PUNE INSTITUTE OF COMPUTER TECHNOLOGY

Department of Computer Engineering

DBMSL (310247)

K-1 Batch Roll no - 31109

Name -Jayesh Gopal Bangad

**Assignment A2**

**Theory**

DDL stands for Data Definition Language. These commands are a subset of broader SQL language dialect. All relational databases have DDL commands for management of database objects.

DDL MySQL Commands create and drop database objects like the following:

 Tables

 Views

 Procedures

 Functions

 Indexes

 Triggers

The full list of DDL MySQL statements is as below:

DDL COMMANDS:

DDL is short form of Data Definition Language, which deals with data schemas and description, of how data can reside in database Various commands in DDL are:

1. Create :

Create table command defines each attribute uniquely.

Each attribute has 3 mandatory things.

(I) Attribute name

(II) Attribute size

(III) Data type Syntax:

Create table tablename (Attribute\_name attribute\_datatype(size),Attribute\_name attribute\_datatype(size),Attribute\_name attribute\_datatype(size).........n)

2. Alter :

By using ALTER command existing table can be modify.

Adding New Columns

Syntax: ALTER TABLE <table name>ADD (<Newcolumn name> <data\_type>(<size>),......n)

Dropping a Column from the Table

Syntax : ALTER TABLE DROP COLUMN \*This command will drop particular column.\*

Modifying Existing Table

Syntax: ALTER TABLE<table name> MODIFY ( <column name><new data type>(<new size>))

Restriction on the ALTER TABLE

1. Using the ALTER TABLE clause the following tasks cannot be performed. Change the name of the table

2. Change the name of the column

3. Decrease the size of a column if table data exists

Drop : The Drop command will destroy table along with the data enteries in it. Syntax: Drop Table <table\_name>

Truncate :

The truncate command deletes all entries existing in tables but keep the structure of table as described.

Syntax: Truncate Table <table\_name>

Rename:

The rename command is used to rename the table

Syntax: Rename <old\_name> <new\_name>

Creating Views:

Database views are created using the CREATE VIEW statement. Views can be created from a single table, multiple tables, or another view. To create a view, a user must have the appropriate system privilege according to the specific implementation.

The basic CREATE VIEW syntax is as follows:

CREATE VIEW view\_name AS SELECT column1, column2.....FROM table\_name WHERE [condition];

You can include multiple tables in your SELECT statement in very similar way as you use them in normal SQL SELECT query.

JDBC :-

The fundamental steps involved in the process of connecting to a database and executing a query consist of the following:

· Import JDBC packages.

· Load and register the JDBC driver.

· Open a connection to the database.

· Create a statement object to perform a query.

· Execute the statement object and return a query resultset.

· Process the resultset.

· Close the resultset and statement objects.

· Close the connection.

These steps are described in detail in the sections that follow

Import JDBC Packages

This is for making the JDBC API classes immediately available to the application program.

The following import statement should be included in the program irrespective of the JDBC driver being used:

import java.sql.\*;

Additionally, depending on the features being used, Oracle-supplied JDBC packages might need to be imported. For example, the following packages might need to be imported while using the Oracle extensions to JDBC such as using advanced data types such as BLOB, and so on. import oracle.jdbc.driver.\*;

import java.sql.\*;

Load and Register the JDBC Driver This is for establishing a communication between the JDBC program and the Oracle database.

This is done by using the static registerDriver() method of the DriverManager class of the JDBC API.

The following line of code does this job:

String url = "jdbc:mysql://192.168.5.101:3306/test";

String userName = "abc"; String password = "abc123";

DriverManager.getConnection(url, userName, password);

Java Database Connectivity JDBC stands for Java Database Connectivity.

JDBC is a Java API to connect and execute the query with the database. It is a part of JavaSE (Java Standard Edition).

JDBC API uses JDBC drivers to connect with the database.

There are four types of JDBC drivers:

o JDBC-ODBC Bridge Driver,

o Native Driver,

o Network Protocol Driver,

and o Thin Driver

A list of popular classes of JDBC API are given below:

 DriverManager class

 Blob class

 Clob class

 Types class

We can use JDBC API to handle database using Java program and can perform the following activities:

 Connect to the database

 Execute queries and update statements to the database

 Retrieve the result received from the database.

