3/134/

Assignment 4

Problem Statement: Design at least 10 SQL queries for suitable database application using SQL DML statements: All types of join, sub-query and View.

Learning Objective:

- To understand types of join, subquery and view
- To understand how to use join with DML commands
- To perform updates on simple view

Learning Outcome: The students will be able to

- Identify and implement types of join, subquery and view
- Implementation and updating of simple view

Software packages and hardware apparatus used:

- MySQL
- 64-bit Linux based open source OS
- 8 GB RAM

Concept related theory:

JOIN:

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 as Self Join.

Types of join:

Cross join: This type of JOIN returns the cartesian product of rows from the tables in

3/134/

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.

Syntax:

```
select column-name-list
from table-name1
CROSS JOIN
table-name2;
```

Inner join: This is a simple JOIN in which result is based on matched data as per the equality condition specified in the query.

Syntax:

```
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.

Syntax:

```
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.

Syntax:

```
select column-name-list from table-name1
LEFT OUTER JOIN table-name2 on
table-name1.column-name = table-name2.column-name;
```

Right Outer Join: The right outer join results 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

Syntax:

```
select column-name-list from table-name1
RIGHT OUTER JOIN table-name2 on
table-name1.column-name = table-name2.column-name;
```

Full Outer Join: The full outer join results a result table with the matched data of two table then remaining rows of both left table and then the right table.

Syntax:

```
select column-name-list from table-name1
FULL OUTER JOIN table-name2 on
table-name1.column-name = table-name2.column-name;
```

Conclusion:

Through this practical, I have implemented various types of joins on tables.