**# Java Postgres Connection**

**# Java code to connect with postgres db**

**# Java postgres database connection client**

**# RDBMS database java client**

If we are working on an application where we are getting user data and same we need to serve later, then we need to store user data somewhere.

So now question is, where to store?

Data base is the answer.

A database is a repository of data, where we can store data permanently or for specific time period (if we are using data purging policy the)and retrieve it later whenever needed by using query commands.

But using command on database UI or shell is fine, how we can use same using programs?

If we want to use query command through program, then we need to create database connection client for same.

In which database, RDBMS or NOSQL?

Its depends on your data format, data size, frequency, application type, architecture and design.

How to create database client in JAVA?

JDBC (Java Database Connectivity) provides an APIs that is useful to write Java programs to connect to any database, retrieve the data from the database, and utilize the data in a java program.

Database server (in our case Postgres)

Database Client

In this section, I am using postgres, open source data base to store person table data.

Please use below git hub link and check readme files with steps –

<https://github.com/jainabhishek0406/JavaPostgresConnection>

**Steps to connect with postgres database server.**

1. **Driver registration**

A database driver is a set of classes and interfaces, written according to JDBC API to communicate with a database.

Check PostgresConnection.java from git repo.

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

1. **Connection to a database**

Check PostgresConnection.java from git repo.

connection = DriverManager.getConnection(url, user, password);

1. **Java SQL statements preparation**

lass PostgresOperation {

public String getQueryToCreateTable()

{

String queryString = "create table person " +

"(name Char(20)," +

"age INT," +

"state Char(10))";

return queryString;

}

public String getQueryToInsertData(){

String queryString = "insert into person(name, age, state) values ('Abhishek Jain', 34, 'MP')" ;

return queryString;

}

}

1. **SQL query execution**

//executing create query to create person table under postgres db.

String queryToCreateTable = postgresOperation.getQueryToCreateTable();

System.out.println("\nCreating PERSON table.");

stmt.executeUpdate(queryToCreateTable);

System.out.println("Table created successfully.");

//executing insert query to insert data into postgres person table.

String queryToInsertData = postgresOperation.getQueryToInsertData();

System.out.println("\nInserting data into table.");

stmt.executeUpdate(queryToInsertData);

System.out.println("Data inserted successfully.");

1. **Retrieving the results**

//executing select query to collect person table data in ResultSet

System.out.println("\ngetting data from person table");

resultSet = stmt.executeQuery( "SELECT \* FROM person;" );

System.out.println("printing data from person table");

while ( resultSet.next() ) {

String name = resultSet.getString("name");

int age = resultSet.getInt("age");

String state = resultSet.getString("state");

//Printing data getting from postgress data base.

System.out.println( "\nNAME = " + name );

System.out.println( "AGE = " + age );

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

System.out.println();

}

1. **Closing the connection.**

resultSet.close();

stmt.close();

connection.close();

All above are basics to learn database connectivity using Java code.

After this you can apply filters, store procedures, pooling mechanism etc.

But before using this first string your basic foundation and then apply enhancements and improvements.

**This is the first step to provide basic foundation for you with approach**

**‘lets grow together’.**

**Github link-**

<https://github.com/jainabhishek0406/JavaPostgresConnection>

<https://github.com/jainabhishek0406/JavaPostgresDB_Operations>