The Advanced Sub Queries

- Jayendra Khatod

Objectives

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

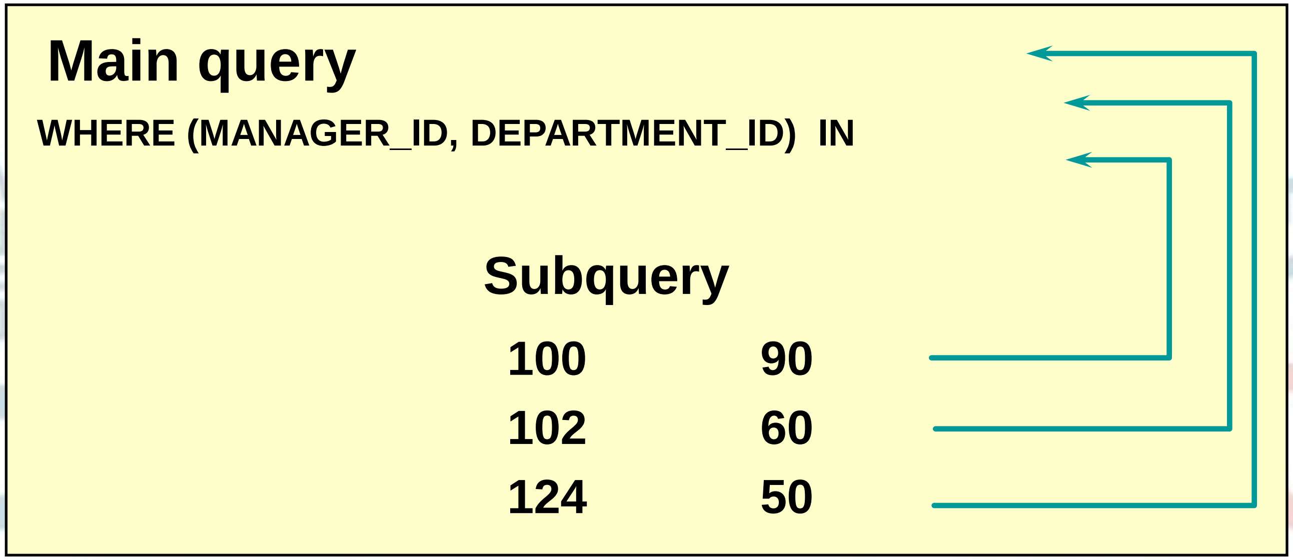
- **Write a multiple-column subquery**
- **Describe and explain the behavior of subqueries when null values are retrieved**
- **Write a subquery in a `FROM` clause**
- **Use scalar subqueries in SQL**
- **Describe the types of problems that can be solved with correlated subqueries**
- **Write correlated subqueries**
- **Update and delete rows using correlated subqueries**
- **Use the `EXISTS` and `NOT EXISTS` operators**
- **Use the `WITH` clause**

Subqueries

```
SELECT    select_list
FROM  table
WHEREexpr 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).**

Multiple-Column Subqueries



Each row of the main query is compared to values from a multiple-row and multiple-column subquery.

Pairwise Comparison Subquery

Display the details of the employees who are managed by the same manager *and* work in the same department as the employees with EMPLOYEE_ID 178 or 174.

```
SELECT       employee_id, manager_id, department_id
FROM   employees
WHERE   (manager_id, department_id) IN
                              (SELECT manager_id, department_id
                               FROM    employees
                               WHERE   employee_id IN (178,174))
```

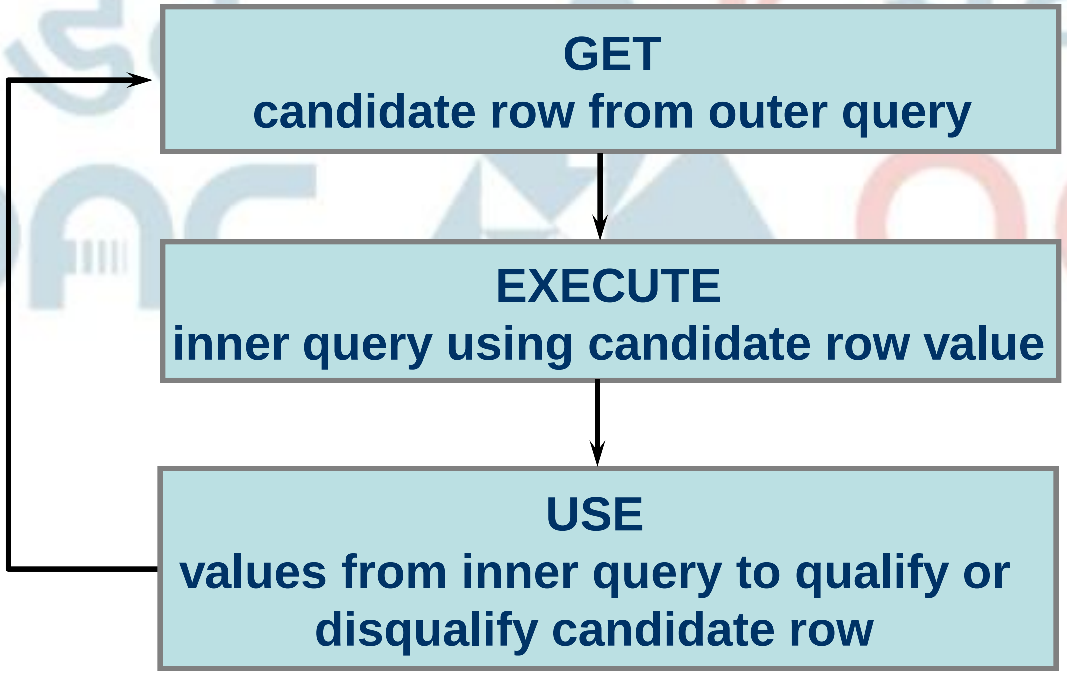
Using a Subquery in the `FROM` Clause

