Assignment - 2

Date of Completion:

Date of Submission:

Title: Implement DDL commands in context of view, index, sequence

Problem Statement: Design & develop SQL & DDL statements which demonstrate the use of SQL objects such as Table, View, Index, Sequence, Synonym

Objective:
1. To understand & implement the various DDL commands.
2. To understand database concepts like view, index, sequence & synonym

S/w & H/w Requirements:
MySQL, Windows 10 (64 bit), i5 processor, JDBC, etc.

Theory:
VIEW:
In SQL, a view is a virtual table based on the result-set of an SQL statement. A view contains rows & colums. Just like a table.

Create View System:

1) Simple View
   View created by involving a single table

2) Complex view:
   View created by involving multiple tables.

3) Drop View
   drop view view_name;

- INDEX:
  Indexes are special lookup tables that database search engine can use to speedup data retrieval
  An index ~~is~~ is a pointer to data in a table.

1) SIMPLE INDEX:
   - create index on one column.

2) Create INDEX on multiple selected columns:- COMPOUND INDEX

3) UNIQUE INDEX-
   created on selected column of database table & doesn't allow duplicate values of that column.

4) SHOW INDEX:
   Shows all indexes created on table.

5) DROP INDEX:
   Drops the given index.

SEQUENCE:

- A sequence is a user defined schema bound object that generates a sequence of numerical values.
- Sequences are frequently used in many databases because application require each row to have a unique ID.

Syntax:
auto-increment.

```
create table table_name( col1 type not NULL
                auto increment, primary key
                (col1), col2 type);
```

SYNONYM

- Supported by Oracle.
- A synonym is an alternative name for objects such as tables, views, sequences, stored procedures, & other database objects.

Syntax:

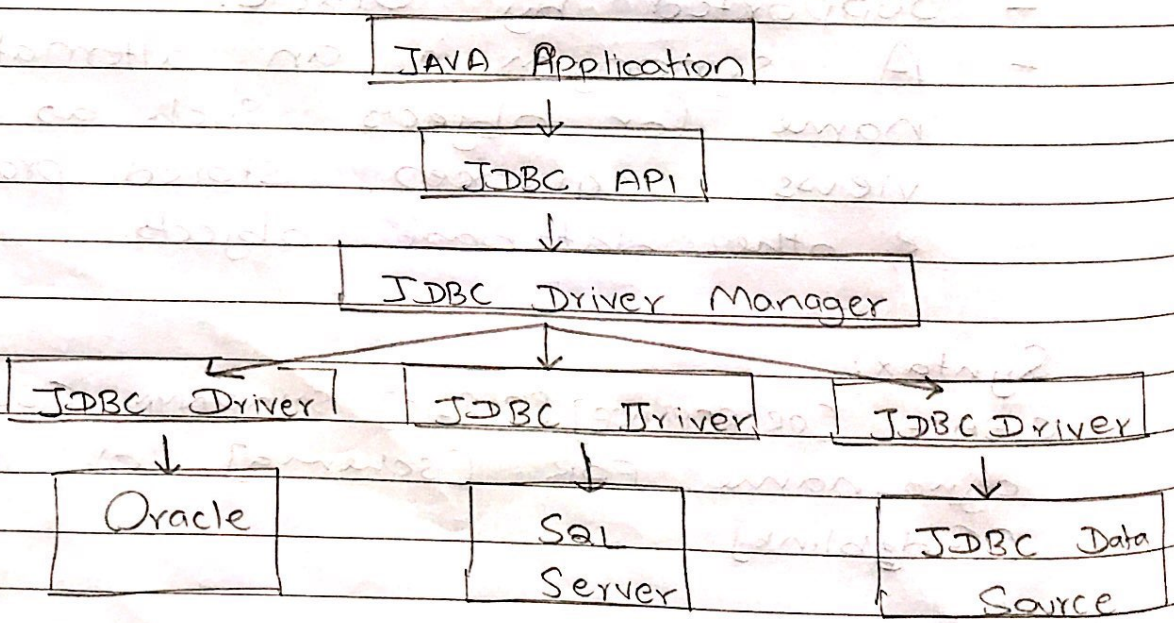
```
CREATE [OR REPLACE] [PUBLIC] SYNONYM [SCHEMA].
syno_name FOR [schema] obj_name
[@dgblink]
```

- JDBC:

JDBC stands for JAVA database connectivity which is a standard JAVA API for database independent connectivity.

Common JDBC components.

1) Driver Manager - This class manages a list of database drivers.
2) Driver - Handels communication with server
3) Connection - Contacting database
4) Statement - Use objects created from this interface to submit the SQL statements to the database.
5) ResultSet - Object data retrieved from database
6) SQL Exception - for error handling.



Conclusion:

1) We understood how to create simple/complex views in a database.
2) We understood indexes, Sequences & synonyms & JDBC connection to MYSQL

