

20MCA134 ADVANCED DBMS LAB

LAB CYCLE 2

Experiment No: 4

Familiarization of Subquery, Joins, Views and Set Operations.

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 EMPLOYEE (ID CHAR(5) PRIMARY KEY, DeptID NUMERIC(2), Name  
CHAR(15), Designation CHAR(15), Basic NUMERIC(10,2), Gender CHAR(1));
```

```
INSERT INTO EMPLOYEE (ID, DeptID, Name, Designation, Basic, Gender) VALUES('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');
```

1. Display the different designations existing in the organisation.

```
SELECT DISTINCT Designation FROM EMPLOYEE;
```

```
mysql> SELECT DISTINCT Designation FROM EMPLOYEE;
+-----+
| Designation |
+-----+
| Typist      |
| Analyst     |
| Clerk       |
| Manager     |
+-----+
4 rows in set (0.00 sec)
```

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

```
SELECT COUNT(DISTINCT Designation) AS Designation_Count FROM EMPLOYEE;
```

```
mysql> SELECT COUNT(DISTINCT Designation) AS Designation_Count FROM EMPLOYEE;
+-----+
| Designation_Count |
+-----+
| 4 |
+-----+
1 row in set (0.00 sec)
```

3. Display ID, name, design,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 EMPLOYEE;

```
mysql> 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 EMPLOYEE;
```

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
114	Menon	Clerk	4	1500.00	112.50000	150.0000	1762.50000
115	Tim	Clerk	4	1500.00	112.50000	150.0000	1762.50000
121	Ruby	Typist	1	2010.00	150.75000	201.0000	2361.75000
123	Mridula	Analyst	2	6000.00	450.00000	600.0000	7050.00000
127	Kiran	Manager	2	4000.00	300.00000	400.0000	4700.00000
156	Mary	Manager	3	4500.00	337.50000	450.0000	5287.50000

8 rows in set (0.00 sec)

4. Display the maximum salary given for female employees.

SELECT MAX(Basic) AS Max_Female_Salary FROM EMPLOYEE WHERE Gender = 'F';

```
mysql> SELECT MAX(Basic) AS Max_Female_Salary FROM EMPLOYEE WHERE Gender = 'F';
```

Max_Female_Salary
6000.00

1 row in set (0.00 sec)

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

ALTER TABLE EMPLOYEE ADD ManagerID CHAR(5);

```
mysql> ALTER TABLE EMPLOYEE ADD ManagerID CHAR(5);
Query OK, 0 rows affected (0.17 sec)
Records: 0 Duplicates: 0 Warnings: 0
```

```
mysql> DESC EMPLOYEE;
```

Field	Type	Null	Key	Default	Extra
ID	char(5)	NO	PRI	NULL	
DeptID	decimal(2,0)	YES		NULL	
Name	char(15)	YES		NULL	
Designation	char(15)	YES		NULL	
Basic	decimal(10,2)	YES		NULL	
Gender	char(1)	YES		NULL	
ManagerID	char(5)	YES		NULL	

7 rows in set (0.01 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 EMPLOYEE SET ManagerID = NULL WHERE ID = '101';
```

```
UPDATE EMPLOYEE SET ManagerID = '101' WHERE ID IN ('102', '121', '156');
```

```
UPDATE EMPLOYEE SET ManagerID = '102' WHERE ID IN ('123', '114', '115');
```

```
UPDATE EMPLOYEE SET ManagerID = '121' WHERE ID = '127';
```

```
mysql> UPDATE EMPLOYEE SET ManagerID = NULL WHERE ID = '101';
```

```
Query OK, 0 rows affected (0.00 sec)
```

```
Rows matched: 1  Changed: 0  Warnings: 0
```

```
mysql> UPDATE EMPLOYEE SET ManagerID = '101' WHERE ID IN ('102', '121', '156');
```

```
Query OK, 3 rows affected (0.04 sec)
```

```
Rows matched: 3  Changed: 3  Warnings: 0
```

```
mysql> UPDATE EMPLOYEE SET ManagerID = '102' WHERE ID IN ('123', '114', '115');
```

```
Query OK, 3 rows affected (0.04 sec)
```

```
Rows matched: 3  Changed: 3  Warnings: 0
```

```
mysql> UPDATE EMPLOYEE SET ManagerID = '121' WHERE ID = '127';
```

```
Query OK, 1 row affected (0.05 sec)
```

```
Rows matched: 1  Changed: 1  Warnings: 0
```

```
mysql> select * from EMPLOYEE;
```

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

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 EMPLOYEE ADD JoiningDate DATE;
```

```
mysql> ALTER TABLE EMPLOYEE ADD JoiningDate DATE;
Query OK, 0 rows affected (0.27 sec)
Records: 0  Duplicates: 0  Warnings: 0

