**Assignment No : 2 (SQL Queries)**

**Consider following Database Schemas**

1. **Branch (branch\_name, branch\_city, assets)**
2. **Account (acc\_no, *branch\_name*, balance)**
3. **Customer (cust\_name, cust\_street, cust\_city)**
4. **Depositor (*cust\_name, acc\_no*)**
5. **Loan (loan\_no, *branch\_name*, amount)**
6. **Borrower (*cust\_name, loan\_no*)**

**Assignment No 2 Part A:**

Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym, different constraints like primary key, foreign key, check constrains, not null, default etc.

**Assignment No 2 Part B:**

Demonstrate the use of concepts like Insert, Select, Update, Delete with operators, functions, and set operator etc. for above mentioned Bank Database Schema and Solve following SQL queries on the Bank database using SQL DML statements.

1. Find the names of all branches in loan relation.
2. Find all loan numbers for loans made at Swargate Branch with loan amount > 25000.
3. Find all customers who have an account or loan or both at bank.
4. Find all customers who have both account and loan at bank.
5. Calculate total loan amount given by bank.
6. Calculate number of accounts at a given bank.
7. Change the value of loan amount to 18000 whose loan\_no is 202206.
8. List all customers in alphabetical order who have loan from Swargate branch.
9. Find the average account balance at each branch.
10. Delete all loans with loan amount between 1200 and 1500.

**Theory**:

A database-management system (DBMS) is a collection of interrelated data and a set of programs to access those data. The collection of data, usually referred to as the database, contains information relevant to an enterprise. The primary goal of a DBMS is to provide a way to store and retrieve database information that is both *convenient* and *efficient*.

**Database Applications:**

* Banking: transactions
* Airlines: reservations, schedules
* Universities: registration, grades
* Sales: customers, products, purchases
* Online retailers: order tracking, customized recommendations
* Manufacturing: production, inventory, orders, supply chain
* Human resources: employee records, salaries, tax deductions

A Database Management System(DBMS) handles the way data is stored, maintained, and retrieved. In the case of a relational database, a Relational Database Management System (RDBMS) performs these tasks.

A relational database management system (RDBMS) is a program that lets you create, update, and administer a relational database. Most commercial RDBMS’s use the Structured Query Language (SQL) to access the database, although SQL was invented after the development of the relational model and is not necessary for its use.

MySQL open source RDBMS overview: MySQL is a popular open source relational database management system (RDBMS) choice for web-based applications.



**DATA DEFINITION LANGUAGE (DDL) QUERIES**

**DDL-**Data Definition Language (DDL) statements are used to define the database structure or schema. Data Definition Language understanding with database schemas and describes how the data should consist in the database, therefore language statements like CREATE TABLE or ALTER TABLE belongs to the DDL. DDL is about “metadata”.

DDL includes commands such as CREATE, ALTER and DROP statements. DDL is used to CREATE, ALTER OR DROP the database objects (Table, Views, Users).

**Data Definition Language (DDL) are used different statements :**

 CREATE –to create objects in the database

 ALTER –alters the structure of the database

 DROP –delete objects from the database

 TRUNCATE –remove all records from a table, including all spaces allocated for the records are removed

 COMMENT –add comments to the data dictionary

 RENAME –rename an object

**CREATE Command**

CREATE is a DDL command used to create databases, tables, triggers and other database objects.

Syntax to Create a Database:

CREATE Database Database\_Name;

Suppose, you want to create a Books database in the SQL database. To do this, you have to write the following DDL Command:

Create Database Books;

Syntax to create a new table:

CREATE TABLE table\_name

(

column\_Name1 data\_type ( size of the column ) ,

column\_Name2 data\_type ( size of the column) ,

column\_Name3 data\_type ( size of the column) ,

...

column\_NameN data\_type ( size of the column )

) ;

Suppose, you want to create a Student table with five columns in the SQL database. To do this, you have to write the following DDL command:

CREATE TABLE Student

(

Roll\_No. Int ,

First\_Name Varchar (20) ,

Last\_Name Varchar (20) ,

Age Int ,

Marks Int ,

) ;

**Constraints**

Constraints are the set of rules defined on tables to ensure data integrity.

