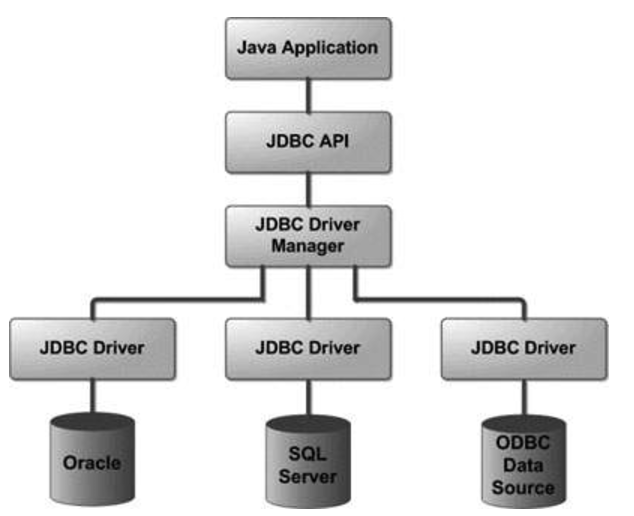
START

1. 配置数据库参数并new 一个数据库连接,执行sql语句，返回结果
2. Making a connection to a database.
3. Creating SQL or MySQL statements.
4. Executing SQL or MySQL queries in the database.
5. Viewing & Modifying the resulting records.
6. 简单列举几个应用场景
7. Java Applications
8. Java Servlets
9. Java ServerPages (JSPs)
10. Enterprise JavaBeans (EJBs).
11. Hibernate/mybatis

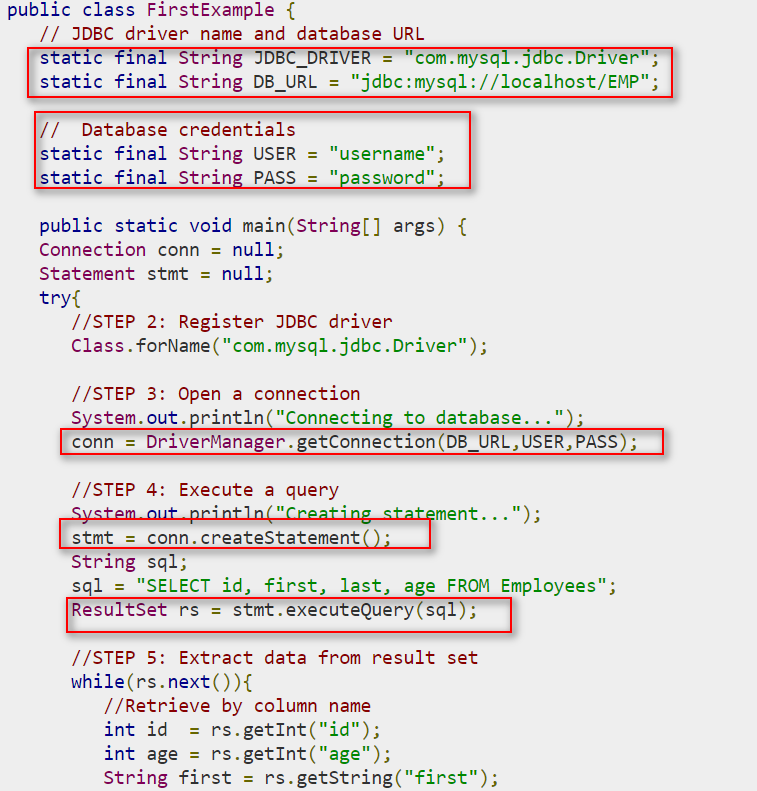
All of these different executables are able to use a JDBC driver to access a database, and take advantage of the stored data.

