BASIC SQL – PART II

Corresponding Reading: Chapter 6.3, 6.4

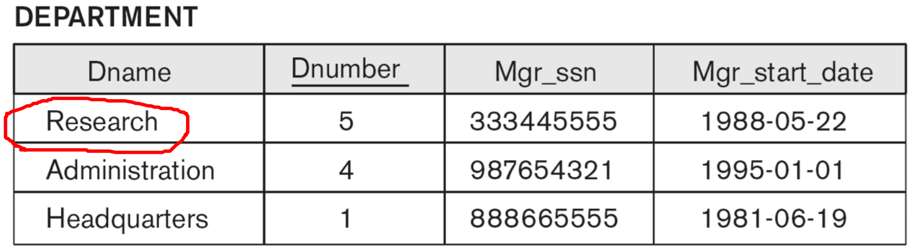
Example: SELECT-PROJECT-JOIN

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

```
SELECT      Fname, Lname, Address
FROM        EMPLOYEE, DEPARTMENT
WHERE       Dname='Research' and Dnumber=Dno;
```

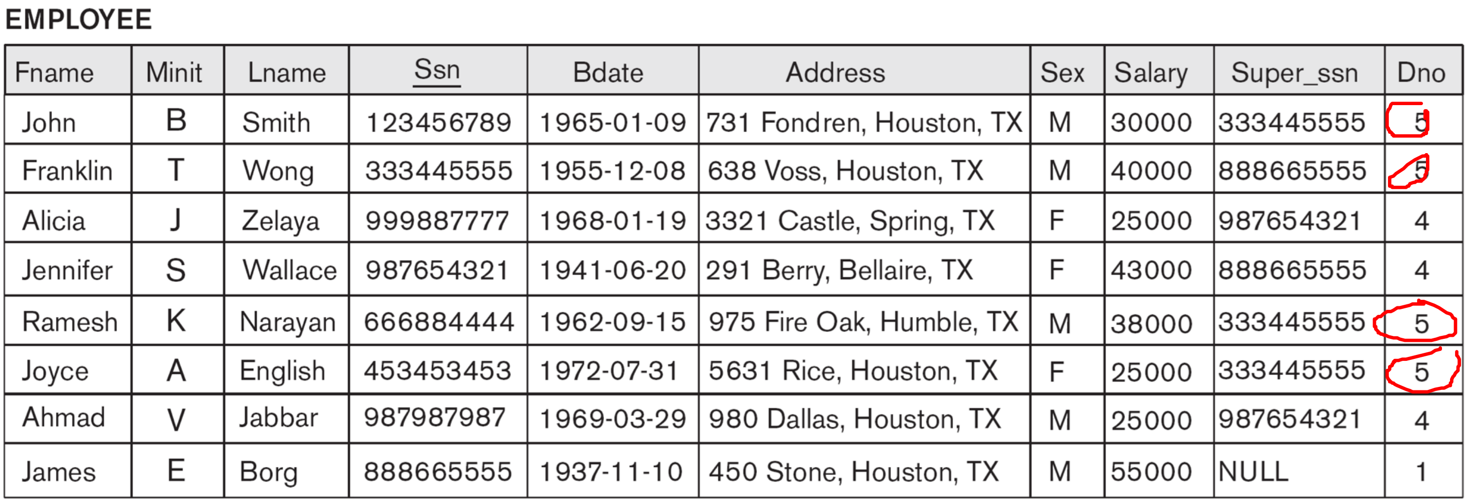
  - Dname='Research' is a selection condition that chooses the particular tuple of interest in the DEPARTMENT table, since Dname is an attribute of DEPARTMENT.
  - Dnumber=Dno is called the join condition since it combines two tuples: one from DEPARTMENT and one from EMPLOYEE.

