Tuesday, December 21, 2021 10:42 AM

JDBC = standard provided by JEE java database connectivity

Commonly used JEE interfaces for JDBC

1. Java.sql.Connection
2. Java.sql.Statement
3. Java.sql.PreparedStatement
4. Java.sql.CallableStatement
5. Java.sql.Resultset
6. Javax.sql.DataSource

These interfaces are used for Database connectivity through Java .
This will be a 2 tier client server distributed application---

1. MySql Server (non java process)
2. Java Client program (jvm java process)
3. Component called as **Connector** ----- it must have implemented all the JEE interfaces with respect to the DB
   a. Mysql-connector.jar

Open mysql installer folder -----
Run the mysqlinstaller.exe application
Click on Add
Connectors
connectorJ
Select the topmost version , click the =>
Install
CHECK the folder , you should find the connector J ----
mysql

C:\program files(x86) \MySQL \ConnectorJ \ mysql-connector.jar

---

Java program <----------->Translator/Convertor }}} **DRIVER**<---------->MySql

Yellow ---java client
Green ---mysql server

Types of JDBC Driver ----------
Type1 Driver ---- This is **outdated** from JDK1.8 onwards (JDBC-ODBCDriver)
---Platform Dependent Driver that worked with ONLY windows ODBC product
---provided by JDK
Type2 Driver --- This was also platform dependent driver provided by DB vendor for that DB
Type3 Driver --- This was platform independent Set of drivers for different Databases
(platform independent version of ODBC)
Type4 Driver ----- WE will use this !!! --- This is platform independent driver provided by the DB vendor for specific DB

Steps to write the Java JDBC Client ----------

1. Add the connector to build path
2. Load the driver class ---- class has to be loaded in the RAM
3. Create JDBC url ---
   URL = uniform resource locator

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

Driver class is in com.mysql.cj.jdbc package which has static block which runs when driver class being used

static block required no call to run. It will run as soon as class is loaded

This is needed to **locate** the SERVER process **resource** from the Client process
The uniform ---common format of the resource locator is as follows ----
Protocol : servicename : details
jdbc : mysql : //localhost:3306/Dbname

4. Get the connection
5. Create Statements---
   Java.sql. Statement interface API(methods) ----
   executeUpdate(sql) ---------- used to execute the DML queries
   executeQuery (sql) --------used to execute SELECT queries
   execute (sql) ----------used to call stored procedures or DDL queries
6. Fire query

HW ----------

1. Create the table from mysql client----------
   Product table
   Columns ---product_id, product_name, product_cost, product_desc, product_expiry_date

   Write a Java Client that connects to the database and inserts records into product table in a loop, till user says "yes"

2. Write a Java Client that connects to the database and asks the user to enter a product id and show the details of that product id on console . (select query with where clause)
3. Write a Java Client that connects to the data base and asks the user to enter a product id and delete the row from the database.
4. Write a java client that shows all products in the table with following info
   a. Product name Expiry Date
5. Write a java client that accepts a product id from the user and also enters a new product description ---update the table so that desc column is modified for that product.