Database Management Systems Laboratory

Assignment A2

Date of Submission: 25-08-2020

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

Roll number: 31301

Name: Aboli Marathe

Class: TE-3

Batch: K-3

PICT, Pune

ASSIGNMENT 2

TITLE SQL DDL Statements

DATE

PROBLEM STATEMENT Design and develop SQL DDL statement which demonstrate the use of SQL objects such as table, view, index, sequence, synonym.

SCHEMA Consider the foll. relational schema.

- Departments (dept_id, dept_name)
- Professors (prof_id, prof_fname, prof_lname, dept_id, designation, salary, doj, email, phone, city)
- Work (prof_id, dept_id, duration)
- Shift (prof_id, shift, working_hours)

Create view, index, sequence and synonym based on above tables.

LEARNING OBJECTIVES
- To understand the working of SQL DDL
- To learn how to create tables in SQL

LEARNING OUTCOMES
- Understand the working of SQL DLL
- Students will be able to create tables.

H/W & S/W REQUIREMENTS 64 bit Fedora Linux, MySQL, monitor, mouse, keyboard.

THEORY

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

- DDL MySQL Commands creat and drop database objec like the follaing:

a) Tables
b) Views
c) Procedures
d) Functios
e) Indexes
f) Truggers

The list of DDL MySQL stmts

- Create
- alter
- rerone
- drop
- truncate
- Comment

i) Create: Create command defines each attribute uniquely. Each attribute has 3 mandatory things:
- ↳ Name
- ↳ Size
- ↳ Datatype

Eg Create table establishment(id Integer);

ii) Alter: By using alter command, existing table can be modified.

Eg Alter table etab add age integer;

iii) Drop: Can destroy element.

Eg Drop table etab;

iv) Truncate: Deletes all entries but keeps the structure.

Eg Truncate table etab;

v) Rename: Used to rename a table.

Eg Rename etab btab;

CREATING VIEWS

- Database views are created using the CREATE VIEW statement. Views can be created from a single table, multiple tables or another view

Syntax

Create view view_name as select column1, column2 from table_name where [condi

- The WITH CHECK option is a create view statement option to ensure that all update and insert satisfy the conditions in the view definition.

INDEX

- An index can be created in a table to find data more quickly and efficiently. Users cannot see indices, they are just used to speed up queries.
- Types of index:
  - ↳ Unique index
  - ↳ Fulltext index
  - ↳ Primary key
  - ↳ Complex Index

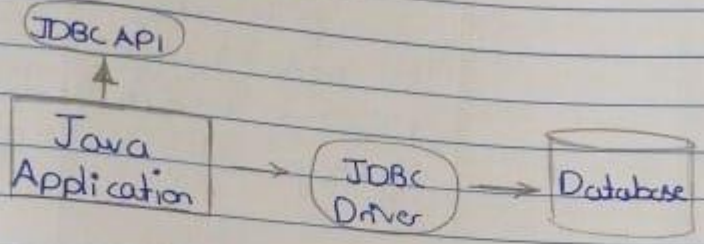
JAVA DATABASE CONNECTIVITY (JDBC)

- In the two-tier model, a Java applet or application talks directly to the data source. This requires a JDBC driver that can communicate with the particular data source being accessed.

- A user's commands are delivered to the database, and the results of those statements are send back to the user.

- Open a connection to datab
- Create a statement object to perfom a query
- Execute the statement object to ~~perform~~ return a query. resultset
- Process the result set.
- Close the result set and statement objects.
- Close the connection.

- There are 4 types of JDBC drivers:
  - ↳ JDBC-ODBC Bridge Driver
  - ↳ Native Driver
  - ↳ Network Protocol Driver
  - ↳ 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 do the following:
  - ↳ Connect to the database
  - ↳ Execute queries and update statements to database
  - ↳ Retrieve result received

SIMPLE VIEW	COMPLEX VIEW
• Contains only one single base table	• Contains >1 base tables
• We cannot use group functions like MAX(), COUNT()	• We can use group functions
• Does not contain groups of data	• It can contains groups of data

CONCLUSION Thus we have successfully studied and implemented DDL statements in MySQL and JDBC connectivity

Code:

```
CREATE TABLE Departments(dept_id INTEGER NOT NULL PRIMARY KEY
AUTO_INCREMENT,dept_name VARCHAR(20));


CREATE TABLE Professors(prof_id INTEGER NOT NULL PRIMARY KEY,prof_fname
VARCHAR(20), prof_lname VARCHAR(20), dept_id INTEGER NOT NULL, designation
VARCHAR(20), salary INTEGER NOT NULL, doj DATE, email VARCHAR(20), phone
BIGINT, city VARCHAR(20), FOREIGN KEY (dept_id)REFERENCES
Departments(dept_id) ON DELETE CASCADE);


CREATE TABLE Works(dept_id INTEGER NOT NULL,prof_id INTEGER NOT NULL,
duration INTEGER, FOREIGN KEY (dept_id)REFERENCES Departments(dept_id) ON
DELETE CASCADE, FOREIGN KEY (prof_id)REFERENCES Professors(prof_id) ON
DELETE CASCADE );


CREATE TABLE  Shift(prof_id INTEGER NOT NULL, shift INTEGER NOT NULL,
working_hours INTEGER NOT NULL, FOREIGN KEY (prof_id)REFERENCES
Professors(prof_id) ON DELETE CASCADE);
```