Example Query: Tables and Output



DEPARTMENT

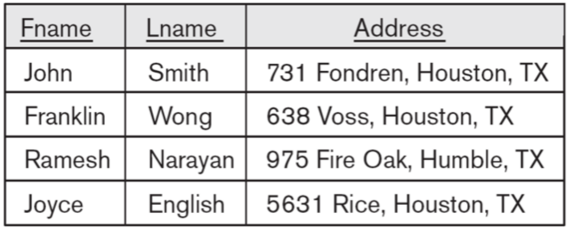
Dname	Dnumber	Mgr_ssn	Mgr_start_date
Research	5	333445555	1988-05-22
Administration	4	987654321	1995-01-01
Headquarters	1	888665555	1981-06-19



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

OUTPUT:



Fname	Lname	Address
John	Smith	731 Fondren, Houston, TX
Franklin	Wong	638 Voss, Houston, TX
Ramesh	Narayan	975 Fire Oak, Humble, TX
Joyce	English	5631 Rice, Houston, TX

Example: SELECT-PROJECT-JOIN (2 Joins)

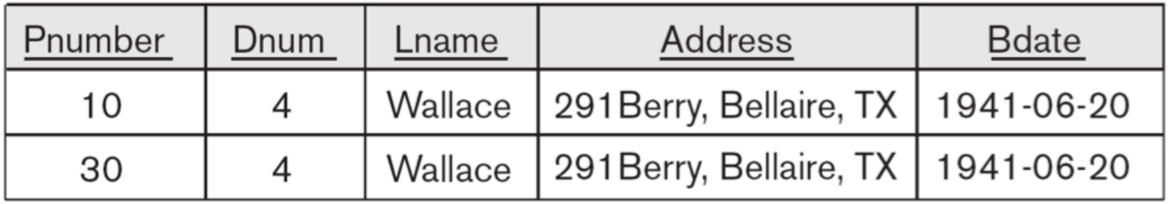
- For every project located in ‘Stafford’, list the project number, the controlling department number, and the department manager’s last name, address, and birth date.

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

- Each tuple in the result is a combination of one project, one department, and one employee that satisfies the join conditions.

Example: Output

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



Pnumber	Dnum	Lname	Address	Bdate
10	4	Wallace	291Berry, Bellaire, TX	1941-06-20
30	4	Wallace	291Berry, Bellaire, TX	1941-06-20

Ambiguous Attribute Names

- The same name can be used for two (or more) attributes as long as the attributes are in different relations.
- If a query refers to two or more attributes with the same name, we MUST qualify the attribute name with the relation name to prevent ambiguity.
- Prefix the relation name to the attribute name and separate with a period.
  - RELATION_A.attribute
  - RELATION_B.attribute

Example:

- List the names and birth dates for each employee and their dependents.

```
SELECT      EMPLOYEE.Fname, EMPLOYEE.Bdate,
            DEPENDENT.Dependent_name,
            DEPENDENT.Bdate
FROM        EMPLOYEE, DEPENDENT
WHERE       DEPENDENT.Essn=EMPLOYEE.Ssn
```

