CSIS05I

Database systems II

Lab (6)

DATA RETRIEVAL - COMPLEX QUERIES

Lab (6)

What is Querying?

Querying is retrieving specific information from a database by using a certain format of commands to provide one or more conditions for the retrieval.

Example 1: How many Employees work in the Research Department?

Example 2: What are the titles of all books written by Author X?

Writing a Query

This is the basic form of a query:

SELECT <Attribute List>
FROM <Table List>
WHERE <Condition>

<Attribute List>: is the list of attribute names whose values we want to retrieve from the database.
<Table List>: is a list of the relation(s) name(s) required to process the query.
<Condition>: is the condition specified by the user to retrieve relevant records ONLY.

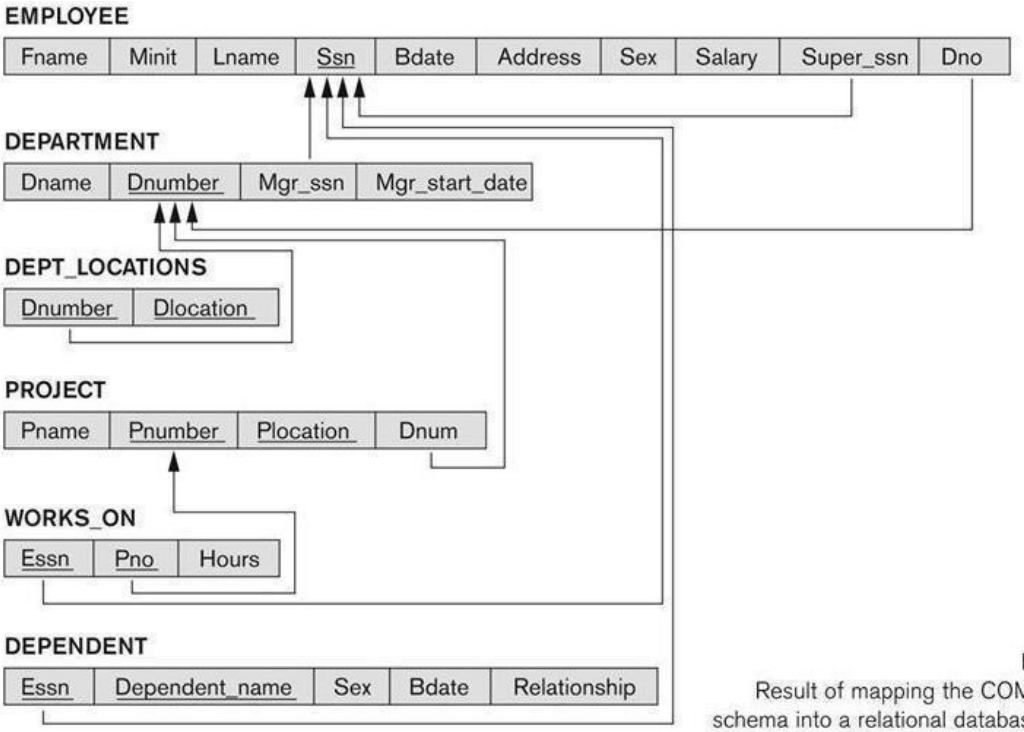


Figure 7.2
Result of mapping the COMPANY ER schema into a relational database schema.

1. Basic Query Examples

In this tutorial, we will retrieve data from multiple relations simultaneously. In order to retrieve data from more than one relation, we use the **JOIN** operator.

Example 1:

Retrieve the first and last names of the employees working in the HR department and their salaries are greater than 2000.

```
SELECT Fname, Lname
FROM Dept JOIN Employee
ON Dept.Dnumber = Employee.Dno
WHERE Dname = 'HR'
AND Salary > '2000';
```

Example 2:

Retrieve the first and last names of employees working in each department. If a department has no employees working in it, null values will be returned for the employee names.

```
SELECT Dname, Fname, Lname
FROM Dept left OUTER JOIN Employee
ON Dept.Dnumber = Employee.Dno;
```

