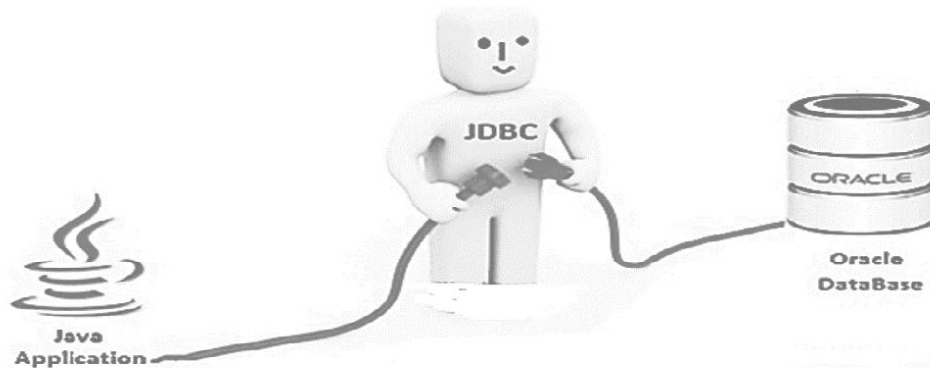


## Unit-5. JDBC

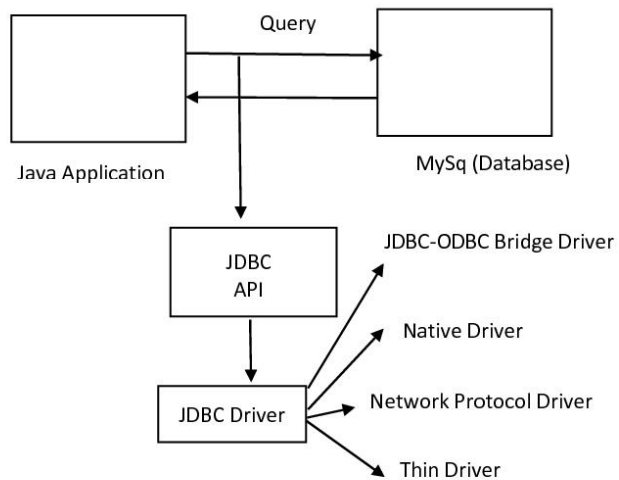
**JDBC:** It is a Java-based data access technology.

**JDBC Driver:**



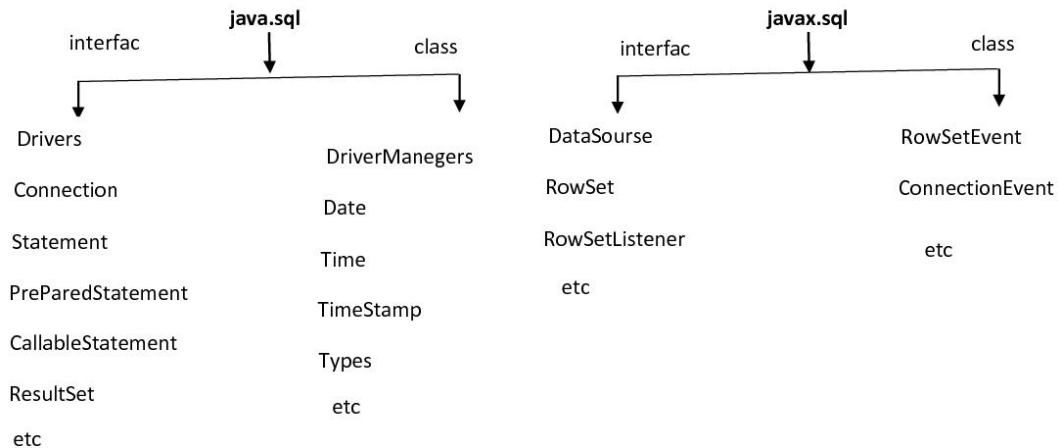
JDBC (Java Database Connectivity) driver is a software component that provides an implementation for interfaces and classes of JDBC API. The classes of JDBC are available in the packages `java.sql` and `javax.sql`.

By using JDBC, we can interact with different types of Relational Databases such as Oracle, MySQL, MS Access, etc.



