JDBC Architecture

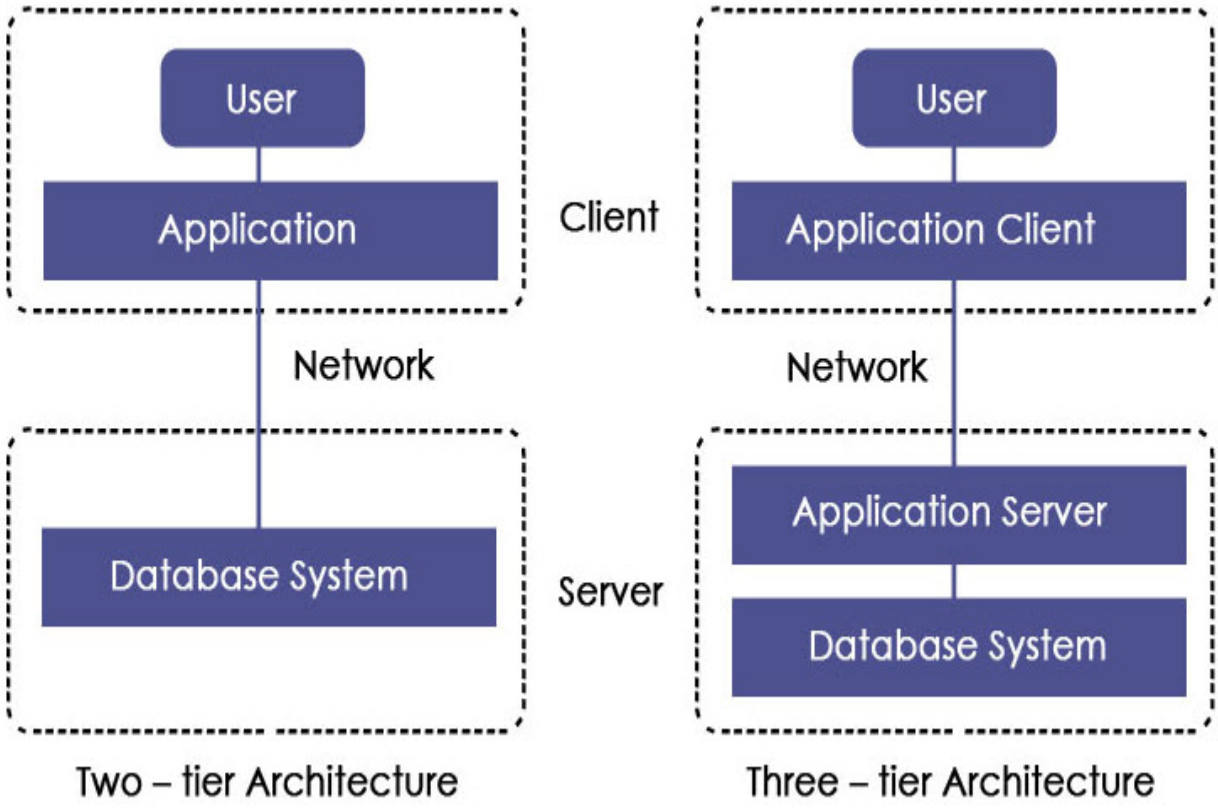
JDBC supports two types of processing models for accessing database i.e. two-tier and three-tier.

1. Two-tier Architecture

This architecture helps java program or application to directly communicate with the database. It needs a JDBC driver to communicate with a specific database. Query or request is sent by the user to the database and results are received back by the user. The database may be present on the same machine or any remote machine connected via a network. This approach is called client-server architecture or configuration.

2. Three-tier Architecture

In this, there is no direct communication. Requests are sent to the middle tier i.e. HTML browser sends a request to java application which is then further sent to the database. Database processes the request and sends the result back to the middle tier which then communicates with the user. It increases the performance and simplifies the application deployment.



