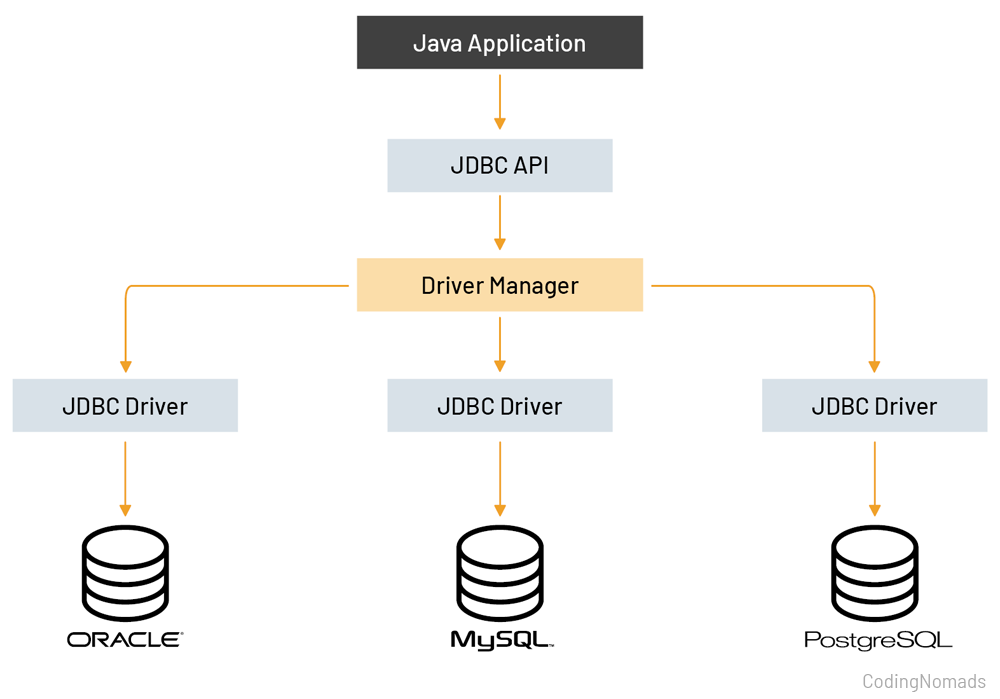
**JDBC (Java Database Connectivity) –**

What is JDBC?

JDBC (Java Database Connectivity) is an API in Java that allows Java programs to interact with relational databases like MySQL, Oracle, and PostgreSQL.



**JDBC Components:**

1. JDBC API – Interfaces like Connection, Statement, ResultSet, PreparedStatement etc.

2. JDBC Driver – Connects Java applications to the database. Types:

**Type 1:** JDBC-ODBC Bridge Driver

**Type 2:** Native-API Driver

**Type 3:** Network Protocol Driver

**Type 4:** Thin Driver (Most commonly used)

**Steps to Connect Java with JDBC**

1. Load the Driver
2. Define the url

3. Establish a Connection

4. Create a Statement

5. Execute the Query

6. Process the Result

7. Close the Connection

There are three types of statements

1. Statement
2. PreparedStatement
3. CallableStatement

**1. Statement (createStatement())**

With a single query, you can insert only one record at a time.

If you want to insert multiple records, you need to execute multiple queries separately

**import** java.sql.Connection;

**import** java.sql.DriverManager;

**import** java.sql.ResultSet;

**import** java.sql.SQLException;

**import** java.sql.Statement;

**public** **class** JdbcStatement {

**public** **static** **void** main(String[] args) {

// **TODO** Auto-generated method stub

**try** {

// Register JDBC Driver

// load the driver

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

// define the URL(Uniform Resource Locator)

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

String username = "root";

String password = "test";

// Establish a Connection

Connection con = DriverManager.*getConnection*(url, username, password);

// Create a Statement Object

Statement stmt = con.createStatement();

stmt.executeUpdate("insert into employee values(9,'hari','sales',7000)");

// execute a query

ResultSet rs=stmt.executeQuery("select \* from employee");

// process the result

**while**(rs.next()) {

System.***out***.print(rs.getInt(1)+" ");

System.***out***.print(rs.getString("name")+" ");

System.***out***.println(rs.getString("dept"));

}

// Close the connection

con.close();

} **catch** (ClassNotFoundException e) {

System.***out***.println(e);

}**catch**(SQLException e) {

System.***out***.println(e);

}

}

}

**2. PreparedStatement (prepareStatement())**

With a single query, you can insert multiple records efficiently.

You prepare the query once and execute it multiple times with different values.

**import** java.sql.\*;

**class** InsertPrepared {

**public** **static** **void** main(String args[]) {

**try** {

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

Connection con = DriverManager.*getConnection*("jdbc:mysql://localhost:3306/project", "root", "test");

PreparedStatement stmt = con.prepareStatement("insert into employee values(?,?,?,?)");

stmt.setInt(1, 101);

stmt.setString(2, "Ratan");

stmt.setString(3, "marketing");

stmt.setInt(4, 10000);

stmt.executeUpdate();

stmt.setInt(1, 102);

stmt.setString(2, "Rajini");

stmt.setString(3, "actor");

stmt.setInt(4, 10000);

stmt.executeUpdate();

stmt.setInt(1, 101);

stmt.setString(2, "Ajith");

stmt.setString(3, "actor");

stmt.setInt(4, 10000);

stmt.executeUpdate();

System.***out***.println(" records inserted");

con.close();

} **catch** (Exception e) {

System.***out***.println(e);

}

}

}

**3. CallableStatement (prepareCall())**

This is used to call stored procedures from the database.

Stored procedures can handle multiple operations in one call, including inserting multiple records.

DELIMITER $$

CREATE PROCEDURE getAllEmployeeDetails()

BEGIN

SELECT \*

FROM employee;

END $$

DELIMITER ;

**import** java.sql.\*;

**public** **class** CallableStatementExample {

**public** **static** **void** main(String[] args) {

// Database connection details

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

String username = "root";

String password = "test";

Connection conn = **null**;

CallableStatement stmt = **null**;

ResultSet rs = **null**;

**try** {

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

// Establish connection

conn = DriverManager.*getConnection*(url, username, password);

String sql = "{call getAllEmployeeDetails()}";

stmt = conn.prepareCall(sql);

// Execute the stored procedure

rs = stmt.executeQuery();

// Process the result set

**while** (rs.next()) {

System.***out***.print(rs.getInt(1)+" ");

System.***out***.print(rs.getString("name")+" ");

System.***out***.println(rs.getString("dept"));

}

}**catch**(ClassNotFoundException e) {

System.***out***.println(e);

}

**catch** (SQLException e) {

e.printStackTrace();

} **finally** {

// Close resources

**try** {

rs.close();

conn.close();

stmt.close();

} **catch** (SQLException e) {

// **TODO** Auto-generated catch block

System.***out***.println(e);

}

}

}

}