Source Code:

JDBC

```java
import java.sql.Connection;


public class Maker {

    public static void main(String[] args) {
        try
        {

                String url = "jdbc:mysql://localhost/pict";
                Connection conn = DriverManager.getConnection(url,"root","root");
                Statement stmt = conn.createStatement();


                stmt.executeUpdate("CREATE TABLE  Shift(prof_id INTEGER NOT NULL, shift INTEGER NOT NULL, working_hours INTEGER NOT NULL, "
                        + "FOREIGN KEY (prof_id)REFERENCES Professors(prof_id) ON DELETE CASCADE);");

                conn.close();
        }
        catch (Exception e)
        {
            System.err.println("Got an exception! ");
            System.err.println(e.getMessage());
        }

    }

}
```

```java
public class Queries {

    public static void main(String[] args) {

            try
            {

                    String url = "jdbc:mysql://localhost/pict";
                    Connection conn = DriverManager.getConnection(url,"root","root");
                    Statement stmt = conn.createStatement();
                    ResultSet rs;
                    System.out.println("\n*****************************************************************************************************************************************************************");
                    rs = stmt.executeQuery("Select * from professors where (city = 'Pune' or city = 'Mumbai') and (prof_fname like 'a%' or prof_fname like 'd%');");

                    System.out.println("\n\n" + "\t Prof id: " +  "\t First Name: " +  "\t Last Name: " +  "\t Department: " +  "\t Designation: " +  "\t Salary: " + "\t DOJ: " +
                    while ( rs.next() )
                    {

                        int id = rs.getInt("prof_id");
                        String fname = rs.getString("prof_fname");
                        String lname = rs.getString("prof_lname");
                        int dept = rs.getInt("dept_id");
                        String designation = rs.getString("designation");
                        int salary = rs.getInt("salary");
                        String doj = rs.getString("doj");
                        String email = rs.getString("email");
                        Long phone = rs.getLong("phone");
                        String city = rs.getString("city");


                        System.out.println("\n\n" + "\t "+  id + "\t\t "+ fname +  "\t\t "+ lname + "\t\t "+  dept +  "\t\t "+ designation +"\t\t "+  salary  + "\t "+doj+ "\t "+

                    }
```

```
        rs = stmt.executeQuery("Select DISTINCT(city) from professors;");
        System.out.println("\n\n" +    "\t City: ");
        while ( rs.next() )
        {


            String city = rs.getString("city");


            System.out.println("\n\n" + "\t "+  city);



        }
```

```
stmt.executeUpdate("Update professors set salary = salary + (salary * 0.05) where doj = '2015-01-01';");


rs = stmt.executeQuery("Select * from professors;");

System.out.println("\n\n" + "\t Prof id: " +  "\t First Name: " +  "\t Last Name: " +  "\t Department: " +  "\t Designation: " +  "\t Salary: " + "\
while ( rs.next() )
{

    int id = rs.getInt("prof_id");
    String fname = rs.getString("prof_fname");
    String lname = rs.getString("prof_lname");
    int dept = rs.getInt("dept_id");
    String designation = rs.getString("designation");
    int salary = rs.getInt("salary");
    String doj = rs.getString("doj");
    String email = rs.getString("email");
    Long phone = rs.getLong("phone");
    String city = rs.getString("city");


    System.out.println("\n\n" + "\t "+  id + "\t\t "+ fname +  "\t\t "+ lname + "\t\t "+  dept +  "\t\t "+ designation +"\t\t "+  salary  + "\t "+dc


}
```

```
        stmt.executeUpdate("Delete from professors where city ='Pune';");


        rs = stmt.executeQuery("Select * from professors;");

        System.out.println("\n\n" + "\t Prof id: " +  "\t First Name: " +  "\t Last Name: " +  "\t Department: " +  "\t Designation: " +  "\t Salary: " + "\t D
        while ( rs.next() )
        {

            int id = rs.getInt("prof_id");
            String fname = rs.getString("prof_fname");
            String lname = rs.getString("prof_lname");
            int dept = rs.getInt("dept_id");
            String designation = rs.getString("designation");
            int salary = rs.getInt("salary");
            String doj = rs.getString("doj");
            String email = rs.getString("email");
            Long phone = rs.getLong("phone");
            String city = rs.getString("city");


            System.out.println("\n\n" + "\t "+  id + "\t\t "+ fname +  "\t\t "+ lname + "\t\t "+  dept +  "\t\t "+ designation +"\t\t "+  salary  + "\t "+doj+ "


        }
```

```
rs = stmt.executeQuery("select prof_fname, prof_lname from professors where city ='Pune' or city ='Mumbai' ;");



System.out.println("\n\n" + "\t First Name: " + "\t Last Name: "  );
while ( rs.next() )
{

    String fname = rs.getString("prof_fname");
    String lname = rs.getString("prof_lname");


    System.out.println("\n\n" +  "\t\t "+ fname +  "\t\t "+ lname );



}
```

```
    rs = stmt.executeQuery("select * from professors where doj='2015-01-01' or doj ='2016-01-01' ;");


    System.out.println("\n\n" + "\t Prof id: " +  "\t First Name: " +  "\t Last Name: " +  "\t Department: " +  "\t Designation: " +  "\t Salary: " + "\