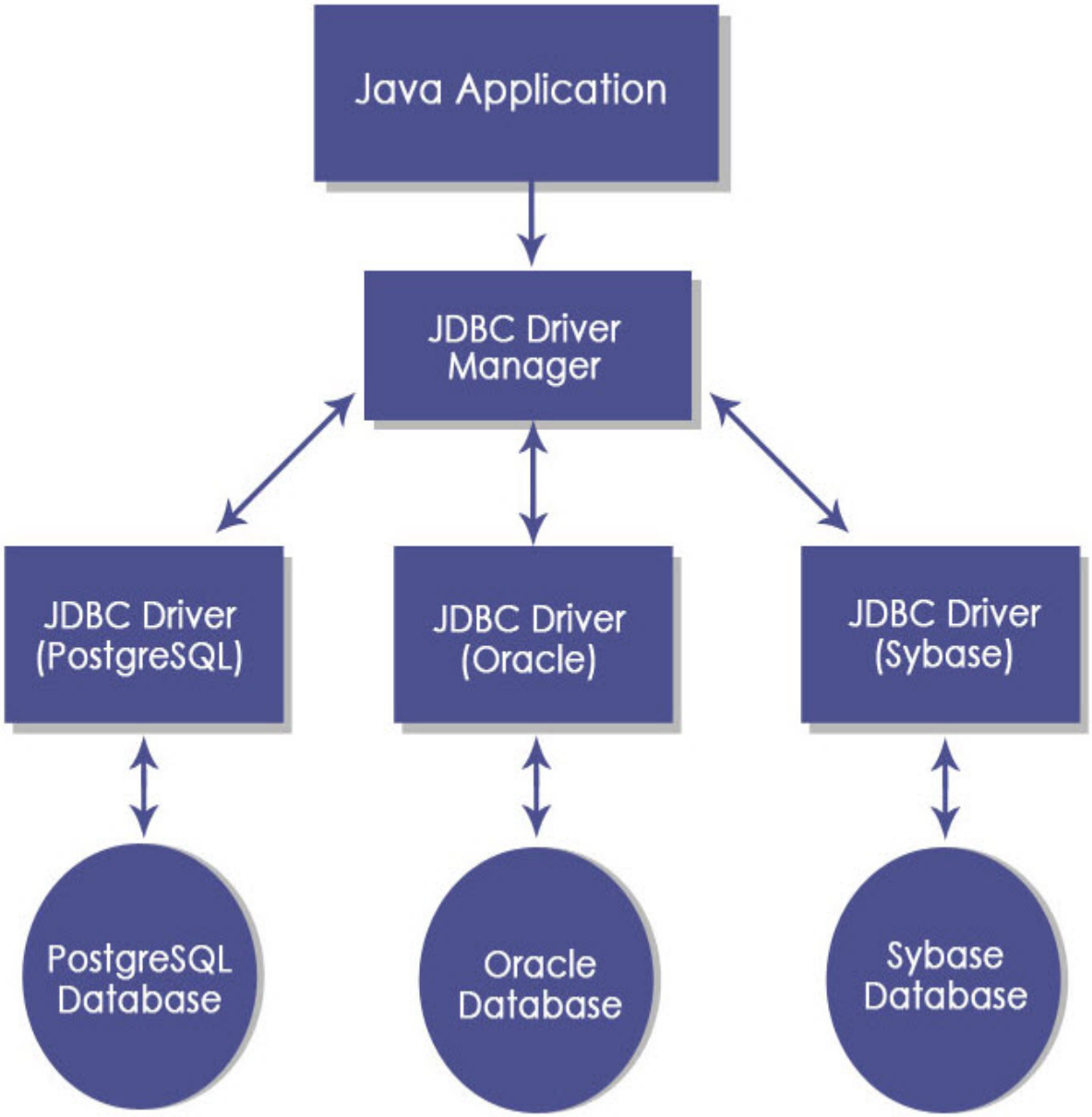
Components of JDBC Architecture

The components are explained below.

- **Driver Manager:** It is a class that contains a list of all drivers. When a connection request is received, it matches the request with the appropriate database driver using a protocol called communication sub-protocol. The driver that matches is used to establish a connection.
- **Driver:** It is an interface which controls the communication with the database server. DriverManager objects are used to perform communication.

- **Connection:** It is an interface which contains methods to contact a database.
- **Statement:** This interface creates an object to submit SQL queries or statements to the database.
- **ResultSet:** This contains the results retrieved after the execution of the SQL statements or queries.
- **SQLException:** Any errors that occur in database application are handled by this class.

Basic JDBC architectural diagram is shown below with the positioning of all components:



Interfaces

The java.sql package consists of many interfaces. Some popular interfaces are mentioned below:

- **Driver Interface:** This interface allows for multiple database drivers. DriverManager objects are created to communicate with the

database. These objects are created by DriverManager.registerDriver();

- **Connection Interface:** Connection interface establishes the connection i.e. session between java program and the database. It has many methods like rollback(), close() etc.
- **Statement Interface:** This interface provides methods for the execution of the SQL queries. It provides factory methods to get a ResultSet object. Some methods of statement interface are executeQuery(), executeUpdate() etc.
- **PreparedStatement Interface:** This interface helps when the SQL queries need to implement many times. It accepts input parameters during runtime.
- **CallableStatement Interface:** This interface is used when stored procedures are to be accessed. It also accepts parameters during run time.
- **ResultSet Interface:** This interface helps to store the result returned after the execution of the SQL queries.