	Dname	Fname	Lname
1	Research	Mohamed	Hassan
2	Research	Ahmed	Omar
3	Research	Aya	Yousef
4	HR	Noha	Ahmed
5	HR	Rana	Hamdy
6	PR	Nadia	Fahmy
7	IT	NULL	NULL
8	Marketing	NULL	NULL
9	Accounting	NULL	NULL

Example 3:

Retrieve the project name as well as the number of hours spent on each project. The project name shouldn't contain any null value.

```
SELECT Hrs, Pname
FROM Works_On right OUTER JOIN Project
ON Works_On.Pno = Project.Pnumber;
```

	Hrs	Pname
1	2	Proj101
2	5	Proj102
3	4	Proj102
4	4	Proj103
5	4	Proj103
6	3	Proj103
7	NULL	Proj104
8	NULL	Proj105
9	NULL	Proj106
10	NULL	Proj107

2. The IN keyword (or OR keyword)

Example 4:

*Retrieve all Employees whose first name is Ahmed **OR** Aya. Both queries provide the same result.*

```
SELECT *
FROM Employee
WHERE Fname IN ('Ahmed', 'Aya');
```

OR

```
SELECT *
FROM Employee
WHERE Fname= 'Ahmed' OR Fname= 'Aya';
```

Example 5:

The NOT IN keyword

Retrieve all Employees who do not have the first name is Ahmed or Aya. Every employee but those with names Ahmed or Aya will be retrieved.

```
SELECT *
FROM Employee
WHERE Fname NOT IN ('Ahmed', 'Aya');
```

3. Having Clause:

The user may wish to retrieve only the values produced by an aggregate function if they satisfy a certain criteria. In order to do so, the WHERE was used, but the problem is that WHERE can only handle individual tuples. As a result, in order to be able to add a condition to a query that produce a single result per set (Aggregate functions), we use the HAVING keyword. Having clause is used when making an aggregate condition.

Note that: *You **MUST** use the **GROUP BY** keyword when using the **HAVING** keyword.*

Example 6:

Retrieve the average salary and number of employees, grouped according to each department, but only retrieve the departments with average salary greater than 1000.

```
SELECT Dno, AVG(Salary) as 'average salary',
COUNT(*) as 'number of employees'
FROM Employee
GROUP BY Dno
HAVING AVG (Salary) > 1000;
```

	Dno	average salary	number of employees
1	1	2800	3
2	2	3430	2
3	3	2280	1

Note that: we used “As” clause to give an **alias** (another name) to the attributes retrieved. For example, when the first names of Employees in the Employee table are retrieved, we can change the 'Fname' attribute to 'First Name' in the output display.

4. Nested query

There are cases in which a user may wish to write a query, searching for a certain value or information that is not explicitly available in any of the relations. So, in order to get the information needed; another query must be executed first, hence the term 'Nested Queries'.

The **inner** query is used to get the unknown data and then the **outer** query uses that result in order to obtain the needed information that was initially requested by the user. In a nested query, a complete SELECT-FROM-WHERE block is written within the WHERE clause of another query.

Example 7:

Retrieve the unique pairs of the Project number(s) and the project name(s) of the project(s) in which Mohamed is working.

```
SELECT DISTINCT Pnumber, Pname
FROM Project
WHERE Pnumber IN ( SELECT Pno
                   FROM Works_On, Employee
                   WHERE Works_On.Essn= Employee.Ssn
                   AND Fname='Mohamed');
```

	Pnumber	Pname
1	101	Proj101
2	102	Proj102

Note that: Distinct keyword is used because in some cases, a query may return duplicate values. To prevent a value from appearing more than once, we use the **DISTINCT** keyword.

Example 8:

Retrieve the employee ID, first and last names of the employees who do not work in HR department, and order by first name and last name.

```
SELECT Ssn, Fname, Lname
FROM Employee
WHERE Dno NOT IN (SELECT Dnumber
                  FROM Dept
                  WHERE Dname = 'HR')
ORDER BY Fname, Lname;
```

Example 9:

Retrieve the first and last names of the employees who have the same first name and gender as a dependent.

```
SELECT e.Fname, e.Lname
FROM Employee e
WHERE e.Ssn IN(SELECT Essn
               FROM Dependent
               WHERE e.Fname= Dependent.Dependent_name
               AND e.Sex = Dependent.Sex);
```

