Name: Tejas Rajesh Machkar
Roll No: 23
Class: TE2 Comp
PRN: F18112025

1.Retrieve the address of customer Fname as 'xyz' and Lname as 'pqr'

```
select * from cust_mer;
+---------+----------+---------+
| cust_no | fname    | lname   |
+---------+----------+---------+
|      23 | xyz      | pqr     |
|      24 | Tejas    | Machkar |
|      35 | Harshika | Mishra  |
+---------+----------+---------+
select * from add_dets;
+---------+------------------------------+-------------------------------
+---------------+----------+---------+
| cust_no | add1                         | add2
| state         | city     | pincode |
+---------+------------------------------+-------------------------------
+---------------+----------+---------+
|      23 | Goodwill Society Porwal Road | Dhanori Pune-411007
| Maharashtra   | Pune     | 411007  |
|      24 | Pratik Nagar society         | Vishrantwadi Pune-411006
| Maharashtra   | Pune     | 411006  |
|      35 | Kaalin wada Sr-102           | Santosh Nagar Mirzapur-611015
| Uttar Pradesh | Mirzapur | 411015  |
+---------+------------------------------+-------------------------------
+---------------+----------+---------+
Select add1,add2,state,city,pincode from add_dets where cust_no = (select
cust_no from cust_mer where fname="xyz" and lname="pqr");
Output:
+------------------------------+---------------------+-------------+-----
-+---------+
| add1                         | add2                | state       | city
| pincode |
+------------------------------+---------------------+-------------+-----
-+---------+
| Goodwill Society Porwal Road | Dhanori Pune-411007 | Maharashtra | Pune
| 411007  |
+------------------------------+---------------------+-------------+-----
-+---------+
-------------------------------------------------------------------------
-----------------
```

2.List the customer holding fixed deposit of amount more than 5000

```
select * from cust_mer;
+---------+----------+---------+
| cust_no | fname    | lname   |
+---------+----------+---------+
|      23 | xyz      | pqr     |
|      24 | Tejas    | Machkar |
|      35 | Harshika | Mishra  |
+---------+----------+---------+
select * from acc_fd_cust_dets;
+---------+-----------+
| cust_no | acc_fd_no |
+---------+-----------+
|      24 |         1 |
|      35 |         2 |
+---------+-----------+
select * from fd_dets;
+-----------+----------+--------+
| acc_fd_no | fd_sr_no | amount |
+-----------+----------+--------+
|         1 |        1 |   4000 |
|         2 |        1 |  10000 |
+-----------+----------+--------+
select fname, lname from cust_mer where cust_no=(select cust_no from
acc_fd_cust_dets where acc_fd_no = (Select acc_fd_no from fd_dets where
amount>5000));
Output:
+----------+--------+
| fname    | lname  |
+----------+--------+
| Harshika | Mishra |
+----------+--------+
--------------------------------------------------------------------------
-----------------
```

3.List the employee details along with branch names to which they belong

```
select * from emp_mstr;
+-------+----------+----------+---------+----------+--------------+-----
-----+
| empno | fname    | mname    | lname   | dept     | desg          |
branchno |
+-------+----------+----------+---------+----------+--------------+-----
-----+
|     1 | Tejas    | Rajesh   | Machkar | Computer | Manager       |
2 |
|     2 | Harshika | Suresh   | Mishra  | Computer | Senior Dev    |
2 |
|     3 | Sumedh   | Ramesh   | Ovhal   | HR       | Recruiter     |
1 |
|     4 | Sayali   | Shailesh | Shinde  | HR       | Party planner |
1 |
+-------+----------+----------+---------+----------+--------------+-----
-----+
select * from branch_mstr;
+----------+---------+
| branchno | name    |
+----------+---------+
|        1 | Denmark |
|        2 | Sydney  |
+----------+---------+
Select emp_mstr.*,branch_mstr.name from emp_mstr join branch_mstr on
emp_mstr.branchno=branch_mstr.branchno;
Output:
+-------+----------+----------+---------+----------+--------------+-----
-----+---------+
| empno | fname    | mname    | lname   | dept     | desg          |
branchno | name    |
+-------+----------+----------+---------+----------+--------------+-----
-----+---------+
|     1 | Tejas    | Rajesh   | Machkar | Computer | Manager       |
2 | Sydney  |
|     2 | Harshika | Suresh   | Mishra  | Computer | Senior Dev    |
2 | Sydney  |
|     3 | Sumedh   | Ramesh   | Ovhal   | HR       | Recruiter     |
1 | Denmark |
|     4 | Sayali   | Shailesh | Shinde  | HR       | Party planner |
1 | Denmark |
+-------+----------+----------+---------+----------+---------------+-----
-----+---------+
-------------------------------------------------------------------------
----------------
```