1. Unique
2. Not null
3. Primary key
4. Default
5. Check
6. Foreign key/reference key

The **UNIQUE** constraint in MySQL does not allow to insert a duplicate value in a column.

Syntax

CREATE TABLE table\_name (col\_name data\_type (size) Unique);

Example

CREATE TABLE Stud (Rno number(4) Unique);

A **NOT NULL** constraint means that a data row must have a value for the column specified as NOT NULL.

Syntax

CREATE TABLE table\_name

(Col\_name Data\_type(size)not null, ... );

Example

Create table stud

(rollno number(4) ,name varchar2(20)not null);

**Primary key**

Each table must normally contain a column or set of columns that uniquely identifies rows of data that are stored in the table. This column or set of columns is referred to as the primary key.

A table can have only one primary key.

Multiple columns can be clubbed under a composite primary key.

Primary key columns is combination of NOT NULL and UNIQUE.

Syntax

CREATE TABLE table\_name ( Col\_name Data\_type(size)CONSTRAINT constraint\_name PRIMARY KEY, ... );

Example

Create table stud (rollno number(4)constraint pk1 primary key,name…)

Create table stud (rollno number(4) primary key, name ….)

In a MySQL table, while inserting data into a table, if no value is supplied to a column, then the column gets the value set as **DEFAULT**.

Syntax

CREATE TABLE table\_name(col\_name

data\_type(size) DEFAULT ‘default\_value’ );

Example

CREATE TABLE Stud (rno number(4) ,name varchar2(20), addr varchar(30) DEFAULT ‘Pune’ );

In a MySQL table, A CHECK constraint controls the values in the associated column. The **CHECK** constraint determines whether the value is valid or not.

Syntax

CREATE TABLEtable\_name(col\_name data\_type(size) Check (condition) );

Example

CREATE TABLE Stud(rno number(4) CHECK (rollno BETWEEN 1 AND 60));

CREATE TABLE Stud(age number(4) CHECK (age>18));

A **FOREIGN** KEY in MySQL creates a link between two tables by one specific column of both tables. The specified column in one table must be a PRIMARY KEY and referred by the column of another table known as FOREIGN KEY.

Syntax

Create table table\_name(col\_name data\_type(size)references table\_name(col\_name));

Example

Createtable stud1 (rollno number(4) references stud(rno));

You can also add constraint after table creation using alter table option

Example

Alter table stud add constraint prk1 primary key(rollno);

You can also drop constraint using Drop command & name of constraint

Example

Drop constraint prk1;

**View**

View is a logical table. It is a physical object which stores data logically. View just refers to data that is stored in base tables.

A view is a logical entity. It is a SQL statement stored in the database in the system tablespace. Data for a view is built in a table created by the database engine in the TEMP tablespace.

**Create View Syntax**

Create View view\_name as select col\_name1,col\_name2 from table\_name [where <condition>]

Example

1>Create view v1 as select name from stud;

2>Create view v2 as select name from stud where addr=‘Nashik’;

**Show View Syntax**

Select col\_name1,.. from View\_name [where condition]

Example

Select \* from v1;

**Drop View Syntax**

Drop View view\_name

Example

Drop View v1;

**Index**

Database index, or just index, helps speed up the retrieval of data from tables. When you query data from a table, first MySQL checks if the indexes exist, then MySQL uses the indexes to select exact physical corresponding rows of the table instead of scanning the whole table.

**Create Index Syntax**

qzz1> Create Index index\_name on table\_name(column\_name)

2> Alter table table\_name add index index\_name (column\_name)

Example

1> Create Index n1 on Stud(Name)

2> Alter table Stud add Index n1 (name)

**Show index Syntax**

Show Index from table\_name

Example

Show Index from Stud;

**Drop Index Syntax**

Alter table table\_name drop Index index\_name

Example

Alter table Stud drop Index n1;

**Sequence**

you can create a column that contains a **sequence** of numbers (1, 2, 3, and so on) by using the AUTO\_INCREMENT attribute.

