, Department

Where DName='Research' **And** DNumber=DNo;

Basic SQL Queries (cont.)

- ◆ Q2: For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthday.

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

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

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

Basic SQL Queries (cont.)

- ◆ Q2: For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthday.

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.

Basic SQL Queries (cont.)

- ◆ Q2: For every project located in 'Stafford', list the project number, the controlling department number, and the department manager's last name, address, and birthday.

PROJECT

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

DEPARTMENT

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

EMPLOYEE

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

Select Pnumber, Dnum, Lname, Address, BDate

From Project, Department, Employee

Where Dnum=Dnumber **And**

MgrSSN=SSN **And**

Plocation='Stafford';

Ambiguous Attribute Name, Aliasing

- ◆ Q1A: Retrieve the name and address of all employees who work for the 'Research' department.

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.

Ambiguous Attribute Name, Aliasing

- ◆ Q1A: Retrieve the name and address of all employees who work for the 'Research' department.

Select Fname, Employee.Name, Address
From Employee, Department
Where Department.Name='Research' **AND**
 Department.DNumber=Employee.DNumber

Ambiguous Attribute Name, Aliasing (cont.)

- ◆ **Alias** for queries referring to the same relation twice
- ◆ Q8: For each employee, retrieve the employee's first and last name and the first and last name of his or her immediate supervisor.

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

Ambiguous Attribute Name, Aliasing (cont.)

- ◆ Q8: For each employee, retrieve the employee's first and last name and the first and last name of his or her immediate supervisor.

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.

Ambiguous Attribute Name, Aliasing (cont.)

- ◆ Q8: For each employee, retrieve the employee's first and last name and the first and last name of his or her immediate supervisor.

EMPLOYEE

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

EMPLOYEE

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

```
Select E.Fname, E.Lname, S.Fname, S.LName  
From Employee As E, Employee As S  
Where E.SuperSSN = S.SSN;
```

Unspecified Where Clause

- ◆ Q9: Select all Employee SSNs

Select SSN

From Employee;

Unspecified Where Clause (cont.)

Select SSN, Dname
From Employee, Department;

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

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

Unspecified Where Clause (cont.)

- ◆ Q10: Select all combinations of Employee SSN and Department Dname in the database

Select SSN, Dname

From Employee, Department;

- ◆ **cross product**: all possible combinations, if
 - more than one relation is specified in the from clause and
 - there is no where-clause

Use of the Asterisk

- ◆ Q1D: Retrieve **all the attributes** of an employee and the attributes of the department he or she works in for every employee of the 'Research' department

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

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

Use of the Asterisk

- ◆ Q1D: Retrieve all the attributes of an employee and the attributes of the department he or she works in for every employee of the 'Research' department

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.

Use of the Asterisk

- ◆ Q1D: Retrieve all the attributes of an employee and the attributes of the department he or she works in for every employee of the 'Research' department

```
Select      *  
From       Employee, Department  
Where      Dname='Research' And Dno=Dnumber;
```

Tables as Sets in SQL

- ◆ Q11: Retrieve the salary of every employee

Select all Salary

From Employee;

Tables as Sets in SQL (cont.)

- ◆ Q11A: Retrieve all distinct salary values

```
Select Distinct      Salary
From      Employee;
```

- ◆ SQL
 - treats a table not as a set but rather as a **multiset**
 - **Duplicate tuples** can appear more than once in a table and in the result of a query
 - Reasons
 - Duplicate elimination is expensive
 - Users requirement
 - For using Aggregate function (e.g. average score)

Tables as Sets in SQL (cont.)

- ◆ Q4: 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	<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

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

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

Tables as Sets in SQL (cont.)

- ◆ Q4: 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	<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.

Tables as Sets in SQL (cont.)

- ◆ Q4: 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	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

(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');

DEPARTMENT

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

PROJECT

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

WORKS_ON

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

Tables as Sets in SQL (cont.)

- ◆ Set operations in SQL
 - UNION: union operation
 - EXCEPT: set difference
 - INTERSECT: intersection
- ◆ Duplicate tuples are eliminated from the result of set operations
- ◆ Union compatible

Substring Pattern Matching

- ◆ Q12: Retrieve all employees whose address is in Houston, Texas

Select FName, LName

From Employee

Where Address Like '%Houston, TX%';

Substring Pattern Matching (cont.)

- ◆ Q12A: Find all employees who were born during the 1950s

```
Select    Fname, Lname
From      Employee
Where      Bdate like '___5_____';
```

Arithmetic Operators

- ◆ Q13: Show the resulting salaries if every employee working on the 'ProductX' project is given a 10 percent raise

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.

Arithmetic Operators

- ◆ Q13: Show the resulting salaries if every employee working on the 'ProductX' project is given a 10 percent raise

```
Select      Fname, Lname, 1.1*Salary
From        Employee, Works_On, Project
Where       SSN=ESSN  and
              Pno=Pnumber and
              Pname='ProductX';
```

- ◆ ||: concatenate operator for appending two string values
- ◆ +, -: for date, time, timestamp by a type-compatible interval

Arithmetic Operators (cont.)

