



Chapter 4

Structure Query Language

SQL - Sub-Queries

Department: Computer
Course: DBMS
Faculty: Sana Shaikh

Using Subqueries to Solve Queries

Using a Subquery to Solve a Problem

Who has a salary greater than Abel's?

Main query:



Which employees have salaries greater
than Abel's salary?



Using a Subquery to Solve a Problem

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

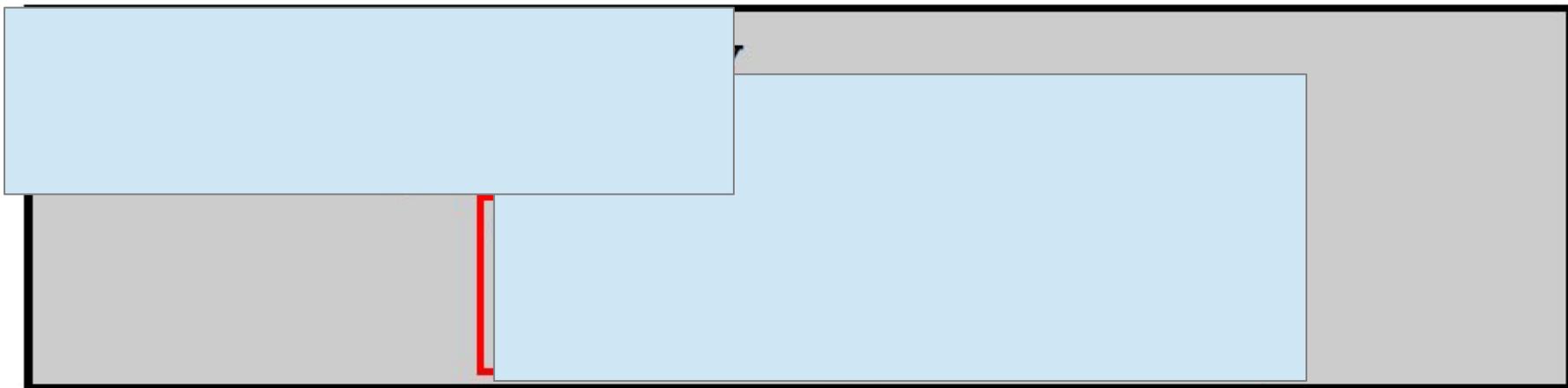
```
SELECT      select_list
FROM        table
WHERE       expr operator
            (SELECT      select_list
             FROM       table) ;
```

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

Using a Subquery



Which employees have salaries greater than Abel's salary?



...@gmail.com) has
not Guide

Using a Subquery



Which employees have salaries greater than Abel's salary?

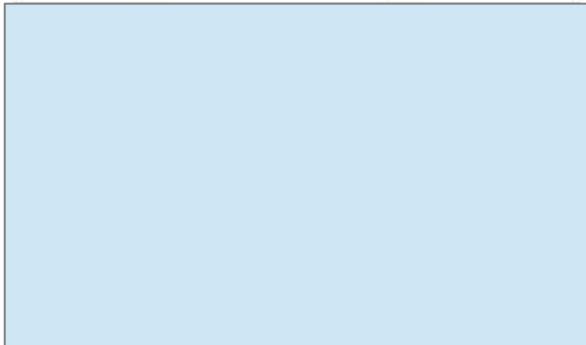
```
(SELECT salary  
FROM employees  
WHERE last name = 'Abel');
```

Using a Subquery



Which employees have salaries greater than Abel's salary?

```
SELECT last_name, salary  
FROM   employees    11000 ←  
WHERE  salary >  
       (SELECT salary  
        FROM   employees  
        WHERE  last_name = 'Abel');
```



dmal.com has
nt Guide

Using a Subquery



Which employees have salaries greater than Abel's salary?

```
SELECT last_name, salary  
FROM   employees    11000 ←  
WHERE  salary >  
       (SELECT salary  
        FROM   employees  
        WHERE  last_name = 'Abel');
```

	LAST_NAME	SALARY
1	Hartstein	13000
2	Higgins	12000
3	King	24000
4	Kochhar	17000
5	De Haan	17000

