

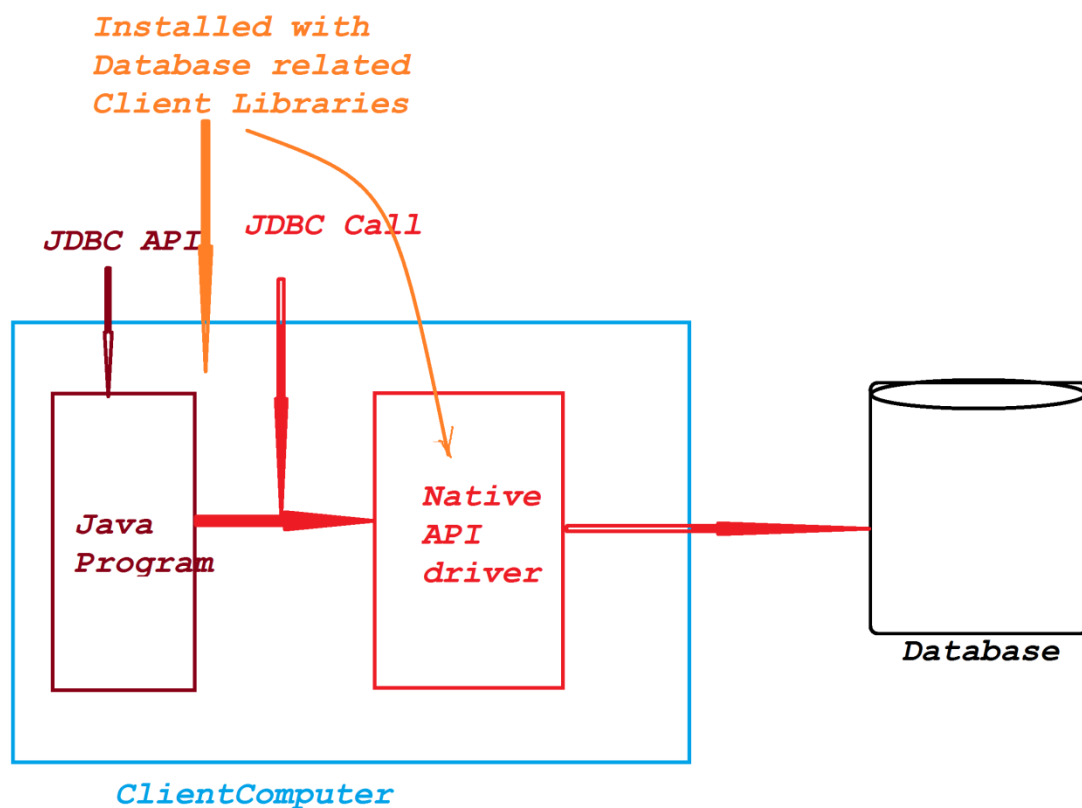
Dt : 23/11/2023

2. Native API driver (Type-2):

=> Native API driver will take the support of Database Client Libraries to establish connection to Database product.

=> when we use Native API driver then the Client Computer must be installed with Database related Client Libraries.

Diagram:



Advantage:

=>In Type-2 driver ODBC-driver is not involved and which saves the execution time and generate HighPerformance of an application.

DisAdvantage:

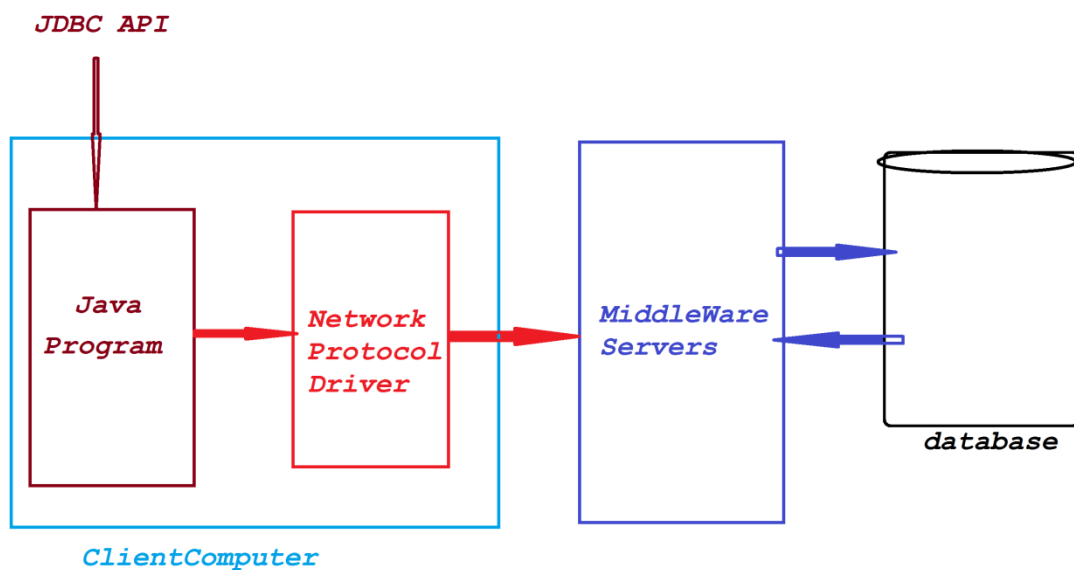
=>The Client Computer must be installed with Database related Client libraries and which makes the application Database dependent.

