Experiment No. 2.1

Student Name: Gaurav Kumar
Branch: MCA - CCD
Semester: II
Subject Name: Advanced Internet Programming Lab

UID: 22MCC20177
Section/Group: 22MCD1 / Grp A
Date of Performance: 09th April 2023
Subject Code: 22CAP-686

1. Aim/Overview of the practical:

Task to be done:

1. Connect
2. Create Database
3. Create Table
4. Insert Records into respective table
5. Update records of particular table of database
6. Delete Records from table.
7. Delete table and database

2. Code for experiment/practical:

DbConnect.java

```java
package SQL;

import java.sql.*;

public class DbConnect {
//    Make static enum for storing values for driver name and values of just 4 popular databases without url
    public enum Database {
        MySQL("com.mysql.cj.jdbc.Driver"),
        Oracle("oracle.jdbc.driver.OracleDriver"),
        PostgreSQL("org.postgresql.Driver"),
        SQLite("org.sqlite.JDBC");
        private final String driver;
        Database(String driver) {
            this.driver = driver;
        }
        public String getDriver() {
            return driver;
        }

    }
    private final String url;
    private final String dbName;
    private final String username;
    private final String password;
    private final Database database;
    private Connection connection;
    private PreparedStatement preparedStatement;
    private ResultSet resultSet;

    public DbConnect(Database database, String url, String dbName, String username, String password) {
        this.database = database;
        this.url = url;
        this.dbName = dbName;
        this.username = username;
        this.password = password;
    }
```

```
    public void buildConnection() throws ClassNotFoundException, SQLException {
        Class.forName(database.getDriver());
        connection = DriverManager.getConnection((url + dbName), username, password);
    }

    public boolean createDatabase(String dbName) throws SQLException {
        preparedStatement = connection.prepareStatement("CREATE DATABASE " + dbName);
        return preparedStatement.execute();
    }

    public boolean createTable(String tableName, String... columns) throws SQLException {
        StringBuilder query = new StringBuilder("CREATE TABLE " + tableName + " (");
        for (int i = 0; i < columns.length; i++) {
            query.append(columns[i]);
            if (i != columns.length - 1) {
                query.append(", ");
            }
        }
        query.append(")");
        preparedStatement = connection.prepareStatement(query.toString());
        return preparedStatement.execute();
    }

    public ResultSet executeQuery(String query) throws SQLException {
        preparedStatement = connection.prepareStatement(query);
        resultSet = preparedStatement.executeQuery();
        return resultSet;
    }

    public ResultSet executeQuery(String query, String... args) throws SQLException {
        preparedStatement = connection.prepareStatement(query);
        for (int i = 0; i < args.length; i++) {
            preparedStatement.setString(i + 1, args[i]);
        }
        resultSet = preparedStatement.executeQuery();
        return resultSet;
    }

    public int executeUpdate(String query) throws SQLException {
        preparedStatement = connection.prepareStatement(query);
        return preparedStatement.executeUpdate();
    }

    public int executeUpdate(String query, String... args) throws SQLException {
        preparedStatement = connection.prepareStatement(query);
        for (int i = 0; i < args.length; i++) {
            preparedStatement.setString(i + 1, args[i]);
        }
        return preparedStatement.executeUpdate();
    }
    public void closeConnection() throws SQLException {
        connection.close();
    }
}
```

Learning outcomes (What I have learnt):

1. **Learn about Database Connectivity JDBC-ODBC.**
2. **Learn about prepareStatement and other functions.**

3. Evaluation Grid:

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.	Demonstration and Performance (Pre Lab Quiz)		5
2.	Worksheet		10
3.	Post Lab Quiz		5