- ◆ Q14: Retrieve all employees in department 5 whose salary is between \$30,000 and \$40,000

```
Select      *  
  
From       Employee  
  
Where      (Salary Between 30000 and 40000) and  
             DNo=5;
```

Ordering of Query Result

- ◆ Q15: Retrieve a list of employees and the projects they are working on, ordered by **descending order** of department and **ascending order** on LName, FName

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

Ordering of Query Result

- ◆ Q15: Retrieve a list of employees and the projects they are working on, ordered by descending order of department and ascending order on LName, FName

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.

Ordering of Query Result

- ◆ Q15: Retrieve a list of employees and the projects they are working on, ordered by descending order of department and ascending order on LName, FName

DEPARTMENT

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

EMPLOYEE

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

WORKS_ON

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

PROJECT

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

Select Dname, Lname, FName, Pname
From Department, Employee, Works_On, Project
Where Dnumber=Dno **and**
SSN=ESSN **and**
PNO=Pnumber
Order By Dname **Desc**, Lname **Asc**, FName **Asc**;

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	<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 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

PROJECT

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

EMPLOYEE

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

DEPARTMENT

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

WORKS_ON

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

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');

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>	<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.

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

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.

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

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

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

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.

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	<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

DEPENDENT

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

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	<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

DEPENDENT

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

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.

Exists Functions

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

DEPENDENT

<u>Essn</u>	<u>Dependent_name</u>	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);

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 E.SSN in (
 Select ESSN
 From Dependent as D
 Where E.Sex = D.Sex and
 E.Fname = D.Dependent_Name);

- ♦ **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);

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 **Exists** (**Select** *
 From Dependent
 Where E.Sex = Sex and
 E.Fname = Dependent_Name);

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	<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.

Exists Functions (cont.)

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

EMPLOYEE

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

DEPENDENT

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

EMPLOYEE

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

DEPENDENT

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

DEPARTMENT

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

```
Select Fname, Lname
From Employee
Where Exists ( Select *
                From Dependent
                Where SSN=ESSN)

And
Exists ( Select *
          From Department
          Where SSN=MgrSSN);
```

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

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	<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.

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
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

Select Fname, Lname
From Employee
Where Not Exists

((**Select** PNumber
 From Project
 Where DNum=5)

Except

(**Select** PNo
 From Works_On
 Where Employee.SSN=ESSN));

PROJECT

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

WORKS_ON

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

- ◆ 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)

EMPLOYEE

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

WORKS_ON

<u>Essn</u>	<u>Pno</u>	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));

PROJECT

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

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)