Written by Nawaraj Prajapati (MCA From JNU)

Initially, Sun Microsystems had released JDBC in JDK 1.1 on Feb 19, 1997. After that, it has been part of the Java Platform.



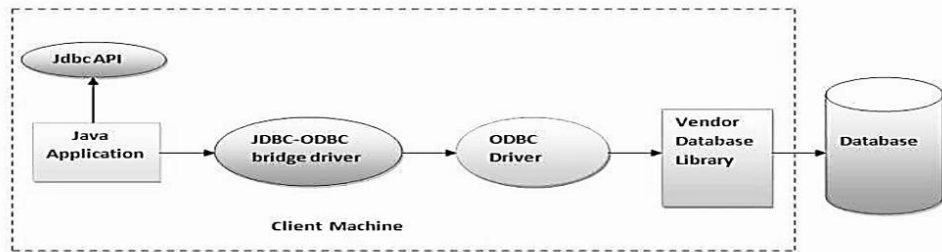
### Types of JDBC Drivers:

There are four type of JDBC drivers.

1. Type 1 : JDBC-ODBC bridge driver
2. Type 2 : Native-API driver (Partially java driver)
3. Type 3 : Network-Protocol driver (fully java driver)
4. Type 4 : Thin driver (fully java driver)

#### 1. Type 1 : JDBC-ODBC Bridge Drivers

Type-1 Driver is also called JDBC-ODBC Driver and Bridge Driver. JDBC-ODBC Driver is a driver provided by Sun Micro Systems as implementation of Driver interface. Sun Micro System has provided JDBC-ODBC driver with the inter dependent on the Microsoft's product ODBC Driver. ODBC Driver is a Open specification it will provide very good environment to interact with the database from JDBC-ODBC Driver. If we want to use JDBC-ODBC driver in our JDBC Application first we have to install the Microsoft product ODBC Driver native library. To interact with the database from Java Application if we use JDBC-ODBC Driver then we should require two type conversions so that JDBC-ODBC driver is slower Driver.



**Advantage:**

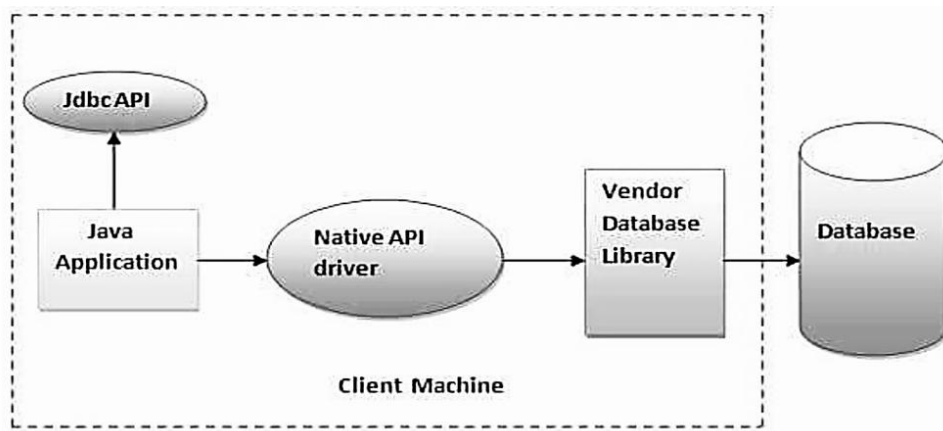
- It can be used with any database for which an ODBC driver is installed.

**Disadvantages:**

- Performance is not good as it converts JDBC method calls into ODBC function calls.
- ODBC driver needs to be installed on the client machine.
- Platform dependent.

**2. Type 2 : Native-API driver (Partially java driver)**

Type-2 driver is also called as Native API Driver, Type -2 Driver was implemented by using java implementations. Native API Driver uses database vendor provided native library. When compared to Type-1 Driver to Type-2 Driver is faster Driver because it should not required two times conversions to interact with database from java applications. If we want to use Type-2 driver in our JDBC Applications, then we have to install the database vendor provided native library.