Syntax

CREATE TABLE table\_name ( column1 datatype NOT NULL AUTO\_INCREMENT,….);

ALTER TABLE table\_name AUTO\_INCREMENT = start\_value;

Example

CREATE TABLE Bills ( Bill\_No INT(11) NOT NULL AUTO\_INCREMENT, name varchar2(20));

ALTER TABLE Bills AUTO\_INCREMENT = 1001

**synonym**

A synonym is an alternative name for objects such as tables, views, sequences, stored procedures, and other database objects.

Eg. Nickname or short name of any person

Note: Synonyms are not possible in MySQL but possible with oracle.

**DML Commands:**

For theory, refer the uploaded PPT on MS Teams (DML PPT.pptx)

**Some of The Most Important SQL Commands**

SELECT - extracts data from a database

UPDATE - updates data in a database

DELETE - deletes data from a database

INSERT INTO - inserts new data into a database

CREATE DATABASE - creates a new database

ALTER DATABASE - modifies a database

CREATE TABLE - creates a new table

ALTER TABLE - modifies a table

DROP TABLE - deletes a table

CREATE INDEX - creates an index (search key)

DROP INDEX - deletes an index

**Assignment No 2 Part A & B: Solution**

mysql> create database **bank**;

Query OK, 1 row affected (1.63 sec)

mysql> show databases;

+--------------------+

| Database |

+--------------------+

| bank |

| batchc |

| bvcoel |

| db1 |

| db2 |

| information\_schema |

| mysql |

| performance\_schema |

| sys |

+--------------------+

10 rows in set (1.19 sec)

mysql> create table **Branch**(branch\_name varchar(25) primary key,branch\_city varchar(20) not null,assets integer);

Query OK, 0 rows affected (16.41 sec)

mysql> desc Branch;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| branch\_name | varchar(25) | NO | PRI | NULL | |

| branch\_city | varchar(20) | NO | | NULL | |

| assets | int | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

3 rows in set (2.46 sec)

mysql> create table **Account**(acc\_no bigint auto\_increment primary key,branch\_name varchar(25) not null,balance integer default 500 check (balance >= 500), foreign key (branch\_name) references Branch(branch\_name));

Query OK, 0 rows affected (2.65 sec)

mysql> alter table account auto\_increment=1;

Query OK, 0 rows affected (0.36 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> desc account;

+-------------+-------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+----------------+

| acc\_no | bigint | NO | PRI | NULL | auto\_increment |

| branch\_name | varchar(25) | NO | MUL | NULL | |

| balance | int | YES | | 500 | |

+-------------+-------------+------+-----+---------+----------------+

3 rows in set (0.00 sec)

mysql> create table **customer**(cust\_name varchar(25) primary key, cust\_city varchar(15) not null);

Query OK, 0 rows affected (2.51 sec)

mysql> desc customer;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| cust\_name | varchar(25) | NO | PRI | NULL | |

| cust\_city | varchar(15) | NO | | NULL | |

+-----------+-------------+------+-----+---------+-------+

2 rows in set (0.01 sec)

mysql> create table **depositor**(cust\_name varchar(25), acc\_no bigint,foreign key (cust\_name) references customer(cust\_name),foreign key (acc\_no) references account(acc\_no));

Query OK, 0 rows affected (3.29 sec)

mysql> desc depositor;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| cust\_name | varchar(25) | YES | MUL | NULL | |

| acc\_no | bigint | YES | MUL | NULL | |

+-----------+-------------+------+-----+---------+-------+

mysql> create table **loan**(loan\_no bigint auto\_increment primary key, branch\_name varchar(25),loan\_amount int, foreign key (branch\_name) references branch(branch\_name));

Query OK, 0 rows affected (1.62 sec)

mysql> alter table loan auto\_increment=202201;

Query OK, 0 rows affected (0.26 sec)

Records: 0 Duplicates: 0 Warnings: 0

mysql> desc loan;