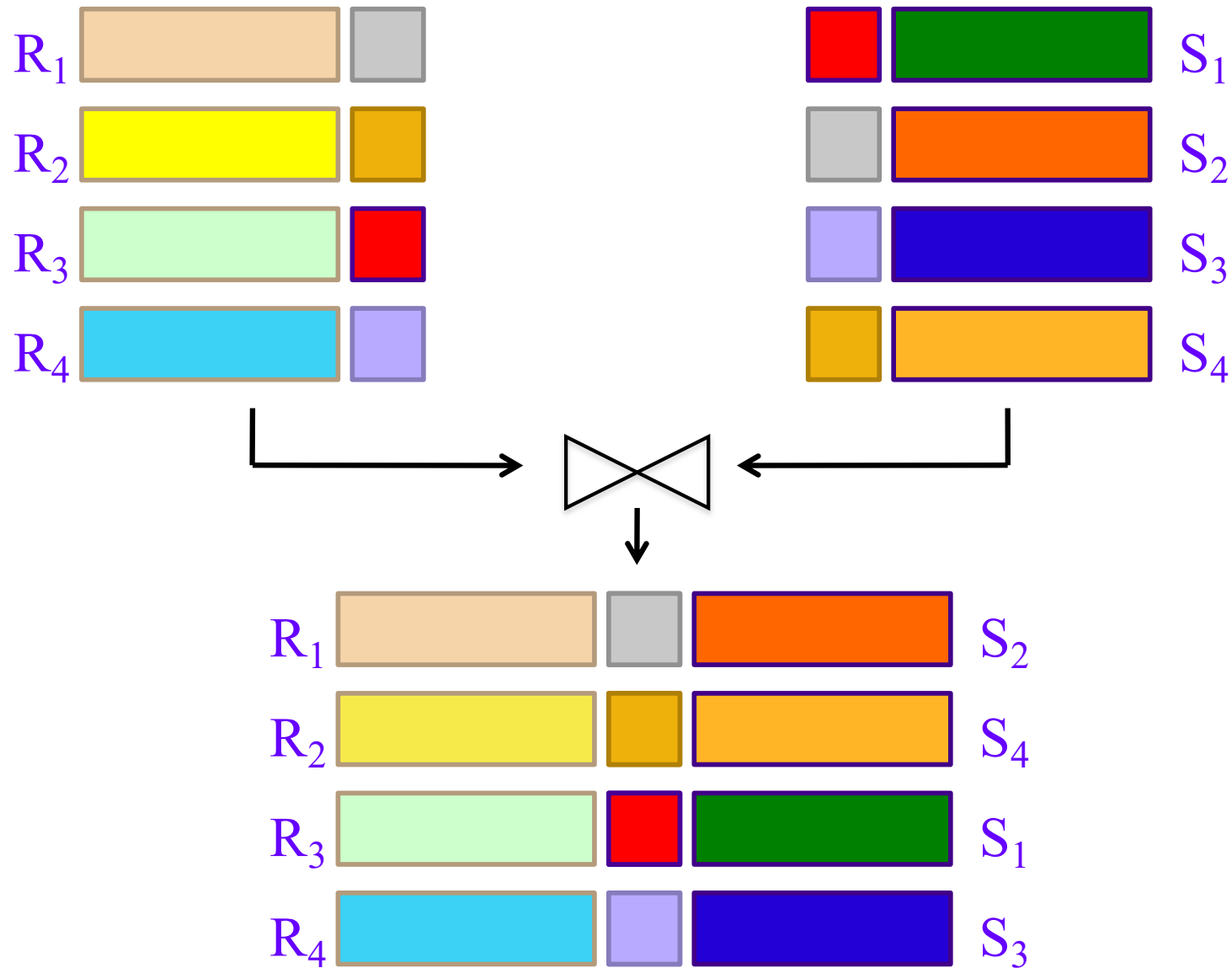
Renaming of Attributes

- ◆ 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	<u>Ssn</u>	Bdate	Address	Sex	Salary	Super_ssn	Dno
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

DEPARTMENT

Dname	<u>Dnumber</u>	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'

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'
```

PROJECT

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

DEPARTMENT

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

EMPLOYEE

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

JOIN 運算是否具有交換律？

是否具有結合律？



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	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
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	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

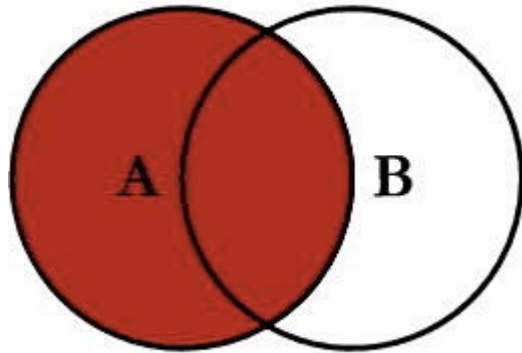
Full Outer Join

學號	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

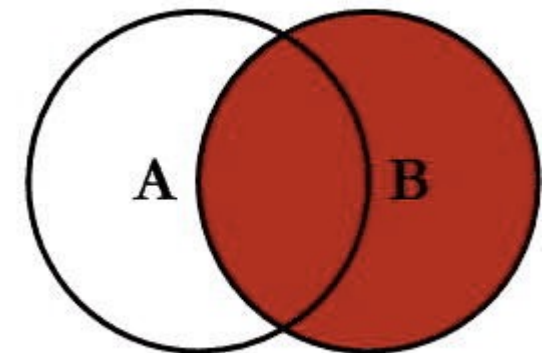
=

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
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

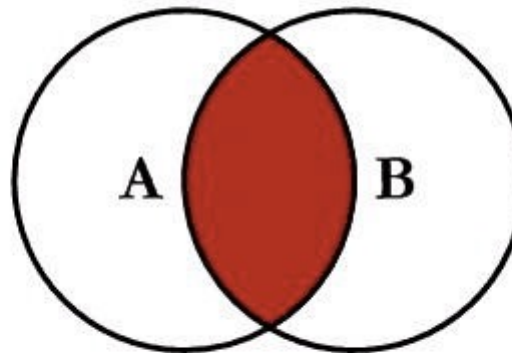
SQL JOINS



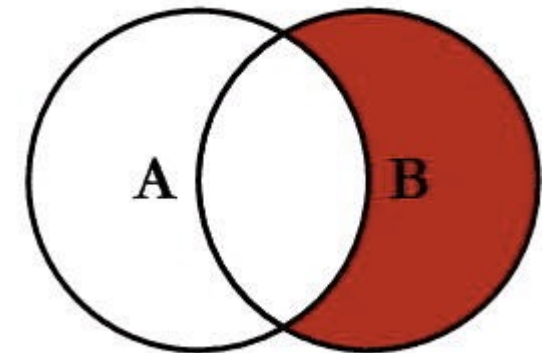
```
SELECT <select_list>
FROM TableA A
LEFT JOIN TableB B
ON A.Key = B.Key
```



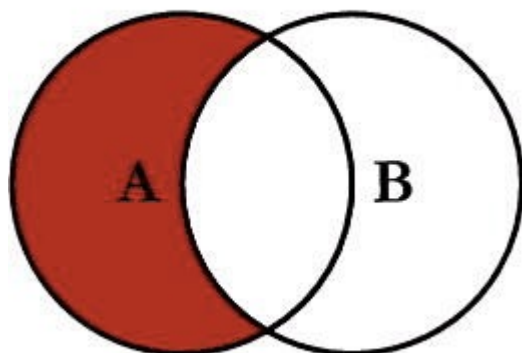
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
```



