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

Table Structure----------

```
select * from account;
+-------------+--------------+---------+
| Acc_no      | branch_name  | balance |
+-------------+--------------+---------+
| C8382381670 | Akurdi       |   13400 |
| C8382381671 | Bund Garden  |   10000 |
| C8392380567 | Vishrantwadi |    5500 |
| C8392380568 | Viman Nagar  |    6500 |
| C8392380570 | Akurdi       |   12500 |
| C8392380670 | Akurdi       |   10500 |
| C8392381670 | Dhanori      |   10500 |
| C8482381665 | Bund Garden  |   55004 |
| C8482381671 | Akurdi       |   50000 |
| C8482381765 | Viman Nagar  |   75004 |
| C8482381766 | Nigdi        |   34000 |
+-------------+--------------+---------+

select * from branch;
+--------------+------------------+----------+
| branch_name  | branch_city      | assets   |
+--------------+------------------+----------+
| Akurdi       | Pimpri-Chinchwad |  6000000 |
| Bund Garden  | Pune             | 12000000 |
| Dhanori      | Mumbai           |  8000000 |
| Nigdi        | Pune             |   120000 |
| Viman Nagar  | Mumbai           | 13000000 |
| Vishrantwadi | Pune             |  9000000 |
+--------------+------------------+----------+

select * from customer;
+-----------+----------------+------------------+
| cust_name | cust_street    | cust_city        |
+-----------+----------------+------------------+
| Baljeet   | Station Road   | Pimpri-Chinchwad |
| Harshika  | Symboisis Road | Pune             |
| Kimaya    | Virar Road     | Mumbai           |
| Nachiket  | SNT Road       | Pimpri-Chinchwad |
| Noopur    | SNT Road       | Pimpri-Chinchwad |
| Noopur    | Station Road   | Pimpri-Chinchwad |
| Rajeev    | Marine Drive   | Mumbai           |
| Sherlock  | 221B-Baker Str | London           |
| Shyam     | Denshaw Road   | Mumbai           |
| Soham     | Law clg Road   | Pune             |
| Tejas     | Porwal Road    | Pune             |
+-----------+----------------+------------------+

select * from depositor;
+-----------+-------------+
| cust_name | Acc_no      |
+-----------+-------------+
| Shyam     | C8482381765 |
| Nachiket  | C8392380670 |
| Noopur    | C8392380570 |
```

```
| Soham     | C8382381671 |
| Rajeev    | C8382381670 |
| Tejas     | C8392380567 |
| Harshika  | C8392380568 |
| Rajeev    | C8392381670 |
| Abhish    | C8482381671 |
| Sherlock  | C8482381766 |
+-----------+-------------+

select * from loan;
+---------+-------------+--------+
| loan_no | branch_name | amount |
+---------+-------------+--------+
| loan123 | Viman Nagar |  35000 |
| loan124 | Bund Garden |  40000 |
| loan125 | Akurdi      |  15000 |
| loan126 | Akurdi      |  11000 |
| loan127 | Viman Nagar |  45000 |
+---------+-------------+--------+

select * from borrower;
+-----------+---------+
| cust_name | loan_no |
+-----------+---------+
| Shyam     | loan123 |
| Soham     | loan124 |
| Noopur    | loan125 |
| Abhish    | loan126 |
| Sumedh    | loan127 |
+-----------+---------+

--------------------------------------------------------------------------------
-

1.SELECT distinct branch_name from loan;
Output:
+-------------+
| branch_name |
+-------------+
| Viman Nagar |
| Bund Garden |
| Akurdi      |
+-------------+
--------------------------------------------------------------------------------
-

2.SELECT loan_no from Loan where branch_name = 'Akurdi' and amount>12000;
Output:
+---------+
| loan_no |
+---------+
| loan125 |
+---------+
--------------------------------------------------------------------------------
-

3.SELECT Borrower.cust_name, Borrowe.loan_no, Loan.amount from Borrower
JOIN Loan ON Borrower.loan_no = Loan.loan_no;
Output:
+-----------+---------+--------+
```

```
| cust_name | loan_no | amount |
+-----------+---------+--------+
| Shyam     | loan123 |  35000 |
| Soham     | loan124 |  40000 |
| Noopur    | loan125 |  15000 |
| Abhish    | loan126 |  11000 |
| Sumedh    | loan127 |  45000 |
+-----------+---------+--------+
```

