



## CSE 311L(Database Management System)

### LAB-Week 02

**Lab Instructor: Marilyn Dip Peris**

#### Topics:

- ▶ Basic SELECT Statement
- ▶ Selecting All Columns, Specific Columns
- ▶ Arithmetic Expressions, Using Arithmetic Operators, Parenthesis
- ▶ Defining a Column Alias

## BASIC QUERIES IN SQL

- SQL has one basic statement for retrieving information from a database; the SELECT statement
- This is *not the same as* the SELECT operation of the relational algebra
- Important distinction between SQL and the formal relational model;
- SQL allows a table (relation) to have two or more tuples that are identical in all their attribute values
- Hence, an SQL relation (table) is a *multi-set* (sometimes called a bag) of tuples; it is *not* a set of tuples
- SQL relations can be constrained to be sets by using the CREATE UNIQUE INDEX command, or by using the DISTINCT option
- Basic form of the SQL SELECT statement is called a *mapping* of a *SELECT-FROM-WHERE block*

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

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

## SIMPLE SQL QUERIES

Basic SQL queries correspond to using the following operations of the relational algebra:

```
SELECT  
PROJECT  
JOIN
```

### Example of a simple query on one relation (company2.sql)

#### Basic SELECT Statement

```
SELECT * | { [DISTINCT] column|expression [alias], ... }  
FROM table;
```

#### Arithmetic Operators

```
SELECT last_name, salary, 12*(salary+100)  
FROM emps;
```

#### Using Column Aliases

```
SELECT last_name "Name", salary*12 "Annual Salary"  
FROM emps;
```

#### Activity 01:

Write a query that displays the last name , weekly salary, department number of the employees. Name the salary column as "Weekly Salary".

**Run:** Populate the table with data given and running company.sql in the mysql prompt

All subsequent examples uses COMPANY database as shown below:

EMPLOYEE									
PNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
<b>DEPARTMENT</b>									
DNAME	DNUMBER	MGRSSN	MGRSTARTDATE						
<b>DEPT_LOCATIONS</b>									
DNUMBER	DLOCATION								
<b>PROJECT</b>									
PNAME	PNUMBER	PLOCATION	DNUM						
<b>WORKS_ON</b>									
ESSN	PNO	HOURS							
<b>DEPENDENT</b>									
ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP					

EMPLOYEE	FNAME	MINIT	LNAME	SSN	BDATE	ADDRESS	SEX	SALARY	SUPERSSN	DNO
John	I	Smith	123456789	1965-01-09	731 Fondren, Houston, TX	M	30000	333445555	5	
Franklin	T	Wong	333445555	1955-12-08	630 Voss, Houston, TX	M	40000	888888888	5	
Alicia	J	Zelaya	989887777	1960-07-19	3021 Castle, Spring, TX	F	25000	987654321	4	
Jennifer	S	Wallace	987654321	1941-06-20	291 Berry, Bellville, TX	F	43000	888888888	4	
Ramosh	K	Narsanyan	666884444	1962-09-15	975 Fire Oak, Humble, TX	M	38000	333445555	5	
Joyce	A	English	453453453	1972-07-31	5831 Rice, Houston, TX	F	25000	333445555	5	
Ahmed	V	Jabbar	987987987	1959-03-29	980 Dallas, Houston, TX	M	25000	987654321	4	
James	E	Borg	888888888	1937-11-10	450 Stone, Houston, TX	M	50000	null	1	

DEPARTMENT	DNAME	DNUMBER	MGRSSN	MGRSTARTDATE	DEPT_LOCATIONS	DNUMBER	DLOCATION
Research	5	333445555	1988-05-22		1	Houston	
Administration	4	987654321	1985-01-01		4	Stafford	
Headquarters	1	888888888	1981-06-19		5	Bellville	
					5	Sugarland	
					5	Houston	

WORKS_ON	ESSN	PNO	HOURS	PROJECT	PNAME	PNUMBER	PLOCATION	DNUM
123456789	1	32.5		ProductX	1	Bellville		5
123456789	2	7.5		ProductY	2	Sugarland		5
666884444	3	40.0		ProductZ	3	Houston		5
453453453	1	20.0		Computerization	10	Stafford		4
453453453	2	20.0		Reorganization	20	Houston		1
333445555	2	10.0		Newbenefits	30	Stafford		4
333445555	3	10.0						
333445555	10	10.0						
333445555	20	10.0						
989887777	30	30.0						
989887777	10	10.0						
987987987	10	35.0						
987987987	30	5.0						
987654321	90	20.0						
987654321	20	15.0						
888888888	20	null						