	Fname	Lname
1	Aya	Yousef

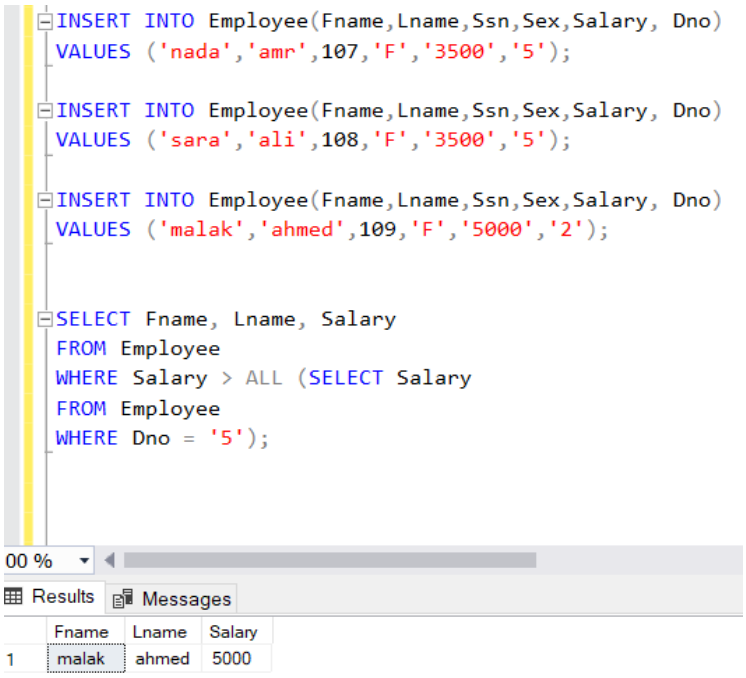
5. The ALL Keyword

ALL operator is used to select all tuples of SELECT STATEMENT. It is also used to compare a value to every value in another value set or result from a subquery.

Example 10:

Retrieve the first and last name of the employee whose salary is greater than that of the employees working in department number 5.

```
SELECT Fname, Lname, Salary
FROM Employee
WHERE Salary > ALL (SELECT Salary
                    FROM Employee
                    WHERE Dno = '5');
```



6. The EXISTS / NOT EXISTS Keywords

The EXISTS keyword is used in order to determine whether a value actually exists in the database before the rest of the query is carried out. It returns a Boolean value, either True or False. If the returned value is true, the rest of the query is carried out; otherwise the rest of the query is not executed and a null is returned (The EXISTS operator is used to test for the existence of any record in a subquery).

Example 11:

Retrieve the first name of employee who exists in the HR department.

```
SELECT Fname
FROM Employee
WHERE EXISTS (SELECT Dname
              FROM Dept
              WHERE Dnumber = Dno AND Dname = 'HR');
```

Example 12:

Retrieve the first and last names of department managers who have at least one dependent, and having Salary equal 3000.

```
SELECT Fname, Lname, Salary
FROM Employee
WHERE EXISTS (SELECT *
              FROM Dependent
              WHERE Essn = Ssn and Salary = 3000)
              AND
              EXISTS ( SELECT *
                       FROM Dept
                       WHERE Dept.Mgr_ssn =
                       Employee.Ssn);
```

Example 13:

Retrieve the first name of employees who do not exist in the HR department.

```
SELECT Fname
FROM Employee
WHERE not EXISTS(SELECT Dname
                 FROM Dept
                 WHERE Dnumber = Dno
                 AND Dname = 'HR');
```

Example 14:

*Retrieve the first and last names of all employees who have **no** dependents.*

```
SELECT Fname, Lname
FROM Employee
WHERE NOT EXISTS ( SELECT *
                   FROM Dependent
                   WHERE Essn = Ssn);
```

Exercises

1. Retrieve the names of the dependents that have employee relatives working in the research department.
2. Retrieve the maximum salary of the employees working on the project that is located in ‘Zamalek’.
3. Retrieve the name of the project having the maximum number of hours.
4. Retrieve the department names without duplication where the working employees have female dependents.
5. Retrieve the names of the employees who live in the same location as that of their project.