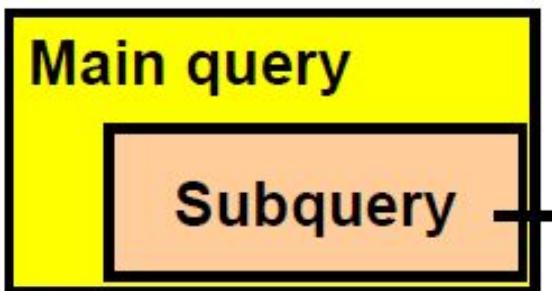
admail.com has
not Guide

Guidelines for Using Subqueries

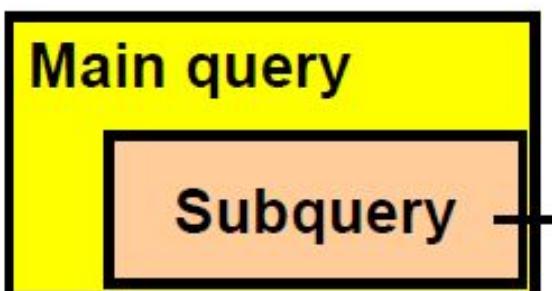
- Enclose subqueries in parentheses.
- Place subqueries on the right side of the comparison condition.
- Use single-row operators with single-row subqueries, and use multiple-row operators with multiple-row subqueries.

Types of Subqueries

- Single-row subquery



- Multiple-row subquery



Single-Row Subqueries

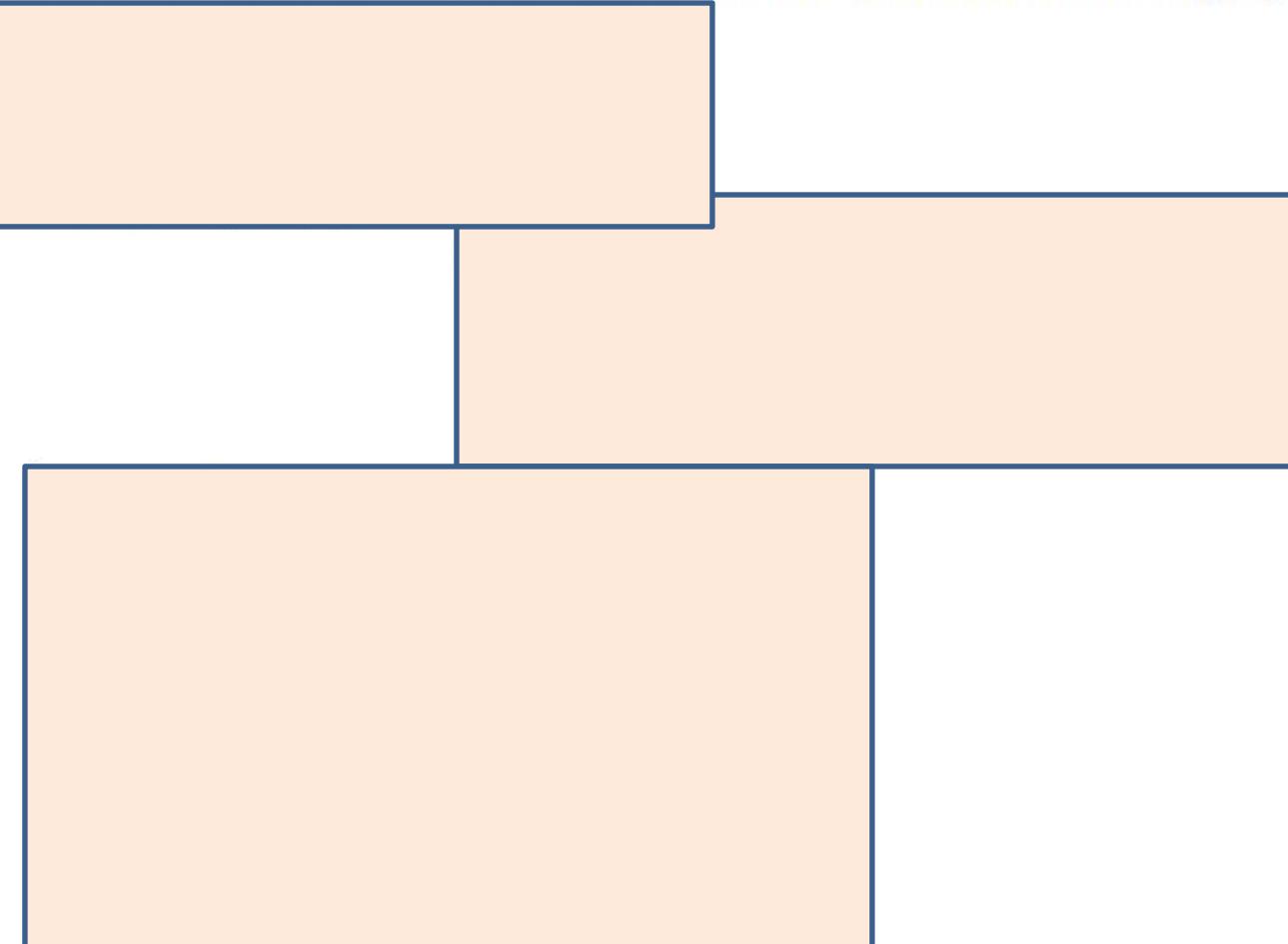
- Return only one row
- Use single-row comparison operators

