# First JDBC Program

A succesfull operation/transaction with the database involves the exceution of **several small steps.** Each step has a **significance** of its own. To write efficient database code, all these steps must be correctly implemented. These steps are:

* **Step 1.** Pre-requisites
* **Step 2.** Connect to database
* **Step 3.** Create Statement
* **Step 4.** Prepare the Query
* **Step 5.** Execute the Query and Collect Data
* **Step 6.** Close Resources

Lets ponder over the above steps one by one

## Step 1. Pre-requisites

Before writing a JDBC API, we need to:

1. Install any database server (here we will use **MySQL** Server).
2. Install a **GUI to operate on** the database server ( here, MySQL GUI - **MySQL Workbench**)
3. While installation, set a **username** and a **password** (We set username: **root** and password: **1234**)
4. Create a schema (We start with **schema** - **school**)
5. Create the first table (We create **table** - **students**)
6. Create columns

|  |  |  |
| --- | --- | --- |
| **S. No** | **Column Name** | **Column Properties** |
| 1. | \_id | Int, Primary Key, Not Null, Unique, Auto Increment |
| 2. | student\_name | Varchar (45), Not Null |
| 3. | student\_class | Int (2), Not Null |
| 4. | student\_fees | Double, Not Null |

1. Make sure that MySQL Server is running before writing the JDBC API.
2. Place the suitable connector jar file in **WEB-INF>>lib** folder. (We use **mysql-connector-java-8.0.23.jar** )

## Step 2. Connect to Database

1. Import the correct packages.

|  |
| --- |
| import java.sql.\*; |

1. Load the JDBC Driver

|  |
| --- |
| Class.forName("com.mysql.cj.jdbc.Driver");  where **com.mysql.cj.jdbc.Driver** is the location of Driver Class |

1. Establish the connection

|  |
| --- |
| Connection conn = DriverManager.getConnection("**jdbc:mysql://localhost:3306/<schema-name>**", "**<username>**", "**<password>**");  where jdbc:mysql://localhost:3306/<schema-name>is **Connection String** |

## Step 3. Create a Statement

|  |
| --- |
| Statement statement = conn.createStatement(); |

## Step 4. Prepare the query

|  |
| --- |
| String insertQuery = "INSERT INTO students (student\_name, student\_class, student\_fees) values('Abhishek Verma', 1, 5000.0)"; |

## Step 5. Execute the query and collect data

|  |
| --- |
| int noOfRowsInserted = stmt.executeUpdate(insertQuery); |

## Step 6. Close the Resources

|  |
| --- |
| statement.close();  connection.close(); |