**JDBC-JAVA DATABASE CONNECTIVITY:**

JDBC stands for java database connectivity. It is a standard API provided by Oracle for java application to interact with different set of databases.

**JAVA APP**

**DATABASES**

**What is API?**

API is predefine classes and interfaces in java

**Why JDBC?**

To store users data permanently we are using JDBC by using java programs .

Example

Class Test{

Public static void main(String [] args){

Int a =15;//this will there in memory when will execute the program after that it will be deleted

String s = “Hello”;

}

}

**How JDBC Works (Architecture of JDBC)**

**Database Driver**

**Databases**

**JDBC Driver Layer**

**JDBC Application Layer**

**Maria DB Driver**

**Oracle Driver**

**SQL Driver**

**JDBC API**

**Java Application**

**JDBC API**

* Import classes and Interfaces:
* Java.sql.DriverManager
* Java.sql.Connection
* Java.sql.Statement
* Java.sql.PreparedStatement
* Java.sql.CallableStatement
* Java.sql.ResultSet
* Java.sql.ResultSetMetaData
* Java.sql.DatabaseMetaData
* Java.sql.SQLException

Software Needed For:

Any code editor eclipse , Notepad++, Netbeans etc

Any Database: MySql ,Oracle,etc

JDBC Connector for respective database

**How to set jar file in your class path:**

**BY Using Command Prompt:**

Step-1 Go to your file location where you downloaded your database driver and copy that path.

Step-2 Go to environment variable and set your class path go to class path and edit new and paste your path what you copied and end with **;.;**

Step-3 for checking connection open your cmd and type there

**javap com.mysql.jdbc.Driver**

**ByUsing Eclipse:**

**Step-1:** Go to Eclipse right click on your project and go to properties then go to Build Path and clicked on libraries.

**Step-2:** If you are getting classpath in libraries then clicked on classpath and right side you will get add external jar file clicked there and browse the location where you have downloaded the mysql connector jar file.

**#Step to connect with the database:**

**1) Load the driver:**

* Class.forName("com.mysql.jdbc.Driver")//we will write this code inside try catch block OR
* DriverManager.registerDriver(new com.mysql.jdbc.Driver());// you can follow anyone of them

**2)** **Create a Connection:**

* Connection conn= DriverManager.getConnection("jdbc:mysql://localhost:3306/dbname","username","password");

**3) Create query:**

* Statement //simple query
* PreparedStatement //parameterized query
* CallableStatement //Stored procedure
* Example:
* String s = "select \* from students";
* Statement stmt = conn.createStatement();
* ResultSet set=stmt.excecuteQuery(q); //we can write update also

**4) Process the data/Execute the Query:**

* While(set.next())
* {

int id = set.getInt("StudentID");//or we can pass number also 1 2 3 row wise

String name = set.getString("StudentName");

System.out.println(id);

System.out.println(name);  
}

**5) Close the Connection**

Conn.close();

**#Example**

package jdbc\_Programs;

import java.sql.\*;

public class FirstJDBC {

public static void main(String[] args) {

try {

//load the driver :

Class.forName("com.mysql.cj.jdbc.Driver");//fully qualified name

//creating a connection

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

String username = "root";

String password = "Aliya@123";

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

if(conn.isClosed()) {

System.out.println("Connection is still closed!!!!");

}

else {

System.out.println("connection created!!!!");

}

}catch(Exception e)

{

e.printStackTrace();

}

}

}

**# How to create the table in database using java code**

package jdbc\_Programs;

import java.sql.\*;

public class FirstTableJDBC {

public static void main(String[] args) {

try {

//Load the Driver

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

//Creating a connection

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

String username = "root";

String password = "Aliya@123";

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

//create a query

String query = "create table table1(tId int(20) primary key auto\_increment,"

+ "tName varchar(30) not null, tCity varchar(400))";

//create a statement

Statement stmt = conn.createStatement();

stmt.executeUpdate(query);// it will create the table in database

System.out.println("table created in database!!!");

conn.close();

}catch(Exception e)

{

e.printStackTrace();

}

}

}

**#How to insert data into table by using Prepared Statement**

**#Example:**

package jdbc\_Programs;

import java.sql.\*;

public class InsertDataIntoTable1 {

public static void main(String[] args) {

try {

//Load the driver

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

//create a connection

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

String username = "root";

String password = "Aliya@123";

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

//create a insert statement

String insert = "insert into table1(tName,tCity) values(?,?)";//this is parameterized query

//get the prepared statement object

PreparedStatement pstmt =conn.prepareStatement(insert);

//Set the values to tables

pstmt.setString(1, "Ashraf");

pstmt.setString(2, "hafeezpet");

pstmt.executeUpdate();

System.out.println("Inserted....");

conn.close();

}catch(Exception e)

