

MES College of Engineering Pune-01
Department of Computer Engineering

Name of Student:	Class:
Semester/Year:	Roll No:
Date of Performance:	Date of Submission:
Examined By:	Experiment No: Part A-02

PART: A) ASSIGNMENT NO: 02

AIM: SQL Queries:

- a. Design and Develop SQL DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym, different constraints etc.
- b. Write at least 10 SQL queries on the suitable database application using SQL DML statements.

Note: Instructor will design the queries which demonstrate the use of concepts like Insert, Select, Update, Delete with operators, functions, and set operator etc.

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.
- Back End: MySQL/ Oracle Database.

THEORY:

MySQL Introduction

- MySQL is an open source, fast, flexible, reliable, RDBMS being used for many small and big businesses.
- MySQL uses a standard form of the well-known SQL data language.
- MySQL is written in C, C++ and works on many operating systems and with many languages including PHP, PERL, C, C++, JAVA, etc.
- MySQL works very quickly and can handle large data sets.

- MySQL Server works in client/server or embedded systems.
- MySQL supports large databases, up to 50 million rows or more in a table. The default file size limit for a table is 4GB, but you can increase it to 8 million terabytes (TB).

1. MySQL Basic Commands

- **To Start MySQL**

```
#mysql -u username -p
```

Enter password:

- **To Access user on Client**

```
#mysql -h Host IP -P 3306 -u username -p
```

Enter password:

- **To Exit MySQL**

```
mysql>Exit; OR #Quit;
```

- **To check version of MySQL**

```
mysql>select version();
```

- **To check current date/time**

```
mysql>select current_date;
```

```
mysql>select now();
```

2. Create, Use, Display and remove a Database

- **To Create a Database**

```
mysql>create database [if not exists] database_name;
```

- **To Use a Database**

```
mysql>use database_name;
```

- **Displaying Databases**

SHOW DATABASES to list all the existing databases in the server.

```
mysql>show databases;
```

```
+-----+
| Database      |
+-----+
| information_schema |
| mysql          |
| performance_schema |
| test           |
.....
```

The databases "mysql", "information_schema" and "performance_schema" are system databases used internally by MySQL. A "test" database is provided during installation for your testing.

- **Removing Databases**

```
mysql>drop database [if exists] database_name;
```

3. MySQL Data Type

- Numeric Data Types
- String Data Types
- Date and Time Data Types

4. Creation of a Table

A table in MySQL is called a “RELATION” and consists of rows and columns addressed as “TUPLES” and “ATTRIBUTES” of the table respectively.

The number of tuples is called “CARDINALITY” and the number of attributes is referred to as “DEGREE” of a relation.

- **Simple Table Creation**

```
mysql>create table table_name (
    <column_name> <data_type>[(size)],
    <column_name> <data_type>[(size)] );
```

- **Creation of Table Using SQL Constraints**

```
mysql>Create table table_name (
    <column_name> <data_type>[(size)] <constraint> ,
    <column_name> <data_type>[(size)] <constraint> );
```

The various constraints that can be issued are:-

- NOT NULL: - Ensures that a column cannot have null values.
- DEFAULT: - Provides a default value for a column when none is specified.
- UNIQUE: - Ensures that all values in a column are different.
- CHECK: - Make sure all values in a column satisfy certain criteria.
- Primary Key: - Used to uniquely identify a row in a table.
- Foreign Key: - Used to ensure referential integrity of the data.

- **To check which table exist in current database**

```
mysql>show tables;
```

- **To view/display table structure**

```
mysql>describe table_name;
```

- **To Drop table**

```
mysql>drop table table_name;
```

- **Loading data from text file (*.txt) into table**

```
mysql>LOAD DATA LOCAL INFILE "Path and Filename" INTO TABLE
Table_name;
```

5. Insert Value in Table

```
mysql>INSERT INTO table_name VALUES (value1,value2,value3,...);
```

OR

```
mysql>INSERT INTO table_name (column1,column2,column3,...)
VALUES (value1, value2, value3,...);
```

6. Retrieving Data from an Existing Table

The Select statement is used to retrieve data from an existing table. There are two type of select statements that can be used:

- **Select Statement Without “Where” Clause**

```
mysql>Select <what_to_select> from <table_name>;
```

- **Select Statement With “Where” Clause**

```
mysql>Select <what_to_select> from <table_name> WHERE constraint;
```

Example: **Select * from pet WHERE pet_name='TOMMY'**; will retrieve all the tuples form the table named “pet” where the value under the field “pet_name”is “TOMMY”.

7. Update Value in table

```
mysql>UPDATE table_name
SET field1=new-value1, field2=new-value2
[WHERE Clause]
```

8. Delete Value from table

```
mysql>DELETE FROM table_name [WHERE Clause]
```

9. Alter Table

- **To add a field**

```
mysql>ALTER TABLE table_name
ADD new_column_name
[ FIRST | AFTER column_name ];
```

- **To modify the data type of a field**

```
mysql>ALTER TABLE table_name
MODIFY column_name <new-data-type>;
```

- **To delete a field**

```
mysql>ALTER TABLE table_name
DROP column_name;
```

- **To set a common value for a field(To set Default value)**

```
mysql>ALTER TABLE table_name ALTER Column_name SET DEFAULT value;
```

- **To change the name of a field**

```
mysql>ALTER TABLE table_name
CHANGE <old_Column_name> <new_column_name> <data-type> ;
```