    while ( rs.next() )
    {

        int id = rs.getInt("prof_id");
        String fname = rs.getString("prof_fname");
        String lname = rs.getString("prof_lname");
        int dept = rs.getInt("dept_id");
        String designation = rs.getString("designation");
        int salary = rs.getInt("salary");
        String doj = rs.getString("doj");
        String email = rs.getString("email");
        Long phone = rs.getLong("phone");
        String city = rs.getString("city");


        System.out.println("\n\n" + "\t "+  id + "\t\t "+ fname +  "\t\t "+ lname + "\t\t "+  dept +  "\t\t "+ designation +"\t\t "+  salary  + "\t "+do


    }

.println("\n**********************************************************************************************************************************************

    rs = stmt.executeQuery("select MAX(salary) from professors where salary between 10000 and 200000;");


    System.out.println("\n\n" + "\t Salary: " );
    while ( rs.next() )
    {
        int salary = rs.getInt("MAX(salary)");

        System.out.println("\n\n" + "\t "+  salary );
```

```
    rs = stmt.executeQuery("select prof_fname, prof_lname,salary , doj from professors order by salary desc;");


    System.out.println("\n\n" + "\t First Name: "  + "\t Last Name: " + "\t Salary: "  + "\t DOJ: " );
    while ( rs.next() )
    {
        String fname = rs.getString("prof_fname");
        String lname = rs.getString("prof_lname");
        int salary = rs.getInt("salary");
        String doj = rs.getString("doj");

        System.out.println("\n\n" + "\t "+  fname + "\t\t" + lname + "\t "+ salary + "\t\t " + doj);



    }


rintln("\n**********************************************************************************************************************************************

    rs = stmt.executeQuery("select prof_fname, prof_lname,dept_id , doj from professors where salary in (30000, 40000,50000);");


    System.out.println("\n\n" + "\t First Name: "  + "\t Last Name: " + "\t Dept Id: "  + "\t DOJ: " );
    while ( rs.next() )
    {
        String fname = rs.getString("prof_fname");
        String lname = rs.getString("prof_lname");
        int dept_id = rs.getInt("dept_id");
        String doj = rs.getString("doj");

        System.out.println("\n\n" + "\t "+  fname + "\t\t" + lname + "\t "+ dept_id + "\t\t " + doj);
```

```java
import java.sql.Connection;
import java.sql.DriverManager;
import java.sql.Statement;


public class Records {

    public static void main(String[] args) {
        try
        {

                String url = "jdbc:mysql://localhost/pict";
                Connection conn = DriverManager.getConnection(url,"root","root");
                Statement stmt = conn.createStatement();


                stmt.executeUpdate("Insert into  Shift values(1,1,6);");
                stmt.executeUpdate("Insert into  Shift values(2,1,6);");
                stmt.executeUpdate("Insert into  Shift values(3,2,6);");
                stmt.executeUpdate("Insert into  Shift values(4,1,6);");
                stmt.executeUpdate("Insert into  Shift values(5,1,5);");
                stmt.executeUpdate("Insert into  Shift values(6,1,6);");
                stmt.executeUpdate("Insert into  Shift values(7,2,6);");
                stmt.executeUpdate("Insert into  Shift values(8,1,6);");
                stmt.executeUpdate("Insert into  Shift values(9,1,4);");
                stmt.executeUpdate("Insert into  Shift values(10,1,6);");

                conn.close();
        }
        catch (Exception e)
        {
            System.err.println("Got an exception! ");
            System.err.println(e.getMessage());
        }


    }

}
```

```java
System.out.println("\n\n\n\nSimple View");
stmt.executeUpdate("Create view Prof_View1 as Select Prof_fname, Prof_lname from Professors;");
rs = stmt.executeQuery("SELECT * FROM Prof_View1;");
System.out.println("\n\n" + "First Name" + "\t\t" + "Last Name" );
while ( rs.next() )
{

    String name = rs.getString("prof_fname");
    String title = rs.getString("prof_lname");


    System.out.println("\n\n" + name + "\t\t" + title );
}




System.out.println("\n\n\n\nComplex View");
stmt.executeUpdate("Create view Prof_Shift_View1 as Select Professors.Prof_fname, Professors.prof_id, shift.shift from Professors INNER JOIN Shift
rs = stmt.executeQuery("SELECT * FROM Prof_Shift_View1;");
System.out.println("\n\n" + "First Name" + "\t\t" + "Id" + "\t\t" + "Shift" );
while ( rs.next() )
{

    String name = rs.getString("prof_fname");
    int id = rs.getInt("prof_id");
    int shift = rs.getInt("shift");


    System.out.println("\n\n" + name + "\t\t" + id + "\t\t" + shift  );
}
/*
```

```
System.out.println("\n\n\n\nSimple Index");
stmt.executeUpdate("CREATE TABLE A1 (Name VARCHAR(10) PRIMARY KEY, c2 Integer, c3 Integer, c4 Integer,   INDEX index_name (c2));");

System.out.println("\n\n\n\nComplex Index");
stmt.executeUpdate("CREATE TABLE A2 (Name VARCHAR(10) PRIMARY KEY, c2 Integer, c3 Integer, c4 Integer,   INDEX index_name (c2,c3,c4));");
System.out.println("\n\n\n\nUnique Index");
stmt.executeUpdate("CREATE TABLE A3 (Name VARCHAR(10) PRIMARY KEY, c2 Integer, c3 Integer, c4 Integer,   UNIQUE INDEX index_name (c2,c3,c4));");
```

```
stmt.executeUpdate("Drop view Prof_View1;");
rs = stmt.executeQuery("SELECT * FROM Prof_View1;");
while ( rs.next() )
{

    int id= rs.getInt("prof_id");



    System.out.println("\n\n" + id );
}
conn.close();
```

