Type	Name	Language	Adv	Disadv
1.	JDBC ODBC bridge driver	OBBC written in C language	Easy to use & Connect	→ Performance degrade JDBC Converted to ODBC → ODBC driver needs to be installed on client machine
2.	Native-API	Partially Java Driver (Converts JdBC method Calls to native Calls of dB API)	Performance upgraded than JDBC-ODBC	→ Native driver needs to be installed on client machine. vendor client library needs to be installed on C.M
3.	Network Protocol driver	Fully Java (uses middleware Converts JDBC Calls to vendor specific DB)	No client side library is required	→ Network support required for client middle tier needs data base Coding
4.	Thin driver	Fully Java driver Converts JDBC to vendor specific data base	Better Performance No software required for client or server	→ Drivers depends on DB.

steps to Connect Java application with Database using JDBC

1. Register driver → uses Class. forname()

   Public static void forname(String className) throws ClassNotFound Exception

2. Create the Connection object

   Connection con = DriverManager. getConnection("jdbc: oracle: thin: @localhost : 1521: xe", "system", "Password");

3. Create the statement object

   Statement st = Con.create Statement();

4. Execute the query

   ResultSet rs = st.executeQuery(" query");

5. close connection

   Con. close();

Statement interface

st. executeQuery () → used to Execute Select Query.

st. ExecuteUpdate () → used to Execute drop, create, insert, update, delete etc

st. Execute () → used to execute queries that may return multiple results.

Prepared Statement:

PreparedStatement st = Con. PrepareStatement ("insert into Emp values (?,?)");

st. set ~~Index~~ (index, value);
st. setString (index, value);
st. setfloat (index, value);
st. SetDouble (index, value);
st. executeQuery () → Select query.
st. executeUpdate() → Create, drop, insert, update, delete etc.

Callable statement:

→ used to called the Stored Procedures and functions.

* Call Stored Procedure using JDBC

CallableStatement st = Con. PrepareCall("{call insertR (?,?)}");

where insertR is a stored Procedure.

st. Execute();

Call Function using JDBC

CallableStatement st = Con. PrepareCall ("{ ? = Call Sum4 (?,
}");

ResultSet: (return type for ExecuteQuery)

ResultSet rs ;

rs.next() → move cursor from one row next to current row.

rs.Previous() → move cursor from one row Previous to current row

rs.first() → first row in result set object

ResultSetMetaData:

rs.getColumnCount() → total no of columns in Result set object

rs.getColumnName(i) → column name at index i

rs.getColumnTypeName(i) → returns Column type name for Specific index

rs.getTableName(i) → returns table name.