4.SELECT Borrower.cust_name from Borrower JOIN Loan ON Borrower.Loan_no = Loan.Loan_no where Loan.branch_name='Akurdi' order by Borrower.cust_name asc;

Output:

```
+-----------+
| cust_name |
+-----------+
| Abhish    |
| Noopur    |
+-----------+
```

5.SELECT cust_name from Depositor
UNION
SELECT cust_name from Borrower;

Output:

```
+-----------+
| cust_name |
+-----------+
| Shyam     |
| Nachiket  |
| Noopur    |
| Soham     |
| Rajeev    |
| Tejas     |
| Harshika  |
| Abhish    |
| Sherlock  |
| Sumedh    |
+-----------+
```

6.SELECT Borrower.cust_name from Borrower JOIN Depositor ON Borrower.cust_name = Depositor.cust_name;

Output:

```
+-----------+
| cust_name |
+-----------+
| Shyam     |
| Noopur    |
| Soham     |
| Abhish    |
+-----------+
```

7.SELECT cust_name from Depositor

```
LEFT JOIN
Borrower USING(cust_name)
WHERE Borrower.cust_name IS NULL;
or
SELECT cust_name from Depositor where  cust_name NOT IN (SELECT cust_name
from Borrower);
Output:
+-----------+
| cust_name |
+-----------+
| Nachiket  |
| Rajeev    |
| Tejas     |
| Harshika  |
| Rajeev    |
| Sherlock  |
+-----------+
--------------------------------------------------------------------------
-

8.SELECT AVG(balance) from Account WHERE branch_name = 'Akurdi';
Output:
+--------------+
| AVG(balance) |
+--------------+
|   21600.0000 |
+--------------+
--------------------------------------------------------------------------
-

9.SELECT AVG(balance) as Avg,branch_name from Account group by
branch_name;
Output:
+------------+--------------+
| Avg        | branch_name  |
+------------+--------------+
| 21600.0000 | Akurdi       |
| 32502.0000 | Bund Garden  |
|  5500.0000 | Vishrantwadi |
| 40752.0000 | Viman Nagar  |
| 10500.0000 | Dhanori      |
| 34000.0000 | Nigdi        |
+------------+--------------+
--------------------------------------------------------------------------
-

10.SELECT COUNT(Depositor.Acc_no) as No_of_Depositor, Account.branch_name
from Depositor JOIN Account ON Depositor.Acc_no=Account.Acc_no group by
Account.branch_name;
Output:
+-----------------+--------------+
| No_of_Depositor | branch_name  |
+-----------------+--------------+
|               2 | Viman Nagar  |
|               4 | Akurdi       |
|               1 | Bund Garden  |
|               1 | Vishrantwadi |
|               1 | Dhanori      |
|               1 | Nigdi        |
+-----------------+--------------+
```

```
--------------------------------------------------------------------------------
-

11.SELECT branch_name from Account group by branch_name HAVING
AVG(balance)>12000;
Output:
+-------------+
| branch_name |
+-------------+
| Akurdi      |
| Bund Garden |
| Viman Nagar |
| Nigdi       |
+-------------+
--------------------------------------------------------------------------------
-

12.SELECT COUNT(*) from customer;
Output:
+----------+
| COUNT(*) |
+----------+
|       11 |
+----------+
--------------------------------------------------------------------------------
-

13.SELECT SUM(amount) from loan;
Output:
+-------------+
| SUM(amount) |
+-------------+
|      146000 |
+-------------+
--------------------------------------------------------------------------------
-

14.DELETE from Loan where amount BETWEEN 10000 AND 30000;
Table Before:
+---------+-------------+--------+
| loan_no | branch_name | amount |
+---------+-------------+--------+
| loan123 | Viman Nagar |  35000 |
| loan124 | Bund Garden |  40000 |
| loan125 | Akurdi      |  15000 |
| loan126 | Akurdi      |  11000 |
| loan127 | Viman Nagar |  45000 |
+---------+-------------+--------+
Table After:
+---------+-------------+--------+
| loan_no | branch_name | amount |
+---------+-------------+--------+
| loan123 | Viman Nagar |  35000 |
| loan124 | Bund Garden |  40000 |
| loan127 | Viman Nagar |  45000 |
+---------+-------------+--------+
Output:
Query OK, 2 rows affected (0.20 sec)
--------------------------------------------------------------------------------
-
```