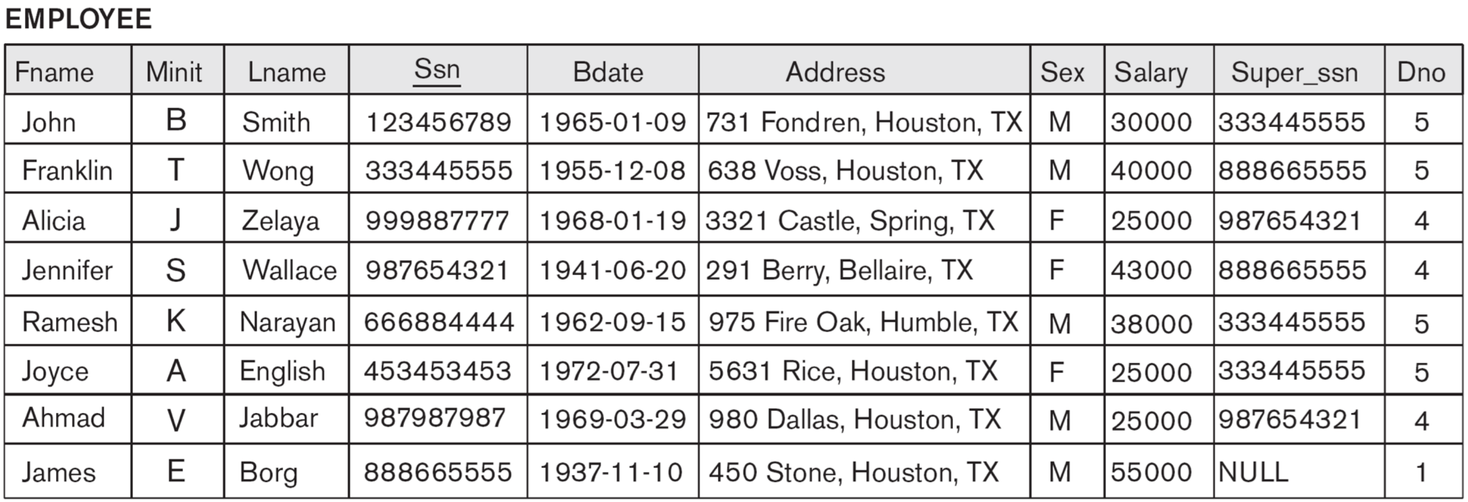
Aliases

- Ambiguity with attribute names also happens when a query refers to the same relation twice.
- Create an alias for each instance of the relation.

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

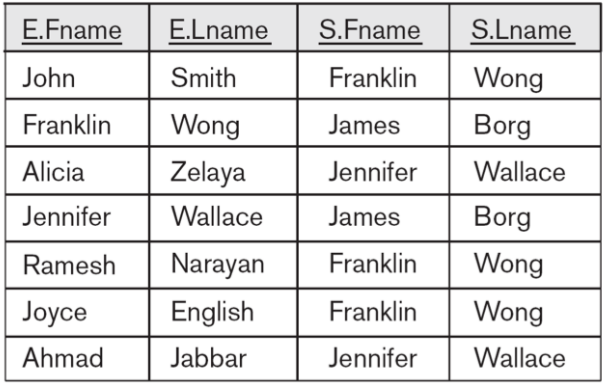
```
SELECT    E.Fname, E.Lname, S.Fname, S.Lname
FROM      EMPLOYEE AS E, EMPLOYEE AS S
WHERE     E.Super_ssn=S.Ssn;
```

Example: Table and Output



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



E.Fname	E.Lname	S.Fname	S.Lname
John	Smith	Franklin	Wong
Franklin	Wong	James	Borg
Alicia	Zelaya	Jennifer	Wallace
Jennifer	Wallace	James	Borg
Ramesh	Narayan	Franklin	Wong
Joyce	English	Franklin	Wong
Ahmad	Jabbar	Jennifer	Wallace

Aliases

- Both of the following queries perform the same action

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

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

Unspecified WHERE Clause

- A missing WHERE clause indicates no condition on tuple selection.

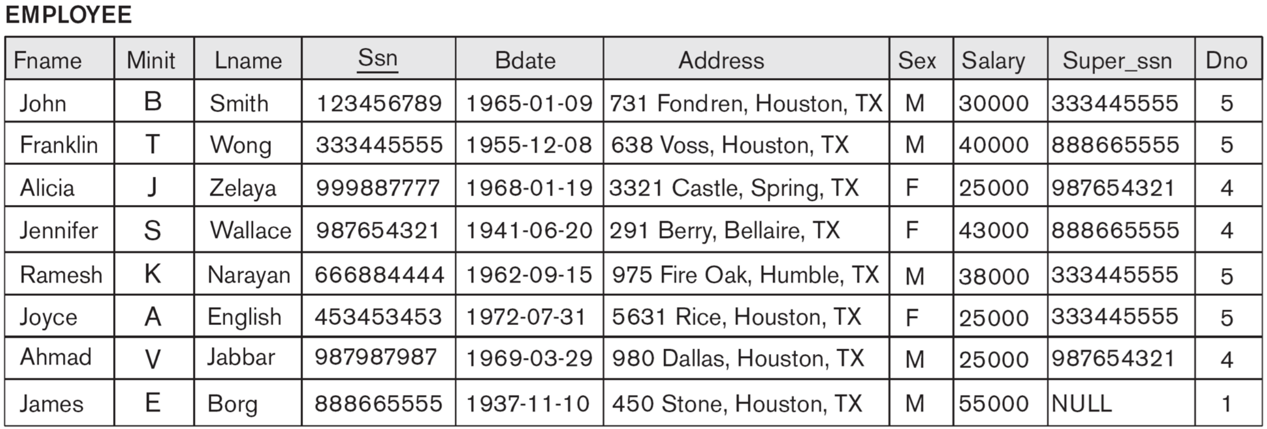
- ALL tuples of the relation specified in the FROM clause qualify and are selected for the query result.

