



Database Design & Applications

The Database Language - SQL Sub-Queries

Objectives

After do the completing this lesson, you should be able to following:

- Describe the types of problem that subqueries can solve
- Define subqueries
- List the types of subqueries
- Write single-row and multiple-row subqueries



Using a Subquery

Who has a salary greater than Abel's?

Main Query:



Which employees have salaries greater than Abel's salary?

Subquery



What is Abel's salary?

Subquery Syntax

```
SELECT      select_list  
FROM        table  
WHERE       expr operator
```

```
(SELECT      select_list  
FROM        table);
```

- The subquery (inner query) executes once before the main query.
- The result of the subquery is used by the main query (outer query).

Using a Subquery

```
SELECT last_name  
FROM employees  
WHERE salary >
```

11000

```
(SELECT salary  
FROM employees  
WHERE last_name = 'Abel');
```

LAST_NAME
King
Kochhar
De Haan
Hartstein
Higgins

Guidelines for Using Subqueries

- Enclose subqueries in parentheses.
- Place subqueries on the right side of the comparison condition.
- The ORDER BY clause in the subquery is not needed unless you are performing Top-N analysis.
- Use single-row operators with single-row subqueries and use multiple-row operators with multiple-row subqueries.



Types of Subqueries

- Single-row subquery



- Multiple-row subquery



Single-Row Subqueries

- Return only one row
- Use single-row comparison operators

Operator	Meaning
=	Equal to
>	Greater than
>=	Greater than or equal to
<	Less than
<=	Less than or equal to
<>	Not equal to

Single-Row Subqueries

```
SELECT last_name, job_id, salary
FROM   employee
WHERE  job_id =
      (SELECT job_id FROM
        employees
        WHERE  employee_id = 141)
AND    salary >
      (SELECT salary FROM
        employees
        WHERE  employee_id = 143);
```

Diagram illustrating the execution of the SQL query. Red arrows indicate the flow of data from the subqueries to the main query conditions:

- The first subquery `(SELECT job_id FROM employees WHERE employee_id = 141)` returns `ST_CLERK`, which is used to filter the `employee` table.
- The second subquery `(SELECT salary FROM employees WHERE employee_id = 143)` returns `2600`, which is used to filter the `salary` column.


LAST_NAME	JOB_ID	SALARY
Rajs	ST_CLERK	3500
Davies	ST_CLERK	3100

Using Group Functions in a Subquery

```
SELECT last_name, job_id, salary
FROM   employee
WHERE  salary =
```

2500

```
SELECT MIN(salary) FROM
employee);
```



LAST_NAME	JOB_ID	SALARY
Vargas	ST_CLERK	2500

Using HAVING Clause with Subqueries

- The MySQL server executes subqueries first.
- The MySQL server returns results into the HAVING clause of the main query.

```
SELECT      department_id, MIN(salary) employee  
FROM  
GROUP BY department_id
```

```
HAVING      MIN(salary)
```

>

```
(SELECT MIN(salary) FROM  
employee  
WHERE    department_id = 30);
```

2500

What is Wrong with this Statement?

- Single-row operator with multiple - row subquery

```
SELECT employee_id, last_name  
FROM   employee  
WHERE  salary =
```

```
(SELECT      MIN(salary)  
 FROM        employee  
  GROUP BY department_id);
```

subquery returns more than one row

Multiple-Row Subqueries

- Return more than one row
- Use multiple-row comparison operators

Operator	Meaning
IN	Equal to any member in the list
ANY	Compare value to each value returned by the subquery
ALL	Compare value to every value returned by the subquery

Using the ANY Operator in Multiple-Row Subqueries

```
SELECT employee_id, last_name, job_id, salary
FROM   employees
WHERE  salary < ANY
      (SELECT salary
       FROM employees
       WHERE job_id = 'IT_PROG')
AND    job_id <> 'IT_PROG';
```

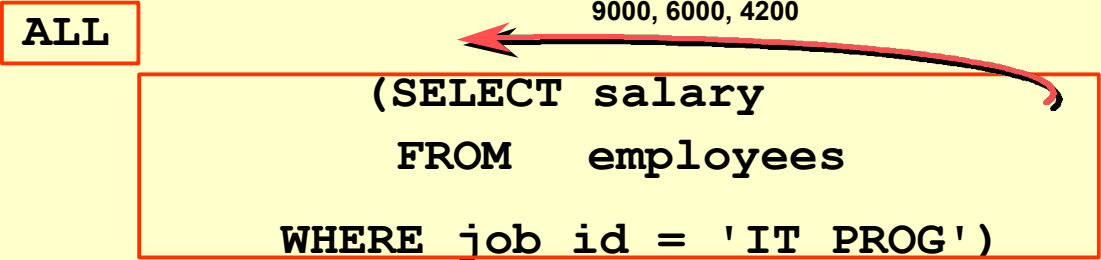
9000, 6000, 4200

EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
124	Mourgos	ST_MAN	5800
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

10 rows selected.

Using the ALL Operator in Multiple-Row Subqueries

```
SELECT employee_id, last_name, job_id, salary
FROM employees WHERE
    salary < ALL
    (SELECT salary
     FROM employees
     WHERE job_id = 'IT_PROG')
AND job_id <> 'IT_PROG';
```



EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
141	Rajs	ST_CLERK	3500
142	Davies	ST_CLERK	3100
143	Matos	ST_CLERK	2600
144	Vargas	ST_CLERK	2500

Null Values Returned by Subquery

Subquery returns no values

```
SELECT last_name, job_id FROM  
employees  
WHERE job_id =
```

```
(SELECT job_id  
FROM employees  
WHERE last_name = 'Haas');
```

0 row(s) returned

THANK YOU!