**Advantage:**

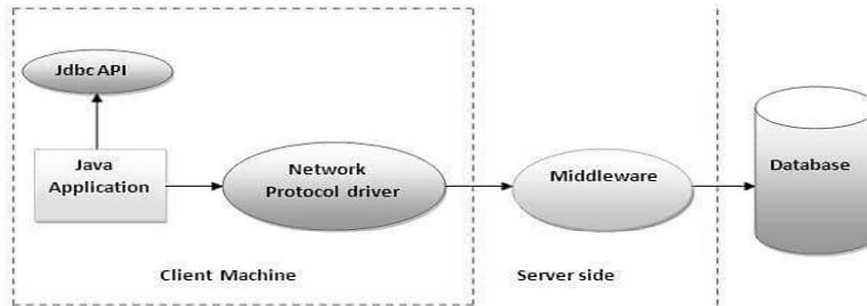
- It is faster than a JDBC-ODBC bridge driver.

**Disadvantages:**

- Platform dependent.
- The vendor client library needs to be installed on the client machine

### 3. Type 3 : Network-Protocol driver (fully java driver)

Type-3 driver is also called as Middle ware database server access driver and Network protocol driver. Type-3 driver should not require any native library installations, it should require middle ware component. Type-3 driver is fastest driver when compared to all the drivers.



#### Advantages:

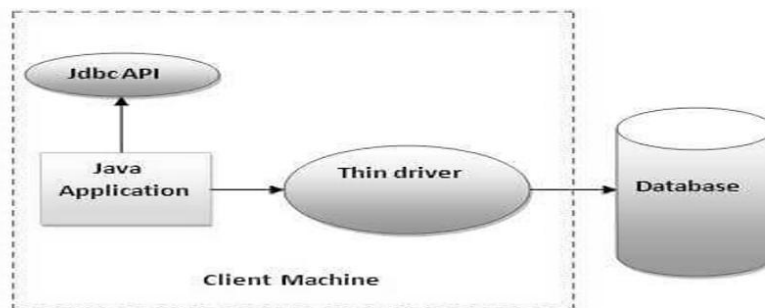
- Platform independent.
- Faster from Type1 and Type2 drivers.
- It follows a three tier communication approach.
- Multiple types of databases can be accessed at the same time.

#### Disadvantage:

Slow performance because of a number of network calls to an intermediate middleware server.

### 4. Type IV : Thin driver (fully java driver):

Type -4 driver is also called as Thin Driver and pure java driver because Type-4 driver was implemented completely by using Java implementations. Type 4 Driver portability is very good as compared to all the remaining drivers. Type 4 driver should not require any native libraries dependencies and it should require on time conversion to interact with the database from java applications. Type 4 driver is the cheap driver among all.



Written by Nawaraj Prajapati (MCA From JNU)

**Advantages:**

- Better performance than all other drivers.
- No software is required at client side or server side.

**Disadvantages:**

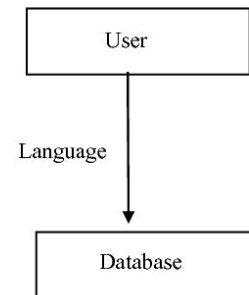
- It is database dependent.
- Multiple types of databases can't be accessed at the same time.

**Structured Query Language (SQL)**

SQL is Structured Query Language, which is a computer language for storing, manipulating and retrieving data stored in a relational database.

SQL is the standard language for Relational Database System. All the Relational Database Management Systems (RDMS) like MySQL, MS Access, Oracle, Postgres and SQL Server use SQL as their standard database language.

SQL was developed by IBM in the early 1970s and became commercially available in 1979. It is globally accepted as the standard relational database management system (RDBMS).