- If more than one relation is specified in the FROM clause, then the **CROSS PRODUCT** (all possible tuple combinations) of these relations is selected.

Unspecified WHERE Clause

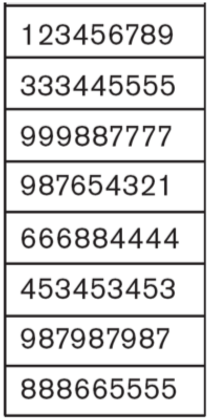
- Example: Select all Employee SSN numbers

```
SELECT      Ssn
FROM        EMPLOYEE;
```



EMPLOYEE

Fname	Minit	Lname	Ssn	Bdate	Address	Sex	Salary	Super_ssn	Dno
John	B	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5
Franklin	T	Wong	333445555	1955-12-08	638 Voss, Houston, TX	M	40000	888665555	5
Alicia	J	Zelaya	999887777	1968-01-19	3321 Castle, Spring, TX	F	25000	987654321	4
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellaire, TX	F	43000	888665555	4
Ramesh	K	Narayan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5
Joyce	A	English	453453453	1972-07-31	5631 Rice, Houston, TX	F	25000	333445555	5
Ahmad	V	Jabbar	987987987	1969-03-29	980 Dallas, Houston, TX	M	25000	987654321	4
James	E	Borg	888665555	1937-11-10	450 Stone, Houston, TX	M	55000	NULL	1

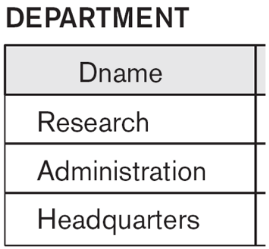


123456789
333445555
999887777
987654321
666884444
453453453
987987987
888665555

Unspecified WHERE Clause

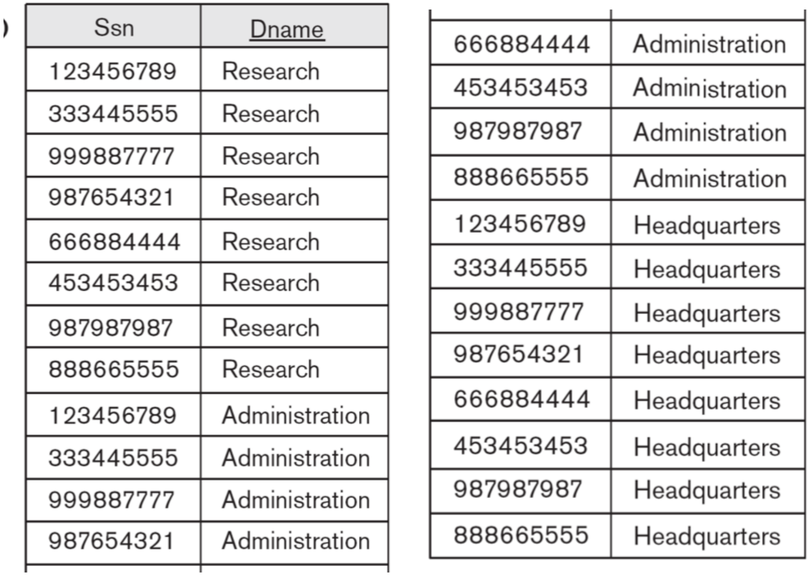
- Example: Select all combinations of an EMPLOYEE Ssn and DEPARTMENT Dname in the database.

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



DEPARTMENT

Dname
Research
Administration
Headquarters



Ssn	Dname
123456789	Research
333445555	Research
999887777	Research
987654321	Research
666884444	Research
453453453	Research
987987987	Research
888665555	Research
123456789	Administration
333445555	Administration
999887777	Administration
987654321	Administration
666884444	Administration
453453453	Administration
987987987	Administration
888665555	Administration
123456789	Headquarters
333445555	Headquarters
999887777	Headquarters
987654321	Headquarters
666884444	Headquarters
453453453	Headquarters
987987987	Headquarters
888665555	Headquarters

Unspecified WHERE Clause

- Extremely important to specify every selection/join condition in the WHERE clause.
- Otherwise incorrect and VERY large relations may result.
- If we were to specify all the attributes in the previous query, we see the actual CROSS PRODUCT.
  - If one relation has M rows and the other relation has N rows, the cross product would result in M*N rows.