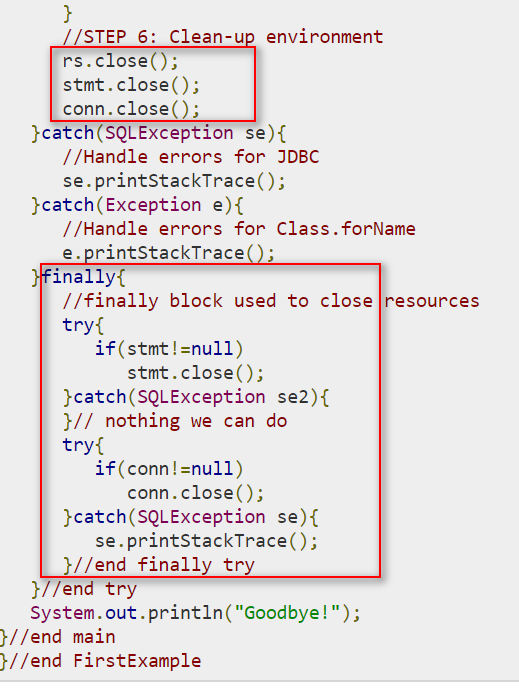
1. JDBC Architecture consists of two layers
2. **JDBC API:** This provides the application-to-JDBC Manager connection.
3. **JDBC Driver API:** This supports the JDBC Manager-to-Driver Connection.

图：



1. Common jdbc components
2. **DriverManager**
3. **Driver**
4. **Connection**
5. **Statement**
6. **ResultSet**
7. **SQLException**
8. Sample code (标红线划重点)





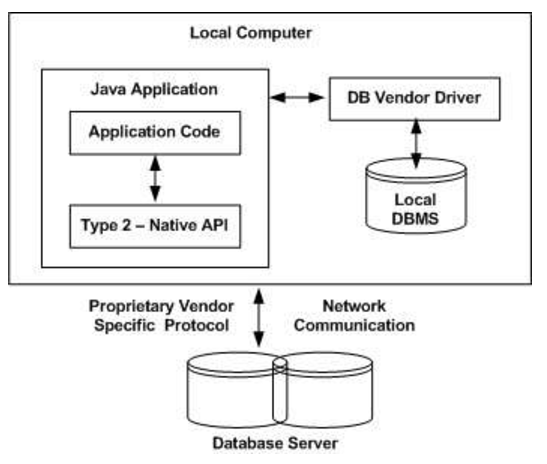
What is jdbc driver

----

using JDBC drivers enable you to open database connections and to interact with it by sending SQL or database commands then receiving results with Java.

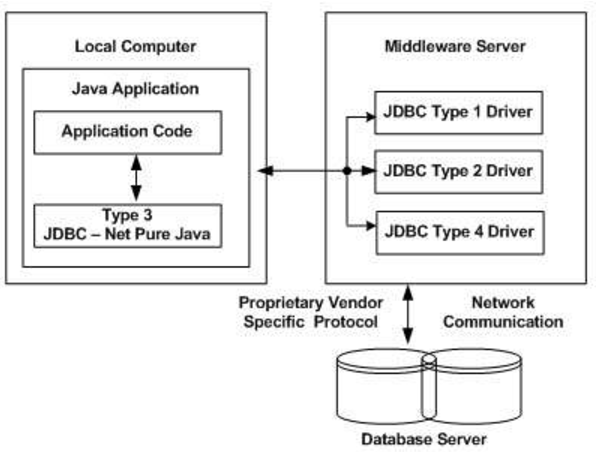
1. Jdbc native API

 JDBC API calls are converted into native C/C++ API calls, If we change the Database, we have to change the native API



1. Jdbc -Net pure java

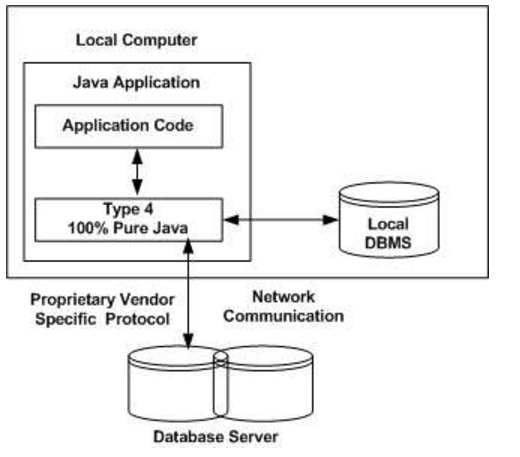
This kind of driver is extremely flexible, since it requires no code installed on the client and a single driver can actually provide access to multiple databases.

