DBMS Assignment A2 and A3

create table Customer (cust_name varchar(15), cust_street varchar(15) , cust_city varchar(15));

create table Branch (branch_name varchar(20) , branch_city varchar(15) , assets int);

create table Account (acc_no int not NULL primary key , branch_name varchar(20) , balance int);

create table Depositer (cust_name varchar(15) , acc_no int , foreign key(acc_no) references Account(acc_no) on delete cascade on update cascade);

create table Loan (loan_no varchar(10) primary key , branch_name varchar(20) , amount int);

create table Borrower (cust_name varchar(15) , loan_no varchar(10) , foreign key(loan_no) references
Loan(loan_no) on delete cascade on update cascade);

Q1. Find the names of all branches in loan relation.

mysql> select branch_name from Loan;

```
+-------------+
| branch_name |
+-------------+
| mumbai      |
| pune        |
| pune        |
| nagpur      |
| indore      |
+-------------+
```

Q2.Find all loan numbers for loans made at Camp Branch with loan amount > 1200.

mysql> select loan_no from Loan where branch_name="pune" and amount > 1200;

```
+---------+
| loan_no |
+---------+
| 22156   |
| 22658   |
+---------+
```

Q3.Find all customers who have a loan from bank.

mysql> select B.cust_name from Borrower B,Loan L where B.loan_no=L.loan_no;

```
+-----------+
| cust_name |
+-----------+
| Dhruvil   |
| Gaurav    |
| Soham     |
+-----------+
```

Q4.Find their names,loan_no and loan amount.

mysql> select B.cust_name,B.loan_no,L.amount from Borrower B, Loan L where B.loan_no=L.loan_no;

```
+-----------+---------+--------+
| cust_name | loan_no | amount |
+-----------+---------+--------+
| Dhruvil   | 22156   |  25000 |
| Gaurav    | 11235   |  26000 |
| Soham     | 55457   |   9000 |
+-----------+---------+--------+
```

Q4. List all customers in alphabetical order who have loan from Camp branch.

mysql> select cust_name from Borrower where loan_no in(select loan_no from Loan where branch_name="Camp") order by cust_name;

```
+-----------+
| cust_name |
+-----------+
| Soham     |
+-----------+
```

Q5. 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 |
+-----------+
| Dhruvil   |
| Gaurav    |
| Sundar    |
```

```
| Golu       |
| Sonu       |
+-----------+
```

Q6. Find all customers who have both account and loan at bank.

mysql> select D.cust_name from Depositor D inner join Borrower B on
D.cust_name=B.cust_name;

```
+-----------+
| cust_name |
+-----------+
| Ram        |
| Gaurav     |
| Soham      |
+-----------+
```

Q7. Find all customer who have account but no loan at the bank.

mysql> select D.cust_name from Depositor D where cust_name not in (select cust_name from
Borrower);

```
+-----------+
| cust_name |
+-----------+
| Sundar     |
| Golu       |
+-----------+
```

Q8. Find average account balance at Camp branch.

mysql> select avg(balance) from Account where branch_name="mumbai";
```
+--------------+
| avg(balance) |
+--------------+
|   30000.0000 |
+--------------+
```

Q9. Find the average account balance at each branch
mysql> select branch_name, avg(balance) from Account group by branch_name;

```
+-------------+--------------+
| branch_name | avg(balance) |
+-------------+--------------+
| MG road     |   10000.0000 |
| nagpur      |   12500.0000 |
| Mumbai      |   20000.0000 |
| pune        |   30000.0000 |
| indore      |   14500.0000 |
+-------------+--------------+
```

Q10. Find no. of depositors at each branch.
mysql> select branch_name,count(D.acc_no) from Account A,Depositor D where A.acc_no = D.acc_no group by branch_name;

```
+-------------+-----------------+
| branch_name | count(D.acc_no) |
+-------------+-----------------+
| MG road     |               1 |
| nagpur      |               1 |
| Mumbai      |               1 |
| pune        |               1 |
| indore      |               1 |
+-------------+-----------------+
```

Q11. Find the branches where average account balance > 12000
mysql> select branch_name,avg(balance) Average_Balance from Account where balance > 12000 group by branch_name;

```
+-------------+-----------------+
| branch_name | Average_Balance |
+-------------+-----------------+
| nagpur      |      12500.0000 |
| Mumbai      |      20000.0000 |
| pune        |      30000.0000 |
| indore      |      14500.0000 |
+-------------+-----------------+
```

Q12. Find number of tuples in customer relation.

mysql> select count(cust_city) No_Of_Tuples from Customer;

```
+--------------+
| No_Of_Tuples |
```

```
+-------------+
|           5 |
+-------------+
```

Q13. Calculate total loan amount given by bank.

mysql> select sum(amount) Total_Loan from Loan;
```
+------------+
| Total_Loan |
+------------+
|     162700 |
+------------+
```

Q14. Delete all loans with loan amount between 1300 and 1500.

mysql> delete from Loan where amount between 20000 and 35000;
mysql> select * from Loan;