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

Single-Row Subqueries

Example

Display the employees whose job ID is the same as that of employee 141:



Single-Row Subqueries

Example

Display the employees whose job ID is the same as that of employee 141:

```
(SELECT job_id  
  FROM employees  
 WHERE employee_id = 141);
```

Single-Row Subqueries

Example

Display the employees whose job ID is the same as that of employee 141:

```
SELECT last_name, job_id  
FROM   employees  
WHERE  job_id =  
       (SELECT job_id  
        FROM   employees  
        WHERE  employee_id = 141);
```

Single-Row Subqueries

Example

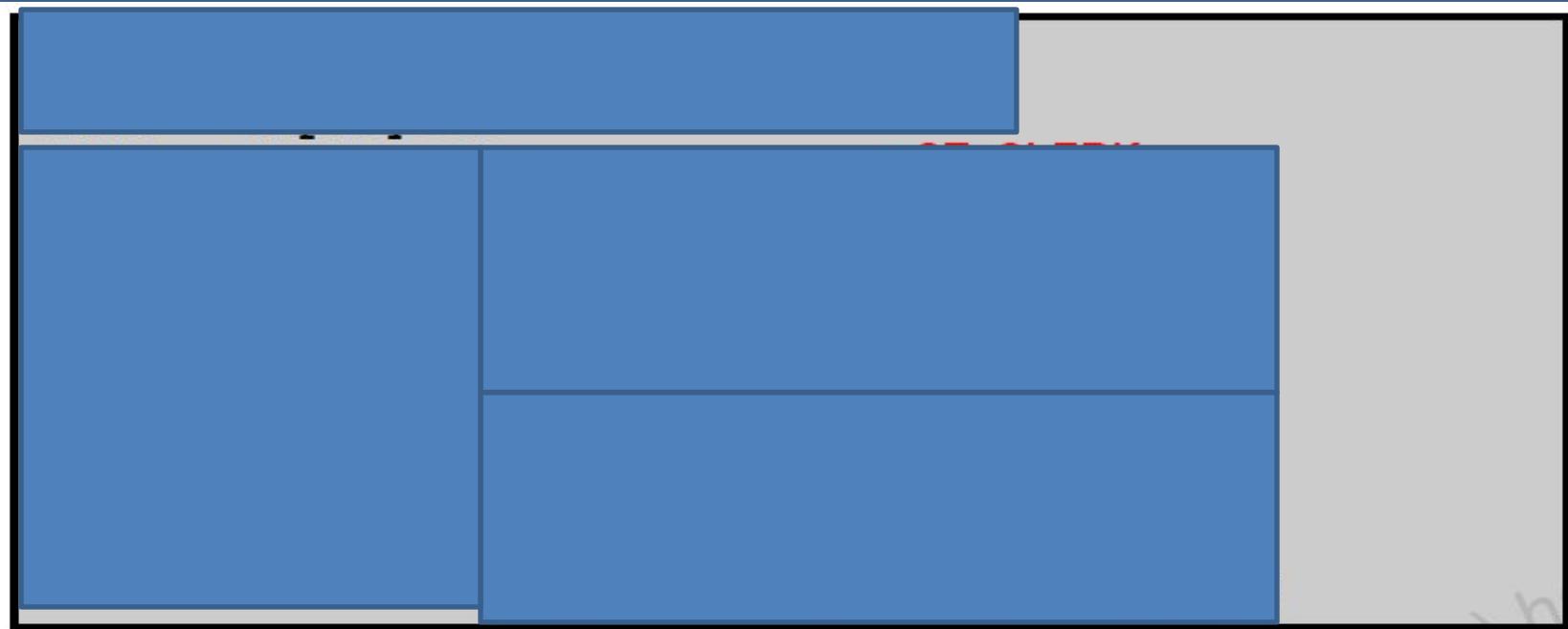
Display the employees whose job ID is the same as that of employee 141:

```
SELECT last_name, job_id  
FROM   employees  
WHERE  job_id =  
       (SELECT job_id  
        FROM   employees  
        WHERE  employee_id = 141);
```

