# Connect to Database

The first step before we can do any database transactions is to **connect with the database.**

### JDBC API to Connect with DB

1. Import the correct packages.

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

1. Load the JDBC Driver

The first thing you need to do before you can open a JDBC connection to a database is to load the JDBC driver for the database. You load the JDBC driver like this:

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

* You only have to load the driver **once for the whole application**.
* You do not need to load it before every connection opened.
* Only before the first JDBC connection opened.

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** |

The following classes and interfaces are used to connect to the database:

|  |  |  |
| --- | --- | --- |
| **S. No** | **Class/Interface Name** | **Function** |
| **1** | Class (C) | Class is a class in which all the drivers should be registered which will be used by the Java Application |
| **2** | forName(String) | forName() is a static method in class Class which loads and register our driver dynamically (by calling DriverManager.registerDriver() ) |
| **3** | Connection (I) | The JDBC Connection class, java. sql. Connection , represents a database connection to a relational database. Before we can read or write data from and to a database via JDBC, we need to open a connection to the database. |
| **4.** | DriverManager (C) | The DriverManager class maintains a list of Driver classes that have registered themselves by calling the method Class.forName() that calls DriverManager.registerDriver() automatically |
| **5.** | getConnection (String, String, String) | This method of Java DriverManager class attempts to establish a connection to the database by using the given database url, username and password |

## Close the Resources

You should explicitly close *Statements*, *ResultSets*, and *Connections* when you no longer need them, unless you declare them in a *try*-with-resources statement (available in JDK 7 and after).

To close a Statement, ResultSet, or Connection object that is not declared in a **try-with-resources** statement, use its close method.

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