+-------------+-------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+----------------+

| loan\_no | bigint | NO | PRI | NULL | auto\_increment |

| branch\_name | varchar(25) | YES | MUL | NULL | |

| loan\_amount | int | YES | | NULL | |

+-------------+-------------+------+-----+---------+----------------+

3 rows in set (0.01 sec)

mysql> create table **borrower**(cust\_name varchar(25), loan\_no bigint, foreign key (cust\_name) references customer(cust\_name), foreign key (loan\_no) references loan(loan\_no));

Query OK, 0 rows affected (4.07 sec)

mysql> desc borrower;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| cust\_name | varchar(25) | YES | MUL | NULL | |

| loan\_no | bigint | YES | MUL | NULL | |

+-----------+-------------+------+-----+---------+-------+

2 rows in set (0.18 sec)

**INSERT VALUES**

mysql> desc branch;

+-------------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+-------+

| branch\_name | varchar(25) | NO | PRI | NULL | |

| branch\_city | varchar(20) | NO | | NULL | |

| assets | int | YES | | NULL | |

+-------------+-------------+------+-----+---------+-------+

3 rows in set (0.00 sec)

mysql> insert into branch values('Katraj','Pune',100000000);

Query OK, 1 row affected (0.13 sec)

mysql> insert into branch values('Swargate','Pune',1055000000);

Query OK, 1 row affected (0.24 sec)

mysql> insert into branch values('Kothrud','Pune',1250000000);

Query OK, 1 row affected (0.17 sec)

mysql> insert into branch values('Hadapsar','Pune',1950000000);

Query OK, 1 row affected (0.14 sec)

mysql> select \* from branch;

+-------------+-------------+------------+

| branch\_name | branch\_city | assets |

+-------------+-------------+------------+

| Hadapsar | Pune | 1950000000 |

| Katraj | Pune | 100000000 |

| Kothrud | Pune | 1250000000 |

| Swargate | Pune | 1055000000 |

+-------------+-------------+------------+

4 rows in set (0.00 sec)

---------------------------------------------------------------------------------------------------------------------------->

mysql> desc account;

+-------------+-------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+----------------+

| acc\_no | bigint | NO | PRI | NULL | auto\_increment |

| branch\_name | varchar(25) | NO | MUL | NULL | |

| balance | int | YES | | 500 | |

+-------------+-------------+------+-----+---------+----------------+

3 rows in set (0.00 sec)

mysql> insert into account(branch\_name, balance) values('Hadapsar',1000);

Query OK, 1 row affected (0.15 sec)

mysql> insert into account(branch\_name, balance) values('Swargate',50000);

Query OK, 1 row affected (1.55 sec)

mysql> insert into account(branch\_name, balance) values('Katraj',15000);

Query OK, 1 row affected (0.52 sec)

mysql> insert into account(branch\_name, balance) values('Kothrud',28000);

Query OK, 1 row affected (0.13 sec)

mysql> insert into account(branch\_name, balance) values('Kothrud',90000);

Query OK, 1 row affected (0.18 sec)

mysql> insert into account(branch\_name, balance) values('Swargate',100000);

Query OK, 1 row affected (0.23 sec)

mysql> insert into account(branch\_name, balance) values('Kothrud',50000);

Query OK, 1 row affected (0.13 sec)

mysql> insert into account(branch\_name, balance) values('Katraj',250000);

Query OK, 1 row affected (0.25 sec)

mysql> insert into account(branch\_name, balance) values('Swargate',1100000);

Query OK, 1 row affected (0.13 sec)

mysql> insert into account(branch\_name, balance) values('Hadapsar',50000);

Query OK, 1 row affected (0.20 sec)

mysql> insert into account(branch\_name, balance) values('Pune');

ERROR 1136 (21S01): Column count doesn't match value count at row 1

mysql> insert into account(branch\_name) values('Pune');

ERROR 1452 (23000): Cannot add or update a child row: a foreign key constraint fails (`bank`.`account`, CONSTRAINT `account\_ibfk\_1` FOREIGN KEY (`branch\_name`) REFERENCES `branch` (`branch\_name`))

mysql> insert into account(branch\_name) values('Katraj');

