PICT, PUNE

DBMS Lab

31118

Name: Shubham Chemate

Assignment 03:

Title: SQL Queries: All types of Join, Sub-Query, & Views.

Problem Statement:

Write at least 10 SQL queries for suitable database application using SQL DML statements.

Objectives:

1) Understand & implement various SQL DML commands.
2) Understand
   i) types of joins
   ii) subquery & it's types
   iii) complex views.

Outcomes:

1) Use mysql to perform various DML commands
2) Understand various joins & complex joins using one of the SQL language (eg. mysql)

Theory Related Concepts:

SQL-Join: A join makes it possible to select data from more than one table by means of single statement.

Types of Join:
1) Inner join
2) Outer joins (left, right, full)
3) Cross join

Inner join:
1) select records that have matching values in both

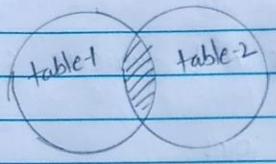
PICT, PUNE

tables.

2) syntax:

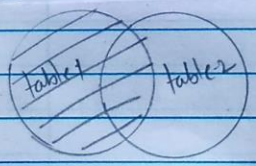
```
SELECT column_name(s)
FROM table_1
INNER JOIN table_2
ON table1.column_name = table.2.column_name.
```

3)

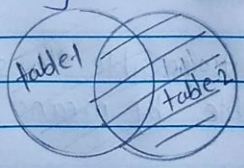


Outer Join:

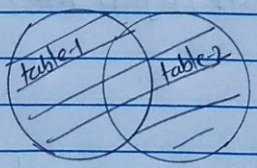
1) Left outer: returns all records from left table & matching records from right table.



2) Right Outer: returns all records from right table & matching records from left table.



3) Full outer: returns all records when there is a match in left / righ table records.



PICT, PUNE

syntax:

```
SELECT column_name(s)
FROM table_1
[LEFT/RIGHT/FULL] JOIN table_2
ON table_1.column_name = table_2.column_name;
```

Cross Join: It returns cartesian product

Syntax:

```
SELECT column_name(s)
FROM table_1
CROSS JOIN table_2;
```

Subquery: As name suggest this is a query within a query.

syntax:

```
SELECT column(s)
FROM table(s)
WHERE column_name OPERATOR
    (SELECT column_name(s)
     FROM table_name(s)
     WHERE condition);
```

Views:

i) A view is an SQL statement stored in database with associated name.

ii) A view is a composition of a table in the form of predefined SQL query.

iii) Creating view:

```
CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name(s)
WHERE condition;
```

PICT, PUNE

iv) Updating view,

```
UPDATE view_name
SET field_name = new_value
WHERE condition;
```

v) deleting rows in view

```
DELETE FROM view_name
WHERE condition;
```

vi) Dropping view,

```
DROP VIEW view_name;
```

Testcases:

consider the following relational schema:

Departments (dept_id, dept_name)

Professors (prof_id, prof_fname, prof_lname, dept_id, designation, salary, dg, email, phone, city)

Works (prof_id, duration)

Shift (prof_id, shift, working_hours)

1) find prof details & department details using NATURAL JOIN.

2) find prof_id, prof_name & shift (INNER JOIN)

3) List all the department details & the corresponding names of the professors in the same department. (LEFT OUTER JOIN)

4) LIST all the professors & corresponding names of the departments. (RIGHT OUTER JOIN)

5) Display professor name, department name, shift, salary where prof_id = 101 (MULTITABLE JOIN)

6) List the total number of professors in each department (count + any join + group by)

PICT, PUNE

7) List the prof_id associated department & department name having name = 'computer' (subquery)

8) find the names of all departments where the professor joined in year 2015 (or doj is 1-1-2015).

Outputs:

Attached separate file containing code & output screenshots of all queries along with schema & data.

Conclusion:

Hence we studied different types of joins, subquery & views. Also written queries to manipulate database.