

Subqueries in WHERE

If the subquery returns only a single value (one column, one row), it can be used like a constant.

```
SELECT *
FROM EMPLOYEE
WHERE Ssn = ( SELECT Mgr_ssn
              FROM DEPARTMENT
              WHERE Dnumber=5 )
```

find the employee who is the manager of department 5

SQL Operations on Relations

operation	meaning	notes
EXISTS R	true if and only if R is not empty	
s IN R	true if and only if s is a row of R	
s > ALL R	true if and only if s is greater than every value in R	also >=, <, <=, =, <>
s > ANY R	true if and only if s is greater than at least one value in R	also >=, <, <=, =, <>

- can also prefix with NOT
 - NOT EXISTS R, s NOT IN R, NOT s > ALL R, etc

All of these operations work when R has only one column.
Many of them also make sense when R has multiple columns, if s has the same columns as R.

Correlated Subqueries

```
SELECT E.Fname, E.Lname, W.Pno, W.Hours
FROM EMPLOYEE E JOIN WORKS_ON W ON E.Ssn=W.Essn
WHERE W.Hours >= ALL ( SELECT Hours
                      FROM WORKS_ON W2
                      WHERE W2.Pno=W.Pno )
```

This is a *correlated subquery* – it must be evaluated for every row of the FROM relation because its result depends on the current row.

Compare to

```
SELECT *
FROM EMPLOYEE
WHERE Ssn = ( SELECT Mgr_ssn
              FROM DEPARTMENT
              WHERE Dnumber=5 )
```

The subquery will return the same value every time, so it could be evaluated once and then treated as a constant.

```
SELECT Fname, Lname, Ssn
FROM EMPLOYEE E
WHERE Sex IN ( SELECT Sex FROM DEPENDENT D WHERE E.Ssn=D.Essn )
```

find employees (first name, last name, ssn) who have a dependent of the same sex

```
SELECT Fname, Lname, Ssn, Pno, Hours
FROM EMPLOYEE E JOIN WORKS_ON W ON E.Ssn=W.Essn
WHERE W.Hours >= ALL ( SELECT Hours
                      FROM WORKS_ON W2
                      WHERE W2.Pno=W.Pno )
```

find the work assignments with the most hours for each project, and report the employee's name, ssn, project number, and hours

```
SELECT *
FROM EMPLOYEE E
WHERE NOT EXISTS ( SELECT *
                  FROM EMPLOYEE E2
                  WHERE E.Ssn <> E2.Ssn AND E.Dno = E2.Dno )
```

find the employees who are the only employee in their department i.e. those for which there are no other employees in the same department

Subqueries in FROM

You can also use a SELECT query to create a relation to be used in the FROM clause.

- the result must be given an alias

```
SELECT E.Fname, E.Lname, W.Pno, W.Hours
FROM EMPLOYEE E JOIN ( SELECT *
                        FROM WORKS_ON W
                        WHERE W.Hours >= ALL
                              ( SELECT Hours
                                FROM WORKS_ON W2
                                WHERE W2.Pno=W.Pno )
                        ORDER BY Hours DESC
                        LIMIT 3 ) W ON E.Ssn=W.Essn
```

of the work assignments with the most hours for each project, find the top 3 and report the employee's name, project number, and hours

Strategies for Writing Queries

Two strategies –

- start with FROM – build a table that contains all of the rows in the desired result and all of the columns you need for the WHERE and SELECT clauses
 - add WHERE to pick the desired rows
 - add SELECT to pick the desired columns
-
- start with FROM – build a table that contains all of the rows in the desired result and all of the columns that you need for the SELECT clause
 - add WHERE to pick the desired rows, using a subquery if the condition involves information not in the FROM table
 - add SELECT to pick the desired columns

Examples

- Find employees who earn more than the company-wide average salary.
- Find employees who are currently assigned to at least one project.
- Find employees who have no dependents on file.
- Find the other employees who work in the same department as employee Franklin Wong.
- Find employees who earn more than the average salary of their own department.
- Find employees who earn more than every employee in the Hardware department.
- List the department name and total project hours for departments whose employees have logged more than 100 hours combined across all their project assignments.
- Find departments whose average salary exceeds the company-wide average salary.
- For each department, find the employee(s) with the highest salary.
- Find employees who are assigned to more projects than the average number of projects per employee company-wide.