

LAB CYCLE 2

Date:03/03/2025

Experiment No: 4

A) AIM:Familiarization of Subquery,Joins,Views and Set Operations.

1. **Subquery:** A subquery is a query nested inside another query, used to retrieve intermediate results that can be utilized by the outer query, often in the SELECT, WHERE, or FROM clauses.
2. **Joins:** Joins are used to combine rows from two or more tables based on a related column, such as an inner join (returning only matching rows) or outer joins (returning all rows from one table and matched rows from the other).
3. **Views:** A view is a virtual table created by a query that defines a specific dataset, allowing users to work with complex queries as if they were regular tables, without altering the underlying data.
4. **Set Operations:** Set operations like UNION, INTERSECT, and EXCEPT are used to combine results from multiple queries, either by merging rows, finding common rows, or eliminating duplicates between them.

Consider the following Database Schema.

Employee (ID character 5, DeptID numeric 2 , Name character 15, Designation character 15, Basic numeric 10,2 , Gender character 1)

```
>>create table Employees(ID varchar(5) PRIMARY KEY,DeptID int,Name varchar(15),Designation varchar(15),Basic decimal(10,2),Gender varchar(1));
```

OUTPUT:

```
mysql> create table employee(ID varchar(5),DeptID int(2),Name varchar(15),Designation varchar(15),Basic float(10,2),Gender varchar(1));
Query OK, 0 rows affected, 2 warnings (0.34 sec)
```

>>INSERT into employees(ID, DeptID, Name , Designation , Basic , Gender) VALUES ('101', 1, 'Ram', 'Typist', 2000 , 'M');

OUTPUT:

```
mysql> select * from employees;
+-----+-----+-----+-----+-----+-----+
| ID   | DeptID | Name   | Designation | Basic   | Gender |
+-----+-----+-----+-----+-----+-----+
| 101  | 1      | Ram    | Typist      | 2000.00 | M      |
| 102  | 2      | Arun   | Analyst     | 6000.00 | M      |
| 121  | 1      | Ruby   | Typist      | 2010.00 | F      |
| 156  | 3      | Mary   | Manager     | 4500.00 | F      |
| 123  | 2      | Mridula | Analyst     | 6000.00 | F      |
| 114  | 4      | Menon  | Clerk       | 1500.00 | M      |
| 115  | 4      | Tim    | Clerk       | 1500.00 | M      |
| 127  | 2      | Kiran  | Manager     | 4000.00 | M      |
+-----+-----+-----+-----+-----+-----+
8 rows in set (0.00 sec)
```

1. Display the different designations existing in the organisation.

>>:SELECT DISTINCT Designation from employees;

OUTPUT:

```
+-----+
| designation |
+-----+
| Typist      |
| Analyst     |
| Manager     |
| Clerk       |
+-----+
4 rows in set (0.00 sec)
```

2. Display the number of different designations existing in the organisation.

>>SELECT count(DISTINCT Designation)from employees;

OUTPUT:

```
+-----+
| count(distinct Designation) |
+-----+
|                               4 |
+-----+
1 row in set (0.00 sec)
```

3. . Display ID, name, designation,deptID and basic, DA, HRA and net salary of all employees with suitable headings as DA, HRA and NET_SAL respectively.(DA is 7.5% of basic, and NET_SAL is Basic + DA+ HRA)

>>SELECT ID,Name,Designation,DeptID,Basic,(Basic * 0.075) AS DA,(Basic * 0.10) AS HRA,(Basic + (Basic * 0.075) + (Basic * 0.10)) AS NET_SAL FROM employees;

OUTPUT:

ID	Name	Designation	DeptID	basic	DA	HRA	NET_SAL
101	Ram	Typist	1	2000.00	150.00000	200.0000	2350.00000
102	Arun	Analyst	2	6000.00	450.00000	600.0000	7050.00000
121	Ruby	Typist	1	2010.00	150.75000	201.0000	2361.75000
156	Mary	Manager	3	4500.00	337.50000	450.0000	5287.50000
123	Mridula	Analyst	2	6000.00	450.00000	600.0000	7050.00000
114	Menon	Clerk	4	1500.00	112.50000	150.0000	1762.50000
115	Tin	Clerk	4	1500.00	112.50000	150.0000	1762.50000
127	Kiran	Manager	2	4000.00	300.00000	400.0000	4700.00000

