DBMS/SQL

Lesson 06 Joins and Subqueries

Capgemini

Lesson Objectives

To understand the following topics:

- Joins
  - SQL: 1999 Compliant Joins
- Types of joins
- Sub-queries
- Tips and Tricks

What are Joins?

If we require data from more than one table in the database, then a join is used.

- Tables are joined on columns, which have the same “data type” and “data width” in the tables.
- The JOIN operator specifies how to relate tables in the query.
  - When you join two tables a Cartesian product is formed, by default.

Types of Joins

Given below is a list of JOINs

Proprietary Joins	**SQL: 1999 Compliant Joins**
Cartesian Product	Cross Joins
Equijoin	Inner Joins (Natural Joins)
Outer-join	Left, Right
Non-equijoin	Join on
Self-join	Join Using

Cartesian Joins

A Cartesian product is a product of all the rows of all the tables in the query.

A Cartesian product is formed when the join condition is omitted or it is invalid

To avoid having Cartesian product always include a valid join condition

Example

```
SELECT Student_Name, Dept_Name
FROM  Student_Master,
Department_Master;
```

Guidelines for Joining Tables

The JOIN condition is written in the WHERE clause

The column names which appear in more than one table should be prefixed with the table name

To improve performance of the query, table name prefix can be include for the other selected columns too

EquiJoin

In an Equijoin, the WHERE statement compares two columns from two tables with the equivalence operator “=”.

This JOIN returns all rows from both tables, where there is a match.

Syntax :

```
SELECT <col1>, <col2>,…
    FROM <table1>,<table2>
    Where <table1>.<col1>=<table2>.<col2>
    [AND <condition>] [ORDER BY <col1>, <col2>,…]
```

EquiJoin - Example

Example 1: To display student code and name along with the department name to which they belong

```
SELECT Student_Code,Student_name,Dept_name
FROM Student_Master ,Department_Master
WHERE Student_Master.Dept_code =Department_Master.Dept_code;
```

Example 2: To display student and staff name along with the department name to which they belong

```
SELECT student_name,staff_name, dept_name
      FROM student_master, department_master,staff_master
      WHERE student_master.dept_code=department_master.dept_code
       and staff_master.dept_code=department_master.dept_code;
```

Non-EquiJoin

A non-equi join is based on condition other than an equality operator
Example: To display details of staff_members who receive salary
in the range defined as per grade

```
SELECT  s.staff_name,s.staff_sal,sl.grade
     FROM staff_master s,salgrade sl
     WHERE staff_sal BETWEEN sl.losal and sl.hisal
```

Outer Join

If a row does not satisfy a JOIN condition, then the row will not appear in the query result.

The missing row(s) can be returned by using OUTER JOIN operator in the JOIN condition.

LEFT JOIN Syntax :-

```
SELECT select_list  FROM t1 LEFT JOIN t2 ON  join_condition;
```

RIGHT JOIN Syntax :-

```
SELECT select_last FROM t1 RIGHT JOIN t2 ON join_condition;
```

UNION Syntax :-

```
SELECT column_name(s) FROM table1
 UNION
SELECT column_name(s) FROM table2;
```

Self Join

In Self Join, two rows from the “same table” combine to form a “resultant row”.

- It is possible to join a table to itself, as if they were two separate tables, by using aliases for table names.
- This allows joining of rows in the same table.

Example: To display staff member information along with their manager information

```
SELECT staff.staff_code, staff.staff_name,
                mgr.staff_code, mgr.staff_name
 FROM staff_master staff, staff_master mgr
 WHERE staff.mgr_code  = mgr.staff_code;
```

What is a SubQuery?

A sub-query is a form of an SQL statement that appears inside another SQL statement.

- It is also called as a “nested query”.

The statement, which contains the sub-query, is called the “parent statement”.

The “parent statement” uses the rows returned by the sub-query.

Subquery - Examples

Example 1: To display name of students from “Mechanics” department.

- Method 1:

```
SELECT Dept_Code
  FROM Department_Master
  WHERE Dept_name = 'Mechanics';
```

- O/P : 40

```
SELECT student_code,student_name
FROM student_master
WHERE dept_code=40;
```

Subquery - Examples

Example 1 (contd.):

- Method 2: Using sub-query

```
SELECT student_code, student_name
        FROM student_master
       WHERE dept_code = (SELECT dept_code
                          FROM department_master
                            WHERE dept_name =
'Mechanics');
```

Where to use Subqueries?

Subqueries can be used for the following purpose :

- To insert records in a target table.
- To create tables and insert records in the table created.
- To update records in the target table.
- To create views.
- To provide values for conditions in the clauses, like WHERE, HAVING, IN, etc., which are used with SELECT, UPDATE and DELETE statements.

Comparison Operators for Subqueries

Types of SubQueries

- Single Row Subquery
- Multiple Row Subquery.

Some comparison operators for subqueries:

Operator	**Description**
IN	Equals to any member of
NOT IN	Not equal to any member of
*ANY	compare value to every value returned by sub-query using operator *
*ALL	compare value to all values returned by sub-query using operator *

Using Comparison Operators - Examples

Example 1: To display all staff details of who earn salary least salary

```
SELECT staff_name, staff_code, staff_sal
     FROM staff_master
        WHERE staff_sal  = (SELECT MIN(staff_sal)
                            FROM staff_master) ;
```

Example 2: To display staff details who earn salary greater than average salary earned in dept 10

```
SELECT staff_code,staff_sal
      FROM staff_master
      WHERE staff_sal > ANY(SELECT AVG(staff_sal)
                            FROM staff_master  GROUP BY
dept_code);
```

Quick Guidelines

For Using Subqueries

- Should be enclosed in parenthesis
- They should be placed on the right side of the comparison condition
- Use operator carefully. Single Row operators for Single Row Subquery and Multiple Row operator for Multiple Row Subquery

Quick Guidelines

If Single row operators are used for a sub query that returns multiple rows, Oracle would throw an error

Restrict using the NOT IN clause, which offers poor performance because the optimizer has to use a nested table scan to perform this activity.

Summary

In this lesson, you have learnt:

- Joins
- Sub-queries

Review – Questions

Question 1: The SQL compliant join which is same as EquiJoin.

- Option 1: Cross Join
- Option 2: Natural Join
- Option 3: Union

Question 2: A sub-query is also sometimes termed as _____.

Review – Questions

Question 3: A sub-query can be used for creating and inserting records.

- True / False

Question 4: If a sub-query returns multiple values, then the valid
operators is/are ____.

- Option 1: =
- Option 2: IN
- Option 3: >
- Option 4: Any