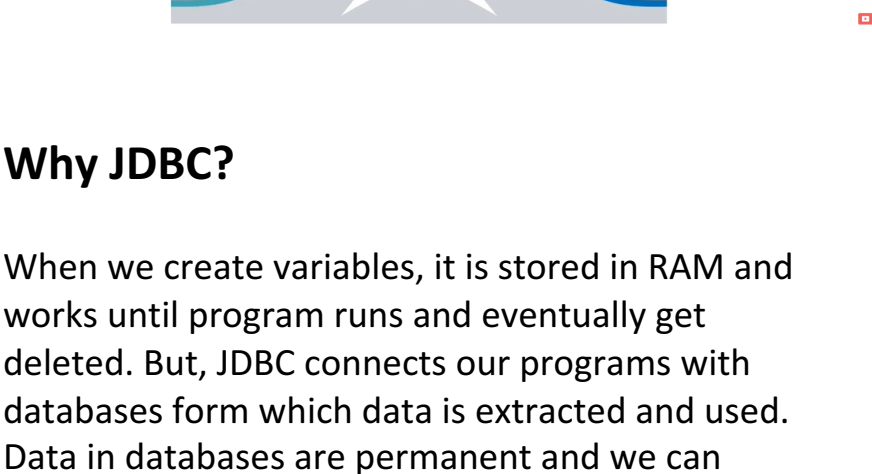


JDBC - Java Database Connectivity

It is standard API for Java applications to interact with different sets of databases.



Why JDBC?

When we create variables, it is stored in RAM and works until program runs and eventually get deleted. But, JDBC connects our programs with databases from which data is extracted and used. Data in databases are permanent and we can perform CRUD operations using JDBC on databases.

How JDBC works?

How JDBC Work(Architecture of JDBC)

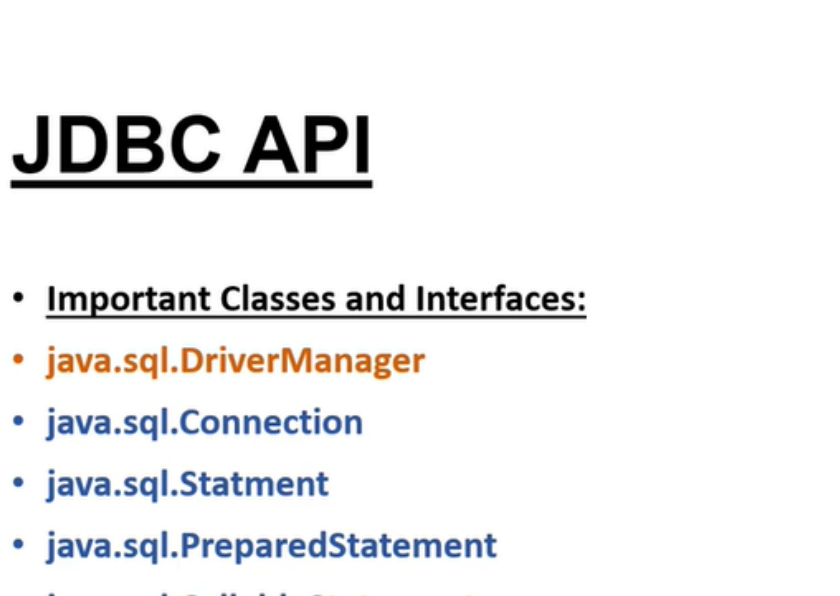


Figure 1 - JDBC Architecture

We have all our JDBC methods in JDBC API. We can write queries in general terms so that all database drivers can understand. And the queries passes to database drivers. And then driver passes on the query to database.

JDBC API

Important Classes and Interfaces:

- **java.sql.DriverManager**
- **java.sql.Connection**
- **java.sql.Statement**
- **java.sql.PreparedStatement**
- **java.sql.CallableStatement**
- **java.sql.ResultSet**
- **java.sql.ResultSetMetaData**
- **java.sql.DatabaseMetaData**
- **java.sql.SQLException**

Software Requirements

Sublime Text
Eclipse
Java 16
JDBC Driver: mysql-connector.jar : 8.0
Database : MySQL

Steps to Connect with Database

- 1. Load the driver**
We can only use the driver after loading it on our machine.

