1.A Establishment Data Connection

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.ResultSet;

import java.sql.Statement;

public class ConnectSQLite {

public static void main(String[] args) {

Connection connection = null;

ResultSet resultSet = null;

Statement statement = null;

try {

Class.forName("org.sqlite.JDBC");

connection = DriverManager.getConnection("jdbc:sqlite:D:\\testdb.db");

statement = connection.createStatement();

resultSet = statement.executeQuery("SELECT EMPNAME FROM EMPLOYEEDETAILS");

while (resultSet.next()) {

System.out.println("EMPLOYEE NAME: " + resultSet.getString("EMPNAME"));

}

} catch (Exception e) {

e.printStackTrace();

} finally {

try {

// Close resources

resultSet.close();

statement.close();

connection.close();

} catch (Exception e) {

e.printStackTrace();

}

}

}

}

2.A SQL Queries Using JDBC

import java.security.MessageDigest;

import java.security.NoSuchAlgorithmException;

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class UserAuthentication {

public static void main(String[] args) {

String dbUrl = "jdbc:sqlite:/path/to/your/database.db";

try (Connection connection = DriverManager.getConnection(dbUrl)) {

String createTableQuery = "CREATE TABLE IF NOT EXISTS User (" +

"UserID TEXT PRIMARY KEY," +

"PasswordHash TEXT)";

connection.createStatement().executeUpdate(createTableQuery);

String inputUserID = "john\_doe";

String inputPassword = "mySecretPassword";

String hashedPassword = hashPassword(inputPassword);

String selectQuery = "SELECT PasswordHash FROM User WHERE UserID = ?";

try (PreparedStatement preparedStatement = connection.prepareStatement(selectQuery)) {

preparedStatement.setString(1, inputUserID);

ResultSet resultSet = preparedStatement.executeQuery();

if (resultSet.next()) {

String storedHash = resultSet.getString("PasswordHash");

if (hashedPassword.equals(storedHash)) {

System.out.println("Access granted!");

} else {

System.out.println("Incorrect password.");

}

} else {

System.out.println("User not found.");

}

}

} catch (SQLException e) {

e.printStackTrace();

}

}

private static String hashPassword(String password) {

try {

MessageDigest md = MessageDigest.getInstance("SHA-256");

byte[] hashBytes = md.digest(password.getBytes());

StringBuilder hexString = new StringBuilder();

for (byte b : hashBytes) {

hexString.append(String.format("%02x", b));

}

return hexString.toString();

} catch (NoSuchAlgorithmException e) {

e.printStackTrace();

return null;

}

}

}

3.A Prepared Statement

import java.sql.Connection;

import java.sql.DriverManager;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.SQLException;

public class SelectQueryExample {

public static void main(String[] args) {

String url = "jdbc:mysql://localhost:3306/mydatabase";

String user = "username";

String password = "password";

String sql = "SELECT \* FROM users WHERE username = ?";

try (

Connection conn = DriverManager.getConnection(url, user, password);

PreparedStatement pstmt = conn.prepareStatement(sql);

) {

pstmt.setString(1, "desired\_username");

try (ResultSet rs = pstmt.executeQuery()) {

while (rs.next()) {

int id = rs.getInt("id");

String username = rs.getString("username");

String email = rs.getString("email");

System.out.println("ID: " + id + ", Username: " + username + ", Email: " + email);

}

}

} catch (SQLException e) {

e.printStackTrace();

}

}

}