=====

3.Network Protocol driver(Type-3):

=>Network Protocol driver will take the support of Middleware Servers to Communicate database product.

Diagram:



Advantage:

=>Middleware Servers will hold database related code.

=>ODBC-driver and Database related Client libraries are not involved.

DisAdvantage:

=>In Network Protocol driver,the network related components and code involved in execution process and degrade the performance of an application.

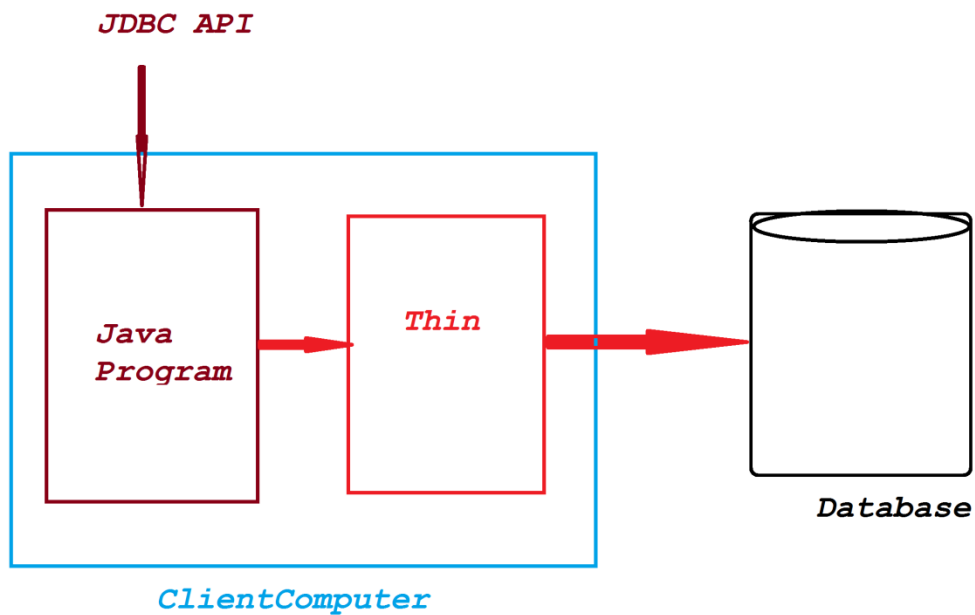
=====

4.Thin driver(Type-4)

=>Thin driver will take the support of Database Network protocol to Communicate with database product directly.

=>Thin driver is Pure Java Driver and Platform independent driver, and which generate HighPerformance of an application.

Diagram:



Advantage:

=>HighPerformance driver

**=>ODBC-driver,database related Client Libraries and MiddleWare Servers
are not involved.**

=====

***imp**

Summary of Objects created from CoreJava:

1.User defined Class Objects

2.String-Objects

3.WrapperClass-Objects

4.Array-Objects

5.Collection<E>-Objects

6.Map<K,V>-Objects

7.Enum<E>-Objects

Summary of Objects created in JDBC:

1.Connection Object

2.Statement Object

3.PreparedStatement Object

4.CallableStatement Object

5.ResultSet Object

(i)Scrollable ResultSet Object

(ii)NonScrollable ResultSet Object

6.RowSet Object

(i)JdbcRowSet Object

(ii)CachedRowSet Object

=>WebRowSet Object

(i)FilteredRowSet Object

(ii)JoinRowSet Object

7.DatabaseMetaData Object

8.ParameterMetaData Object

9.ResultSetMetaData Object

10.RowSetMetaData Object

=====

=

faq:

define Serialization process?

=>The process of converting Object state into binary Stream or Byte

Stream is known as Serialization process.

Coding Rule:

=>To perform Serialization process the class must be implemented from

'java.io.Serializable' interface.

=====

=>Based on Serialization process the Objects in Java are categorized

into two types:

(i)Serializable Objects

(ii)NonSerializable Objects

(i)Serializable Objects:

=>The Objects which are generated from implementation classes of

"java.io.Serializable" interface are Serializable Objects.

Ex:

All CoreJava Objects

(ii)NonSerializable Objects:

=>The Objects which are generated from NonImplementation classes of "java.io.Serializable" interface are NonSerializable Objects.

Ex:

All JDBC Objects

=====

faq:

why we have to perform Serialization process?

=>Through Serialization process we can make objects available in the form of Stream and can be moved on the Network from one location to another location.

Conclusion:

=>Serializable CoreJava Objects can be moved on the Network directly, but NonSerializable JDBC Objects cannot be moved on the Network directly

=====

===