JDBC(java to database connectivity)

It is an API used to connected and execute the query with database from java

*it is a specification from sun micro systems ,that provides a standard abstraction to communication with various database

API(application programming interface)

It is an interface which is used to establish the connection between two systems

How does JDBC helps to connect java to database(theory)

When we have both java application and database ,and we want those to communicate each other ,but java does not understand sql language and database does not understand java language .

Now we need one mechanism which takes java statements and converts it into database understandable language ,and to convert database response into java understandable language, for that we have our jdbc api

Now jdbc api will take java statements and converts into database understable language and sends the request ,now database will understand it and does respective work

When database will take the information ,it gives to jdbc api and it will converts particular database into java understandable language

So we will jdbc will act as interface between java to database

- Jdbc api will convert java language into database understandable language ,but it never converts java to particular database language

For that we need driver software

Note:

Jdbc api alone cannot convert java into particular database language ,inorder to do it we need a driver software,which is provided by database vendors

Since it is a third party application ,inorder to use it we need to integrate it ,inorder to integrate we need to add the dependency

Driver:

It is a class which is present in org.postgresql package(since we are using postgresql database)

We need this ,inorder to convert database language into particular postgresql database language

Five to achieve connection between java to database using jdbc

1.load (or) register the driver

2.create connection

3.establish statement

4.execute query

5.close connection

1.load (or) register the driver

Inorder to use the particular driver ,we need to bring that into our java memory

We can load our driver class into java memory either in 2 ways ,either we can load (or) register but not both at a time

a.Register Driver Class

inorder to register it

step1.create the object of driver class

Driver driver= new Driver();

*import it from org.postgresql package

Step2. Ask the DriverManager to register the driver

DriverManager is a class present in java.sql package

In that particular class we have a static method that is – registerDriver(reference)

Which accept object reference as a parameter

Now

DriverManager.regesterDriver(driver);

-pass the driver object reference to the particular method ,then it is going to register your driver class into java memory

As soon as we wrote that line ,it will cause you ,checked exception(classNotFoundException) always try to handle it by using try and catch

Loading the Driver

Inorder to load the driver

-Class.forName("org.postgresql.Driver");

Class is class present in java.lang package, in that class we have a static method called as forName() which will accept a string argument ,where we will pass a fullyQualifiesClass Name of the particular driver provided by database vendors

Now this method is going to load our driver class into memeory

it will cause you ,checked exception(classNotFoundException) always try to handle it by using try and catch

2.create connection

To establish a connection between java to database we have different ways

We create a connection by passing three strings

1.url 2.username 3.password

url- “jdbc:postgresql://localhost:5432/databasename”

username-“postgres”

password-“root”

We have getConnection() is public static method present in DriverManager class ,that method is used to build the connection between java to database

Mainly we have 3 getconnection() ,which is overloaded

1.getConnection(string url, String username ,String password) ;

Here we have a getconnection method ,which accept three string arguments ,where will pass url ,inorder to get database location and password and username inorder to unlock database

DriverManager.getConnection(url,username,password);

This line will throw you the checked exception ,try to handle it by using try and catch

2.getConnection(String url)

This method accept single string where we will pass user,password,url in one string

String url-
“jdbc:postgresql://localhost:5432/databasename?user=postgres&password=root”

3.getConnection(properties file,String url);

These methods are the example for built in overloaded methods

These methods will return the object of Connection interface

Connection connection=DriverManager.getConnection();

Since connection is an interface we cannot able to create an object directly ,so we have get connection method

Connection is an interface present in java.sql package

*connection objects gets created only if url,password,username are accurate ,if any one is wrong the getconnection() not all create an object

Instead it will throw you an sqlExeception

3.create Statement:

We can create statement in 2 ways

1.Statement 2.PreparedStaement

We can create statement by calling the method that

1.Is present in connection

- Statement statement = connection.createStatement();
- Statement is a interface present in java.lang package
- We can get statement object by calling createStatement method
- createStatement() throws SQLException
- generally statement object used to execute static querry

2.creating statement using PreparedStatement

PreapresStatement is an interface present in java.sql package
It is child of Statement interface

- we can get the preparedStatement object by calling prepareStatement()
- prepareStatement method throws the sql exception
- this statement generally used to execute the dynamic querries
- it is used to execute parameterised query ,we use the delimeter(?)

and pass the value to it by using setter method

we prefer prepareStatement over an Statement to execute any querry dynamically

4,Executequerry

This step is important inorder to insert the query inside the database

Here we have three methods inorder to execute the querries

1.execute(String querry):

Used to execute both select and non-select queries

-it return Boolean value

Returns true if query is successfully inserted

2.exceuteUpdate(String query)

This method used to execute only non-select queries

Return type of this method is int

It will return the value that how many rows are effected

3.executeQuery(String query)

This method is used to execute only select queries

Where this method will return the ResultSet object

5.close connection

It is one of the important part in our jdbc

Because if we don't close the connection ,it will leads the connection memory leakage

-inorder to release other database resources ,the connection may be holding on to connect

-the safe pattern in java is to close connection always

ResultSet:

It is an interface ,present in java.sql package

It holds the set of data's return by executeQuery()

-result set will act as cursor ,where it will point above the table initially

And to make cursor to point first row of the given table we will make use of next() ,where this method helps us to move the cursor to nextline

next();

- it is present inside ResultSet interface ,where it leads Boolean return type
- if there any row existed in the next position it will return true
- if there any row is not existed in the nextposition ,it will return us a false

to fecth the data from resultset we have
getX(column index (or) column id)
-these method will fecth the value from particular column based on its index (or) based on column name
X—represents datatype of a data

Batch Execution (or)Batch processing

When ever we have multiple querries which need to be executed in one call we are going to make use of batch processing
In batch execution all querries combined together and executed at once
*batch processing allows us to group related sql statement into a batch and submit them with one call to the database

Advanatage :

Make the execution performance fast
-helps in increasing the data consistency
-we use addBatch() to add multiplr querries
-we call executeBatch() to execute ,the querries into database