15.DELETE from Branch,Loan,Account where amount branch_name='Nigdi';
Table before:
Account:-+-------------+-------------+---------+
| Acc_no | branch_name | balance |
+-------------+-------------+---------+
C8382381670	Akurdi	13400
C8382381671	Bund Garden	10000
C8392380567	Vishrantwadi	5500
C8392380568	Viman Nagar	6500
C8392380570	Akurdi	12500
C8392380670	Akurdi	10500
C8392381670	Dhanori	10500
C8482381665	Bund Garden	55004
C8482381671	Akurdi	50000
C8482381765	Viman Nagar	75004
C8482381766	Nigdi	34000
+-------------+-------------+---------+
Branch:+-------------+-----------------+----------+
| branch_name | branch_city | assets |
+-------------+-----------------+----------+
Akurdi	Pimpri-Chinchwad	6000000
Bund Garden	Pune	12000000
Dhanori	Mumbai	8000000
Nigdi	Pune	120000
Viman Nagar	Mumbai	13000000
Vishrantwadi	Pune	9000000
+-------------+-----------------+----------+
Loan:+---------+------------+--------+
| loan_no | branch_name | amount |
+---------+------------+--------+
loan123	Viman Nagar	35000
loan124	Bund Garden	40000
loan127	Viman Nagar	45000
+---------+------------+--------+
Tables After:
Account:+-------------+-------------+---------+
| Acc_no | branch_name | balance |
+-------------+-------------+---------+
C8382381670	Akurdi	13400
C8382381671	Bund Garden	10000
C8392380567	Vishrantwadi	5500
C8392380568	Viman Nagar	6500
C8392380570	Akurdi	12500
C8392380670	Akurdi	10500
C8392381670	Dhanori	10500
C8482381665	Bund Garden	55004
C8482381671	Akurdi	50000
C8482381765	Viman Nagar	75004
+-------------+-------------+---------+
Branch:+-------------+-----------------+----------+
| branch_name | branch_city | assets |
+-------------+-----------------+----------+
Akurdi	Pimpri-Chinchwad	6000000
Bund Garden	Pune	12000000
Dhanori	Mumbai	8000000
Viman Nagar	Mumbai	13000000
Vishrantwadi	Pune	9000000
+-------------+-----------------+----------+
Loan:+---------+------------+--------+

loan_no	branch_name	amount
loan123	Viman Nagar	35000
loan124	Bund Garden	40000
loan127	Viman Nagar	45000

16.CREATE VIEW cust as SELECT * from customer;
Select * from Cust;
Output:

cust_name	cust_street	cust_city
Baljeet	Station Road	Pimpri-Chinchwad
Harshika	Symboisis Road	Pune
Kimaya	Virar Road	Mumbai
Nachiket	SNT Road	Pimpri-Chinchwad
Noopur	SNT Road	Pimpri-Chinchwad
Noopur	Station Road	Pimpri-Chinchwad
Rajeev	Marine Drive	Mumbai
Sherlock	221B-Baker Str	London
Shyam	Denshaw Road	Mumbai
Soham	Law clg Road	Pune
Tejas	Porwal Road	Pune

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

DBMS-Assignment-02 & 03

- Question:

Q1) How can we make use of CREATE statements to create multiple object?