	LAST_NAME	JOB_ID
1	Rajs	ST_CLERK
2	Davies	ST_CLERK
3	Matos	ST_CLERK
4	Vargas	ST_CLERK

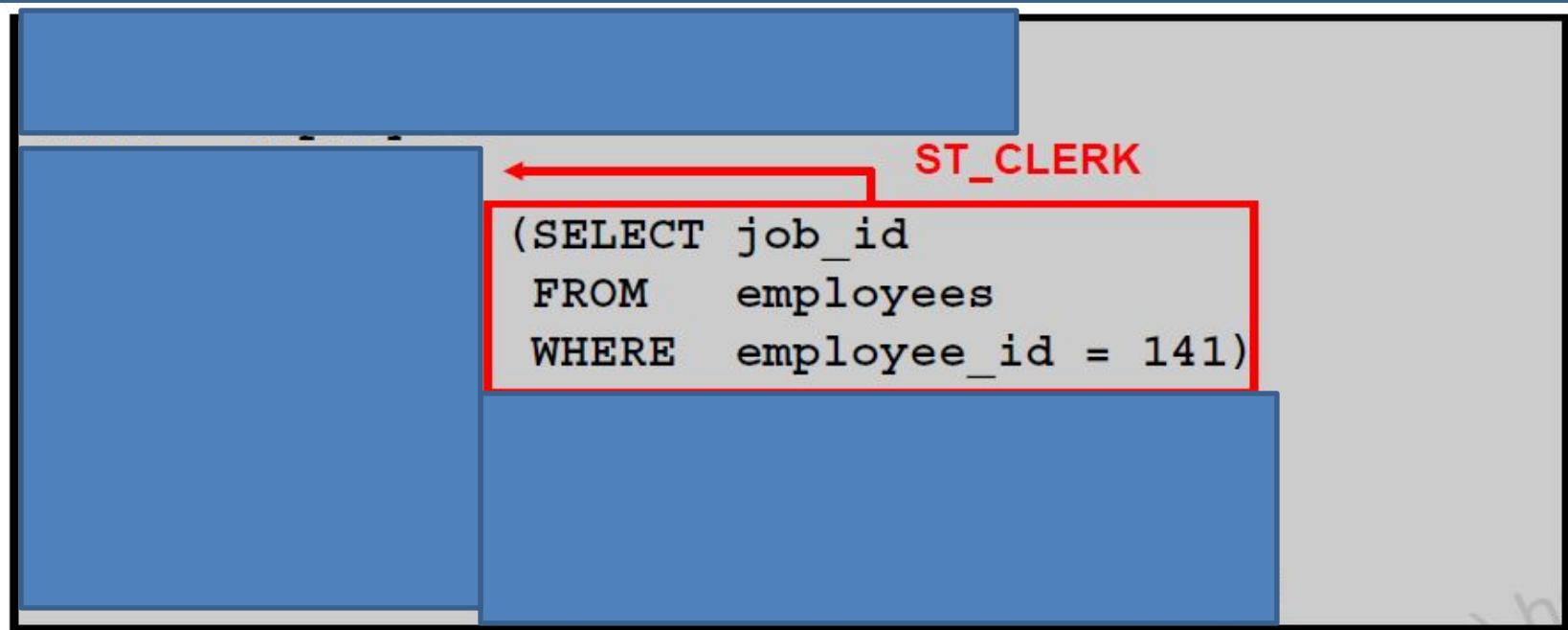
Executing Single-Row Subqueries

A SELECT statement can be considered as a query block. The example in the slide displays employees whose job ID is the same as that of employee 141 and whose salary is greater than that of employee 143.