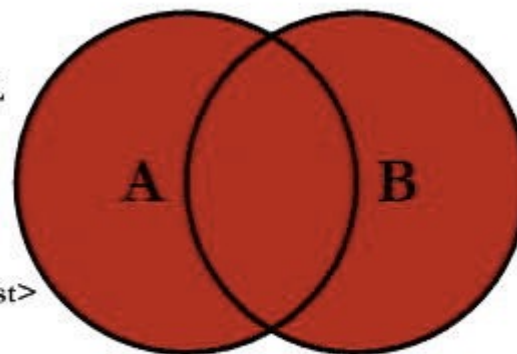
```
SELECT <select_list>
FROM TableA A
INNER JOIN TableB B
ON A.Key = B.Key
```



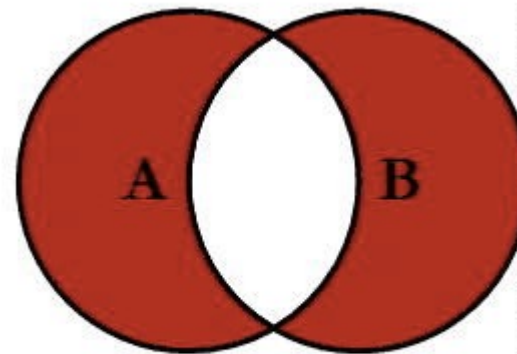
```
SELECT <select_list>
FROM TableA A
RIGHT JOIN TableB B
ON A.Key = B.Key
WHERE A.Key IS NULL
```



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

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 EMPLOYEE.SSN=DEPEDENT.ESSN) >= 2;
```

EMPLOYEE

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

DEPENDENT

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

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	<u>Ssn</u>	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

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

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

WORKS_ON

<u>Essn</u>	<u>Pno</u>	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	<u>Pnumber</u>	Plocation	Dnum
-------	----------------	-----------	------

WORKS_ON

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

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

Pname	<u>Pnumber</u>	...	<u>Essn</u>	<u>Pno</u>	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

Pname	<u>Pnumber</u>	...	<u>Essn</u>	<u>Pno</u>	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 (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

PROJECT

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

WORKS_ON

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

EMPLOYEE

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

```
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
-------	-------	-------	------------	-------	---------	-----	--------	-----------	-----

```
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

DEPARTMENT

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

EMPLOYEE

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



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

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>]

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

Update in SQL

Updates in SQL

- ◆ three SQL commands to modify the database
 - INSERT
 - DELETE
 - UPDATE

Updates in SQL (cont.)

- ◆ **E.g.** 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')
```

- ◆ The constraints specified in the DDL commands are automatically enforced by the DBMS when updates are applied to the database

U3A

- ◆ **Insertion of multiple tuples** resulting from a query into a relation
<E.g.> to create a temporary table that has the name, number of employees, and total salaries for each department.

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

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 ;
```

DELETE

- ◆ Removes tuples from a relation
- ◆ Includes a WHERE-clause to select the tuples to be deleted
- ◆ The number of tuples deleted depends on the number of tuples in the relation that satisfy the WHERE-clause
- ◆ 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
- ◆ **Referential integrity** should be enforced

U4A/U4B/U4C/U4D

**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

U5/U6

U5: 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
```

U6: 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' )
```

View in SQL

Relational Views in SQL

- A view is a single **virtual table** that is derived from other tables –
 - (1) **base tables** or
 - (2) previously defined **views**
- ◆ A view does **not necessarily exist** in physical form, which limits the possible update operations that can be applied to views
- ◆ There are no limitations on querying a view
- ◆ **CREATE VIEW** command specify a view by specifying a (virtual) table name and a defining query
- ◆ The view attribute names can be inherited from the attribute names of the tables in the defining query

Relational Views in SQL (cont.)

- ◆ One advantages of a view is to **simplify** the **specification** of queries.
- ◆ Views can also be used as a **security** and **authorization** mechanism
- ◆ DBMS responsible for keeping the view up-to-date if the base tables on which the view is defined are modified
- ◆ It is the responsibility of **DBMS**, not the user, to make sure that the view is **up to date**
- ◆ View is **not realized** at the time of **view definition**, but at the time we specify a query on the view

V1

- ◆ **CREATE VIEW WORKS_ON1 AS**
SELECT FNAME, LNAME, PNAME, HOURS
FROM EMPLOYEE, PROJECT, WORKS_ON
WHERE SSN=ESSN **AND** PNO=PNUMBER ;

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

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

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

QUERIES ON VIEWS

- ◆ Retrieve the last name and first name of all employees who work on 'ProjectX'.