```
+---------+-------------+--------+
| loan_no | branch_name | amount |
+---------+-------------+--------+
| 22658   | pune        |  67500 |
| 33128   | nagpur      |  35200 |
| 55457   | indore      |   9000 |
+---------+-------------+--------+
```

Q15. Delete all tuples at every branch located in pune.

mysql> delete A,L from Account A inner join Loan L on A.branch_name=L.branch_name where A.branch_name = "pune";
mysql> select * from Account;

```
+--------+-------------+---------+
| acc_no | branch_name | balance |
+--------+-------------+---------+
| 112011 | MG road     |   10000 |
| 112701 | nagpur      |   12500 |
| 701905 | Mumbai      |   20000 |
| 741905 | indore      |   14500 |
+--------+-------------+---------+
```

mysql> select * from Loan;

```
+---------+-------------+--------+
| loan_no | branch_name | amount |
+---------+-------------+--------+
| 33128   | nagpur      |  35200 |
```

```
| 55457   | indore      |  9000 |
+---------+-------------+-------+
```

Q.16. Create synonym for customer table as cust_name.
mysql> create view cust_name as select * from customer;
mysql> select * from cust_name;

```
+-----------+----------------+-----------+
| cust_name | cust_street    | cust_city |
+-----------+----------------+-----------+
| Ram       | M G Road       | Pune      |
| shyam     | Phule Market   | Mumbai    |
| sundar    | ramtekari      | Nagpur    |
| golu      | atre road      | pune      |
| sonu      | mahakal highway | Indore   |
+-----------+----------------+-----------+
```

MES College of Engineering Pune-01

Department of Computer Engineering

Name of Student: Dhruvil Shah	Class: TE Comp 1
Semester/Year: 5th/2020	Roll No: 047
Date of Performance:	Date of Submission:
Examined By:	Experiment No: Part A-02 ,A-03

GROUP: A ASSIGNMENT NO: 02 & 3

AIM:

1. Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym.

2. Design at least 10 SQL queries for suitable database application using SQL DML statements

Insert, Select, Update, Delete with operators, functions, and set operator.

OBJECTIVES:

- To develop basic, intermediate and advanced Database programming skills.

- To develop basic Database administration skill.

APPRATUS:

- Operating System recommended: 64-bit Open source Linux or its derivative

- Front End: Java/PHP/Python

- Backend: MySql: 5.5.54

IMPLEMENTATION:

A. Account(Acc_no, branch_name,balance)

B. branch(branch_name,branch_city,assets)

C. customer(cust_name,cust_street,cust_city)

D. Depositor(cust_name,acc_no)

E. Loan(loan_no,branch_name,amount)

F. Borrower(cust_name,loan_no)

Solve following query: Create above tables with appropriate constraints like primary key, foreign key, unique, not null etc.
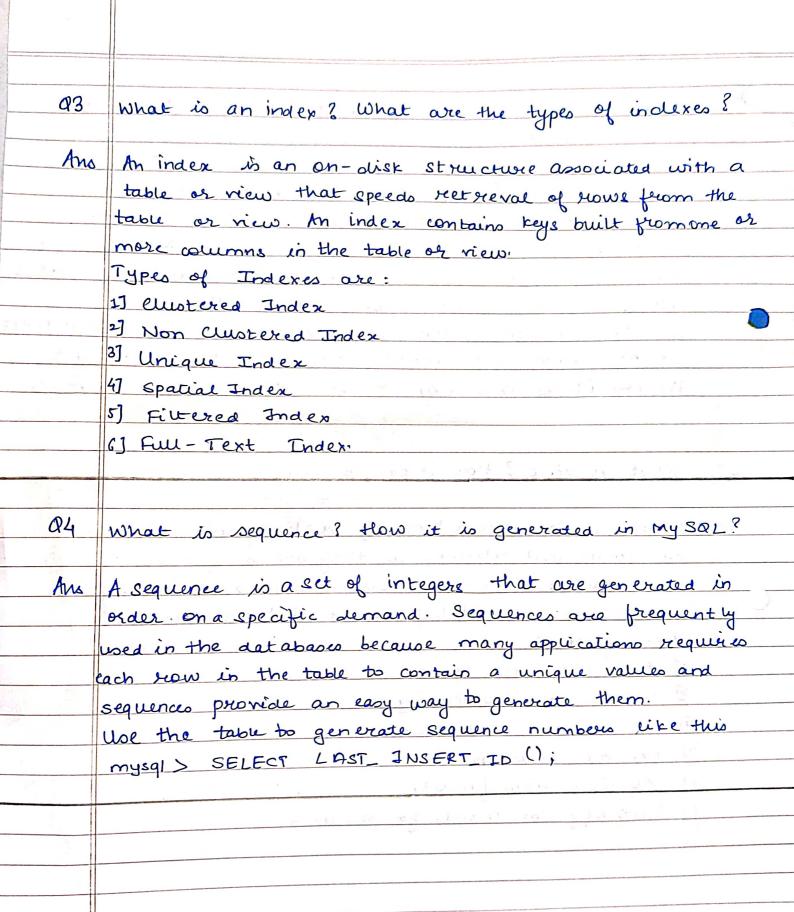
Q1. Find the names of all branches in loan relation.

Q2.Find all loan numbers for loans made at Akurdi Branch with loan amount > 12000.

Q3.Find all customers who have a loan from bank. Find their names,loan_no and loan amount.

Q4. List all customers in alphabetical order who have loan from Akurdi branch.

Q5. Find all customers who have an account or loan or both at bank.

Q6. Find all customers who have both account and loan at bank.

Q7. Find all customer who have account but no loan at the bank.

Q8. Find average account balance at Akurdi branch.

Q9. Find the average account balance at each branch

Q10. Find no. of depositors at each branch.

Q11. Find the branches where average account balance > 12000

Q12. Find number of tuples in customer relation.

Q13. Calculate total loan amount given by bank.

Q14. Delete all loans with loan amount between 1300 and 1500.

Q15. Delete all tuples at every branch located in Nigdi.

CONCLUSION:

QUESTIONS:

1. How we can make use of Create statement to create multiple objects?
2. What is view ? How it can helpful to user?
3. What is an Index?What are the types of indexes ?
4. What is Sequence ? How it is generated in MySql?
5. How to create synonyms in MySql?