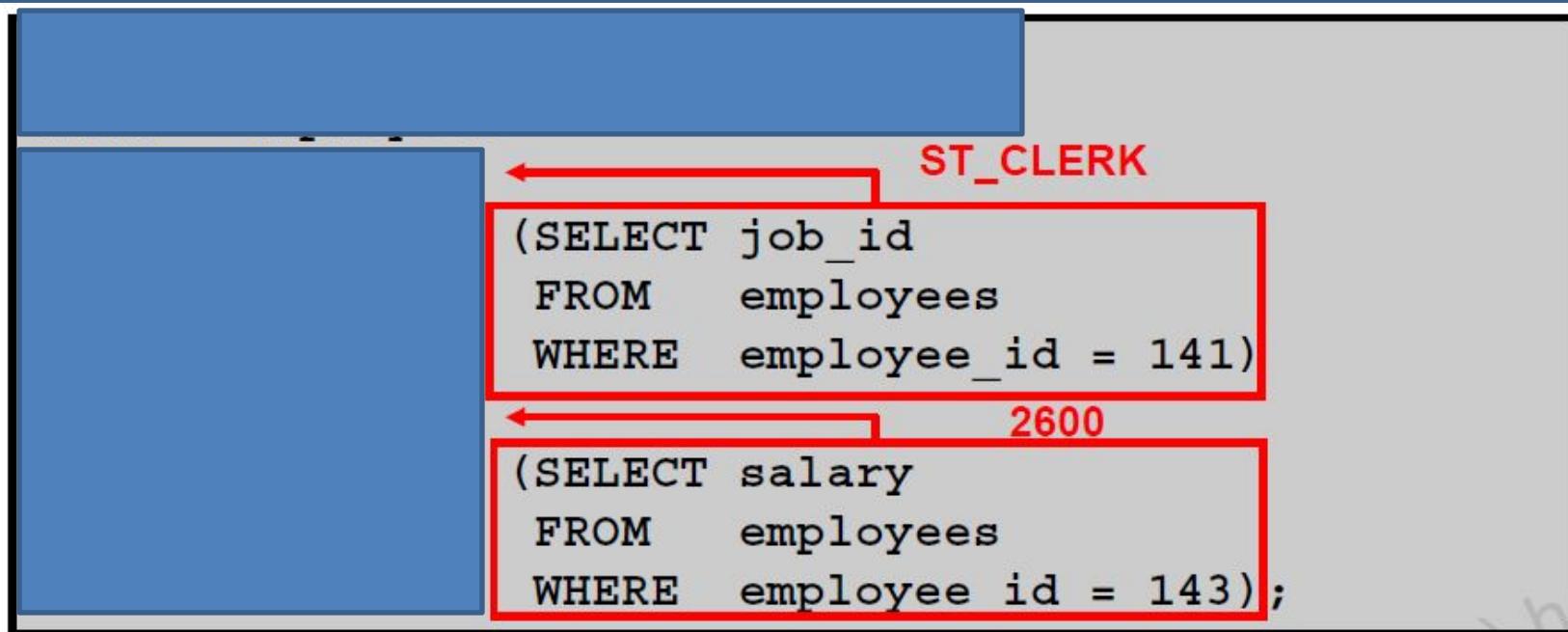
Executing Single-Row Subqueries

A SELECT statement can be considered as a query block. The example in the slide displays employees whose job ID is the same as that of employee 141 and whose salary is greater than that of employee 143.



Executing Single-Row Subqueries

A SELECT statement can be considered as a query block. The example in the slide displays employees whose job ID is the same as that of employee 141 and whose salary is greater than that of employee 143.



Executing Single-Row Subqueries

A SELECT statement can be considered as a query block. The example in the slide displays employees whose job ID is the same as that of employee 141 and whose salary is greater than that of employee 143.

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  job_id = ST_CLERK
       (SELECT job_id
        FROM   employees
        WHERE  employee_id = 141)
AND    salary > 2600
       (SELECT salary
        FROM   employees
        WHERE  employee_id = 143);
```

Executing Single-Row Subqueries

A SELECT statement can be considered as a query block. The example in the slide displays employees whose job ID is the same as that of employee 141 and whose salary is greater than that of employee 143.