```
QV1: SELECT  PNAME, FNAME, LNAME
          FROM    WORKS_ON1
          WHERE   PNAME='ProjectX' ;
```

```
* CREATE VIEW    WORKS_ON1 AS
  SELECT          FNAME, LNAME, PNAME, HOURS
  FROM            EMPLOYEE, PROJECT, WORKS_ON
  WHERE           SSN=ESSN AND PNO=PNUMBER ;
```

V2

**CREATE VIEW DEPT_INFO (DEPT_NAME, NO_OF_EMPS, TOTAL_SAL)
AS**

**SELECT DNAME, COUNT(*), SUM(SALARY)
FROM DEPARTMENT, EMPLOYEE
WHERE DNUMBER=DNO
GROUP BY DNAME ;**

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

}

Drop View

A view is removed using the **DROP VIEW** command

V1A: DROP VIEW WORKS_ON1 ;

V2A: DROP VIEW DEPT_INFO ;

View 是 Virtual Table ,
DBMS 如何實作 View 的功能 ?



View Implementation

- ◆ Two main approaches

- Query modification

- Modifying view query into a query on the underlying base tables
 - Disadvantage: inefficient for views of complex queries

- View materialization

- **Physically** creating a temporary view table
 - **Incremental update** view when base tables are updated
 - If view is not queried for a period of time, system automatically remove the physical view table

View Implementation (cont.)

- ◆ Query modification: modifying view query into a query on the underlying base tables

```
CREATE VIEW    WORKS_ON1 AS
SELECT FNAME, LNAME, PNAME, HOURS
FROM    EMPLOYEE, PROJECT, WORKS_ON
WHERE   SSN = ESSN AND PNO = PNUMBER ;

SELECT PNAME, FNAME, LNAME
FROM    WORKS_ON1
WHERE   PNAME = 'ProjectX' ;
```



```
SELECT PNAME, FNAME, LNAME
FROM    EMPLOYEE, PROJECT, WORKS_ON
WHERE   SSN = ESSN AND PNO = PNUMBER
        AND PNAME='ProjectX' ;
```

Updating of Views

- ◆ A view update operation may be mapped in multiple ways to update operations on the defining base relations - **ambiguity**
- ◆ updating views is still an active research area

Ex: To update the WORKS_ON1 view by modifying the PNAME attribute of 'John Smith' from 'ProductX' to 'ProductY'.

```
UV1:  UPDATE  WORKS_ON1  
      SET      PNAME = 'ProductY'  
      WHERE  LNAME='Smith' AND  
            FNAME='John' AND  
            PNAME='ProductX'
```

View Update 如何對應到其 Base Table 的 Update ?

John Smith 參與的計畫不變，只是改名為 Product Y
還是

John Smith 參與的計畫改變，改參與Product Y 計畫？



Updating of Views (cont.)

- ◆ Updates on the base relations to give the desired update on the view
- ◆ Two possibilities: (1) Change the name of the 'ProductX' tuple in the PROJECT relation to 'ProductY'

```
UPDATE   PROJECT
SET      PNAME = 'ProductY'
WHERE    PNAME = 'ProductX'
```

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

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

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

Updating of Views (cont.)

(2) Relate 'John Smith' to the 'ProductY' PROJECT tuple in place of the 'ProductX' PROJECT tuple

```
UPDATE   WORKS_ON
SETPNO = ( SELECT  PNUMBER
             FROM    PROJECT
             WHERE PNAME = 'ProductY' )
WHERE ESSN = ( SELECT SSN
               FROM    EMPLOYEE
               WHERE LNAME = 'Smith'
                   AND FNAME = 'John' )
AND PNO = ( SELECT PNUMBER
             FROM PROJECT
             WHERE PNAME = 'ProductX' )
```

Updating of Views (cont.)

- ◆ Some view updates may not make much sense

UV2: UPDATE DEPT_INFO

SET TOTAL_SAL = 100000

WHERE DNAME = 'Research' ;

- ◆ In general, we cannot guarantee that any view can be updated
- ◆ A view update is unambiguous only if **one update** on the base relations can accomplish the desired update effect on the view
- ◆ If a view update can be mapped to **more than one update** on the underlying base relations, we must have a certain procedure to choose the desired update

Updating of Views (cont.)

- ◆ A view with a single defining table is **updatable** if the view attributes contain the **primary key**
- ◆ Views defined on multiple tables using **joins** are generally **not updatable**
- ◆ Views defined using **grouping** and **aggregate** functions are **not updatable**

WITH

- ◆ 假設薪水最高的員工不只一位，列出所有這些員工的SSN。
 - **CREATE VIEW** Max_Salary(Value)
AS SELECT Max(Salary)
FROM Employee;

SELECT SSN
FROM Employee, Max_Salary
WHERE Employee.Salary = Max_Salary.Value;
 - **WITH** Max_Salary(Value)
AS SELECT Max(Salary)
FROM Employee
SELECT SSN
FROM Employee, Max_Salary
WHERE Employee.Salary = Max_Salary.Value;

Assertion

Constraint: salary of an employee must not be greater than the salary of the manager of the department that the employee works for

```
CREATE ASSERTION SALARY_CONSTRAINT  
CHECK (NOT EXISTS (SELECT * FROM EMPLOYEE E,  
                EMPLOYEE M, DEPARTMENT D  
                WHERE E.SALARY > M.SALARY AND  
                E.DNO=D.DNUMBER AND D.MGRSSN=M.SSN)));
```

