Advanced Database

Complex SQL Retrieval Queries

- SQL allows queries that check whether an attribute value is NULL. Rather than using = or < > to compare an attribute value to NULL, SQL uses the comparison operators **IS** or **IS NOT.**

Query 18. Retrieve the names of all employees who do not have supervisors.

- SELECT Fname, Lname
- FROM EMPLOYEE
- WHERE Super_ssn **IS** NULL

Complex SQL Retrieval Queries

- **Query 4. Make a list of all project numbers for projects that involve an employee whose last name is 'Smith', either as a worker or as a manager of the department that controls the project.**

```
    SELECT DISTINCT Pnumber
    FROM PROJECT
      WHERE Pnumber
IN
        ( SELECT Pnumber
        FROM PROJECT, DEPARTMENT, EMPLOYEE
        WHERE Dnum=Dnumber AND Mgr_ssn=Ssn AND Lname='Smith' )
OR
        Pnumber IN ( SELECT Pno
        FROM WORKS_ON, EMPLOYEE
         WHERE Essn=Ssn AND Lname='Smith' );
```

Complex SQL Retrieval Queries

This query will select the Essns of all employees who work the same (project, hours) combination on some project that employee ‘John Smith’ (whose Ssn = ‘123456789’) works on.

```
SELECT DISTINCT Essn
    FROM WORKS_ON
     WHERE (Pno, Hours)
IN
    ( SELECT Pno, Hours
     FROM WORKS_ON
     WHERE Essn=‘123456789’ );
```

Complex SQL Retrieval Queries

An example is the following query, which returns the names of employees whose salary is greater than the salary of all the employees in department 5:

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

Complex SQL Retrieval Queries

- **Query 16. Retrieve the name of each employee who has a dependent with the same first name and is the same sex as the employee.**

```
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 );
```

Complex SQL Retrieval Queries

- **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. For example, Q16 may be written as in Q16A:**

```
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;
```

The EXISTS Function in SQL

- We illustrate the use of EXISTS—and NOT EXISTS—with some examples. First, we formulate Query 16 in an alternative form that uses EXISTS as in Q16B:

```
SELECT E.Fname, E.Lname
FROM EMPLOYEE AS E
WHERE EXISTS ( SELECT *
               FROM DEPENDENT AS D
              WHERE E.Ssn=D.Essn AND E.Sex=D.Sex
              AND E.Fname=D.Dependent_name);
```

The EXISTS Function in SQL

- In general, EXISTS(Q) returns **TRUE** if there is at least one tuple in the result of the nested query Q, and it returns **FALSE** otherwise.

Query 6. Retrieve the names of employees who have no dependents.

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

Explicit Sets and Renaming of Attributes in SQL

Query 7. List the names of managers who have at least one dependent.

```
SELECT Fname, Lname
 FROM EMPLOYEE
 WHERE EXISTS ( SELECT *
                FROM DEPENDENT
                WHERE Ssn=Essn ) AND EXISTS
                ( SELECT *
                FROM DEPARTMENT
                WHERE Ssn=Mgr_ssn );
```

Explicit Sets and Renaming of Attributes in SQL

- **Q8A shows how query Q8 from the previous section can be slightly changed to retrieve the last name of each employee and his or her supervisor, while renaming the resulting attribute names as Employee_name and Supervisor_name.**

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

Explicit Sets and Renaming of Attributes in SQL

Query 17. Retrieve the Social Security numbers of all employees who work on project numbers 1, 2, or 3.

```
SELECT DISTINCT Essn
 FROM WORKS_ON
 WHERE Pno IN (1, 2, 3);
```

Aggregate Functions in SQL

- A number of built-in aggregate functions exist: **COUNT, SUM, MAX, MIN, and AVG.**
- These functions can be used in the SELECT clause or in a HAVING clause

Query 19. Find the sum of the salaries of all employees, the maximum salary, the minimum salary, and the average salary.

```
SELECT SUM (Salary), MAX (Salary), MIN (Salary), AVG (Salary)
 FROM EMPLOYEE;
```

Aggregate Functions in SQL

- **Query 20. Find the sum of the salaries of all employees of the ‘Research’ department, as well as the maximum salary, the minimum salary, and the average salary in this department.**

```
SELECT      SUM (Salary), MAX (Salary), MIN (Salary), AVG (Salary)
FROM         (EMPLOYEE JOIN DEPARTMENT ON Dno=Dnumber)
WHERE        Dname=‘Research’;
```

Aggregate Functions in SQL

- **Queries 21 and 22. 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’;
```

Aggregate Functions in SQL

- **Query 23. Count the number of distinct salary values in the database.**

```
SELECT   COUNT (DISTINCT Salary)
 FROM     EMPLOYEE;
```

Aggregate Functions in SQL

- For example, to retrieve the names of all employees who have two or more dependents (Query 5), we can write the following:

```
SELECT Lname, Fname
FROM EMPLOYEE
WHERE ( SELECT COUNT (*)
        FROM DEPENDENT
        WHERE Ssn=Essn ) >= 2;
```

The GROUP BY and HAVING Clauses

- **Query 24. 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;
```

The GROUP BY and HAVING Clauses

- Q25 shows how we can use a join condition in conjunction with GROUP BY. In this case, the grouping and functions are applied after the joining of the two relations.
- **Query 25. For each project, retrieve the project number, the project name, and the number of employees who work on that project**.

```
SELECT     Pnumber, Pname, COUNT (*)
FROM       PROJECT, WORKS_ON
WHERE      Pnumber=Pno
GROUP  BY    Pnumber, Pname;
```

The GROUP BY and HAVING Clauses

- **Query 26. For each project on which more than two employees work, retrieve the project number, the project name, and the number of employees who work on the project.**

```
SELECT        Pnumber, Pname, COUNT (*)
FROM          PROJECT, WORKS_ON
WHERE         Pnumber=Pno
GROUP BY      Pnumber, Pname
HAVING COUNT (*) > 2;
```

The GROUP BY and HAVING Clauses

- **Query 27. For each project, retrieve the project number, the project name, and the number of employees from department 5 who work on the project.**

```
SELECT      Pnumber, Pname, COUNT (*)
FROM        PROJECT, WORKS_ON, EMPLOYEE
WHERE       Pnumber=Pno AND Ssn=Essn AND Dno=5
GROUP BY    Pnumber, Pname;
```

The GROUP BY and HAVING Clauses

- **Query 28. For each department that has more than five employees, retrieve the department number and the number of its employees who are making more than $40,000.**

```
SELECT      Dnumber, COUNT (*)
FROM        DEPARTMENT, EMPLOYEE
WHERE       Dnumber=Dno AND Salary>40000 AND ( SELECT Dno
FROM        EMPLOYEE
GROUP BY    Dno
HAVING COUNT (*) > 5);
```