Assignment No .3

Title of Assignment: SQL Queries – all types of Join, Sub-Query and View:

Write at least10 SQL queries for suitable database application using SQL DML statements.

Note: Instructor will design the queries which demonstrate the use of concepts like all types of Join, Sub-Query and View

Course Objective:

Implement SQL queries for given requirements, using different SQL concepts

Course Outcome:

C306.3 Implement SQL queries for given requirements, using different SQL concepts

Software Required: - Mysql

Theory: -

Join in SQL

SQL Join is used to fetch data from two or more tables, which is joined to appear as single set of data. SQL Join is used for combining column from two or more tables by using values common to both tables. **Join** Keyword is used in SQL queries for joining two or more tables. Minimum required condition for joining table, is **(n-1)** where **n**, is number of tables. A table can also join to itself known as, **Self Join**.

Types of Join:-

The following are the types of JOIN that we can use in SQL.

- Inner
- Outer
- Left
- Right

Cross JOIN or Cartesian Product

This type of JOIN returns the cartesian product of rows from the tables in Join. It will return a table which consists of records which combines each row from the first table with each row of the second table.

Cross JOIN Syntax is,

SELECT column-name-list

from *table-name1*

CROSS JOIN

table-name2;

INNER Join or EQUI Join

This is a simple JOIN in which the result is based on matched data as per the equality condition

specified in the query.

Inner Join Syntax is,

SELECT column-name-list

from *table-name1*

INNER JOIN

table-name2

WHERE table-name1.column-name = table-name2.column-name;

Natural JOIN

Natural Join is a type of Inner join which is based on column having same name and same datatype present in both the tables to be joined.

Natural Join Syntax is,
SELECT *

from *table-name1*

NATURAL JOIN

table-name2;

Outer JOIN

Outer Join is based on both matched and unmatched data. Outer Joins subdivide further into,

- Left Outer Join
- Right Outer Join
- Full Outer Join

Left Outer Join

The left outer join returns a result table with the **matched data** of two tables then remaining rows of the **left** table and null for the **right** table's column.

Left Outer Join syntax is,

SELECT column-name-list

from *table-name1*

LEFT OUTER JOIN

table-name2

on table-name1.column-name = table-name2.column-name;

Left outer Join Syntax for **Oracle** is,

select column-name-

listfrom *table-name1*,

table-name2

on table-name1.column-name = table-name2.column-name(+);

Left Outer Join query will be,

SELECT * FROM class LEFT OUTER JOIN class_info ON (class.id=class_info.id);

The result table will look like,

ID	NAME	ID	ADDRESS
1	Abhi	1	DELHI
2	Adam	2	MUMBAI
3	Alex	3	CHENNAI

4	Anu	Null	Null

5	Ashish	Null	Null

Right Outer Join

The right outer join returns a result table with the **matched data** of two tables then remaining rows of the **right table** and null for the **left** table's columns.

select column-name-list

from *table-name1*

RIGHT OUTER JOIN

table-name2

on table-name1.column-name = table-name2.column-name;

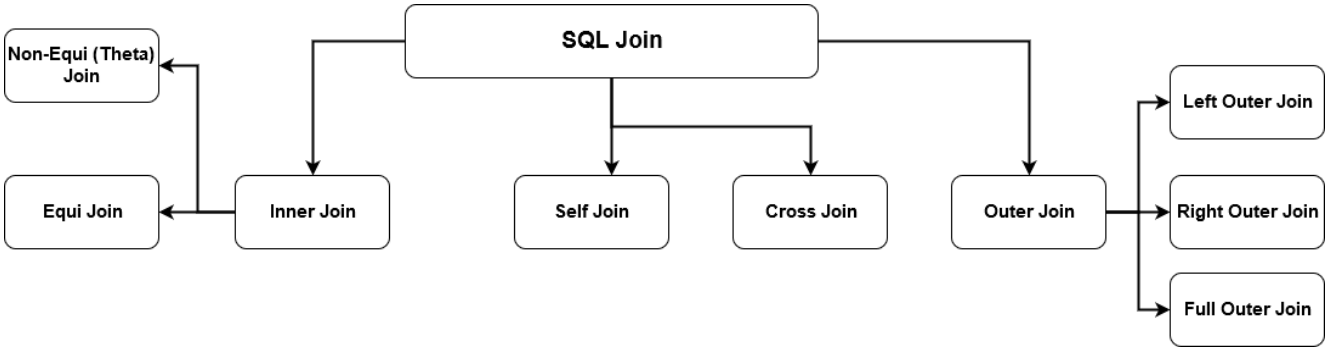
Right outer Join Syntax for **Oracle** is,

select column-name-

listfrom *table-name1,*

table-name2

on table-name1.column-name(+) = table-name2.column-name;



SQL Subquery

Subquery or **Inner query** or **Nested query** is a query in a query. SQL subquery is usually added in the WHERE Clause of the SQL statement. Most of the time, a subquery is used when you know how to search for a value using a SELECT statement, but do not know the exact value in the database.