```
CREATE DOMAIN D_NUM AS INTEGER  
CHECK (D_NUM > 0 AND D_NUM < 21);
```

Additional Features of SQL

Additional Features of SQL

- ◆ Granting, revoking of privileges to users to access certain relations
- ◆ **Embed** SQL in programming language C, C++, COBOL, PASCAL based on cursor
- ◆ Transaction control granularity for concurrency control, recovery
- ◆ Physical database design parameters, file structures for relations, access paths as indexes

Creating Indexes in SQL

- ◆ An SQL base relation generally corresponds to a stored file
- ◆ Statements can create and drop indexes on base relations
- ◆ These statements have been removed from SQL2 because they specify physical access paths - not logical concepts
- ◆ One or more indexing attributes are specified for each index
- ◆ **CREATE INDEX** statement is used
- ◆ Each index is given an *index name*

```
I1:      CREATE INDEX LNAME_INDEX
          ON  EMPLOYEE(LNAME);
```

Rank			DBMS	Database Model	Score		
Oct 2021	Sep 2021	Oct 2020			Oct 2021	Sep 2021	Oct 2020
1.	1.	1.	Oracle	Relational, Multi-model	1270.35	-1.19	-98.42
2.	2.	2.	MySQL	Relational, Multi-model	1219.77	+7.24	-36.61
3.	3.	3.	Microsoft SQL Server	Relational, Multi-model	970.61	-0.24	-72.51
4.	4.	4.	PostgreSQL	Relational, Multi-model	586.97	+9.47	+44.57
5.	5.	5.	MongoDB	Document, Multi-model	493.55	-2.95	+45.53
6.	6.	8.	Redis	Key-value, Multi-model	171.35	-0.59	+18.07