8 rows in set (0.00 sec)

4. Display the maximum salary given for female employees.

>>SELECT MAX(Basic)from employees WHERE Gender='F';

OUTPUT:

```
+-----+
| max(basic) |
+-----+
|      6000.00 |
+-----+
1 row in set (0.01 sec)
```

5. Add a column manager-id into the above table.

>>ALTER TABLE Employees ADD COLUMN Manager_ID INT;

OUTPUT:

Field	Type	Null	Key	Default	Extra
ID	varchar(5)	YES		NULL	
DeptID	int	YES		NULL	
Name	varchar(15)	YES		NULL	
Designation	varchar(15)	YES		NULL	
Basic	decimal(10,2)	YES		NULL	
Gender	varchar(1)	YES		NULL	
manager_id	int	YES		NULL	

7 rows in set (0.00 sec)

6. Update values of manager id of employees as null for 101, 101 for 102, 121, 156. 102 for 123,114,115.121 for 127.

>>UPDATE employees SET Manager_ID= NULL where ID='101';

UPDATE employees SET Manager_ID= 101 where ID IN ('102','121','156'); UPDATE employees SET Manager_ID= 102 where ID IN ('123','114','115'); UPDATE employees SET Manager_ID= 121 where ID='127';

OUTPUT:

ID	DeptID	Name	Designation	Basic	Gender	manager_id
101	1	Ram	Typist	2000.00	M	NULL
102	2	Arun	Analyst	6000.00	M	101
121	1	Ruby	Typist	2010.00	F	101
156	3	Mary	Manager	4500.00	F	101
123	2	Mridula	Analyst	6000.00	F	102
114	4	Menon	Clerk	1500.00	M	102
115	4	Tim	Clerk	1500.00	M	102
127	2	Kiran	Manager	4000.00	M	121

8 rows in set (0.00 sec)

7. Add a column joining date to the above table and update appropriate values for the joining date field.

>>ALTER TABLE employees ADD COLUMN Joining_date DATE;

OUTPUT:

Field	Type	Null	Key	Default	Extra
ID	varchar(5)	YES		NULL	
DeptID	int	YES		NULL	
Name	varchar(15)	YES		NULL	
Designation	varchar(15)	YES		NULL	
Basic	decimal(10,2)	YES		NULL	
Gender	varchar(1)	YES		NULL	
manager_id	int	YES		NULL	
joining_date	date	YES		NULL	

8 rows in set (0.01 sec)

>>UPDATE employees SET Joining_date = '2000-12-01' WHERE ID = '101';
 UPDATE employees SET Joining_date = '2000-12-02' WHERE ID = '102';
 UPDATE employees SET Joining_date = '2000-12-03' WHERE ID = '121';
 UPDATE employees SET Joining_date = '2000-11-03' WHERE ID = '156';
 UPDATE employees SET Joining_date = '2000-01-03' WHERE ID = '123';
 UPDATE employees SET Joining_date = '2001-01-03' WHERE ID = '114';
 UPDATE employees SET Joining_date = '2001-07-03' WHERE ID = '115';
 UPDATE employees SET Joining_date = '1999-07-03' WHERE ID = '127';

OUTPUT:

```
mysql> select * from employees;
```

ID	DeptID	Name	Designation	Basic	Gender	manager_id	joining_date
101	1	Ram	Typist	2000.00	M	NULL	2000-12-01
102	2	Arun	Analyst	6000.00	M	101	2000-12-02
121	1	Ruby	Typist	2010.00	F	101	2000-12-03
156	3	Mary	Manager	4500.00	F	101	2000-11-03
123	2	Mridula	Analyst	6000.00	F	102	2000-01-03
114	4	Menon	Clerk	1500.00	M	102	2001-01-03
115	4	Tim	Clerk	1500.00	M	102	2001-07-03
127	2	Kiran	Manager	4000.00	M	121	1999-07-03

