Assignment Database

```java
package Assignments.assignment2.Code;

import java.sql.*;


//Create a table product with fields: id, name, vendor, price, manufacture date, expiry date

public class CreateTableDatabase {
    public static void main(String[] args) throws Exception{
        Class.forName("org.mariadb.jdbc.Driver");
        Connection conn;
        String url="jdbc:mariadb://localhost:3306/databaseAssignment";
        conn= DriverManager.getConnection(url,"root","root");
        if(conn!=null){
            String sql = "CREATE TABLE Product " +
                    "(id INTEGER PRIMARY KEY, " +
                    "name varchar(30), " +
                    "vendor varchar(30)," +
                    "price double,"+
                    "mandufacture_date date,"+
                    "expiry_date date)";
            Statement stmt=conn.createStatement();
            int result=stmt.executeUpdate(sql);
            System.out.println("Table created");
        }
        else{
            System.out.println("Error creating table");
        }
    }
}
```

```
MariaDB [databaseAssignment]> desc Product
    -> ;
+-------------------+-------------+------+-----+---------+-------+
| Field             | Type        | Null | Key | Default | Extra |
+-------------------+-------------+------+-----+---------+-------+
| id                | int(11)     | NO   | PRI | NULL    |       |
| name              | varchar(30) | YES  |     | NULL    |       |
| vendor            | varchar(30) | YES  |     | NULL    |       |
| price             | double      | YES  |     | NULL    |       |
| mandufacture_date | date        | YES  |     | NULL    |       |
| expiry_date       | date        | YES  |     | NULL    |       |
+-------------------+-------------+------+-----+---------+-------+
6 rows in set (0.002 sec)
```

```java
package Assignments.assignment2.Code;

import java.sql.*;


//1. Write sql to insert 5 record into product table

public class Insert {
    public static void main(String[] args) throws Exception{
        Class.forName("org.mariadb.jdbc.Driver");
        Connection conn;
        String url="jdbc:mariadb://localhost:3306/databaseAssignment";
        conn=DriverManager.getConnection(url,"root","root");
        if(conn!=null){
            Statement stmt=conn.createStatement();
            String sql="INSERT INTO Product VALUES (100,'Jojan','CG',2000,'2020-01-
14','2021-01-14')," +
                    "(101,'Nalin','BG',200,'2019-01-14','2021-01-14')," +
                    "(102,'Dipesh','Uni',2230,'2018-04-14','2022-01-14')," +
                    "(103,'Dristy','Broke',300,'2016-05-14','2024-01-14')," +
                    "(104,'Subin','Newa',150,'2021-03-14','2023-01-14')," +
                    "(105,'Hari','Erer',10,'2019-01-14','2022-01-14')";
            int result=stmt.executeUpdate(sql);
            if(result!=0){
                System.out.println("inserted successfully");
            }
            else{
                System.out.println("Not inserted");
            }
        }
        else{
            System.out.println("error while inserting");
        }
    }
}
```

```java
package Assignments.assignment2.Code;

import java.sql.*;

// Write sql to insert 5 record into product table
public class InsertPrepared {
    public static void main(String[] args) throws Exception{
        Class.forName("org.mariadb.jdbc.Driver");
        String url="jdbc:mariadb://localhost:3306/databaseAssignment";
        Connection conn = DriverManager.getConnection(url, "root", "root");
        if(conn!=null){
            String sql="INSERT INTO Product VALUES (?,?,?,?,?,?)";
            PreparedStatement stmt=conn.prepareStatement(sql);
            stmt.setInt(1,100);
            stmt.setString(2,"Jojan");
            stmt.setString(3,"CG");
            stmt.setInt(4,200);
            stmt.setDate(5, java.sql.Date.valueOf("2020-12-01"));
            stmt.setDate(6,java.sql.Date.valueOf("2021-12-01"));

//            stmt.setInt(1,101);
//            stmt.setString(2,"Nalin");
//            stmt.setString(3,"BG");
//            stmt.setInt(4,200);
//            stmt.setDate(5, java.sql.Date.valueOf("2019-12-01"));
//            stmt.setDate(6,java.sql.Date.valueOf("2021-12-01"));

            int result=stmt.executeUpdate();
            if(result!=0){
                System.out.println("inserted successfully");
            }
            else{
                System.out.println("Not inserted");
            }
            System.out.println("successfull");

            stmt.close();
            conn.close();
        }
        else{
            System.out.println("Error");
        }
    }
}
```

```
MariaDB [databaseAssignment]> select * from Product;
+-----+--------+--------+-------+-------------------+-------------+
| id  | name   | vendor | price | mandufacture_date | expiry_date |
+-----+--------+--------+-------+-------------------+-------------+
| 100 | Jojan  | CG     |  2000 | 2020-01-14        | 2021-01-14  |
| 101 | Nalin  | BG     |   200 | 2019-01-14        | 2021-01-14  |
| 102 | Dipesh | Uni    |  2230 | 2018-04-14        | 2022-01-14  |
| 103 | Dristy | Broke  |   300 | 2016-05-14        | 2024-01-14  |
| 104 | Subin  | Newa   |   150 | 2021-03-14        | 2023-01-14  |
| 105 | Hari   | Erer   |    10 | 2019-01-14        | 2022-01-14  |
+-----+--------+--------+-------+-------------------+-------------+
6 rows in set (0.001 sec)
```

```java
package Assignments.assignment2.Code;

import java.sql.*;

// Write sql to update the price of all product by 10%

public class Update {
    public static void main(String[] args) throws Exception{
        Class.forName("org.mariadb.jdbc.Driver");
        String url="jdbc:mariadb://localhost:3306/databaseAssignment";
        Connection conn = DriverManager.getConnection(url, "root", "root");
        if(conn!=null){