6. Which are the different commands used to modify database object?
7. List down the different operators that support MySql
8. What is difference between Delete,Drop and Truncate?
9. List down different MySql functions.
10. Explain in details column level constraints in MySql.

Q1 How we can make use of create statement to create multiple objects ?

Ans You can create several tables and views and grant privileges in one operation using the CREATE SCHEMA statement. The CREATE SCHEMA statement is useful if you want to guarantee the creation of several tables, views and grants in one operation. Specifically, the CREATE SCHEMA statement can include only CREATE TABLE, CREATE VIEW and GRANT statements. You must have the privileges necessary to issue the included statements.

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

Ans Views in SQL are a kind of virtual tables. A view also has rows and columns as they are in a real table in the database. We can create a view by selecting fields from one or more tables present in the database.
Uses of VIEW :
1] Restrict data access
2] Hiding data complexity
3] Store complex queries
4] Rename columns.
5] Multiple view facility.
6] Simplify commands for the user.

Q3 What is an index? What are the types of indexes?

Ans An index is an on-disk structure associated with a table or view that speeds retrieval of rows from the table or view. An index contains keys built from one or more columns in the table or view.

Types of Indexes are:
1] Clustered Index
2] Non Clustered Index
3] Unique Index
4] Spatial Index
5] Filtered Index
6] Full-Text Index.

Q4 What is sequence? How it is generated in MySQL?

Ans A sequence is a set of integers that are generated in order. On a specific demand. Sequences are frequently used in the databases because many applications requires each row in the table to contain a unique values and sequences provide an easy way to generate them.

Use the table to generate sequence numbers like this

mysql > SELECT LAST_INSERT_ID ();

Q5 How to create synonym's in MYSQL?

Ans Use the CREATE SYNONYM statement to create synonym, which in an alternative name for a table, view, sequence. procedure, stored function, package, materialized view.
Syntax:
CREATE SYNONYM table_name 1 for table_name. schema

Q6 Which are the different commands used to modify database object?

Ans The different commands used to modify database object is ALTER.
Syntax:
ALTER TABLE table_name MODIFY "column-name" "New Data Type"

Q7 List down the different operators that support MYSQL

Ans Arithmetic Operators: '+', '-', '*', '/', '%'
Comparison operators: '>', '<', '=', '!=', '>=', '<=', '!>', '!<', '<>'.
Logical Operators: BETWEEN, EXISTS, OR, AND, NOT, IN, ALL, ANY, LIKE, UNIQUE

Q8 What is difference between Delete, Drop and Truncate.

Ans DROP :

Drop is a DDL command and is used to remove table definition and indexes, data, constraints, trigers, etc. that table.

Syntax :

DROP TABLE table-name;

TRUNCATE :

Truncate is DDL command and it is used to delete all the tuples from the table.

Syntax

TRUNCATE TABLE table-name.

DELETE :

Delete is a DML command and it is used to delete one or more tuples of a table.

Syntax

DELETE from ;

Q9 List down different MySOL functions

Ans 1] Avg ( )

2] MAX ( )

3] MIN ( )

4] SUM ( )

5] COUNT ( )

Q10 Explain in detail column level constraints in MYSQL.

Ans.

Constraint	Description
1) NOT NULL	It specifies that the column cannot contain a null value
2) UNIQUE	It specifies that a column cannot be inserted with duplicate values.
3) DEFAULT	It specifies a default value if no value is inserted.
4) PRIMARY KEY	This constraint for a table enforces the table to accept unique data for a specific column and creates a unique index.
5) FOREIGN KEY	Creates a link between two tables by one specific column of both tables.
6) CHECK	It determines whether the value is valid or not from a logical expression.