Source code

```
import java.sql.*;
import java.util.*;

public class App {
    static Scanner in = new Scanner(System.in);
    static exe ex = new exe();
    public static void main(String[] args) {
        int repeat=1, choice;

        System.out.println("1.Create table");
        System.out.println("2.Create Simple view");
        System.out.println("3.Create Simple index");
        System.out.println("4.Create sequence");
        System.out.println("5.Create synonym");
        System.out.println("6.exit");
        System.out.println("7.Create Complex view");
        System.out.println("8.Create Compound index");
        System.out.println("9.Create Unique index");

        ex.reset();
        while(repeat==1){
            System.out.print("Enter option : ");
            choice = Integer.parseInt(in.nextLine());
            switch (choice) {
                case 1:
                    ex.create_table();
                    break;
                case 2:
                    ex.create_view();
```

```
                break;
            case 3:
                ex.create_index();
                break;
            case 4:
                ex.create_seq();
                break;
            case 5:
                ex.create_synonym();
                break;
            case 7:
                ex.create_complex_view();
                break;
            case 8:
                ex.create_compound_index();
                break;
            case 9:
                ex.create_unique_index();
                break;
            default:
                break;
            }
        System.out.println("Again? (1/0) : ");
        repeat = Integer.parseInt(in.nextLine());
      }
   }
}


import java.sql.*;

public class exe {
```

```
    void create_table(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/A1_pro
fessor_schema","root","Hello@123");
            Statement stmt = con.createStatement();
            stmt.executeUpdate("drop table if exists sample");
            stmt.executeUpdate("create table sample (id int not
null,value varchar(20))");

            ResultSet r = stmt.executeQuery("select * from sample");
            while(r.next()) {
                System.out.println(r.getInt(1) + "\t" + r.getString(2));
            }

            con.close();
        }catch(Exception e){System.out.println(e);}
    }

    void create_view(){
        try{
            String stmnt = "create or replace view comp as select fname,
lname from professor where dept_id=1";
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/A1_pro
fessor_schema","root","Hello@123");
            Statement stmt = con.createStatement();
            stmt.executeUpdate(stmnt);
            ResultSet r = stmt.executeQuery("select * from comp");
```

```
            while(r.next()) {
                System.out.println(r.getString(1) + "\t" + r.getString(2));
            }
            con.close();
        }catch(Exception e){System.out.println(e);}
    }

    void create_index(){
        try{
            String stmnt = "CREATE INDEX ind_1 ON
professor(fname)";
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/A1_pro
fessor_schema","root","Hello@123");
            Statement stmt = con.createStatement();
            stmt.executeUpdate(stmnt);
            ResultSet r = stmt.executeQuery("show index from
professor");
            while(r.next()){
                System.out.println(r.getString(3) + "\t" + r.getString(5));
            }
            con.close();
        }catch(Exception e){System.out.println(e);}
    }

    void create_seq(){
        try{
            String stmnt = "CREATE TABLE employees (emp_no INT
AUTO_INCREMENT PRIMARY KEY,f_name
VARCHAR(50),l_name VARCHAR(50));";
            Class.forName("com.mysql.cj.jdbc.Driver");
```

```
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/A1_pro
fessor_schema","root","Hello@123");
            Statement stmt = con.createStatement();
            stmt.executeUpdate(stmnt);
            con.close();
        }catch(Exception e){System.out.println(e);}
    }

    // write in writeups[ only avaliable in oracle not mysql ]
    void create_synonym(){
        try{
            String stmnt = "CREATE TABLE employees (emp_no INT
AUTO_INCREMENT PRIMARY KEY,f_name
VARCHAR(50),l_name VARCHAR(50));";
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/A1_pro
fessor_schema","root","Hello@123");
            Statement stmt = con.createStatement();
            stmt.executeUpdate("drop table if exists employees");
            stmt.executeUpdate(stmnt);
            con.close();
        }catch(Exception e){System.out.println(e);}
    }

    void create_complex_view(){
        try{
            String stmnt = "create or replace view comp as select
p.fname, p.lname from professor p, department d where
p.dept_id=d.dept_id";
            Class.forName("com.mysql.cj.jdbc.Driver");
```

```
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/A1_pro
fessor_schema","root","Hello@123");
            Statement stmt = con.createStatement();
            stmt.executeUpdate(stmnt);
            ResultSet r = stmt.executeQuery("select * from comp");
            while(r.next()) {
               System.out.println(r.getString(1) + "\t\t\t" +
r.getString(2));
            }
            con.close();
        }catch(Exception e){System.out.println(e);}
    }

    void create_compound_index(){
        try{
            String stmnt = "CREATE INDEX ind_2 ON
professor(fname, lname)";
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/A1_pro
fessor_schema","root","Hello@123");
            Statement stmt = con.createStatement();
            stmt.executeUpdate(stmnt);
            ResultSet r = stmt.executeQuery("show index from
professor");
            while(r.next()){
               System.out.println(r.getString(3) + "\t" + r.getString(5));
            }
            con.close();
        }catch(Exception e){System.out.println(e);}
    }
```

```
    void create_unique_index(){
        try{
            String stmnt = "CREATE unique INDEX ind_3 ON
department(dept_id)";
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/A1_pro
fessor_schema","root","Hello@123");
            Statement stmt = con.createStatement();
            stmt.executeUpdate(stmnt);
            ResultSet r = stmt.executeQuery("show index from
department");
            while(r.next()){
                System.out.println(r.getString(3) + "\t" + r.getString(5));
            }
            con.close();
        }catch(Exception e){System.out.println(e);}
    }

    void reset(){
        try{
            Class.forName("com.mysql.cj.jdbc.Driver");
            Connection con =
DriverManager.getConnection("jdbc:mysql://localhost:3306/A1_pro
fessor_schema","root","Hello@123");
            Statement stmt = con.createStatement();

            stmt.executeUpdate("drop table if exists employees");
            stmt.executeUpdate("drop index ind_1 on professor");
            stmt.executeUpdate("drop index ind_2 on professor");
            stmt.executeUpdate("drop index ind_3 on department");
```

```
            stmt.executeUpdate("DROP VIEW IF EXISTS comp");
            stmt.executeUpdate("drop table if exists sample");

            con.close();
        }catch(Exception e){System.out.println(e);}
    }


}
```