Query OK, 1 row affected (0.17 sec)

mysql> insert into account(branch\_name,balance) values('Katraj',400);

ERROR 3819 (HY000): Check constraint 'account\_chk\_1' is violated.

mysql> select \* from account;

+--------+-------------+---------+

| acc\_no | branch\_name | balance |

+--------+-------------+---------+

| 1 | Hadapsar | 1000 |

| 2 | Swargate | 50000 |

| 3 | Katraj | 15000 |

| 4 | Kothrud | 28000 |

| 5 | Kothrud | 90000 |

| 6 | Swargate | 100000 |

| 7 | Kothrud | 50000 |

| 8 | Katraj | 250000 |

| 9 | Swargate | 1100000 |

| 10 | Hadapsar | 50000 |

| 12 | Katraj | 500 |

+--------+-------------+---------+

11 rows in set (0.00 sec)

---------------------------------------------------------------------------------------------------------------------------->

mysql> desc customer;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| cust\_name | varchar(25) | NO | PRI | NULL | |

| cust\_city | varchar(15) | NO | | NULL | |

+-----------+-------------+------+-----+---------+-------+

2 rows in set (0.00 sec)

mysql> insert into customer values('Ram','Pune');

Query OK, 1 row affected (0.27 sec)

mysql> insert into customer values('Suraj','Mumbai');

Query OK, 1 row affected (0.15 sec)

mysql> insert into customer values('Priya','Mumbai');

Query OK, 1 row affected (0.18 sec)

mysql> insert into customer values('Pooja','Nasik');

Query OK, 1 row affected (0.28 sec)

mysql> insert into customer values('Karan','Pune');

Query OK, 1 row affected (0.10 sec)

mysql> insert into customer values('Raman','Nagpur');

Query OK, 1 row affected (0.51 sec)

mysql> insert into customer values('Divya','Pune');

Query OK, 1 row affected (0.11 sec)

mysql> insert into customer values('Pihu','Nasik');

Query OK, 1 row affected (0.13 sec)

mysql> insert into customer values('Suman','Nagpur');

Query OK, 1 row affected (0.11 sec)

mysql> insert into customer values('Deepika','Pune');

Query OK, 1 row affected (0.08 sec)

mysql> Select \* from customer;

+-----------+-----------+

| cust\_name | cust\_city |

+-----------+-----------+

| Deepika | Pune |

| Divya | Pune |

| Karan | Pune |

| Pihu | Nasik |

| Pooja | Nasik |

| Priya | Mumbai |

| Ram | Pune |

| Raman | Nagpur |

| Suman | Nagpur |

| Suraj | Mumbai |

+-----------+-----------+

10 rows in set (0.00 sec)

--------------------------------------------------------------------------------------------------------------------------------->

mysql> desc depositor;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| cust\_name | varchar(25) | YES | MUL | NULL | |

| acc\_no | bigint | YES | MUL | NULL | |

+-----------+-------------+------+-----+---------+-------+

2 rows in set (0.00 sec)

mysql> insert into depositor values('Suman',2);

Query OK, 1 row affected (0.13 sec)

mysql> insert into depositor values('Divya',6);

Query OK, 1 row affected (0.12 sec)

mysql> insert into depositor values('Karan',8);

Query OK, 1 row affected (0.17 sec)

mysql> insert into depositor values('Ram',9);

Query OK, 1 row affected (0.10 sec)

mysql> insert into depositor values('Deepika',10);

Query OK, 1 row affected (0.11 sec)

mysql> select \* from depositor;

+-----------+--------+

| cust\_name | acc\_no |

+-----------+--------+

| Suman | 2 |

| Divya | 6 |

| Karan | 8 |

| Ram | 9 |

| Deepika | 10 |

+-----------+--------+

5 rows in set (0.00 sec)

mysql> desc loan;