Subqueries are an alternate way of returning data from multiple tables.
Subqueries can be used with the following SQL statements along with the comparision operators like =, <, >, >=, <= etc.

- SELECT
- INSERT
- UPDATE
- DELETE
- **SQL Subquery Example:**

- 1) Usually, a subquery should return only one record, but sometimes it can also return multiple records when used with operators LIKE IN, NOT IN in the where clause. The query syntax would be like,
- SELECT first_name, last_name, subject
  FROM student_details
  WHERE games NOT IN ('Cricket', 'Football');
- **Subquery** output would be similar to:

first_name	last_name	subject
Shekar	Gowda	Badminton
Priya	Chandra	Chess

SQL Subquery; INSERT Statement

3) Subquery can be used with INSERT statement to add rows of data from one or more tables to another table. Lets try to group all the students who study Maths in a table 'maths_group'.

```
INSERT INTO maths_group(id, name)
SELECT id, first_name || ' ' || last_name
FROM student_details WHERE subject= 'Maths'
```

SQL Subquery; SELECT Statement

4) A subquery can be used in the SELECT statement as follows. Lets use the product and order_items table defined in the sql_joins section.

```
select p.product_name, p.supplier_name, (select order_id from order_items where product_id =
101) as order_id from product p where p.product_id = 101
```

product_name	supplier_name	order_id
Television	Onida	5103

Correlated Subquery

A query is called correlated subquery when both the inner query and the outer query are interdependent. For every row processed by the inner query, the outer query is processed as well. The inner query depends on the outer query before it can be processed.

```
SELECT p.product_name FROM product p
WHERE p.product_id = (SELECT o.product_id FROM order_items o
WHERE o.product_id = p.product_id);
```

Subquery Notes

Nested Subquery

1) You can nest as many queries you want but it is recommended not to nest more than 16 subqueries in oracle

Non-Corelated Subquery

2) If a subquery is not dependent on the outer query it is called a non-correlated subquery

Subquery Errors

3) Minimize subquery errors: Use drag and drop, copy and paste to avoid running subqueries with spelling and database typos. Watch your multiple field SELECT comma use, extra or to few getting SQL error message "Incorrect syntax".

SQL Subquery Comments

Adding SQL Subquery comments are good habit (/* your command comment */) which can saveyou time, clarify your previous work .. results in less SQL headaches.

SQL Views

A VIEW is a virtual table, through which a selective portion of the data from one or more tablescan be seen. Views do not contain data of their own. They are used to restrict access to the database or to hide data complexity. A view is stored as a SELECT statement in the database.

DML operations on a view like INSERT, UPDATE, DELETE affects the data in the originaltable upon which the view is based.

The Syntax to create a sql view is

CREATE VIEW view_nameAS

SELECT column_list
FROM table_name [WHERE condition];

- ***view_name*** is the name of the VIEW.
- The SELECT statement is used to define the columns and rows that you want to displayin the view.
- **For Example:** to create a view on the product table the sql query would be like,CREATE

VIEW view_product

AS SELECT product_id, product_nameFROM

product;

Conclusion:

Students are able to implement SQL queries all types of Join, Sub-Query and View for given requirements, using different SQL concepts

Activity to be Submitted by Students

1. Consider the schema for MovieDatabase:
ACTOR (Act_id, Act_Name, Act_Gender)
DIRECTOR (Dir_id, Dir_Name,
Dir_Phone)
MOVIES (Mov_id, Mov_Title, Mov_Year, Mov_Lang,
Dir_id) MOVIE_CAST (Act_id, Mov_id, Role)
RATING (Mov_id,
Rev_Stars) Write SQL queries to
1. List the titles of all movies directed by‘Hitchcock’.
2. Find the movie names where one or more actors acted in two or moremovies.
3. List all actors who acted in a movie before 2000 and also in a

movieafter 2015 (use JOINoperation).
4. Find the title of movies and number of stars for each movie that has at least one rating and find the highest number of stars that movie received. Sort the result by movie title.
5. Update rating of all movies directed by 'Steven Spielberg' to5.

2. Apply Self Join to Student(Roll,name,address,branch,Class)

3. From the following table, create a view for all salespersons. Return salesperson ID, name, and city.

Sample table: salesman

salesman_id	name	city	commission
5001	James Hoog	New York	0.15
5002	Nail Knite	Paris	0.13
5005	Pit Alex	London	0.11
5006	Mc Lyon	Paris	0.14
5007	Paul Adam	Rome	0.13
5003	Lauson Hen	San Jose	0.12