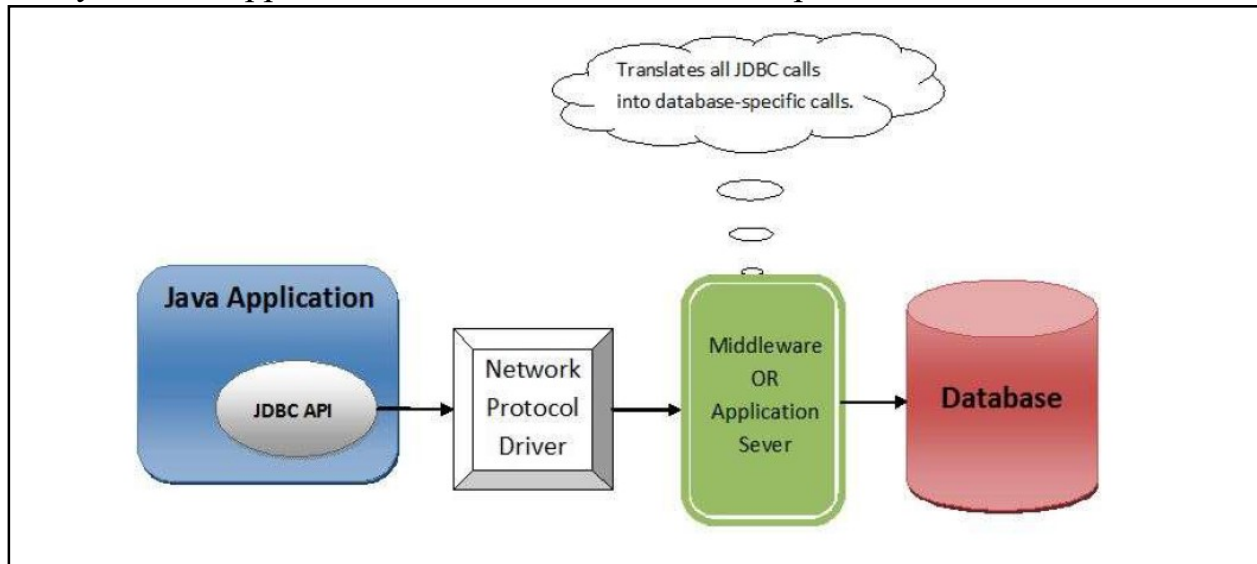


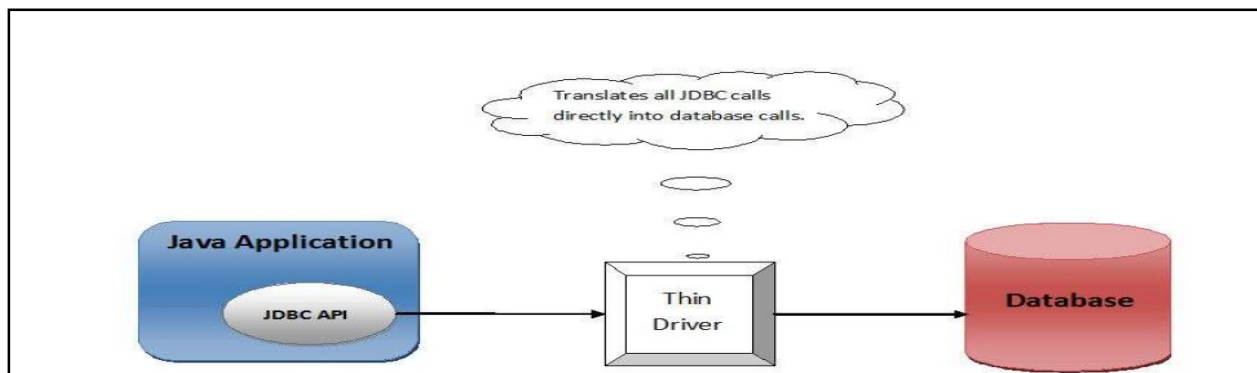
## **Type 3 JDBC Driver( The Network Protocol driver)**

Type 3 JDBC Drivers make use of middle ware or application server that translates all JDBC calls into database specific calls. One of the main advantage of this driver is that it is entirely written in java language. So no portability issues. But it is costly as extra application server or middleware component has to be maintained.



## **Type 4 JDBC Driver(Thin Driver)**

Type 4 JDBC Driver is also called Thin Driver as it directly converts JDBC calls into database specific calls. This driver is most popular among all 4 type of JDBC drivers. This driver is preferred over Type 3 Driver as it removes extra layer of communication (Application Server / Middle ware) and this makes it faster than the Type 3 JDBC Driver. A type 4 JDBC driver is an all Java driver which connects directly to the database. It is implemented for a specific database product. Today, most JDBC drivers are type 4 drivers.



## All JDBC programs do the following:

**Step 1)** load the JDBC driver

**Step 2)** Specify the name and location of the database being used

**Step 3)** Connect to the database with a Connection object

**Step 4)** Execute a SQL query using a Statement object

**Step 5)** Get the results in a ResultSet object

**Step 6)** Finish by closing the ResultSet ,Statement and Connection objects

## Steps to connect to the database in Java:

### 1.Load the driver

The forName() method of Class class is used to register the driver class. This method is used to dynamically load the driver class.

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

### 2.Define the Connection URL

Connection

```
con=DriverManager.getConnection("jdbc:mysql://localhost:3306/gkp","root","root");
```

### 3. Establish the Connection

```
DriverManager.getConnection  
(mysqlURL, username,password);
```

### 4. Create a Statement

```
Statement stmt=con.createStatement();
```

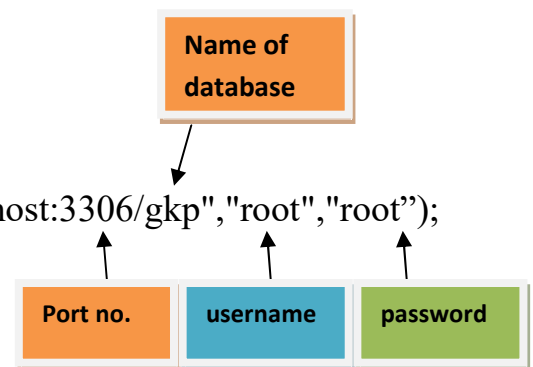
### 5. Execute the query

```
String query = "SELECT column FROM table";
```

```
ResultSet resultSet =statement.executeQuery(query);
```

### 6. Close the connection object

```
connection.close();
```



## Note:

Mysql database has been used throughout Jdbc, although we can use some other database as well.