- **To change the name of a table**

```
mysql>ALTER TABLE old_table_name
RENAME TO <new_table_name> ;
```

10. MySQL-View

- **Creating View**

```
#CREATE OR REPLACE
[ALGORITHM = {MERGE | TEMPTABLE | UNDEFINED}] VIEW
[database_name].[view_name]
AS [SELECT statement]

OR

#CREATE VIEW view_name AS
SELECT column_name(s)
FROM table_name
WHERE condition
```

11. MySQL Index

- **Creating A Mysql Index - New Table**

If you are creating a new MySQL table you can specify a column to index by using the INDEX term as we have below.

```
#CREATE TABLE table_name(col_name1,Col_name2, INDEX (col_name));
```

We have created two fields: name and employee ID (index).

```
#CREATE TABLE employee_records ( name VARCHAR(50), employeeID INT,
INDEX (employeeID));
```

- **Creating A Mysql Index - Existing Table**

You can also add an index to an older table that you think would benefit from some indexing. The syntax is very similar to creating an index in a new table. First, let's create the table.

```
#CREATE INDEX [index name] ON [table name]([column name]);
```

Example:

```
#CREATE TABLE employee_records2 (name VARCHAR(50), employeeID INT);
#CREATE INDEX id_index ON employee_records2(employeeID)
```

- **Unique Index**

```
CREATE UNIQUE INDEX index_name ON table_name ( column1, column2,...);
CREATE UNIQUE INDEX AUTHOR_INDEX ON tutorials_tbl (tutorial_author)
CREATE UNIQUE INDEX AUTHOR_INDEX ON tutorials_tbl (tutorial_author DESC)
```

- **ALTER command to add and drop INDEX:**

There are four types of statements for adding indexes to a table:

- Primary Key

```
#ALTER TABLE tbl_name ADD PRIMARY KEY (column_list):
```

- Unique Index

```
#ALTER TABLE tbl_name ADD UNIQUE index_name (col_list):
```

- Simple Index

```
#ALTER TABLE tbl_name ADD INDEX index_name (col_list):
```

- FULLTEXT index that is used for text-searching purposes

```
#ALTER TABLE tbl_name ADD FULLTEXT index_name (col_list):
```

- **Displaying INDEX Information:**

```
SHOW INDEX FROM table_name\G
```

IMPLEMENTATION:

Consider following relation and solve the queries: Create different tables given below with appropriate constraints like primary key, foreign key, check constraints, not null etc.

Account (**Acc_no**, **branch_name**, balance)

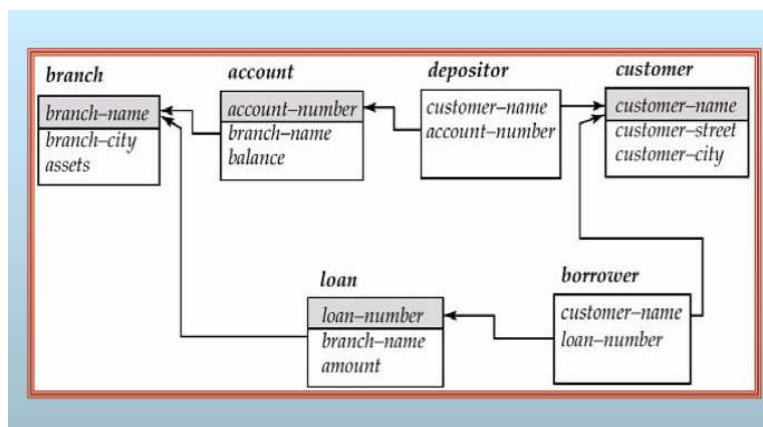
Branch (**branch_name**, branch_city, assets)

Customer (**cust_name**, cust_street, cust_city)

Depositor (**cust_name**, **acc_no**)

Loan (**loan_no**, **branch_name**, amount)

Borrower (**cust_name**, **loan_no**)



1. Find the names of all branches in loan relation.
2. Find all loan numbers for loans made at Akurdi Branch with loan amount > 12000.
3. Find all customers who have a loan from bank. Find their names, loan_no and loan amount.
4. List all customers in alphabetical order who have loan from Akurdi branch.
5. Find all customers who have an account or loan or both at bank.
6. Find all customers who have both account and loan at bank.
7. Find all customers who have account but no loan at the bank.
8. Find the average account balance at each branch
9. Find no. of depositors at each branch.
10. Find name of Customer and city where customer name starts with Letter P.
11. Display distinct cities of branch.
12. Find the branches where average account balance > 12000
13. Find number of tuples in customer relation.
14. Calculate total loan amount given by bank.
15. Delete all loans with loan amount between 1300 and 1500.
16. Delete all tuples at every branch located in Nigdi.

CONCLUSION:**QUESTIONS:**

1. What are different types of databases? Explain any one (open-source) database.
2. What are DDL, DML, DCL, and TCL Languages?
3. What are primary key, unique key and foreign key?
4. How we can make use of Create statement to create multiple objects?
5. What is View? How is can helpful to user?
6. What is an Index? What are the types of indexes?
7. What is Sequence? How it is generated in MySQL?
8. What are different the Query Optimization technique's?
9. How to create synonyms in MySQL?
10. Which are the different commands used to modify database object?
11. List down the different operators that support MySQL
12. What is difference between Delete, Drop and Truncate?