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

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

```
SELECT * FROM EMPLOYEE ORDER BY JoiningDate ASC;
```

```
UPDATE EMPLOYEE SET JoiningDate = '2022-01-01' WHERE ID = 101;
UPDATE EMPLOYEE SET JoiningDate = '2023-05-10' WHERE ID = 102;
UPDATE EMPLOYEE SET JoiningDate = '2021-08-15' WHERE ID = 114;
UPDATE EMPLOYEE SET JoiningDate = '2020-03-22' WHERE ID = 115;
UPDATE EMPLOYEE SET JoiningDate = '2019-07-30' WHERE ID = 121;
UPDATE EMPLOYEE SET JoiningDate = '2022-12-15' WHERE ID = 123;
UPDATE EMPLOYEE SET JoiningDate = '2021-06-25' WHERE ID = 127;
UPDATE EMPLOYEE SET JoiningDate = '2020-11-01' WHERE ID = 156;
```

```
mysql> SELECT * FROM EMPLOYEE ORDER BY JoiningDate ASC;
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | DeptID | Name   | Designation | Basic   | Gender | ManagerID | JoiningDate |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 121 | 1      | Ruby   | Typist      | 2010.00 | F      | 101       | 2019-07-30  |
| 115 | 4      | Tim    | Clerk       | 1500.00 | M      | 102       | 2020-03-22  |
| 156 | 3      | Mary   | Manager     | 4500.00 | F      | 101       | 2020-11-01  |
| 127 | 2      | Kiran  | Manager     | 4000.00 | M      | 121       | 2021-06-25  |
| 114 | 4      | Menon  | Clerk       | 1500.00 | M      | 102       | 2021-08-15  |
| 101 | 1      | Ram    | Typist      | 2000.00 | M      | NULL      | 2022-01-01  |
| 123 | 2      | Mridula | Analyst     | 6000.00 | F      | 102       | 2022-12-15  |
| 102 | 2      | Arun   | Analyst     | 6000.00 | M      | 101       | 2023-05-10  |
+-----+-----+-----+-----+-----+-----+-----+-----+
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.

```
ALTER TABLE EMPLOYEE ADD CONSTRAINT fk_dept_id FOREIGN KEY (DeptID)
REFERENCES DEPARTMENT(DeptID);
```

```
mysql> show columns from EMPLOYEE like 'DeptID';
+-----+-----+-----+-----+-----+-----+
| Field | Type | Null | Key | Default | Extra |
+-----+-----+-----+-----+-----+-----+
| DeptID | int  | YES  | MUL | NULL    |       |
+-----+-----+-----+-----+-----+-----+
1 row 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 DEPARTMENT (DeptID, DNAME) VALUES
(1, 'DESIGN'),
(2, 'CODING'),
(3, 'TESTING'),
(4, 'RESEARCH');
```

```
mysql> select * from DEPARTMENT;
+-----+-----+
| 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 EMPLOYEE.Name, DEPARTMENT.DNAME FROM EMPLOYEE JOIN  
DEPARTMENT ON EMPLOYEE.DeptID = DEPARTMENT.DeptID;
```

```
mysql> SELECT EMPLOYEE.Name, DEPARTMENT.DNAME FROM EMPLOYEE JOIN DEPARTMENT ON EMPLOYEE.DeptID = DEPARTMENT.DeptID;  
+-----+  
| 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 DNAME FROM DEPARTMENT WHERE DeptID = (SELECT DeptID FROM  
EMPLOYEE WHERE Name = 'Arun');
```

```
mysql> SELECT DNAME FROM DEPARTMENT WHERE DeptID = (SELECT DeptID FROM EMPLOYEE WHERE Name = 'Arun');  
+-----+  
| DNAME |  
+-----+  
| CODING |  
+-----+  
1 row in set (0.00 sec)
```

13. Display the salary given by DESIGN department.

```
SELECT SUM(Basic) FROM EMPLOYEE WHERE DeptID = (SELECT DeptID FROM  
DEPARTMENT WHERE DNAME = 'DESIGN');
```

```
mysql> SELECT SUM(Basic) FROM EMPLOYEE WHERE DeptID = (SELECT DeptID FROM DEPARTMENT WHERE DNAME = 'DESIGN');  
+-----+  
| SUM(Basic) |  
+-----+  
| 4010.00 |  
+-----+  
1 row in set (0.01 sec)
```

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

```
SELECT * FROM EMPLOYEE WHERE Designation = 'Typist' AND DeptID = (SELECT DeptID  
FROM DEPARTMENT WHERE DNAME = 'DESIGN');
```

```
mysql> SELECT * FROM EMPLOYEE WHERE Designation = 'Typist' AND DeptID = (SELECT DeptID FROM DEPARTMENT WHERE DNAME = 'DESIGN');  
+-----+  
| ID | DeptID | Name | Designation | Basic   | Gender | ManagerID | JoiningDate |  
+-----+  
| 101 | 1 | Ram | Typist      | 2000.00 | M      | NULL      | 2022-01-01 |  
| 121 | 1 | Ruby | Typist      | 2010.00 | F      | 101       | 2019-07-30 |  
+-----+  
2 rows in set (0.00 sec)
```

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

SELECT Name, Basic FROM EMPLOYEE WHERE DeptID = (SELECT DeptID FROM DEPARTMENT WHERE DNAME = 'RESEARCH');