DEPENDENT	ESSN	DEPENDENT_NAME	SEX	BDATE	RELATIONSHIP
333445555		Alice	F	1988-04-05	DAUGHTER
333445555		Theodore	M	1983-10-25	SON
333445555		Joy	F	1988-05-03	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	1987-06-05	SPOUSE

### Example of a simple query on one relation

**Query 0: Retrieve the birth date and address of the employee whose name is 'John Smith'.**

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

The SELECT-clause specifies the projection attributes and the WHERE-clause specifies the selection condition However, the result of the query may contain duplicate tuples

### Example of a simple query on two relations

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

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

Similar to a SELECT-PROJECT-JOIN sequence of relational algebra operations (DNAME='Research') is a selection condition (corresponds to a SELECT operation in relational algebra) (DNUMBER=DNO) is a join condition (corresponds to a JOIN operation in relational algebra)

### Example of a simple query on three relations

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

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

In Q2, there are two join conditions The join condition DNUM=DNUMBER relates a project to its controlling department The join condition MGRSSN=SSN relates the controlling department to the employee who manages that department

### ALIASES, \* AND DISTINCT, EMPTY WHERE-CLAUSE

- In SQL, we can use the same name for two (or more) attributes as long as the attributes are in different relations
- A query that refers to two or more attributes with the same name must qualify the attribute

name with the relation name by prefixing the relation name to the attribute name **Example:**  
EMPLOYEE.LNAME, DEPARTMENT.DNAME

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

**Example**

**Query 3: For each employee, retrieve the employee's name, and the name of his or her immediate supervisor.**

Q3: SELECT E.FNAME, E.LNAME, S.FNAME, S.LNAME FROM EMPLOYEE AS E, EMPLOYEE AS S WHERE E.SUPERSSN=S.SSN

In Q3, the alternate relation names E and S are called aliases or tuple variables for the EMPLOYEE relation. We can think of E and S as two different copies of EMPLOYEE; E represents employees in role of supervisees and S represents employees in role of supervisors.

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

**UNSPECIFIED WHERE-clause**

A missing WHERE-clause indicates no condition; hence, all tuples of the relations in the FROM-clause are selected. This is equivalent to the condition WHERE TRUE

Example:

**Query 4: Retrieve the SSN values for all employees.**

Q4: SELECT SSN FROM EMPLOYEE

If more than one relation is specified in the FROM-clause and there is no join condition, then the CARTESIAN PRODUCT of tuples is selected

Example:

Q5: SELECT SSN, DNAME FROM EMPLOYEE, DEPARTMENT

**Note:** It is extremely important not to overlook specifying any selection and join conditions in the WHERE-clause; otherwise, incorrect and very large relations may result

**USE OF \***

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

Examples:

**Retrieve all the attribute values of EMPLOYEES who work in department 5.**

Q1a: SELECT \* FROM EMPLOYEE WHERE DNO=5

**Retrieve all the attributes of an employee and attributes of DEPARTMENT he works in for every employee of ‘Research’ department.**

Q1b: SELECT \* FROM EMPLOYEE, DEPARTMENT WHERE DNAME='Research'  
AND DNO=DNUMBER

### **USE OF DISTINCT**

SQL does not treat a relation as a set; duplicate tuples can appear. To eliminate duplicate tuples in a query result, the keyword DISTINCT is used

Example: the result of **Q1c** may have duplicate SALARY values whereas **Q1d** does not have any duplicate values

Q1c: SELECT SALARY FROM EMPLOYEE

Q1d: SELECT DISTINCT  
SALARY FROM EMPLOYEE

### **Activity 02:**

Find the results in SQL for these queries:

- 1) Find the first name and Last name of the employees who are supervised by “Franklin Wong”?
- 2) Find the last and first name of the female employees who have a dependent with the same first name as themselves?
- 3) For each department find out the department manager’s last name, his start date and the name his dependents (if any)?
- 4) For each employee find out the employee’s last and first name, the department name in which he works and the project name he works in and the number of hours he work in those projects.