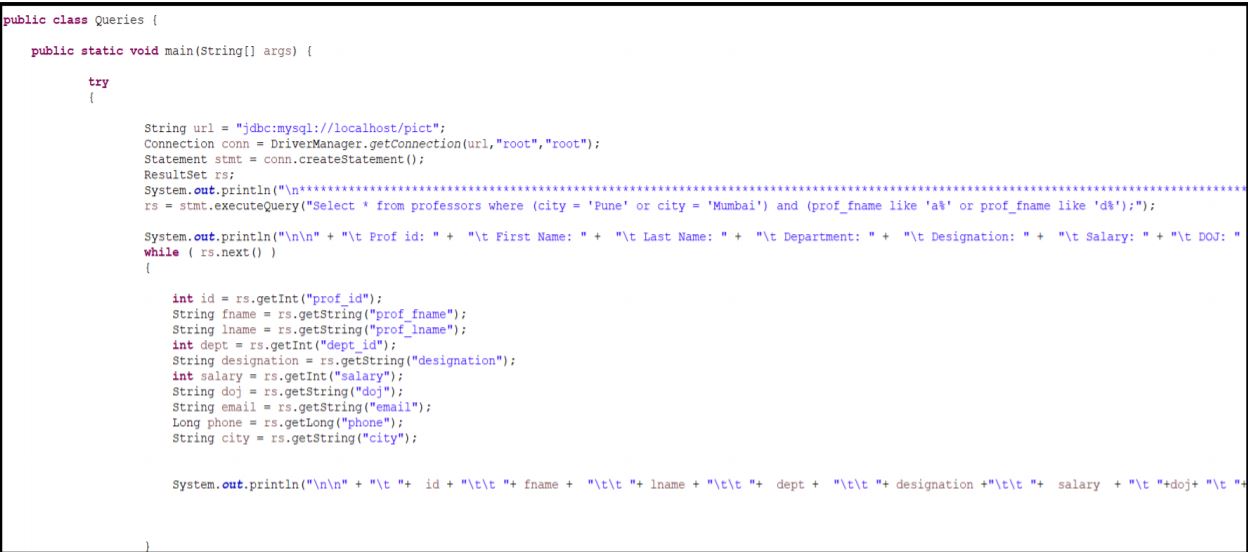
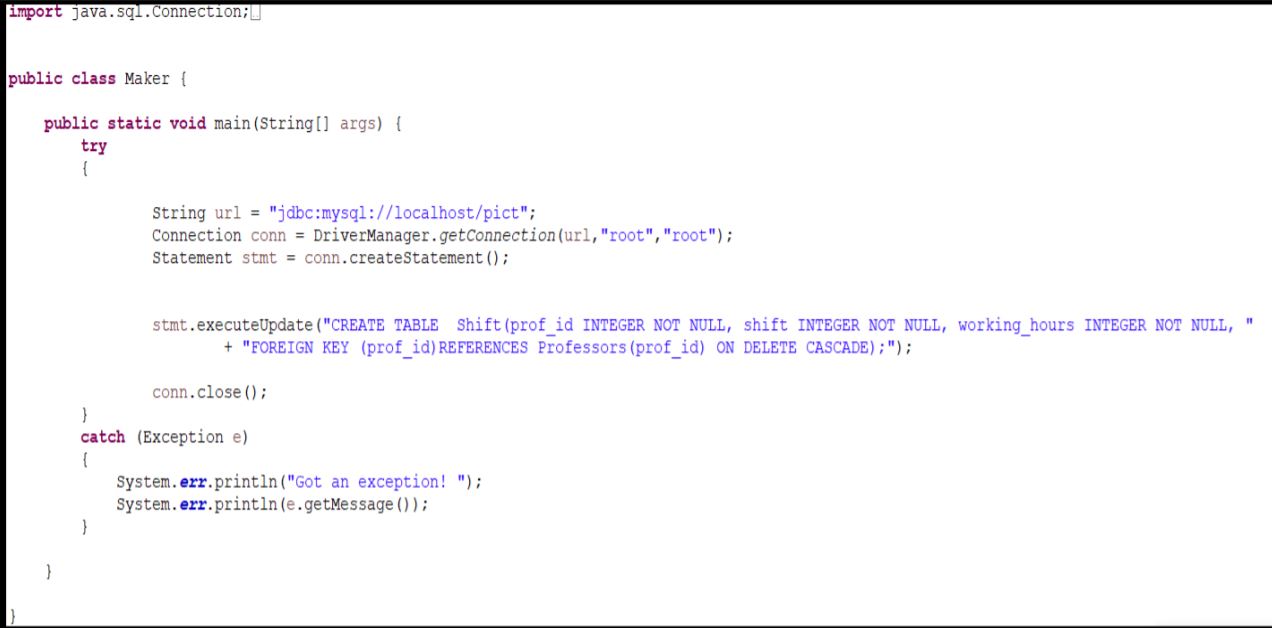
**Simple View**

When we create a view on a single table, it is called a simple view.