A-1) We can use CREATE statement to create multiple tables and views:

Eg:
```
CREATE TABLE dept (
    dept_name VARCHAR(10),
    building_no NUMBER,
    fees INTEGER);

CREATE VIEW faculty as
    SELECT id, name, dept_name
    FROM instructor;
```

Q2) What is a view? How can it be helpful to a user?

A-2) A view is a virtual table. It is a data object that doesn't contain any data, contents of a view are derived from a base table. They're operated just like the base table but dont contain any data of their own. The query stored in a view is computed everytime the view is refereced, thus we always recieve the consistent data directly from base table.

> Uses of views:

i) Security: When we want a user to access or view only some fields of a base table then views can be helpful.

ii) Query Simplicity: A view can contain data from several tables, thus it's helpful in converting multitable queries into single table queries.

iii) Structural Simplicity: Views can provide a user a personalised view of the DB, presenting the DB as virtual tables that make sense to that user.

iv) Consistency: Views present a consistent, unchanged view of the underlying tables even if source tables are split, restructured or renamed.

v) Data integrity: When data is modified via views, some integrity constraints are used as in the base tables

Q3) What's an index? What're the type of indices?

A.3) i) Index is a data structure that ~~implies~~ improves the speed of operations in a table.

ii) They can create one or more columns.

iii) Indexes are tables which keep primary key or index field as a pointer to reach a particular record in actual table.

iv) Syntax: CREATE INDEX index_name ON table_name (col_name(s));

> Types of indices

i) Single-column indices

ii) Unique-indices: Doesn't allow duplicate values, to be inserted into base table.

iii) Composite indices: multicolumn index.

Q4) What's a sequence? How's it generated in MYSQL?

A.4) A sequence is a set of int's that are generated and supported by some DBMS to produce unique values on demand.

2) Used in many apps that require each row in a table to contain a unique value and sequences provide an easy way to generate them.

3) A sequence in MYSQL is generated by setting the auto-~~generated~~ increment attribute to a column which generally a PK.

Eg: CREATE table student(
rollno INT AUTO-INCREMENT PRIMARY KEY;
name VARCHAR(20));

Q5) How to create synonyms in ~~any~~ MYSQL?

A.5) i) Synonyms can be created using create_synonym_db() procedure

ii) Given a schema name, this procedure creates a synonym schema containing views that refer to all the tables and views in the original schema.

iii) Parameters:

- in_db_name VARCHAR(64): The name of the schema for which to

create synonym.

- in-synonym VARCHAR(64): The name to use for synonym schema. This schema name mustn't already exist.

Q6) Which different commands are used to modify a db object?

A.6) i) ALTER: Command to make changes in an already existing db design.

ii) PROP: Command to delete a table, column.

iii) RENAME: Command to rename an existing db obj like tables or columns.

Q7) List down the different operators supported by MYSQL.

A.7) i> Comparison operators:

- Equality (=)
- IS and NULL-safe comparison
- IS and BOOLEAN comparison
- Greater than and less than >, <
- BETWEEN
- IN

ii> Logical operator:

NOT, AND, OR, XOR

iii> Arithmetic operator:

+, -, *, /

iv) TEXT operators:

- LIKE, SOUNDS LIKE, REGEXP

v) Bitwise operators

Q8) Differentiate between DELETE, DROP and TRUNCATE.

A.8) i> DELETE command is used to remove some/all rows from the table.

ii> TRUNCATE removes all rows from the table.

iii> DROP removes a table from the DB.

DROP and TRUNCATE are DDL commands whereas DELETE is DML