7.	7.	6.	IBM Db2	Relational, Multi-model	165.96	-0.60	+4.06
8.	8.	7.	Elasticsearch	Search engine, Multi-model	158.25	-1.98	+4.41
9.	9.	9.	SQLite	Relational	129.37	+0.72	+3.95
10.	10.	10.	Cassandra	Wide column	119.28	+0.29	+0.18
11.	11.	11.	Microsoft Access	Relational	116.38	-0.56	-1.87
12.	12.	12.	MariaDB	Relational, Multi-model	102.59	+1.90	+10.82
13.	13.	13.	Splunk	Search engine	90.61	-0.99	+1.21
14.	14.	15.	Hive	Relational	84.74	-0.83	+15.19
15.	15.	17.	Microsoft Azure SQL Database	Relational, Multi-model	79.72	+1.46	+15.32
16.	16.	16.	Amazon DynamoDB	Multi-model	76.55	-0.38	+8.14
17.	17.	14.	Teradata	Relational, Multi-model	69.83	+0.15	-5.96
18.	21.	64.	Snowflake	Relational	58.26	+6.19	+52.32
19.	18.	21.	Neo4j	Graph	57.87	+0.24	+6.53
20.	19.	19.	SAP HANA	Relational, Multi-model	55.28	-0.96	+1.04
21.	20.	23.	FileMaker	Relational	52.84	+0.52	+5.46
22.	22.	20.	Solr	Search engine, Multi-model	51.17	+1.36	-1.31
23.	23.	18.	SAP Adaptive Server	Relational, Multi-model	48.59	+1.57	-6.58
24.	24.	22.	HBase	Wide column	45.20	+0.14	-3.16
25.	25.	24.	Google BigQuery	Relational	43.79	-0.13	+9.38

Database Programming

- ◆ Approaches to DB programming
 - Embedding DB commands in a general-purpose programming language
 - Embedded SQL
 - Dynamic SQL
 - SQLJ
 - Using a library of DB functions (API)
 - SQL/CLI (Call Level Interface, ODBC)
 - JDBC
 - Designing a brand-new language
 - Oracle PL/SQL

Impedance Mismatch

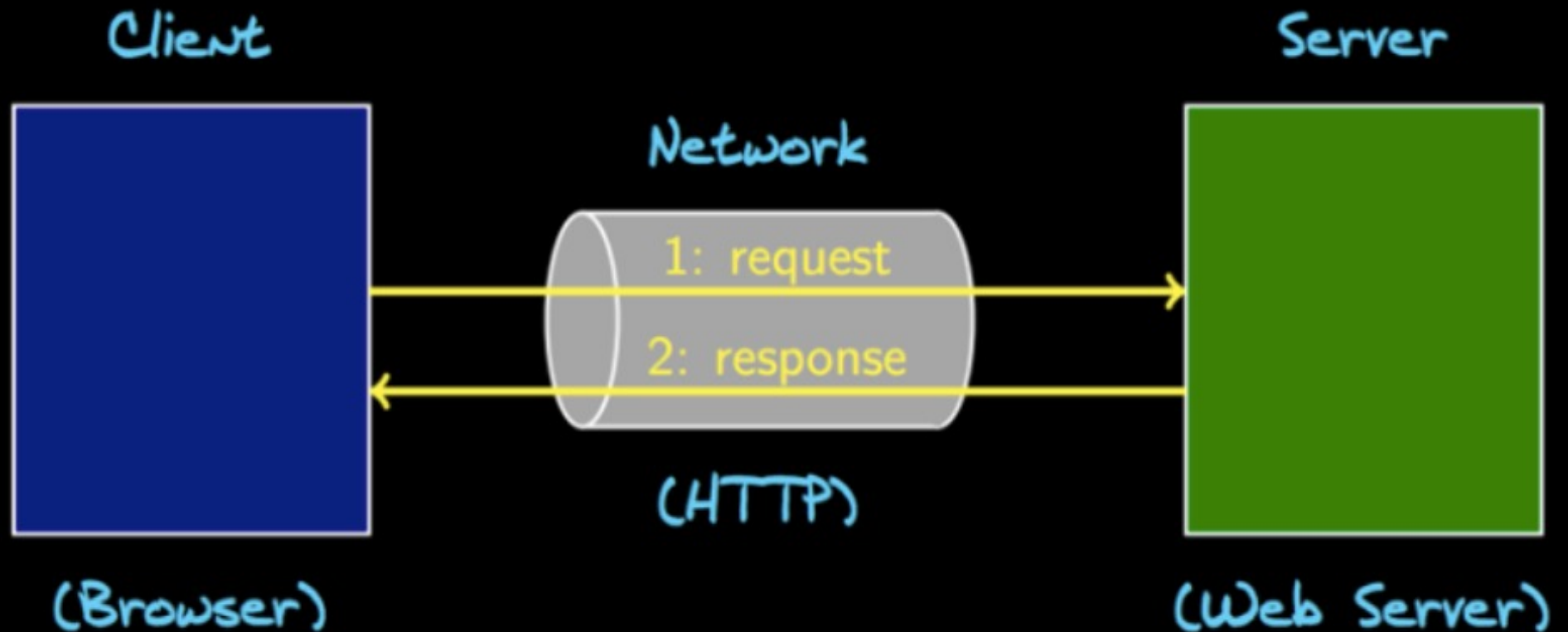
