



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Experiment No.9
Demonstrate Database connectivity
Date of Performance:
Date of Submission:



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

Aim :- Write a java program to connect Java application with the MySQL database

Objective :- To learn database connectivity

Theory:

Database used : MySql

1. Driver class: The driver class for the mysql database is `com.mysql.jdbc.Driver`.
2. Connection URL: The connection URL for the mysql database is `jdbc:mysql://localhost:3306/loan management` where `jdbc` is the API, `mysql` is the database, `localhost` is the server name on which mysql is running, can also use IP address, 3306 is the port number and `loan management` is the database name.
3. Username: The default username for the mysql database is `Hiren`.
4. Password: It is the password given by the user at the time of installing the mysql database. Password used is “ “.

To connect a Java application with the MySQL database, follow the following steps.

- First create a database and then create a table in the mysql database.
- To connect java application with the mysql database, `mysqlconnector.jar` file is required to be loaded.
- download the jar file `mysql-connector.jar`
- add the jar file to the same folder as the java program.
- Compile and run the java program to retrieve data from the database.

Conclusion: Data has been retrieved successfully from a table by establishing database connectivity of java program with mysql database.

2. Explain steps to connect a java application with the MySQL database

- Download and install MySQL Connector/J.
- Include the MySQL Connector/J JAR file in your Java project.
- Import the necessary classes from the `'java.sql'` package.
- Use the `'DriverManager.getConnection()'` method to establish a connection to the MySQL database.
- Provide the JDBC URL, username, and password for authentication.
- Perform database operations using `'Statement'` or `'PreparedStatement'` objects.
- Close the connection and resources after completing the database operations.



Implementation:

```
import java.sql.*;

public class MusicLibraryManager {
    // JDBC URL, username, and password of MySQL server
    private static final String JDBC_URL = "jdbc:mysql://localhost:3306/music_library";
    private static final String USERNAME = "your_username";
    private static final String PASSWORD = "your_password";

    public static void main(String[] args) {
        try (Connection connection = DriverManager.getConnection(JDBC_URL, USERNAME,
PASSWORD)) {
            System.out.println("Connected to the database.");
            // Example SQL query to retrieve data
            String sql = "SELECT * FROM Song";

            try (Statement statement = connection.createStatement();
                ResultSet resultSet = statement.executeQuery(sql)) {

                // Process the result set
                while (resultSet.next()) {
                    int songId = resultSet.getInt("SongID");
                    String title = resultSet.getString("Title");
                    int duration = resultSet.getInt("DurationInSeconds");
                    System.out.println("Song ID: " + songId + ", Title: " + title + ", Duration: " + duration + "
seconds");
                }
            } catch (SQLException e) {
                System.err.println("Error connecting to the database: " + e.getMessage());
            }
        }
    }
}
```

Conclusion: Data has been retrieved successfully from a table by establishing database connectivity of java program with mysql database.

2. Explain steps to connect a java application with the MySQL database

- Download and install MySQL Connector/J.



Vidyavardhini's College of Engineering and Technology

Department of Artificial Intelligence & Data Science

- Include the MySQL Connector/J JAR file in your Java project.
- Import the necessary classes from the `java.sql` package.
- Use the `DriverManager.getConnection()` method to establish a connection to the MySQL database.
- Provide the JDBC URL, username, and password for authentication.
- Perform database operations using `Statement` or `PreparedStatement` objects.
 - Close the connection and resources after completing the database operations.