LAB CYCLE 2

Date:03/03/2025

Experiment No: 4

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)

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

```
mysql> show tables;
+----+
| Tables_in_24mca31 |
+----+
| department |
| employee |
| employees |
+----+
3 rows in set (0.00 sec)
```

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

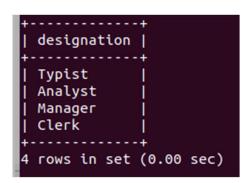
OUTPUT:

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

1.Display the different designations existing in the organisation.

SQL:SELECT DISTINCT Designation from employees;

OUTPUT:



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

SQL: SELECT count(DISTINCT Designation) from employees;

```
count(distinct Designation) |

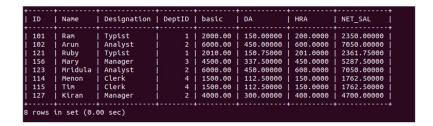
4 |

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)

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

OUTPUT:



4. Display the maximum salary given for female employees.

SQL: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.

SQL: ALTER TABLE Employees ADD COLUMN Manager_ID INT;

OUTPUT:

Field	+ Туре	Null	Key	Default	Extra
DeptID Name Designation Basic Gender	varchar(5) int varchar(15) varchar(15) decimal(10,2) varchar(1) int	YES YES YES YES YES YES YES YES		NULL NULL NULL NULL NULL NULL 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.

```
SQL: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 De	ptID Name		-+ Basic		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	į F	102
114	4 Menon	Clerk	1500.00	M	102
115	4 Tim	Clerk	1500.00	i M	102
127	2 Kiran	Manager	4000.00	M	121
+			-+		+
8 rows in s	et (0.00 sec)				

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

SQL:ALTER TABLE employees ADD COLUMN Joining_date DATE;

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

SQL:UPDATE employees SET Joining_date = '2000-12-01' WHERE ID = '101';

OUTPUT:

D	DeptID		Designation				
101		Ram	Typist		M		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
14	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. Display the details of employees according to their seniority.

SQL:select * FROM employees ORDER BY Joining_date ASC;

ID	DeptID	Name	Designation	Basic	Gender	manager_id	joining_date
127	2	Kiran	Manager	4000.00	М	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
+		+		+	+	+	

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.

SQL:CREATE TABLE departments(DeptID INT PRIMARY KEY, DName varchar(20));

OUTPUT:



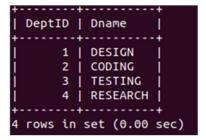
SQL:ALTER Table employees ADD FOREIGN KEY (DeptID) REFERENCES departments(DeptID);

OUTPUT:

+	Null	+	+
Field Type		Key	Default Extra
id	NO YES YES YES YES YES YES	PRI MUL 	NULL

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.

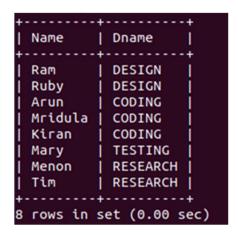
SQL: INSERT into departments(DeptID,DName) VALUES (1,'DESIGN');



11. Display the employee name and department name.

SQL:SELECT E.Name,D.Dname FROM employees E JOIN departments D ON E.DeptID = D.DeptID;

OUTPUT:



12. Display the department name of employee Arun.

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



13. Display the salary given by DESIGN department.

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

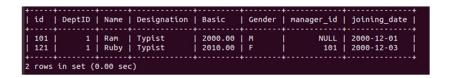
OUTPUT:



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

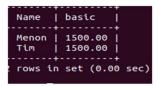
SQL:SELECT E.* FROM employees E JOIN departments D ON E.DeptID = D.DeptID where E.Designation='Typist' AND D.Dname='DESIGN';

OUTPUT:



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

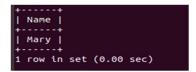
SQL: SELECT E.Name, E.Basic FROM employees E JOIN departments D ON E.DeptID = D.DeptID where D.Dname='Research';



16. List the female employees working in TESTING department.

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

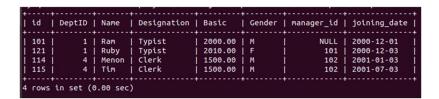
OUTPUT:



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

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

OUTPUT:



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

SQL: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:



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

SQL: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);



20. Display the second maximum salary.

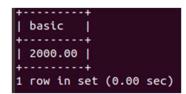
SQL:select max(Basic) as second_max_salary from employees where Basic<(select max(Basic)from employees);

OUTPUT:

21. Display the second minimum salary.

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

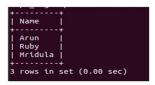
OUTPUT:



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

SQL: 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:



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

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

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