There are five types of SQL commands:

1. DDL: Data Definition Language
2. DQL: Data Query Language
3. DML: Data Manipulation Language
4. DCL: Data Control Language
5. TCL: Transaction Control Language

**1. DDL: Data Definition Language (CREATE, DROP, TRUNCATE, ALTER)**

**CREATE:** It is used to create a new table.

Syntax of CREATE:

```
CREATE TABLE TableName (  
    Column1 datatype,  
    Column2 datatype,  
    Column3 datatype,  
    ....  
    ColumnN datatype  
);
```

Example of CREATE:

```
CREATE TABLE Emp_Information
(
    Emp_ID int,
    Emp_Name varchar(255),
    Address varchar(255),
    City varchar(255),
    Country varchar(255)
);
```

**DROP:** It is used to delete an entire table.

Syntax of DROP:

```
DROP TABLE TableName;
```

Examples of DROP:

```
DROP Table Emp_Information;
```

**TRUNCATE:** It is used to remove all records from a table.

Syntax of TRUNCATE:

```
TRUNCATE TABLE TableName;
```

Example of TRUNCATE:

```
TRUNCATE Table Employee_Info;
```

**ALTER:** It is used to modify an existing database.

Syntax of ALTER:

```
ALTER TABLE TableName
ADD ColumnName Datatype;
```

Example of ALTER:

**ADD Column BloodGroup:**

```
ALTER TABLE Employee_Info
ADD BloodGroup varchar(255);
```

**DROP Column BloodGroup:**

```
ALTER TABLE Employee_Info
DROP COLUMN BloodGroup ;
```

Written by Nawaraj Prajapati (MCA From JNU)

## 2. DQL: Data Query Language

**SELECT:** It is used to retrieve data from the database.

Syntax of SELECT:

```
SELECT Column1, Column2, ...ColumnN  
FROM TableName;
```

Example of SELECT:

```
SELECT * FROM Emp_Information;
```

## 3. DML: Data Manipulation Language(INsert, UPDate, DELETE)

**INSERT:** It is used to insert data into a table.

Syntax of INSERT:

```
INSERT INTO TableName (Column1, Column2, Column3, ...,ColumnN)  
VALUES (value1, value2, value3, ...);
```

Example of INSERT:

```
INSERT INTO Emp_Information  
VALUES ('07', 'Abhiraj,', 'parsi tol', 'Kalaiya', Nepal);
```

**UPDATE:** It is used to update existing data within a table.

Syntax of UPDATE:

```
UPDATE TableName  
SET Column1 = Value1, Column2 = Value2, ...  
WHERE Condition;
```

Example of UPDATE:

```
UPDATE Emp_Information  
SET Emp_Name = 'Aisha', City= Nepal  
WHERE Emp_ID = 2;
```

**DELETE:** It is used to delete records from a database table.

Syntax of DELETE:

```
DELETE FROM table_name;
```

Example of DELETE:

```
DELETE FROM Emp_Information;
```

Written by Nawaraj Prajapati (MCA From JNU)

**4. DCL: Data Control Language (GRANT, REVOKE)**

**GRANT:** It is used to give user to access privileges to the database.

**REVOKE:** It is used to take back permissions from the user.

**5. TCL: Transaction Control Language**

**COMMIT:** It is used to save all the transactions to the database.

**ROLLBACK:** It is used to undo transaction that have not already been saved to the database.

**SAVE POINT:** It is used to roll the transaction back to a certain point without rolling back the entire transaction.

**Configuring JDBC:**

Configuring JDBC correctly is an important step in ensuring that your application is able to access the database correctly. The first step is to download and install the JDBC driver for the database you are using. Once the driver is installed, you will need to configure the connection parameters, such as the database URL, username, and password. The next step is to create a JDBC connection. This is done by creating a Connection object, which is used to establish a connection to the database. Once the connection is established, you can use the Connection object to execute SQL statements and retrieve data from the database.

**Primary key:**

Primary key will not allow any duplicate values, null values and its data will be arranged in ascending order due to clustered index. Only one primary key is allowed per table.

**Unique key:**

Unique key is a key which will not accept any duplicate values, but it will accept one null value; we can place any number of unique keys per table.

**Foreign key:**

Foreign key is used to refer primary key defined field in another table and it allows duplication.

**Null:**

Which all null values

**Not null:**

Which doesn't allow Null values.

**Attributes:**

It represents some property of an entity that describes an entity such as employee's name or salary

**Entity:**

It represents a real world object such as an employee or a project.



### Difference between Oracle and mysql:

1. Oracle is private company where as mysql is open source and only few features were exist compare to oracle.
2. Oracle support sub queries. Mysql doesn't support sub queries.
3. Oracle can create view table. Mysql can't create view table.
4. Mysql support auto increment function. Oracle doesn't support auto increment function.
5. Mysql is generally used by web application whereas oracle is used by companies.