            Statement stmt=conn.createStatement();
            String sql1="SELECT * FROM Product";
            ResultSet rs=stmt.executeQuery(sql1);
//          int num=0;
            while (rs.next()){
//              num=num+1;
                int id=rs.getInt("id");
                double p=rs.getDouble("price");
                double tmp=p*0.1;
                p=p+tmp;
//              System.out.println(p);
                String sql2="UPDATE Product SET price = "+ p +"WHERE id ="+id;
                stmt.executeUpdate(sql2);
            }
        }
        else {
            System.out.println("error");
        }
    }
}
```

```java
package Assignments.assignment2.Code;

import java.sql.*;

// Write sql to update the price of all product by 10%

public class UpdatePrepared {
    public static void main(String[] args) throws Exception{
        Class.forName("org.mariadb.jdbc.Driver");
        String url="jdbc:mariadb://localhost:3306/databaseAssignment";
        Connection conn = DriverManager.getConnection(url, "root", "root");
        if (conn!=null){
            String sql="UPDATE Product SET price = ? WHERE id = ?";
            PreparedStatement stmt=conn.prepareStatement(sql);
            String sql1="SELECT * FROM Product";
            ResultSet rs=stmt.executeQuery(sql1);
            while (rs.next()){
                int id=rs.getInt("id");
                double p=rs.getDouble("price");
                double tmp=p*0.1;
                p=p+tmp;
                stmt.setDouble(1,p);
                stmt.setInt(2,id);
                stmt.executeUpdate();
            }
        }
        else{
            System.out.println("Error");
        }
    }
}
```

```
MariaDB [databaseAssignment]> select * from Product;
+-----+--------+--------+-------+-------------------+-------------+
| id  | name   | vendor | price | mandufacture_date | expiry_date |
+-----+--------+--------+-------+-------------------+-------------+
| 100 | Jojan  | CG     |  2200 | 2020-01-14        | 2021-01-14  |
| 101 | Nalin  | BG     |   220 | 2019-01-14        | 2021-01-14  |
| 102 | Dipesh | Uni    |  2453 | 2018-04-14        | 2022-01-14  |
| 103 | Dristy | Broke  |   330 | 2016-05-14        | 2024-01-14  |
| 104 | Subin  | Newa   |   165 | 2021-03-14        | 2023-01-14  |
| 105 | Hari   | Erer   |    11 | 2019-01-14        | 2022-01-14  |
+-----+--------+--------+-------+-------------------+-------------+
6 rows in set (0.001 sec)
```

```java
package Assignments.assignment2.Code;

import java.sql.*;

//  Write sql to delete product whose price is greater than 200.

public class Delete {
    public static void main(String[] args) throws Exception{
        Class.forName("org.mariadb.jdbc.Driver");
        String url="jdbc:mariadb://localhost:3306/databaseAssignment";
        Connection conn = DriverManager.getConnection(url, "root", "root");
        if(conn!=null){
            Statement stmt=conn.createStatement();
            String sql="DELETE FROM Product WHERE price > 200";
//            ResultSet rs=stmt.executeQuery(sql);
            int result=stmt.executeUpdate(sql);
            if(result!=0){
                System.out.println("Deleted successfully");
            }
            else{
                System.out.println("Not inserted");
            }
        }
        else{
            System.out.println("Error");
        }
    }
}
```

```java
package Assignments.assignment2.Code;

import java.sql.*;

// Write sql to delete product whose price is greater than 200.

public class DeletePrepared {
    public static void main(String[] args) throws Exception{
        Class.forName("org.mariadb.jdbc.Driver");
        String url="jdbc:mariadb://localhost:3306/databaseAssignment";
        Connection conn = DriverManager.getConnection(url, "root", "root");
        if(conn!=null){
            String sql="DELETE FROM Product WHERE price > ?";
            PreparedStatement stmt=conn.prepareStatement(sql);
            stmt.setInt(1,200);
            int result=stmt.executeUpdate();
            if(result!=0){
                System.out.println("Deleted successfully");
            }
            else{
                System.out.println("Not inserted");
            }
            conn.close();
            stmt.close();
        }
        else{
            System.out.println("Error");
        }

    }
}
```

```
MariaDB [databaseAssignment]> select * from Product;
+-----+-------+--------+-------+-------------------+-------------+
| id  | name  | vendor | price | mandufacture_date | expiry_date |
+-----+-------+--------+-------+-------------------+-------------+
| 104 | Subin | Newa   |   165 | 2021-03-14        | 2023-01-14  |
| 105 | Hari  | Erer   |    11 | 2019-01-14        | 2022-01-14  |
+-----+-------+--------+-------+-------------------+-------------+
2 rows in set (0.001 sec)
```

```java
package Assignments.assignment2.Code;

import java.sql.*;
// Write sql to show all the product.
public class Select {
    public static void main(String[] args) throws Exception{
        Class.forName("org.mariadb.jdbc.Driver");
        String url="jdbc:mariadb://localhost:3306/databaseAssignment";
        Connection conn = DriverManager.getConnection(url, "root", "root");
        if(conn!=null){
            String sql="SELECT * FROM Product";
            Statement stmt=conn.createStatement();

            ResultSet rs=stmt.executeQuery(sql);
            while (rs.next()){
                System.out.println("Name is : "+rs.getString("name"));
                System.out.println("vendor is : "+rs.getString("vendor"));
                System.out.println("price is : "+rs.getInt("price"));
                System.out.println("Manufacture date is :
"+rs.getDate("mandufacture_date"));
                System.out.println("Expiry date is : "+rs.getDate("expiry_date"));
            }
        }
    }
}
```

