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

**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.

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

Ans
1. Download MySQL JDBC Driver: Download the MySQL JDBC driver (mysql-connector.jar) from the MySQL website or Maven repository.
2. Create a Database and Table: Use MySQL client tools or command-line interface to create a database and table within MySQL. For example:

```sql
CREATE DATABASE mydatabase;
USE mydatabase;
CREATE TABLE mytable (
    id INT PRIMARY KEY,
    name VARCHAR(50)
);
```

3. Load MySQL JDBC Driver: Load the MySQL JDBC driver class using `Class.forName()`. This step is necessary to register the driver with the DriverManager.

```java
Class.forName("com.mysql.cj.jdbc.Driver");
```

4. Establish Connection: Use `DriverManager.getConnection()` to establish a connection to the MySQL database by providing the JDBC URL, username, and password.

```java
String url = "jdbc:mysql://localhost:3306/mydatabase";
String username = "root";
String password = "password";
Connection connection = DriverManager.getConnection(url, username, password);
```

5. Create Statement: Create a Statement object to execute SQL queries against the database.

```java
Statement statement = connection.createStatement();
```

6. Execute Query: Use the `executeQuery()` method to execute a SELECT query and retrieve data from the database.

```java
ResultSet resultSet = statement.executeQuery("SELECT * FROM mytable");
```

7. Process Results: Iterate through the ResultSet to retrieve data from each row and process it accordingly.

```java
while (resultSet.next()) {
    int id = resultSet.getInt("id");
    String name = resultSet.getString("name");
    System.out.println("ID: " + id + ", Name: " + name);
}
```

8.Close Resources: Close the ResultSet, Statement, and Connection objects in a `finally` block or using try-with-resources to release database resources.

```java
resultSet.close();
statement.close();
connection.close();
```

These steps demonstrate how to establish a connection, execute a query, and retrieve data from a MySQL database using JDBC in a Java application.