Output:

Records

```
mysql> select * from professors;
+---------+------------+------------+---------+-------------+--------+------------+-----------------+------------+--------+
| prof_id | prof_fname | prof_lname | dept_id | designation | salary | doj        | email           | phone      | city   |
+---------+------------+------------+---------+-------------+--------+------------+-----------------+------------+--------+
|       2 | Anu        | Marathe    |       1 | HOD         | 400000 | 2001-12-12 | data1@gmail.com | 9245413291 | Mumbai |
|       3 | Anu        | Siya       |       3 | TA          | 300000 | 2002-12-12 | data3@gmail.com | 1245413291 | Delhi  |
|       4 | Raj        | Siya       |       2 | RA          | 100000 | 2010-12-12 | da@gmail.com    | 3245413291 | Mumbai |
|       5 | Raj        | Pawar      |       7 | HOD         | 100000 | 2015-12-12 | dan@gmail.com   | 3225413291 | Mumbai |
|       6 | Shan       | LeCunn     |       6 | Dean        | 700000 | 2017-12-12 | dane@gmail.com  | 3255413291 | NY     |
+---------+------------+------------+---------+-------------+--------+------------+-----------------+------------+--------+
5 rows in set (0.00 sec)

mysql> select * from departments;
+---------+-----------+
| dept_id | dept_name |
+---------+-----------+
|       1 | A         |
|       2 | B         |
|       3 | C         |
|       4 | D         |
|       5 | E         |
|       6 | F         |
|       7 | G         |
|       8 | H         |
|       9 | I         |
|      10 | J         |
+---------+-----------+
10 rows in set (0.00 sec)

mysql> select * from works;
+---------+---------+----------+
| dept_id | prof_id | duration |
+---------+---------+----------+
|       3 |       3 |       20 |
|       4 |       2 |        0 |
|       6 |       6 |      145 |
|       7 |       2 |       45 |
|      10 |       6 |      115 |
+---------+---------+----------+
5 rows in set (0.00 sec)

mysql> select * from shift;
+---------+-------+--------------+
| prof_id | shift | working_hours |
+---------+-------+--------------+
|       2 |     1 |            6 |
|       3 |     2 |            6 |
|       4 |     1 |            6 |
|       5 |     1 |            5 |
|       6 |     1 |            6 |
+---------+-------+--------------+
5 rows in set (0.00 sec)
```

Table Descriptions

```
Simple View

First Name              Last Name

Anu             Marathe

Anu             Siya

Raj             Siya

Raj             Pawar

Shan            LeCunn
```

```
Complex View

First Name              Id              Shift

Anu             2               1

Anu             3               2

Raj             4               1

Raj             5               1

Shan            6               1
```

After dropping view:

```
Got an exception!
Table 'pict.prof_view1' doesn't exist
```

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.