### JDBC Statements:

There are two types of statements.

1. Statement
2. PreparedStatement

#### 1. Statement:

Statement is an interface, you can create an object of this interface using the createStatement() method of the Connection interface. The Statement interface can't accept any parameters. Statement is compile multiple times so its preformation is slow.

#### Syntax of Statement:

```
Statement stmt = null;
try {
    stmt = conn.createStatement();
    ...
}
catch (SQLException e) {
    ...
}
finally {
    stmt.close();
}
```

The important method of Statement interface

#### ○ public boolean execute(String url) :

This method is used for all type of SQL statement (eg.Select,Insert,Update etc.).This method returns a boolean value.if you don't know which method is used (executeQuery() or executeUpdate()) then you should go for execute() method.

#### ○ public ResultSet executeQuery(String url):

This method is used for the Select A statement which retrieves some data from the database. This method returns a ResultSet object.

#### ○ public int executeUpdate(String url):

If you want to modify in your database, then you should go for executeUpdate() method. This method returns an int value which indicates a number of rows affected.

## 2. PreparedStatement:

PreparedStatement is SQL statement that is pre-compiled by the database. PreparedStatement improve the performance of SQL commands that are executed multiple times. Once compiled, PreparedStatement can be customized prior to each execution by altering pre-defined SQL parameters.

### Syntax of PreparedStatement:

```
PreparedStatement pstmt = null;
try {
    String SQL = "Update Employees SET age = ? WHERE id = ?";
    pstmt = conn.prepareStatement(SQL);
    ...
}
catch (SQLException e) {
    ...
}
finally {
    ...
}
```

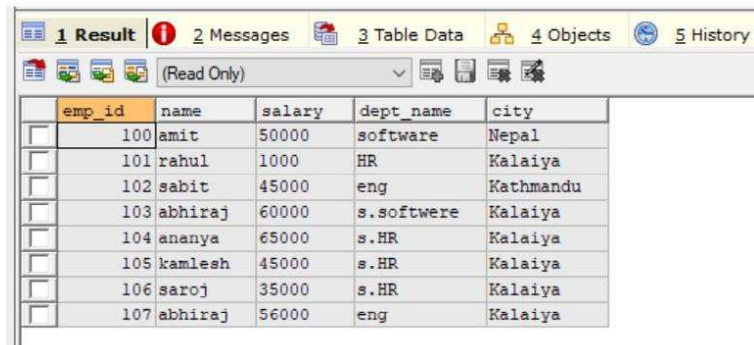
### Query order of execution:

1. FROM
2. JOIN
3. WHERE
4. GROUP BY
5. HAVING
6. SELECT
7. DISTINCT
8. ORDER BY
9. LIMIT

1. FROM/JOIN: The FROM and/or JOIN clauses are first executed to identify the data of interest.
2. Where: The clause is used to filter rows based on specific conditions.
3. GROUP BY: The GROUP BY clause is used to group data based on the values ?? of one or more columns.
4. HAVING: The HAVING clause filters grouped data based on the specified conditions.
5. SELECT: The SELECT clause defines the columns to be returned in the result set.
6. DISTINCT: The DISTINCT clause removes duplicates from final output result.
7. ORDER BY: The ORDER BY clause is used to sort the values ?? retrieved in ascending or descending order.

Written by Nawaraj Prajapati (MCA From JNU)

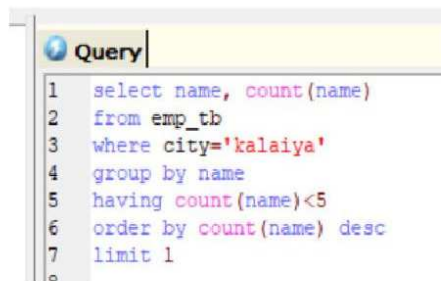
8. LIMIT/OFFSET clause: Finally, LIMIT and/or OFFSET clauses are used to keep or discard a specified number of rows.