{

}

}

}

**#inserting Data to Table with dynamic input JDBC**

Here we have to use buffer reader for inserting dynamic data in the table

And buffer reader present inside java.io package

package jdbc\_Programs;

import java.io.\*;

import java.sql.\*;

public class InsertDataIntoTable1 {

public static void main(String[] args) {

try {

//Load the driver

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

//create a connection

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

String username = "root";

String password = "";

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

//create a insert statement

String insert = "insert into table1(tName,tCity) values(?,?)";//this is parameterized q

//get the prepared statement object

PreparedStatement pstmt =conn.prepareStatement(insert);

//buffer reader to read the dynamic value inside table

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

System.out.println("Enter Name:");

String name = br.readLine();

System.out.println("Enter City");

String city = br.readLine();

//Set the values to tables

pstmt.setString(1, name);

pstmt.setString(2, city);

pstmt.executeUpdate();

System.out.println("Inserted....");

conn.close();

}catch(Exception e)

{

e.printStackTrace();

}

}

}

**#Inserting image to Data Base using java**

First we have to create the table name called image

**Create table images (id int primary key auto\_increment, pic blob)**

package jdbc\_Programs;

import java.io.FileInputStream;

import java.sql.\*;

public class ImageSave {

public static void main(String[] args) {

try {

//load the driver

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

//create a connection

Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb","root","Aliya@123");

//write a query statement

String query = "insert into images(pic) values(?)";

PreparedStatement ptsmt = conn.prepareStatement(query);

//this is not valid for images

//ptsmt.setString(1, "values");

//for image will write valid images

FileInputStream fis = new FileInputStream("image.jpg");

ptsmt.setBinaryStream(1,fis,fis.available());

ptsmt.executeUpdate();

System.out.println("done....");

conn.close();

}catch(Exception e)

{

e.printStackTrace();

}

}

}

**#Inserting Large image to Database using java App in database:**

**#ConnectionProvider.java**

package jdbc\_Programs;

//Singleton Class

import java.sql.Connection;

import java.sql.DriverManager;

public class ConnectionProvider {

private static Connection conn;

public static Connection getConnection() {

try {

if(conn == null) {

//load the driver

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

conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/mydb","root","Aliya@123");

}

}catch(Exception e)

{

e.printStackTrace();

}

return conn;

}

}

**#LargeImageApp.java**

package jdbc\_Programs;

import java.io.File;

import java.io.FileInputStream;

import java.sql.Connection;

import java.sql.PreparedStatement;

import javax.swing.JFileChooser;

import javax.swing.JOptionPane;

public class LargeImageApp {

public static void main(String[] args) {

try

{

Connection c = ConnectionProvider.getConnection();

String query="insert into images(pic) values(?)";

PreparedStatement pstmt = c.prepareStatement(query);

//Java swing

JFileChooser jfc = new JFileChooser();//it will choose the file

jfc.showOpenDialog(null);// it will show the dioulogue box

File file = jfc.getSelectedFile();

FileInputStream fis = new FileInputStream(file);

pstmt.setBinaryStream(1,fis,fis.available());

pstmt.executeUpdate();

JOptionPane.showMessageDialog(null, "success");

}catch(Exception e)

{

e.printStackTrace();

}

}

}

**#Updating Data Of Table:**

**#Select Data from Table:**

package jdbc\_Programs;

import java.sql.Connection;

import java.sql.ResultSet;

import java.sql.RowId;

import java.sql.Statement;

public class SelectingDataFromTable {

public static void main(String[] args) {

try {

Connection conn = ConnectionProvider.getConnection();

String query = "select \* from table1";

Statement stmt=conn.createStatement();

ResultSet set = stmt.executeQuery(query);

while(set.next()) {

int id = set.getInt(1);

String name = set.getString(2);

String city = set.getString(3);

System.out.println(name+":"+id+":"+city);

}

}catch(Exception e) {

e.printStackTrace();

}

}

}

**#What is Result Set in JDBC:**

Result set is an interface it store the data in the table formate.

**#How to get object of ResultSet:**

**ResultSet set = stmt.executeQuery("select \* from table1");**

**#CRUD Operation By using JDBC:**

**Start.java**  
package studentapp.crudoperation;

import java.io.BufferedReader;

import java.io.IOException;

import java.io.InputStreamReader;

public class Start {

public static void main(String[] args) throws NumberFormatException, IOException {

System.out.println("Welcome to Student App!!!");

BufferedReader br = new BufferedReader(new InputStreamReader(System.in));

while(true) {

System.out.println("Press 1 to ADD student!!!");

System.out.println("Press 2 to DELETE student!!!");

System.out.println("Press 3 to DISPLAY student!!!");

System.out.println("Press 4 to Exit App!!!");

int c = Integer.parseInt(br.readLine());

if(c==1)

{

//add student

System.out.println("Enter the username!!!");

String name = br.readLine();

System.out.println("Enter User Phone!!!!");

String phone = br.readLine();

System.out.println("Enter user city!!!!");

String city = br.readLine();