```
SELECT last_name, job_id, salary
FROM   employees
WHERE  job_id = ST_CLERK
       (SELECT job_id
        FROM   employees
        WHERE  employee_id = 141)
AND    salary > 2600
       (SELECT salary
        FROM   employees
        WHERE  employee_id = 143);
```

	LAST_NAME	JOB_ID	SALARY
1	Rajs	ST_CLERK	3500
2	Davies	ST_CLERK	3100

Using Group Functions in a Subquery

The example in the slide displays the employee last name, job ID, and salary of all employees whose salary is equal to the minimum salary. The MIN group function returns a single value (2500) to the outer query.

Using Group Functions in a Subquery

```
SELECT last_name, job_id, salary  
FROM   employees ← 2500  
WHERE  salary =  
       (SELECT MIN(salary)  
        FROM   employees);
```

	LAST_NAME	JOB_ID	SALARY
1	Vargas	ST_CLERK	2500

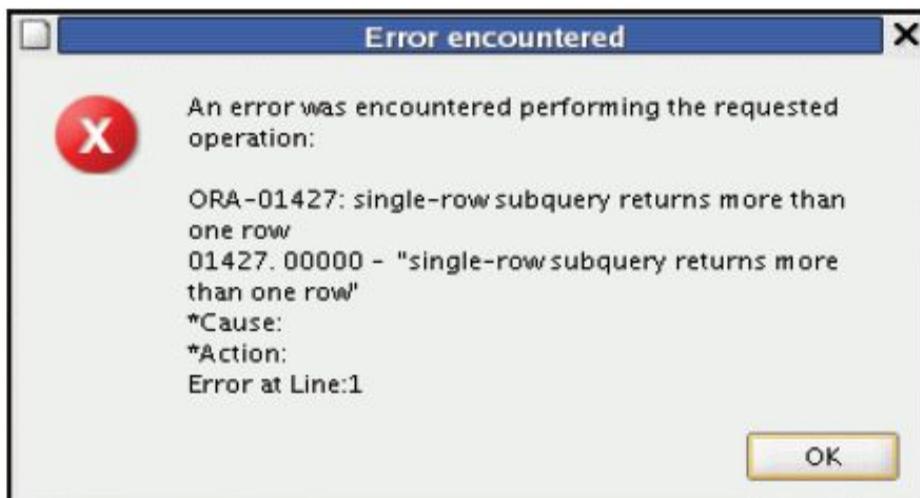
The example in the slide displays the employee last name, job ID, and salary of all employees whose salary is equal to the minimum salary. The MIN group function returns a single value (2500) to the outer query.

What Is Wrong with This Statement?

```
SELECT employee_id, last_name  
FROM   employees  
WHERE  salary =  
       (SELECT    MIN(salary)  
        FROM      employees  
        GROUP BY department_id);
```

What Is Wrong with This Statement?

```
SELECT employee_id, last_name  
FROM   employees  
WHERE  salary =  
       (SELECT    MIN(salary)  
        FROM      employees  
        GROUP BY department_id);
```



Single-row operator with multiple-row subquery

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

Multiple-Row Subqueries

Subqueries that return more than one row are called multiple-row subqueries. You use a multiple-row operator, instead of a single-row operator, with a multiple-row subquery. The multiple-row operator expects one or more values:

Example

Find the employees who earn the same salary as the minimum salary for each department.

Multiple-Row Subqueries

Subqueries that return more than one row are called multiple-row subqueries. You use a multiple-row operator, instead of a single-row operator, with a multiple-row subquery. The multiple-row operator expects one or more values:

```
SELECT last_name, salary, department_id  
FROM employees  
WHERE salary IN (SELECT MIN(salary)  
                  FROM employees  
                  GROUP BY department_id);
```

Example

Find the employees who earn the same salary as the minimum salary for each department.

Multiple-Row Subqueries

Subqueries that return more than one row are called multiple-row subqueries. You use a multiple-row operator, instead of a single-row operator, with a multiple-row subquery. The multiple-row operator expects one or more values:

```
SELECT last_name, salary, department_id  
FROM   employees  
WHERE  salary IN (SELECT    MIN(salary)  
                   FROM      employees  
                   GROUP BY department_id);
```

Example

Find the employees who earn the same salary as the minimum salary for each department. The inner query is executed first, producing a query result. The main query block is then processed and uses the values that were returned by the inner query to complete its search condition. In fact, the main query appears to the Oracle server as follows:

```
SELECT last_name, salary, department_id  
FROM   employees  
WHERE  salary IN (2500, 4200, 4400, 6000, 7000, 8300,  
                  8600, 17000);
```

Multiple-Row Subqueries (continued)

The ANY operator (and its synonym, the SOME operator) compares a value to *each* value returned by a subquery. The slide example displays employees who are not IT programmers and whose salary is less than that of any IT programmer. The maximum salary that a programmer earns is \$9,000.

<ANY means less than the maximum. >ANY means more than the minimum. =ANY is equivalent to IN.

Using the ANY Operator in Multiple-Row Subqueries

```
SELECT employee_id, last_name, job_id, salary
FROM   employees      9000, 6000, 4200
WHERE  salary < ANY
       (SELECT salary
        FROM   employees
        WHERE  job_id = 'IT_PROG')
AND    job_id <> 'IT_PROG';
```

	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	144	Vargas	ST_CLERK	2500
2	143	Matos	ST_CLERK	2600
...				
9	206	Gietz	AC_ACCOUNT	8300
10	176	Taylor	SA_REP	8600

Multiple-Row Subqueries (continued)

The ALL operator compares a value to *every* value returned by a subquery. The slide example displays employees whose salary is less than the salary of all employees with a job ID of IT_PROG and whose job is not IT_PROG.

>ALL means more than the maximum, and <ALL means less than the minimum.

The NOT operator can be used with IN, ANY, and ALL operators.

Using the ALL Operator in Multiple-Row Subqueries

```
SELECT employee_id, last_name, job_id, salary
FROM   employees      9000, 6000, 4200
WHERE  salary < ALL
          (SELECT salary
           FROM   employees
           WHERE  job_id = 'IT_PROG')
AND    job_id <> 'IT_PROG';
```

	EMPLOYEE_ID	LAST_NAME	JOB_ID	SALARY
1	141 Rajs	ST_CLERK	3500	
2	142 Davies	ST_CLERK	3100	
3	143 Matos	ST_CLERK	2600	
4	144 Vargas	ST_CLERK	2500	

Discussion Questions



Exercise Questions

2. Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in ascending order by salary.

3. Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains a *u*. Place your SQL statement in a text file named `lab_06_03.sql`. Run your query.

Exercise Questions

2. Create a report that displays the employee number, last name, and salary of all employees who earn more than the average salary. Sort the results in ascending order by salary.

```
SELECT employee_id, last_name, salary  
FROM   employees  
WHERE  salary > (SELECT AVG(salary)  
                  FROM   employees)  
ORDER BY salary;
```

3. Write a query that displays the employee number and last name of all employees who work in a department with any employee whose last name contains a *u*. Place your SQL statement in a text file named lab_06_03.sql. Run your query.

```
SELECT employee_id, last_name  
FROM   employees  
WHERE  department_id IN (SELECT department_id  
                         FROM   employees  
                         WHERE  last_name like '%u%');
```

4. The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

Modify the query so that the user is prompted for a location ID. Save this to a file named lab_06_04.sql.

4. The HR department needs a report that displays the last name, department number, and job ID of all employees whose department location ID is 1700.

```
SELECT last_name, department_id, job_id  
FROM   employees  
WHERE  department_id IN (SELECT department_id  
                           FROM   departments  
                           WHERE  location_id = 1700);
```

Modify the query so that the user is prompted for a location ID. Save this to a file named lab_06_04.sql.

```
SELECT last_name, department_id, job_id  
FROM   employees  
WHERE  department_id IN (SELECT department_id  
                           FROM   departments  
                           WHERE  location_id = &Enter_location);
```

5. Create a report for the HR department that displays the last name and salary of every employee who reports to King.

6. Create a report for the HR department that displays the department number, last name, and job ID for every employee in the Executive department.

5. Create a report for the HR department that displays the last name and salary of every employee who reports to King.

```
SELECT last_name, salary  
FROM employees  
WHERE manager_id = (SELECT employee_id  
                      FROM employees  
                      WHERE last_name = 'King');
```

6. Create a report for the HR department that displays the department number, last name, and job ID for every employee in the Executive department.