The screenshot shows a database query result window with a toolbar at the top containing icons for Result, Messages, Table Data, Objects, and History. Below the toolbar is a dropdown menu set to "(Read Only)". The main area displays a table with the following data:

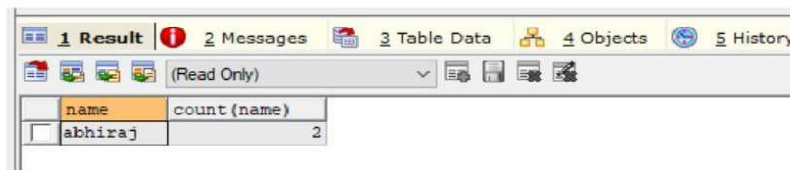
emp_id	name	salary	dept_name	city
100	amit	50000	software	Nepal
101	rahul	1000	HR	Kalaiya
102	sabit	45000	eng	Kathmandu
103	abhiraj	60000	s.software	Kalaiya
104	ananya	65000	s.HR	Kalaiya
105	kamlesh	45000	s.HR	Kalaiya
106	saroj	35000	s.HR	Kalaiya
107	abhiraj	56000	eng	Kalaiya

select name, count(name) (5)  
from emp\_tb (1)  
where city='kalaiya' (2)  
group by name (3)  
having count(name)<5 (4)  
order by count(name) desc (6)  
limit 1 (7)



The screenshot shows a query editor window with a toolbar at the top containing a "Query" button. Below the toolbar is a text area containing the following SQL query:

```
1 select name, count(name)
2 from emp_tb
3 where city='kalaiya'
4 group by name
5 having count(name)<5
6 order by count(name) desc
7 limit 1
8
```



The screenshot shows a database query result window with a toolbar at the top containing icons for Result, Messages, Table Data, Objects, and History. Below the toolbar is a dropdown menu set to "(Read Only)". The main area displays a table with the following data:

name	count(name)
abhiraj	2

### Steps to connect to the database in java:

#### 1. Register driver class:

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

For example:

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

#### 2. Creating connection:

The getConnection() method of DriverManager class is used to connection with the database.

For example:

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost/db_servletpage", "root", "admin");
```

#### 3. Creating statement:

The createStatement() method of Connection interface is used to create statement. The object of statement is responsible of execute queries with the database.

For example:

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

#### 4. Executing queries:

Once we have created a Statement object, we can use it to execute a sql statement.

For example:

```
stm.execute("insert into tb_servletpage values('"+name+"','"+address+"')");
```

## RowSets:

RowSet is child interface of ResultSet. We can use RowSet to handle a group of records in more effective way than ResultSet. RowSet interface present in javax.sql package.

### Types of RowSets:

There are 2 types of RowSets

1. Connected RowSets
2. Disconnected RowSets

#### 1. Connected RowSets

Connected RowSets are just like ResultSets. To access RowSet data compulsory connection should be available to database. We cannot serialize Connected RowSets.

For example: JdbcRowSet

**JdbcRowSet:** A JdbcRowSet is a thin wrapper around a ResultSet that makes it possible to use the result set as a JavaBeans component.

#### 2. Disconnected RowSets

Without having Connection to the database we can access RowSet data. We can serialize Disconnected RowSets.

For example: CachedRowSet, WebRowSet, FilteredRowSet, JoinRowSet

**CachedRowSet:** A CachedRowSet stores data in memory so you can work on the data without keeping the connection open all the time. CachedRowSet is the super interface of the ones below.

**WebRowSet:** A WebRowSet can read and write data in XML format, making it possible to transfer the data through tiers in a web application.

**FilteredRowSet:** A FilteredRowSet allows filtering data without having to write SQL SELECT queries.

**JoinRowSet:** A JoinRowSet combines data from different RowSet objects, which is equivalent to SQL JOIN queries.

### Differences between ResultSet and RowSet

ResultSet		RowSet	
1	ResultSet present in java.sql Package.	1	RowSet present in javax.sql Package.
2	It cannot be serialized.	2	A RowSet object can be serialized.
3	By default, ResultSet object is not scrollable or, updatable.	3	By default, RowSet object is scrollable and updatable.
4	ResultSet object cannot be passed over network.	4	You can pass a RowSet object over the network.