4.List the employee details along with contact details using left outer join & right join

```
select * from emp_mstr;
+-------+----------+----------+---------+----------+---------------+-----
-----+
| empno | fname    | mname    | lname   | dept     | desg          |
branchno |
+-------+----------+----------+---------+----------+---------------+-----
-----+
|     1 | Tejas    | Rajesh   | Machkar | Computer | Manager       |
2 |
|     2 | Harshika | Suresh   | Mishra  | Computer | Senior Dev    |
2 |
|     3 | Sumedh   | Ramesh   | Ovhal   | HR       | Recruiter     |
1 |
|     4 | Sayali   | Shailesh | Shinde  | HR       | Party planner |
1 |
+-------+----------+----------+---------+----------+---------------+-----
-----+
select * from cntc_dets;
+-------+-----------+------------+
| empno | cntc_type | cntc_data  |
+-------+-----------+------------+
|     1 | Mobile    | 9764661522 |
|     2 | email     | hansa.mail |
|     3 | Mobile    | 9654234521 |
|     4 | mail      | saya.mail  |
+-------+-----------+------------+
select
emp_mstr.empno,emp_mstr.fname,emp_mstr.mname,emp_mstr.lname,cntc_dets.cnt
c_type,cntc_dets.cntc_data from emp_mstr right join cntc_dets on
emp_mstr.empno=cntc_dets.empno;
Output:
+-------+----------+----------+---------+-----------+------------+
| empno | fname    | mname    | lname   | cntc_type | cntc_data  |
+-------+----------+----------+---------+-----------+------------+
|     1 | Tejas    | Rajesh   | Machkar | Mobile    | 9764661522 |
|     2 | Harshika | Suresh   | Mishra  | email     | hansa.mail |
|     3 | Sumedh   | Ramesh   | Ovhal   | Mobile    | 9654234521 |
|     4 | Sayali   | Shailesh | Shinde  | mail      | saya.mail  |
+-------+----------+----------+---------+-----------+------------+
-------------------------------------------------------------------------
----------------
```

5.List the customer who do not have bank branches in their vicinity

```
select * from cust_mer;
+---------+----------+---------+
| cust_no | fname    | lname   |
+---------+----------+---------+
|      23 | xyz      | pqr     |
|      24 | Tejas    | Machkar |
|      35 | Harshika | Mishra  |
+---------+----------+---------+
select * from add_dets;
+---------+------------------------------+-------------------------------
+---------------+----------+---------+
| cust_no | add1                         | add2
| state         | city     | pincode |
+---------+------------------------------+-------------------------------
+---------------+----------+---------+
|      23 | Goodwill Society Porwal Road | Dhanori Pune-411007
| Maharashtra   | Pune     | 411007  |
|      24 | Pratik Nagar society         | Vishrantwadi Pune-411006
| Maharashtra   | Pune     | 411006  |
|      35 | Kaalin wada Sr-102           | Santosh Nagar Mirzapur-611015
| Uttar Pradesh | Mirzapur | 411015  |
+---------+------------------------------+-------------------------------
+---------------+----------+---------+
select fname,lname from cust_mer where cust_no in(select cust_no from
add_dets where pincode not in (select pincode from branch_add_dets));
Output:
+----------+--------+
| fname    | lname  |
+----------+--------+
| Harshika | Mishra |
+----------+--------+
-------------------------------------------------------------------------
----------------
```

6.
A.Create View on borrower table by selecting any two columns and perform insert update delete operations

```
CREATE VIEW BORROWER_VIEW AS SELECT * FROM Borrower;
  Select * from BORROWER_VIEW;
  Output:
      +-----------+---------+
      | cust_name | loan_no |
      +-----------+---------+
      | Shyam     | loan123 |
      | Soham     | loan124 |
      | Sumedh    | loan127 |
      +-----------+---------+
-------------------------------------------------------------------------
----------------
```

i.

```
insert into loan values ("loan128","Bund Garden",8000);
INSERT INTO BORROWER_VIEW VALUES('TANMAY JAGTAP', 116);
Tables before:
BORROWER_VIEW:
+-----------+---------+
| cust_name | loan_no |
+-----------+---------+
| Shyam     | loan123 |
| Soham     | loan124 |
| Sumedh    | loan127 |
+-----------+---------+
Loan:
+---------+-------------+--------+
| loan_no | branch_name | amount |
+---------+-------------+--------+
| loan123 | Viman Nagar |  35000 |
| loan124 | Bund Garden |  40000 |
| loan127 | Viman Nagar |  45000 |
+---------+-------------+--------+
```