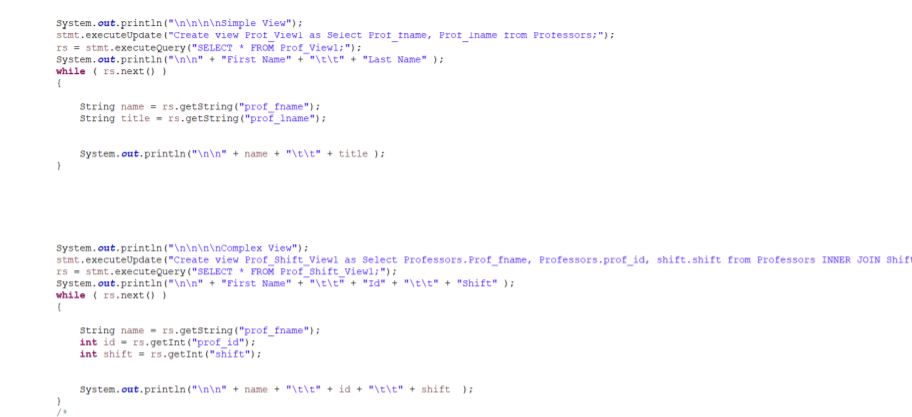
**Complex View**

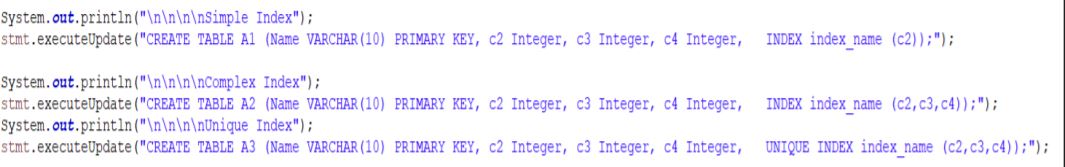
When we create a view on more than one table, it is called a complex view.

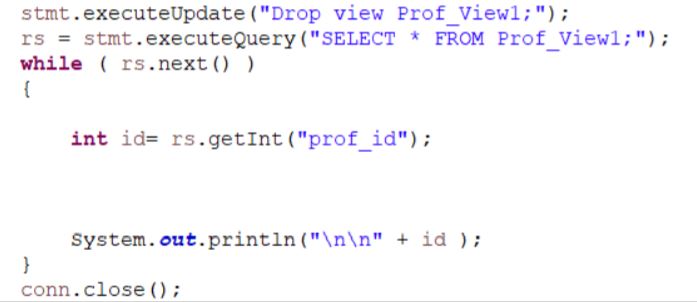
Source Code:

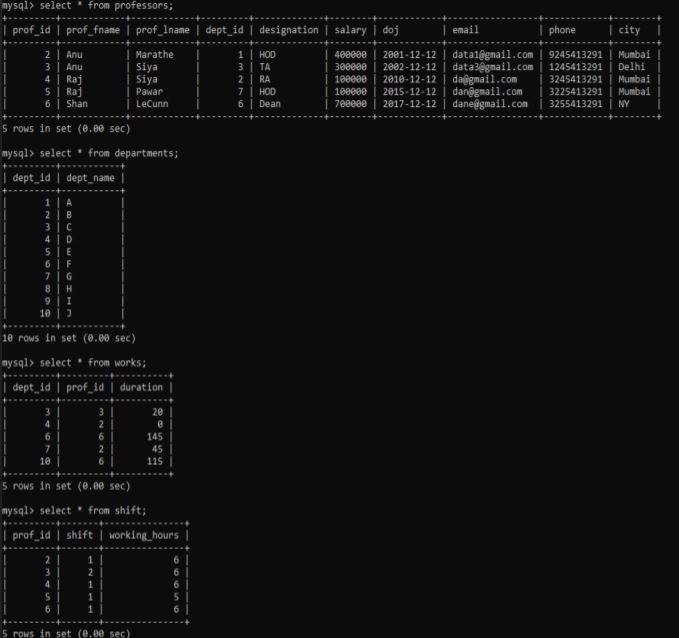




****

****

****

****

**Conclusion:**

Thus we have successfully learnt how to create tables and use DDL commands in MySQL, and JDBC connectivity and applied our knowledge on the real world example of books data system.