5) Explain basic step to make connection to database. Explain with program to make connection.

→ The basic step to make connection to database are:-

- Load Driver
- Create connection to database URL
- Create the statements
- Execute statement
- Process the result
- Close the statement
- Close connection

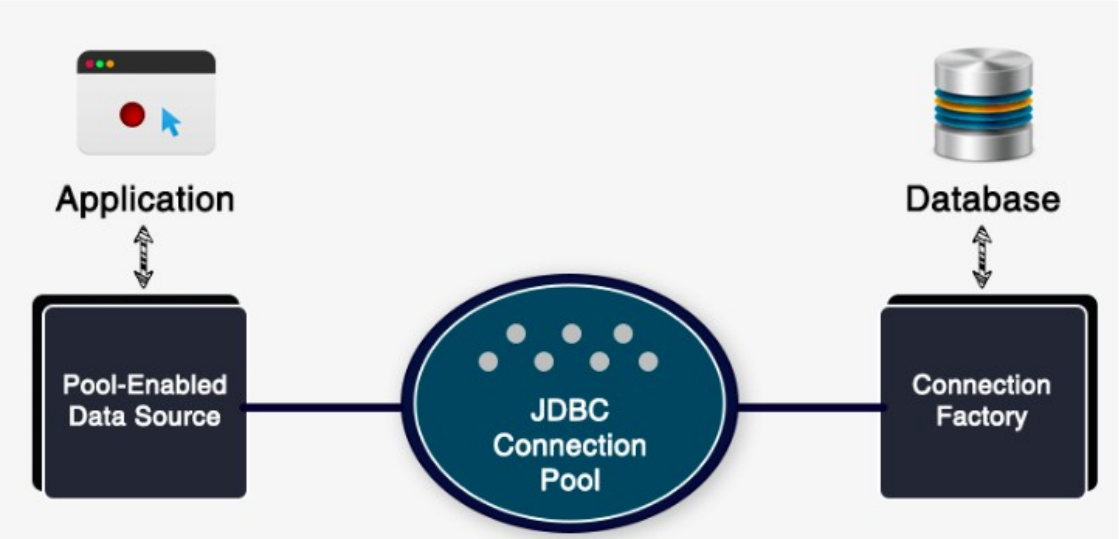
→ Eg :-

```
package Database;

import java.sql.Connection;
import java.sql.DriverManager;

public class ConnectionDemo {
   public static void main(String[] args) throws Exception {

      //step 1: Load the driver
      Class.forName("com.mysql.cj.jdbc.Driver");
      // step 2:Get the conncetion by passing the URL
      String url="jdbc:mysql://localhost:3306/free_blog_shoot";
      Connection con= DriverManager.getConnection(url,"root","redhat");
      System.out.println("Connected Successfully");
      con.close();
   }
}
```

6) Why connection pooling is required?

→Connection pools allows you to scare resources such as database connections, to be shared among multiple users. If you have 1000 users each requiring a connection now and then, if you don't have connection pool the database suffers. Keeping connection open constantly for each user is no good. Opening and closing per-use is expensive,overhead for maintaining as OS resources to create,maintain and close connections to data store(in a relative sense). With a connection pool, Users "borrow" the connection for as long as they need it, and then return it to the pool, you might be able to support 1000 users with 10 connections depending on application.

Instead of opening and closing connections for every request, connection pooling uses a cache of database connections that can be reused when future requests to the database are required. It lets your database scale effectively as the data stored there and the number of clients accessing it grow. Traffic is never constant, so pooling can better manage traffic peaks without causing outages. Your production database shouldn’t be your bottleneck.



Note:- In context of Java Web Applications

In Java web applications DB connection pools are provided as libraries. The DB connection pool manager keeps a number of DB connections open in memory. When the data access code closes a connection, the connection pool manager doesn’t close the connection, instead, it returns it to the pool. When the next call requests a connection, it returns an unused connection from the pool, rather than creating a new one. This improves the latency of your application.