8 rows in set (0.00 sec)

8. Display the details of employees according to their seniority.

>>select * FROM employees ORDER BY Joining_date ASC;

OUTPUT:

ID	DeptID	Name	Designation	Basic	Gender	manager_id	joining_date
127	2	Kiran	Manager	4000.00	M	121	1999-07-03
123	2	Mridula	Analyst	6000.00	F	102	2000-01-03
156	3	Mary	Manager	4500.00	F	101	2000-11-03
101	1	Ram	Typist	2000.00	M	NULL	2000-12-01
102	2	Arun	Analyst	6000.00	M	101	2000-12-02
121	1	Ruby	Typist	2010.00	F	101	2000-12-03
114	4	Menon	Clerk	1500.00	M	102	2001-01-03
115	4	Tim	Clerk	1500.00	M	102	2001-07-03

8 rows in set (0.00 sec)

9. Create a new table DEPARTMENT with fields DEPTID and DNAME. Make DEPTID as the primary key and make DEPTID in employee table to refer to the DEPARTMENT table.

```
>>CREATE TABLE departments(DeptID INT PRIMARY KEY, DName varchar(20));
```

OUTPUT:

Field	Type	Null	Key	Default	Extra
DeptID	int	NO	PRI	NULL	
Dname	varchar(20)	YES		NULL	

```
>>ALTER Table employees ADD FOREIGN KEY (DeptID) REFERENCES  
departments(DeptID);
```

OUTPUT:

Field	Type	Null	Key	Default	Extra
id	int	NO	PRI	NULL	
DeptID	int	YES	MUL	NULL	
Name	varchar(15)	YES		NULL	
Designation	varchar(15)	YES		NULL	
Basic	decimal(10,2)	YES		NULL	
Gender	varchar(1)	YES		NULL	
manager_id	int	YES		NULL	
joining_date	date	YES		NULL	

8 rows in set (0.01 sec)

10. Insert values into the DEPARTMENT table. Make sure that all the existing values for DEPTID in emp is inserted into this table. Sample values are DESIGN, CODING, TESTING, RESEARCH.

```
>>INSERT into departments(DeptID,DName) VALUES (1,'DESIGN');
```

OUTPUT:

DeptID	Dname
1	DESIGN
2	CODING
3	TESTING
4	RESEARCH

4 rows in set (0.00 sec)

11. Display the employee name and department name.

```
>>SELECT E.Name,D.Dname FROM employees E JOIN departments D ON E.DeptID =  
D.DeptID;
```

OUTPUT:

Name	Dname
Ram	DESIGN
Ruby	DESIGN
Arun	CODING
Mridula	CODING
Kiran	CODING
Mary	TESTING
Menon	RESEARCH
Tim	RESEARCH

8 rows in set (0.00 sec)

12. Display the department name of employee Arun.

```
>>SELECT D.Dname FROM employees E JOIN departments D ON E.DeptID =  
D.DeptID where E.Name='Arun';
```

OUTPUT:

```
+-----+
| Dname |
+-----+
| CODING |
+-----+
1 row in set (0.00 sec)
```

13. Display the salary given by DESIGN department.

>>SELECT SUM(E.Basic) FROM employees E JOIN departments D ON E.DeptID = D.DeptID where D.Dname='DESIGN';

OUTPUT:

```
+-----+
| sum(E.Basic) |
+-----+
| 4010.00 |
+-----+
1 row in set (0.00 sec)
```

14. . Display the details of typist working in DESIGN department.