+-------------+-------------+------+-----+---------+----------------+

| Field | Type | Null | Key | Default | Extra |

+-------------+-------------+------+-----+---------+----------------+

| loan\_no | bigint | NO | PRI | NULL | auto\_increment |

| branch\_name | varchar(25) | YES | MUL | NULL | |

| loan\_amount | int | YES | | NULL | |

+-------------+-------------+------+-----+---------+----------------+

3 rows in set (0.00 sec)

//Inserting multiple rows in a table

mysql> insert into loan(branch\_name,loan\_amount) values('Swargate',200000), ('Katraj',50000), ('Katraj',25000),('Hadapsar',500000), ('Swargate',100000), ('Swargate',20000),('Katraj',250000), ('Swargate',80000),('Hadapsar',2500),('Katraj',1500);

mysql> select \* from loan;

+---------+-------------+-------------+

| loan\_no | branch\_name | loan\_amount |

+---------+-------------+-------------+

| 202201 | Swargate | 200000 |

| 202202 | Katraj | 50000 |

| 202203 | Katraj | 25000 |

| 202204 | Hadapsar | 500000 |

| 202205 | Swargate | 100000 |

| 202206 | Swargate | 18000 |

| 202207 | Katraj | 250000 |

| 202208 | Swargate | 80000 |

| 202209 | Hadapsar | 2500 |

| 202210 | Katraj | 1500 |

+---------+-------------+-------------+

10 rows in set (0.00 sec)

mysql> desc borrower;

+-----------+-------------+------+-----+---------+-------+

| Field | Type | Null | Key | Default | Extra |

+-----------+-------------+------+-----+---------+-------+

| cust\_name | varchar(25) | YES | MUL | NULL | |

| loan\_no | bigint | YES | MUL | NULL | |

+-----------+-------------+------+-----+---------+-------+

2 rows in set (0.18 sec)

mysql> insert into borrower values('Deepika',202201);

Query OK, 1 row affected (0.32 sec)

mysql> insert into borrower values('Divya',202205);

Query OK, 1 row affected (0.18 sec)

mysql> insert into borrower values('Karan',202206);

Query OK, 1 row affected (0.19 sec)

mysql> insert into borrower values('Ram',202208);

Query OK, 1 row affected (0.12 sec)

mysql> insert into borrower values('Priya',202204);

Query OK, 1 row affected (0.31 sec)

mysql> select \* from borrower;

+-----------+---------+

| cust\_name | loan\_no |

+-----------+---------+

| Deepika | 202201 |

| Divya | 202205 |

| Karan | 202206 |

| Ram | 202208 |

| Priya | 202204 |

+-----------+---------+

5 rows in set (0.00 sec)

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

**1. Find the names of all branches in loan relation.**

mysql> select distinct branch\_name from loan;

+-------------+

| branch\_name |

+-------------+

| Hadapsar |

| Katraj |

| Swargate |

+-------------+

3 rows in set (0.01 sec)

**2. Find all loan numbers for loans made at Swargate Branch with loan amount > 25000.**

mysql> select loan\_no from loan where branch\_name = 'Swargate' and loan\_amount > 25000;

+---------+

| loan\_no |

+---------+

| 202201 |

| 202205 |

| 202208 |

+---------+

3 rows in set (0.10 sec)

**3. Find all customers who have an account or loan or both at bank.**

mysql> (select cust\_name from depositor) union (select cust\_name from borrower);

+-----------+

| cust\_name |

+-----------+

| Deepika |

| Divya |

| Karan |

| Ram |

| Suman |

| Priya |

+-----------+

6 rows in set (0.14 sec)

**4. Find all customers who have both account and loan at bank.**

mysql> select distinct cust\_name from borrower where cust\_name in (select cust\_name from depositor);

+-----------+

| cust\_name |

+-----------+

| Deepika |

| Divya |

| Karan |

| Ram |

+-----------+

4 rows in set (0.12 sec)

**5. Calculate total loan amount given by bank.**

mysql> select sum(loan\_amount) from loan;

+------------------+

| sum(loan\_amount) |

+------------------+

| 1225000 |

+------------------+

1 row in set (0.06 sec)