Asterisk (*) in SQL Queries

- To retrieve all attribute values of selected tuples, we do not have to list each of the attribute names in the SELECT clause.   We just specify *

- Example: Select all attributes of any employee who works in the DEPARTMENT number 5.

```
SELECT      *
FROM        EMPLOYEE
WHERE       Dno=5;
```

Asterisk (*) in SQL Queries

- Example: Retrieve all attributes of an employee and the attributes of the DEPARTMENT in which they work for every employee of the Research department.

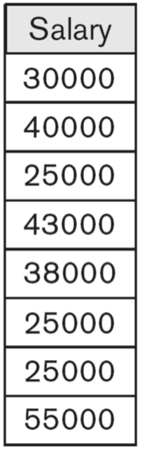
```
SELECT    *
FROM      EMPLOYEE, DEPARTMENT
WHERE     Dname='Research' AND
          Dno=Dnumber;
```

DISTINCT values

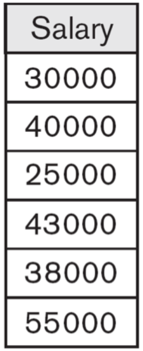
- SQL does not automatically remove duplicate tuples in the results of a query.
- If you want to eliminate duplicate tuples, specify DISTINCT in the SELECT clause.

```
SELECT   Salary
FROM     EMPLOYEE;
```

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



Salary
30000
40000
25000
43000
38000
25000
25000
55000



Salary
30000
40000
25000
43000
38000
55000

Comparison Operator - BETWEEN

- For convenience, SQL has a BETWEEN operator that allows us to check values between lower and bounds.

- Example: 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;
```

Same as: ((Salary>=30000) AND (Salary<=40000))

Ordering of Query Results

- SQL allows the user to order the tuples in the result of a query by the values of one or more attributes in the query result using the **ORDER BY** clause.
- Example:
  - Retrieve list of employees and projects they are working on
  - Ordered by department and:
    - Within each department, ordered alphabetically by last name, then first name.

Ordering of Query Results

- Example:

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

Ordering of Query Results

- Default ordering is in ASCENDING order of values.
- Specify the keyword **DESC** if you want descending order
- The keyword **ASC** can be used to explicitly specify ascending order.
- Example:

```
ORDER BY D.Dname DESC, E.Lname ASC, E.Fname ASC
```

Modifying the Database

- Three commands for modifying the database
  - INSERT
  - DELETE
  - UPDATE

INSERT Command

- **INSERT** is used to add a single tuple to a relation
- Specify **relation name** and a **list of values** for the tuple
- Values should be listed in the same order in which the corresponding attributes were specified in the CREATE TABLE command.
- Example:

```
INSERT INTO  EMPLOYEE
VALUES ('Richard','K','Marini','653298653','1962-12-30',
        '98 Oak Forest, Katy, TX', 'M', 37000,
        '653298653', 4);
```

INSERT – Specify attribute names

- Specify attribute names that correspond to the values provided in the INSERT command.
- Useful if a relation has many attributes and you only need to specify a few of these attributes.
- Be careful with NOT NULL and key constraints
- Example:

```
INSERT INTO EMPLOYEE (Fname, Lname, Dno, Ssn)
VALUES  (‘Richard’,’Marini’,4,’653298653’);
```

DELETE

- DELETE command removes tuples from a relation
- Includes a WHERE clause to select tuples to be deleted.
- Without a WHERE clause all tuples are deleted
- Examples:

```
DELETE FROM     EMPLOYEE
WHERE           Lname='Brown';

DELETE FROM     EMPLOYEE;
```

UPDATE

- Modifies attribute values of one or more selected tuples.
- WHERE clause selects the tuples to be modified
- **SET** clause specifies the attributes to be modified and their new values
- Example:

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