- ◆ Problem occurs because of difference between the DB model & PL model
 - Attribute data type vs. data type of PL: binding
 - Mapping between query result data structure & data structure in PL: cursor (iterator variable) is used to loop over the tuples in a query result
 - Impedance mismatch is less a problem when a special DB PL is designed.

Typical Sequence of Interaction in DB Programming

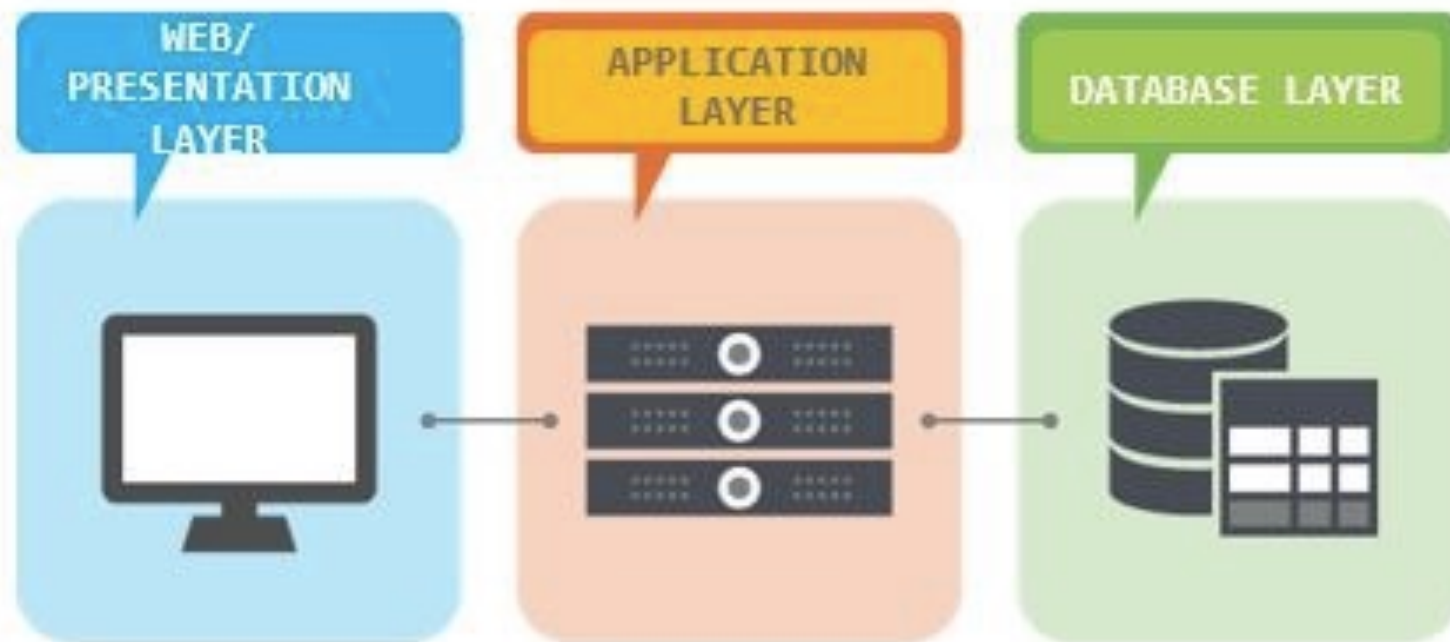
- ◆ **Client/server** model
 - Client program handles the logic of a software application
 - Client includes some calls to one or more database server to access or update data
 - **Front-end** vs. **Back-end**
- ◆ Common sequence
 - Client program establish or open a connection to the database server
 - Once the connection is established, program interact with the DB by submitting queries, updates or other commands
 - Close the connection

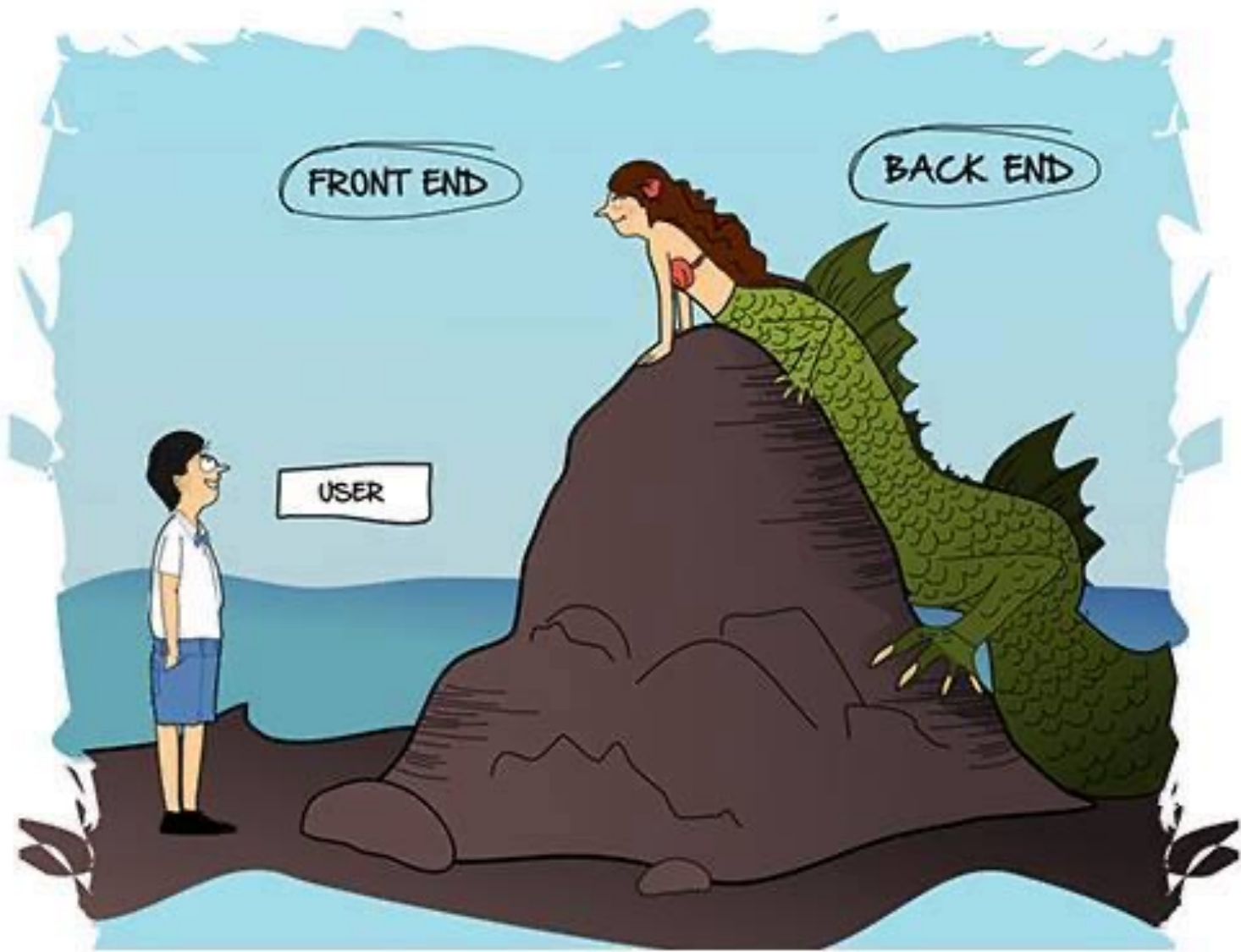
Client-Server

There is a request/response protocol associated with any client-server architecture:



Client-Server (cont.)





Front end vs. Back end.



Summary

- ◆ SQL: Tuple relational calculus + algebra
- ◆ DDL, DML, View, Indexing, ...
- ◆ Embedd SQL + Host language
- ◆ Syntax

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

SQL queries run in this order

FROM + JOIN



WHERE



GROUP BY



HAVING



SELECT (window functions
happen here !)



ORDER BY



LIMIT