```
SELECT  a.last_name, a.salary,
        a.department_id, b.salavg
FROM    employees a, (SELECT   department_id,
                       AVG(salary) salavg
                       FROM     employees
                       GROUP BY department_id) b
WHERE   a.department_id = b.department_id
AND     a.salary > b.salavg;
```

LAST_NAME	SALARY	DEPARTMENT_ID	SALAVG
Hartstein	13000	20	9500
Mourgos	5800	50	3500
Hunold	9000	60	6400
Zlotkey	10500	80	10033.3333
Abel	11000	80	10033.3333
King	24000	90	19333.3333
Higgins	12000	110	10150

Correlated Subqueries

Correlated subqueries are used for row-by-row processing. Each subquery is executed once for every row of the outer query.


GET
candidate row from outer query
EXECUTE
inner query using candidate row value
USE
values from inner query to qualify or disqualify candidate row

Correlated Subqueries

```
SELECT column1, column2, ...
FROM   table1 outer
WHERE  column1 operator
                   (SELECT  colum1, column2
                       FROM     table2
                       WHERE    expr1 =
                                     outer.expr2);
```

The subquery references a column from a table in the parent query.

Using Correlated Subqueries

Find all employees who earn more than the average salary in their department.

```
SELECT last_name, salary, department_id
FROM   employees outer
WHERE  salary >
              (SELECT AVG(salary)
               FROM    employees
               WHERE   department_id =
                       outer.department_id) ;
```

Each time a row from the outer query is processed, the inner query is evaluated.

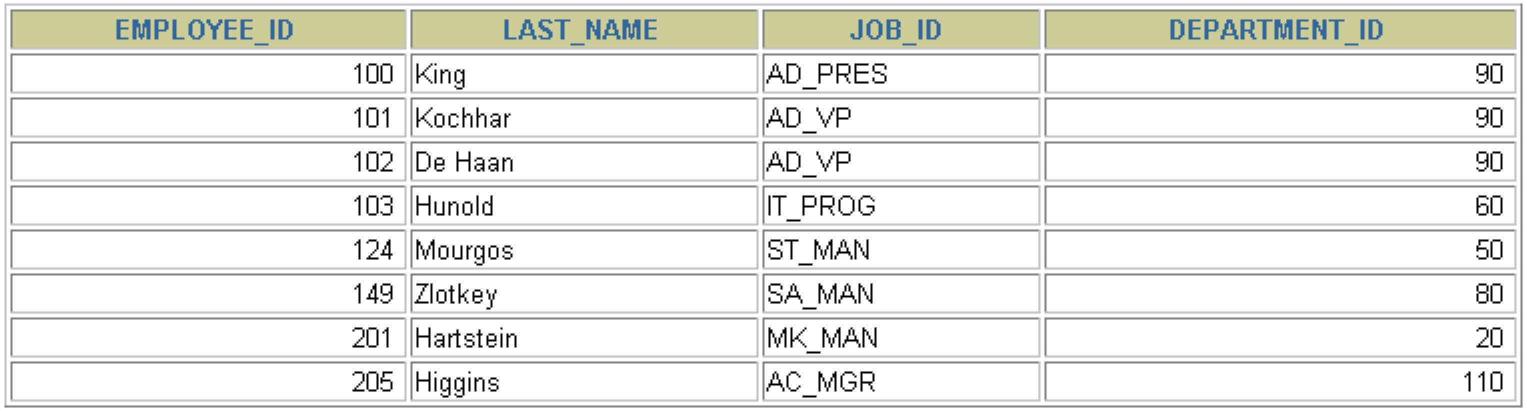
Using the `EXISTS` Operator

- The `EXISTS` operator tests for existence of rows in the results set of the subquery.
- If a subquery row value is found:
  - The search does not continue in the inner query
  - The condition is flagged `TRUE`
- If a subquery row value is not found:
  - The condition is flagged `FALSE`
  - The search continues in the inner query

Using the EXISTS Operator

Find employees who have at least one person reporting to them.

```
SELECT employee_id, last_name, job_id, department_id
FROM   employees outer
WHERE  EXISTS ( SELECT 'X'
                 FROM   employees
                 WHERE  manager_id =
                        outer.employee_id);
```



EMPLOYEE_ID	LAST_NAME	JOB_ID	DEPARTMENT_ID
100	King	AD_PRES	90
101	Kochhar	AD_VP	90
102	De Haan	AD_VP	90
103	Hunold	IT_PROG	60
124	Mourgos	ST_MAN	50
149	Zlotkey	SA_MAN	80
201	Hartstein	MK_MAN	20
205	Higgins	AC_MGR	110

Using the NOT EXISTS Operator

Find all departments that do not have any employees.

```
SELECT department_id, department_name
FROM departments d
WHERE NOT EXISTS (SELECT 'X'
                  FROM   employees
                  WHERE  department_id
                         = d.department_id);
```

DEPARTMENT_ID	DEPARTMENT_NAME
190	Contracting

Summary

In this lesson, you should have learned the following:

- A multiple-column subquery returns more than one column.
- Multiple-column comparisons can be pairwise.
- A multiple-column subquery can also be used in the `FROM` clause of a `SELECT` statement.

Summary (Continued..)

- **Correlated subqueries are useful whenever a subquery must return a different result for each candidate row.**
- **The `EXISTS` operator is a Boolean operator that tests the presence of a value.**
- **You can use the `WITH` clause to use the same query block in a `SELECT` statement when it occurs more than once**

Thank You !