>>:SELECT E.* FROM employees E JOIN departments D ON E.DeptID = D.DeptID where E.Designation='Typist' AND D.Dname='DESIGN';

OUTPUT:

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | DeptID | Name | Designation | Basic | Gender | manager_id | joining_date |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 101 | 1 | Ram | Typist | 2000.00 | M | NULL | 2000-12-01 |
| 121 | 1 | Ruby | Typist | 2010.00 | F | 101 | 2000-12-03 |
+-----+-----+-----+-----+-----+-----+-----+-----+
2 rows in set (0.00 sec)
```

15. . Display the salary of employees working in RESEARCH department.

>>SELECT E.Name,E.Basic FROM employees E JOIN departments D ON E.DeptID = D.DeptID where D.Dname='Research';

OUTPUT:

```
+-----+-----+
| Name | basic |
+-----+-----+
| Menon | 1500.00 |
| Tim | 1500.00 |
+-----+-----+
2 rows in set (0.00 sec)
```


16. . List the female employees working in TESTING department.

```
>>SELECT E.Name FROM employees E JOIN departments D ON E.DeptID = D.DeptID
where E.Gender='F' AND D.Dname='Testing';
```

OUTPUT:

```
+-----+
| Name |
+-----+
| Mary |
+-----+
1 row in set (0.00 sec)
```

17. Display the details of employees not working in CODING or TESTING department.

```
>>SELECT E.* FROM employees E LEFT JOIN departments D ON E.DeptID =
D.DeptID where D.Dname NOT IN('Coding','Testing');
```

OUTPUT:

```
+-----+-----+-----+-----+-----+-----+-----+-----+
| id | DeptID | Name | Designation | Basic | Gender | manager_id | joining_date |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 101 | 1 | Ram | Typist | 2000.00 | M | NULL | 2000-12-01 |
| 121 | 1 | Ruby | Typist | 2010.00 | F | 101 | 2000-12-03 |
| 114 | 4 | Menon | Clerk | 1500.00 | M | 102 | 2001-01-03 |
| 115 | 4 | Tim | Clerk | 1500.00 | M | 102 | 2001-07-03 |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.00 sec)
```

18. . Display the names of department giving maximum salary.

```
>>select D.Dname from employees E join departments D on E.DeptID=D.DeptID group by
D.Dname having max(E.Basic)=(select max(Basic) from employees);
```

OUTPUT:

```
+-----+
| Dname |
+-----+
| CODING |
+-----+
1 row in set (0.00 sec)
```

19. Display the names of departments with minimum number of employees.

```
>>select D.Dname from departments D join employees E on E.DeptID=D.DeptID group by
D.Dname having count(E.ID)=(select min(Dept_count) from (select count(ID) as
Dept_count from employees group by DeptID)as t);
```

OUTPUT:

```
+-----+
| Dname |
+-----+
| TESTING |
+-----+
1 row in set (0.00 sec)
```

20. Display the second maximum salary.

```
>>select max(Basic) as second_max_salary from employees where Basic<(select  
max(Basic)from employees);
```

OUTPUT:

```
+-----+  
| basic |  
+-----+  
| 4500.00 |  
+-----+  
1 row in set (0.00 sec)
```

21. Display the second minimum salary.

```
>>select min(Basic) as second_min_salary from employees where Basic>(select min(Basic)  
from employees);
```

OUTPUT:

```
+-----+  
| basic |  
+-----+  
| 2000.00 |  
+-----+  
1 row in set (0.00 sec)
```

22. Display the names of employees getting salary greater than the average salary of their department.

```
>>select E.Name from employees E join (select DeptID,avg(Basic)as avg_salary from  
employees group by DeptID)as Dept_avg on E.DeptID=Dept_avg.DeptID where  
E.Basic>Dept_avg.avg_salary;
```

OUTPUT:

```
+-----+  
| Name |  
+-----+  
| Arun |  
| Ruby |  
| Mridula |  
+-----+  
3 rows in set (0.00 sec)
```

23. Display the names of employees working under the manager Ram.

```
>>select E.Name from employees E join employees m on E.manager_id=m.ID where  
m.Name='Ram';
```

OUTPUT:

```
+-----+  
| Name |  
+-----+  
| Arun |  
| Ruby |  
| Mary |  
+-----+  
3 rows in set (0.00 sec)
```