```
mysql> SELECT Name, Basic FROM EMPLOYEE WHERE DeptID = (SELECT DeptID FROM DEPARTMENT WHERE DNAME = 'RESEARCH');
+-----+-----+
| 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 * FROM EMPLOYEE WHERE Gender = 'F' AND DeptID = (SELECT DeptID FROM DEPARTMENT WHERE DNAME = 'TESTING');

```
mysql> SELECT * FROM EMPLOYEE WHERE Gender = 'F' AND DeptID = (SELECT DeptID FROM DEPARTMENT WHERE DNAME = 'TESTING');
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | DeptID | Name | Designation | Basic | Gender | ManagerID | JoiningDate |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 156 | 3 | Mary | Manager | 4500.00 | F | 101 | 2020-11-01 |
+-----+-----+-----+-----+-----+-----+-----+-----+
1 row in set (0.00 sec)
```

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

SELECT * FROM EMPLOYEE WHERE DeptID NOT IN (SELECT DeptID FROM DEPARTMENT WHERE DNAME IN ('CODING', 'TESTING'));

```
mysql> SELECT * FROM EMPLOYEE WHERE DeptID NOT IN (SELECT DeptID FROM DEPARTMENT WHERE DNAME IN ('CODING', 'TESTING'));
+-----+-----+-----+-----+-----+-----+-----+-----+
| ID | DeptID | Name | Designation | Basic | Gender | ManagerID | JoiningDate |
+-----+-----+-----+-----+-----+-----+-----+-----+
| 101 | 1 | Ram | Typist | 2000.00 | M | NULL | 2022-01-01 |
| 114 | 4 | Menon | Clerk | 1500.00 | M | 102 | 2021-08-15 |
| 115 | 4 | Tim | Clerk | 1500.00 | M | 102 | 2020-03-22 |
| 121 | 1 | Ruby | Typist | 2010.00 | F | 101 | 2019-07-30 |
+-----+-----+-----+-----+-----+-----+-----+-----+
4 rows in set (0.01 sec)
```

18. Display the names of department giving maximum salary.

SELECT DNAME FROM DEPARTMENT WHERE DeptID = (SELECT DeptID FROM EMPLOYEE GROUP BY DeptID ORDER BY SUM(Basic) DESC LIMIT 1);

```
mysql> SELECT DNAME FROM DEPARTMENT WHERE DeptID = (SELECT DeptID FROM EMPLOYEE GROUP BY DeptID ORDER BY SUM(Basic) DESC LIMIT 1);
+-----+
| DNAME |
+-----+
| CODING |
+-----+
1 row in set (0.00 sec)
```

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

```
SELECT DNAME FROM DEPARTMENT WHERE DeptID = (SELECT DeptID FROM EMPLOYEE GROUP BY DeptID ORDER BY COUNT(*) ASC LIMIT 1);
```

```
mysql> SELECT DNAME FROM DEPARTMENT WHERE DeptID = (SELECT DeptID FROM EMPLOYEE GROUP BY DeptID ORDER BY COUNT(*) ASC LIMIT 1);
+-----+
| DNAME |
+-----+
| TESTING |
+-----+
1 row in set (0.00 sec)
```

20. Display the second maximum salary.

```
SELECT DISTINCT Basic FROM EMPLOYEE ORDER BY Basic DESC LIMIT 1 OFFSET 1;
```

```
mysql> SELECT DISTINCT Basic FROM EMPLOYEE ORDER BY Basic DESC LIMIT 1 OFFSET 1;
+-----+
| Basic |
+-----+
| 4500.00 |
+-----+
1 row in set (0.00 sec)
```

21. Display the second minimum salary.

```
SELECT DISTINCT Basic FROM EMPLOYEE ORDER BY Basic ASC LIMIT 1 OFFSET 1;
```

```
mysql> SELECT DISTINCT Basic FROM EMPLOYEE ORDER BY Basic ASC LIMIT 1 OFFSET 1;
+-----+
| 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 Name FROM EMPLOYEE E WHERE Basic > (SELECT AVG(Basic) FROM EMPLOYEE WHERE DeptID = E.DeptID);
```

```
mysql> SELECT Name FROM EMPLOYEE E WHERE Basic > (SELECT AVG(Basic) FROM EMPLOYEE WHERE DeptID = E.DeptID);
+-----+
| Name |
+-----+
| Arun |
| Ruby |
| Mridula |
+-----+
3 rows in set (0.00 sec)
```


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

SELECT Name FROM EMPLOYEE WHERE ManagerID = (SELECT ID FROM EMPLOYEE WHERE Name = 'Ram');

```
mysql> SELECT Name FROM EMPLOYEE WHERE ManagerID = (SELECT ID FROM EMPLOYEE WHERE Name = 'Ram');
+-----+
| Name |
+-----+
| Arun |
| Ruby |
| Mary |
+-----+
3 rows in set (0.00 sec)
```