Tables after:

BORROWER_VIEW:

```
+---------------+---------+
| cust_name     | loan_no |
+---------------+---------+
| Shyam         | loan123 |
| Soham         | loan124 |
| Sumedh        | loan127 |
| Tanmay Jagtap | loan128 |
+---------------+---------+
```

Loan:

```
+---------+-------------+--------+
| loan_no | branch_name | amount |
+---------+-------------+--------+
| loan123 | Viman Nagar |  35000 |
| loan124 | Bund Garden |  40000 |
| loan127 | Viman Nagar |  45000 |
| loan128 | Bund Garden |   8000 |
+---------+-------------+--------+
-------------------------------------------------------------------------
-----------------
```

```
ii.UPDATE BORROWER_VIEW SET cust_name="ALI ANEES" WHERE
loan_no="loan128";
   Output:
+-----------+---------+
| cust_name | loan_no |
+-----------+---------+
| Shyam     | loan123 |
| Soham     | loan124 |
| Sumedh    | loan127 |
| ALI ANEES | loan128 |
+-----------+---------+
-------------------------------------------------------------------------
-----------------
```

```
iii.DELETE FROM Loan WHERE loan_no="loan128";
      Output:
BORROWER_VIEW:
+-----------+---------+
| cust_name | loan_no |
+-----------+---------+
| Shyam     | loan123 |
| Soham     | loan124 |
| Sumedh    | loan127 |
+-----------+---------+
-------------------------------------------------------------------------
-----------------
```

B. Create view on borrower and depositor table by selecting any one column from each table

```
CREATE VIEW BOR_DEP AS SELECT acc_no, Depositor.cust_name, loan_no FROM
Depositor INNER JOIN Borrower ON Depositor.cust_name=Borrower.cust_name;
select * from BOR_DEP;
Output:
+-------------+-----------+---------+
| acc_no      | cust_name | loan_no |
+-------------+-----------+---------+
| C8482381765 | Shyam     | loan123 |
| C8382381671 | Soham     | loan124 |
+-------------+-----------+---------+
-------------------------------------------------------------------------
-----------------
```

```
i.
INSERT INTO Loan VALUES ("loan128","Bund Garden",8000);
INSERT INTO Borrower VALUES('ALIASGAR ANEES', "loan128");
INSERT INTO Depositor VALUES('ALIASGAR ANEES', "364");
SELECT * FROM BOR_DEP;
Output:
+-------------+----------------+---------+
| acc_no      | cust_name      | loan_no |
+-------------+----------------+---------+
| C8482381765 | Shyam          | loan123 |
| C8382381671 | Soham          | loan124 |
| 364         | ALIASGAR ANEES | loan128 |
+-------------+----------------+---------+
-------------------------------------------------------------------------
-----------------
```

```
ii.
UPDATE Depositor SET cust_name='ALI ANIS' WHERE acc_no="364";
UPDATE Borrower SET cust_name='ALI ANIS' WHERE loan_no="loan128";
SELECT * FROM BOR_DEP;
Output:
+-------------+-----------+---------+
| acc_no      | cust_name | loan_no |
+-------------+-----------+---------+
| C8482381765 | Shyam     | loan123 |
| C8382381671 | Soham     | loan124 |
| 364         | ALI ANIS  | loan128 |
+-------------+-----------+---------+
-------------------------------------------------------------------------
-----------------
```

```
iii.
DELETE FROM Depositor WHERE acc_no="364";
DELETE FROM loan WHERE loan_no="loan128";
SELECT * FROM BOR_DEP;
Output:
+-------------+-----------+---------+
| acc_no      | cust_name | loan_no |
+-------------+-----------+---------+
| C8482381765 | Shyam     | loan123 |
| C8382381671 | Soham     | loan124 |
+-------------+-----------+---------+
```

Name: Tejas. R. Machkar
RollNo: 23 PRN: F18112025
Batch: P

DBMS - Assignment 4

• Questions:

Q1) What's JOIN clause?

A.1) A join clause is used to combine rows from two or more tables., based on related column between the tables.

