JDBC setup on windows with sqlite

1]Download sqlite →

SQLite Download Page

(A bundle of command-line tools for managing SQLite database files, including the command-line shell program, the sqldiff.exe program, and the sqlite3_analyzer.exe program. 64-bit.)

2]Download driver for sqlite →

SQLite JDBC driver 3.34 (npackd.org)

3]now in 1st step it get downloaded as a zip file, unzip it and move the 3 files from this file into c drive under new folder name as “sqlite”

4]Also create a folder called MyJava in c drive only into which we are going to store jdbc program.

Into this folder copy the file which got downloaded in step 2 i.e (sqlite-jdbc-3.34.0)

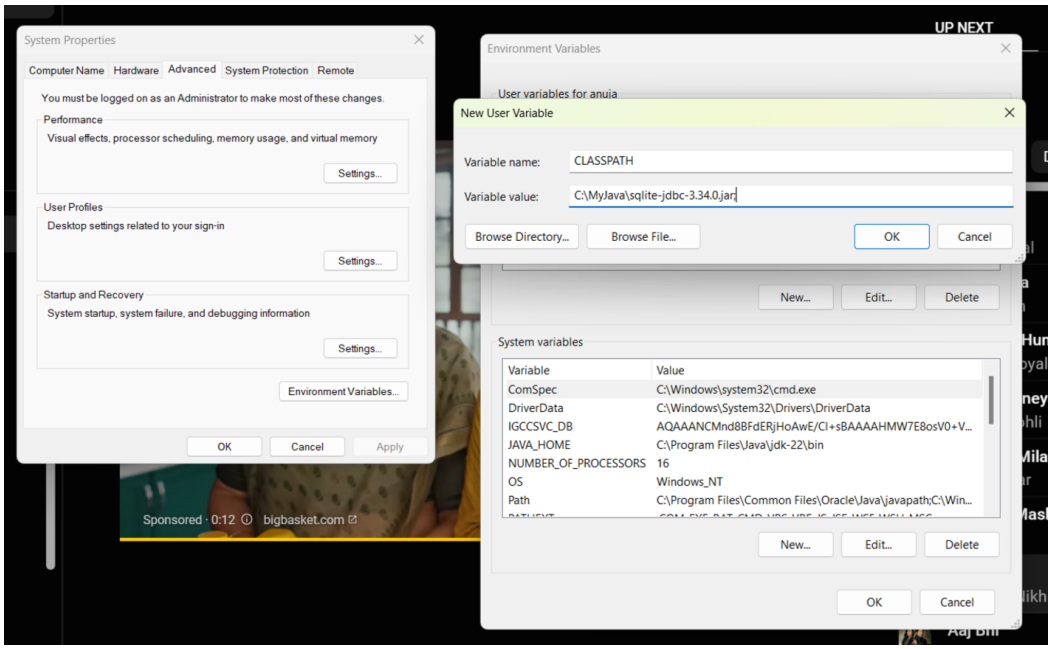
5] Now in terminal set the classpath for the above file :

C:\>cd MyJava

C:\MyJava>set classpath=c:\MyJava\sqlite-jdbc-3.34.0.jar;

6]Now go to advanced system settings in settings →

Environmental variables → Create new



6]To run sqlite →

C:\sqlite>sqlite3

7] To create db file

sqlite> .open java.db

sqlite> .databases
main: C:\sqlite\java.db r/w

8]JDBC program →

```
import java.sql.*;

public class DatabaseOperations {
    private static final String DB_URL = "jdbc:sqlite:C://sqlite
```

```
    public static void main(String[] args) {
        try {
            // Register JDBC driver
            Class.forName("org.sqlite.JDBC");

            // Establish connection
            Connection con = DriverManager.getConnection(DB_URL

            // Create table if not exists
            createTable(con);

            // Insert data into the table
            insertData(con);

            // Delete record for Roll_No 5
            deleteRecord(con, 5);

            // Update city from Sangli to Pune
            updateCity(con, "Sangli", "Pune");

            // Display names of students with marks greater thar
            displayStudentsAbove60(con);

            // Display students according to their marks (Descer
            displayStudentsDescending(con);

            // Close connection
            con.close();
        } catch (ClassNotFoundException e) {
            System.err.println("Error loading SQLite JDBC drive
        } catch (SQLException e) {
            System.err.println("Database error: " + e.getMessage
        } catch (Exception e) {
            System.err.println("Error: " + e.getMessage());
        }
```

```
    }

    private static void createTable(Connection con) throws SQLEx
        String createTableQuery = "CREATE TABLE IF NOT EXISTS St
                "Roll_No INTEGER PRIMARY KEY," +
                "Name TEXT," +
                "City TEXT," +
                "Grade CHAR," +
                "Marks REAL)";
        Statement stmt = con.createStatement();
        stmt.executeUpdate(createTableQuery);
        stmt.close();
    }

    private static void insertData(Connection con) throws SQLExc
        String insertQuery = "INSERT INTO Student (Roll_No, Name
        PreparedStatement pstmt = con.prepareStatement(insertQue

        // Inserting data for each student
        insertStudent(pstmt, 1, "Atul", "Sangli", 'A', 90.50);
        insertStudent(pstmt, 2, "Sangram", "Sangli", 'B', 70.25)
        insertStudent(pstmt, 3, "Satya", "Mumbai", 'B', 61.36);
        insertStudent(pstmt, 4, "Jaydeep", "Pune", 'B', 60.95);
        insertStudent(pstmt, 5, "Prashant", "Sangli", 'C', 55.20
        insertStudent(pstmt, 6, "Abhi", "Pune", 'C', 55.84);

        pstmt.close();
    }

    private static void insertStudent(PreparedStatement pstmt, i
        pstmt.setInt(1, rollNo);
        pstmt.setString(2, name);
        pstmt.setString(3, city);
        pstmt.setString(4, String.valueOf(grade));
        pstmt.setDouble(5, marks);
        pstmt.executeUpdate();
```

```
    }

    private static void deleteRecord(Connection con, int rollNo)
        String deleteQuery = "DELETE FROM Student WHERE Roll_No
        PreparedStatement pstmt = con.prepareStatement(deleteQue
        pstmt.setInt(1, rollNo);
        pstmt.executeUpdate();
        pstmt.close();
    }

    private static void updateCity(Connection con, String oldCit
        String updateQuery = "UPDATE Student SET City = ? WHERE
        PreparedStatement pstmt = con.prepareStatement(updateQue
        pstmt.setString(1, newCity);
        pstmt.setString(2, oldCity);
        pstmt.executeUpdate();
        pstmt.close();
    }

    private static void displayStudentsAbove60(Connection con) t
        String query = "SELECT Name FROM Student WHERE Marks > 6
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
        System.out.println("Students with marks greater than 60
        while (rs.next()) {
            String name = rs.getString("Name");
            System.out.println(name);
        }
        rs.close();
        stmt.close();
    }

    private static void displayStudentsDescending(Connection con
        String query = "SELECT * FROM Student ORDER BY Marks DES
        Statement stmt = con.createStatement();
        ResultSet rs = stmt.executeQuery(query);
```

```
        System.out.println("Students sorted by marks (Descending
        while (rs.next()) {
            int rollNo = rs.getInt("Roll_No");
            String name = rs.getString("Name");
            String city = rs.getString("City");
            char grade = rs.getString("Grade").charAt(0);
            double marks = rs.getDouble("Marks");
            System.out.println(rollNo + "\t" + name + "\t" + cit
        }
        rs.close();
        stmt.close();
    }
}
```