**6. Calculate number of accounts at a given bank.**

**mysql> select count(\*) from account;**

**+----------+**

**| count(\*) |**

**+----------+**

**| 11 |**

**+----------+**

**1 row in set (0.24 sec)**

**7.** **Change the value of loan amount to 18000 whose loan\_no is 202206.**

mysql> select \* from loan;

+---------+-------------+-------------+

| loan\_no | branch\_name | loan\_amount |

+---------+-------------+-------------+

| 202201 | Swargate | 200000 |

| 202202 | Katraj | 50000 |

| 202203 | Katraj | 25000 |

| 202204 | Hadapsar | 500000 |

| 202205 | Swargate | 100000 |

| 202206 | Swargate | 20000 |

| 202207 | Katraj | 250000 |

| 202208 | Swargate | 80000 |

+---------+-------------+-------------+

8 rows in set (0.03 sec)

mysql> update loan set loan\_amount = 18000 where loan\_no = 202206;

Query OK, 1 row affected (0.72 sec)

Rows matched: 1 Changed: 1 Warnings: 0

mysql> select \* from loan;

+---------+-------------+-------------+

| loan\_no | branch\_name | loan\_amount |

+---------+-------------+-------------+

| 202201 | Swargate | 200000 |

| 202202 | Katraj | 50000 |

| 202203 | Katraj | 25000 |

| 202204 | Hadapsar | 500000 |

| 202205 | Swargate | 100000 |

| 202206 | Swargate | 18000 |

| 202207 | Katraj | 250000 |

| 202208 | Swargate | 80000 |

+---------+-------------+-------------+

8 rows in set (0.41 sec)

**8. List all customers in alphabetical order who have loan from Swargate branch.**

mysql> select cust\_name from borrower, loan where borrower.loan\_no = loan.loan\_no and branch\_name = "Swargate" order by cust\_name;

+-----------+

| cust\_name |

+-----------+

| Deepika |

| Divya |

| Karan |

| Ram |

+-----------+

4 rows in set (0.22 sec)

**9. Find the average account balance at each branch**

mysql> select branch\_name, avg (balance) from account group by branch\_name ;

+-------------+---------------+

| branch\_name | avg (balance) |

+-------------+---------------+

| Hadapsar | 25500.0000 |

| Katraj | 88500.0000 |

| Kothrud | 56000.0000 |

| Swargate | 416666.6667 |

+-------------+---------------+

4 rows in set (0.47 sec)

**10. Delete all loans with loan amount between 1200 and 1500.**

mysql> select \* from loan;

+---------+-------------+-------------+

| loan\_no | branch\_name | loan\_amount |

+---------+-------------+-------------+

| 202201 | Swargate | 200000 |

| 202202 | Katraj | 50000 |

| 202203 | Katraj | 25000 |

| 202204 | Hadapsar | 500000 |

| 202205 | Swargate | 100000 |

| 202206 | Swargate | 20000 |

| 202207 | Katraj | 250000 |

| 202208 | Swargate | 80000 |

| 202209 | Hadapsar | 2500 |

| 202210 | Katraj | 1500 |

+---------+-------------+-------------+

10 rows in set (0.00 sec)

mysql> delete from loan where loan\_amount >= 1200 and loan\_amount <= 1500;

Query OK, 1 row affected (0.01 sec)

mysql> select \* from loan;

+---------+-------------+-------------+

| loan\_no | branch\_name | loan\_amount |

+---------+-------------+-------------+

| 202201 | Swargate | 200000 |

| 202202 | Katraj | 50000 |

| 202203 | Katraj | 25000 |

| 202204 | Hadapsar | 500000 |

| 202205 | Swargate | 100000 |

| 202206 | Swargate | 20000 |

| 202207 | Katraj | 250000 |

| 202208 | Swargate | 80000 |

| 202209 | Hadapsar | 2500 |

+---------+-------------+-------------+

9 rows in set (0.00 sec)

**Conclusion**:

In this assignment, we have studied and demonstrated various DDL and DML statements in SQL.