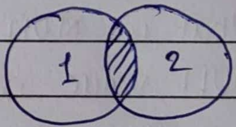
• A related column between the tables is important for the JOIN clause.

Q2) What are are different kinds of JOIN? Explain in detail.

A.2) i) INNER JOIN: The inner join keyword selects the records that have matching values in both the tables.

• Syntax:

```
SELECT column_name(s) FROM table_1
INNER JOIN table_2 ON
table1.colname = table2.colname.
```

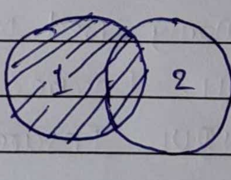


ii) LEFT OUTER JOIN: This keyword returns all records from the ~~table~~ left table and matched records from the right table.

• The result is none from the right side, if there's no match

• Syntax:

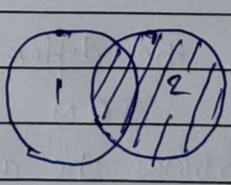
```
Select col_name(s) from table1
LEFT OUTER JOIN table2 ON
table1.col_name = table2.col_name;
```



iii) RIGHT OUTER JOIN: This keywords returns all records from the right table and matched records from left table. The result is null for left side when there's no match.

• Syntax:

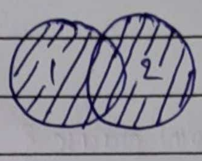
```
SELECT col_name(s) FROM table1
RIGHT OUTER JOIN table2 ON
table1.col_name = table2.col_name
```



iv) FULL OUTER JOIN: It returns all records from both sides when there's a match in either side.

4> FULL ~~O~~OUTER JOIN : It returns all records from both sides when there's a match in either side.

- Syntax : SELECT colname(s) from table1
  FULL OUTER JOIN table2 ON
  table1.column = table2.column



Q3) What's cross Join?

A.3) • CROSS JOIN produces a result set which's the product of 2 rows of 2 associated tables when no 'where' clause is used with CROSS JOIN.

- The result is known as a cartesian product.
- Syntax: SELECT * from table1 CROSS JOIN table2;

Q4) What is NULL value and how is it different from zero value?

A.4) NULL value is used to represent a missing value. A NULL value in a table, is the value in a field that appears to be blank. A field with zero value is not blank and therefore, a NULL value is different from zero value.

Q5) What are the diff MYSQL constraints?

A.5) Commonly used MYSQL constraints

1) NOT NULL: Ensures that column cannot have NULL value.
2) ~~UNQ~~ UNIQUE: Ensures values in a column are unique.
3) PRIMARY KEY: A combination of not null and unique.
4) FORFIGN KEY: Identifies a row in another table uniquely.
5) DEFAULT: Sets a default value for column when no value is specified.

Q6> What's the difference between ~~jon~~ join and union?

A.6

JOIN	UNION
1) Join combines the attributes of the tuples present in 2 different relations ~~sh~~ that share some common attributes.	1) Union combines tuples of the relation that are present in the query.

2) It's applicable when the two involved relations have atleast one common attribute.	2) Applicable when no of columns present query are same and corresponding attributes have same domain.
3) The length of resultant tuples is more as compared to length of tuples involved relations.	3) The no of resultant tuples is more as compared to no of tuples in each involved relation.

Q7) List different aggregation function in MYSQL.

A.7) i) AVG()

ii> COUNT()

iii> MAX()

iv> MIN()

v> SUM()

Q8) Differentiate between unique key and primary key?

A.8) Primary key	Unique key
1) Can't accept NULL values.	1) Can accept only one NULL value.
2) By default, it's a clustered index.	2) By default, it's a non-clustered index.
3) Only one per table.	3) More than one pertable is allowed.
4) Can be made foriegn key into another table.	4) Can be made a foreign key into another table.

Q9) Explain 'where' and 'having' clause.

A.9) i> WHERE: • It's used to filter only those records that are fulfilled by a specific condition given by the user.

- It's used to restrict no of rows displayed.
- Syntax: SELECT column name(s) from table.name
  WHERE condition;

ii> HAVING: • It's used in SELECT statement to specify filter conditions for a group of rows or aggregates. It's often used with GROUP BY clause to filter groups based on a specific condition.

- If 'group by' clause is emitted, HAVING behaves like a WHERE clause.
- Syntax: Select col-name(s) from table-name
  WHERE condition GROUP BY col_name(s)
  HAVING condition;