//create student object to store student

Student st = new Student(name, phone, city);

boolean answer=StudentDao.insertStudentToDB(st);

if(answer) {

System.out.println("student added successfully!!!!!");

}else {

System.out.println("something went wrong!!!!");

}

System.out.println(st);

}else if(c==2)

{

//delete student

System.out.println("Enter student id to be deleted!!!!!!!");

int userId=Integer.parseInt(br.readLine());

boolean f =StudentDao.deleteStudent(userId);

if(f) {

System.out.println("Deleted......");

}else {

System.out.println("Something Went Wrong!!!!");

}

}else if(c==3)

{

//display student

StudentDao.showAllStudent();

}else if(c==4)

{

//exit

break;

}else {

}

}

System.out.println("Thankyou for using my application!!!!!!!! ");

System.out.println("See you soon Bye Bye!!!!!!!!!!!");

}

}

**Student.java**

package studentapp.crudoperation;

public class Student {

private int student\_id;

private String student\_name;

private String student\_phone;

private String student\_city;

public Student(int student\_id, String student\_name, String student\_phone, String student\_city) {

this.student\_id = student\_id;

this.student\_name = student\_name;

this.student\_phone = student\_phone;

this.student\_city = student\_city;

}

public Student(String student\_name, String student\_phone, String student\_city) {

this.student\_name = student\_name;

this.student\_phone = student\_phone;

this.student\_city = student\_city;

}

public int getStudent\_id() {

return student\_id;

}

public void setStudent\_id(int student\_id) {

this.student\_id = student\_id;

}

public String getStudent\_name() {

return student\_name;

}

public void setStudent\_name(String student\_name) {

this.student\_name = student\_name;

}

public String getStudent\_phone() {

return student\_phone;

}

public void setStudent\_phone(String student\_phone) {

this.student\_phone = student\_phone;

}

public String getStudent\_city() {

return student\_city;

}

public void setStudent\_city(String student\_city) {

this.student\_city = student\_city;

}

@Override

public String toString() {

return "Student [student\_id=" + student\_id + ", student\_name=" + student\_name + ", student\_phone="

+ student\_phone + ", student\_city=" + student\_city + "]";

}

}

**#ConnectionGenerator.java**

package studentapp.crudoperation;

import java.sql.Connection;

import java.sql.DriverManager;

public class ConnectionGenerator {

static Connection conn;

public static Connection create() {

try {

//load the driver

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

//create the connection

String user = "root";

String url = "jdbc:mysql://localhost:3306/student\_manage";

String password = "Aliya@123"; conn=DriverManager.getConnection(url,user,password);

}catch(Exception e) {

e.printStackTrace();

}

return conn;

}

}

**#StudentDao.java**

package studentapp.crudoperation;

import java.sql.Connection;

import java.sql.PreparedStatement;

import java.sql.ResultSet;

import java.sql.Statement;

public class StudentDao {

public static boolean insertStudentToDB(Student st)

{

boolean f = false;

//Jdbc Code....

try {

Connection con = ConnectionGenerator.create();

String insert = "insert into students (sname,sphone,scity) values(?,?,?)";

//prepared statement

PreparedStatement pstmt = con.prepareStatement(insert);

//set the value of parameter

pstmt.setString(1, st.getStudent\_name());

pstmt.setString(2, st.getStudent\_phone());

pstmt.setString(3,st.getStudent\_city());

//excecute...

pstmt.executeUpdate();

f = true;

}catch(Exception e)

{

e.printStackTrace();

}

return f;

}

public static boolean deleteStudent(int userId) {

boolean f = false;

//Jdbc Code....

try {

Connection con = ConnectionGenerator.create();

String insert = "delete from students where sid=?";

//prepared statement

PreparedStatement pstmt = con.prepareStatement(insert);

//set the value of parameter

pstmt.setInt(1,userId);

//excecute...

pstmt.executeUpdate();

f = true;

}catch(Exception e)

{

e.printStackTrace();

}

return f;

}

public static void showAllStudent() {

//Jdbc Code....

try {

Connection con = ConnectionGenerator.create();

String select = "select \* from students";

//For selecting the data we will use statements!!!

Statement stmt = con.createStatement();

//Now will write execute query!!!

ResultSet set = stmt.executeQuery(select);

//Now will write while loop for display all data!!!

while(set.next()) {

int id = set.getInt(4);

String name = set.getString("sname");

String phone = set.getString("sphone");

String city = set.getString("scity");

System.out.println("ID: " + id);

System.out.println("Name: " + name);

System.out.println("Phone: " + phone);

System.out.println("City: " + city);

System.out.println("+++++++++++++++++++++++++++++++++");

}

}catch(Exception e)

{

e.printStackTrace();

}

}

}

**Stored Procedure:**

A stored procedure is a prepared SQL code that you can save, so the code can be reused over and over agai**n.**

**Syntax of Stored Procedure:**

Delimiter $$

Create Procedure sp\_name()

BEGIN

---statements---

END $$

DELIMITER;