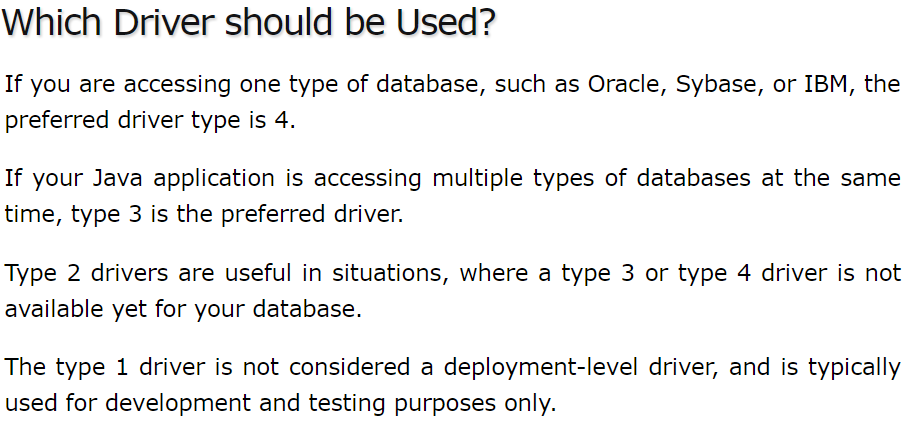


You can think of the application server as a JDBC "proxy," meaning that it makes calls for the client application. As a result, you need some knowledge of the application server's configuration in order to effectively use this driver type.

1. 100 % Pure java

a pure Java-based driver communicates directly with the vendor's database through socket connection. This is the highest performance driver available for the database and is usually provided by the vendor itself.





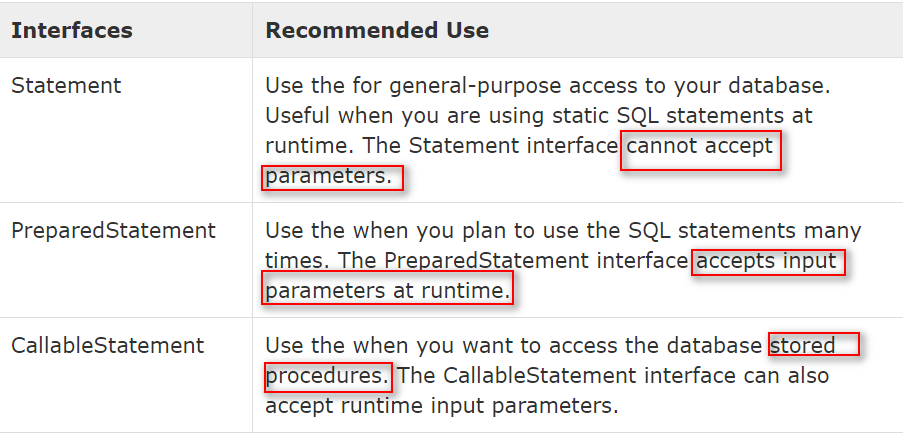
TIPS:

1. Using only a Database url. A second form of the DriverManager.getConnection( ) method requires only a database URL
2. Closing JDBC Connection

it is required explicitly to close all the connections to the database to end each database session

1. Statements , PreparedStatement and CallableStatement

The\_JDBC *Statement, CallableStatement,* and *PreparedStatement* interfaces define the methods and properties that enable you to send SQL or PL/SQL commands and receive data from your database.



TIPS:CALLABLEStatement