1st Method
Class.forName("Driver_name")
e.g. Class.forName("com.mysql.jdbc.Driver")

2nd Method
DriverManager.registerDriver(new com.mysql.cj.jdbc.Driver())
- 2. Create a connection**
Connection gets established between program and database

import java.sql.*;
DriverManager.getConnection("url","username","password");
e.g. ("jdbc:mysql://localhost:3306/jdbctest","root","password")
- 3. Create a query**
We will fire SQL queries after establishing connection. We will need one of three things below:
 - a. Statement - If we need simple query.
 - b. PreparedStatement - If we have question marks/complex query.
 - c. CallableStatement - If we need functions/procedures from database.
 - a. Statement Example
String q = " select * from students" ;
Statement stmt = conn.createStatement();
 - stmt.executeQuery() - when we expect only data
e.g. ResultSet set = stmt.executeQuery();
It will pull all data in set.
 - stmt.executeUpdate() - when we expect update and not data
▪ stmt.execute();
- 4. Process the data**
We will process our data with help of set. We have methods for it.

set.next() - If there is next row available?
set.getInt()
set.getString()
e.g.
while(set.next()){
 int id = set.getInt(ColumnName/"ColumnName");
 String name = set.getString(ColumnName/"ColumnName");
 System.out.println(id + " " + name);
}

5. Close the connection
conn.close();

Setup Connection using Java App

```
1 import java.sql.*;  
2 class FirstJDBC{  
3     public static void main(String[] args) {  
4         try{  
5             //Load the Driver  
6             Class.forName("com.mysql.cj.jdbc.Driver");  
7  
8             //Create connection  
9             String url = "jdbc:mysql://localhost:3306/jdbctest";  
10            String user = "root";  
11            String pass = "password";  
12            Connection conn = DriverManager.getConnection(url,user,  
13                pass);  
14            if(conn.isClosed()){  
15                System.out.println("...Closed");  
16            }else{  
17                System.out.println("Created...");  
18            }  
19        }catch(Exception e){  
20            e.printStackTrace();  
21        }  
22    }  
23 }  
24 }
```

```
PS C:\Users\USER\OneDrive\Desktop\jdbc> javac .\FirstJDBC.java  
PS C:\Users\USER\OneDrive\Desktop\jdbc> java FirstJDBC  
Created...  
PS C:\Users\USER\OneDrive\Desktop\jdbc> |
```

Creating Table using Java Program

```
1 import java.sql.*;  
2 class InsertJDBC{  
3     public static void main(String[] args) {  
4         try{  
5             //Load the Driver  
6             Class.forName("com.mysql.cj.jdbc.Driver");  
7  
8             //Create connection  
9             String url = "jdbc:mysql://localhost:3306/  
10            jdbctest";  
11            String user = "root";  
12            String pass = "password";  
13            Connection conn = DriverManager.getConnection(  
14                url,user,pass);  
15  
16            //Create a query  
17            String q = "create table table1(tid int(20)  
18                primary key auto_increment, tname varchar(  
19                200) not null, tCity varchar(400))";  
20  
21            //Create a Statement  
22            Statement stmt = conn.createStatement();  
23            stmt.executeUpdate(q);  
24            System.out.println("Table Created");  
25  
26            //Close Connection  
27            conn.close();  
28        }catch(Exception e){  
29            e.printStackTrace();  
30        }  
31    }  
32 }
```

```
PS C:\Users\USER\OneDrive\Desktop\jdbc> javac .\InsertJDBC.java  
PS C:\Users\USER\OneDrive\Desktop\jdbc> java .\InsertJDBC.java  
Table Created  
PS C:\Users\USER\OneDrive\Desktop\jdbc>
```

```
mysql> use jdbctest;  
Database changed  
mysql> show tables;  
+-----+  
| Tables_in_jdbctest |  
+-----+  
| table1              |  
+-----+  
1 row in set (0.26 sec)  
  
mysql> desc table1  
->  
+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+  
| tid   | int           | NO   | PRI | NULL    | auto_increment |  
| tname | varchar(200)  | NO   |     | NULL    |               |  
| tCity | varchar(400)  | YES  |     | NULL    |               |  
+-----+  
3 rows in set (0.09 sec)  
  
mysql>
```

Inserting Data to Table with dynamic Input

```
1 //Insert data into table using PreparedStatement  
2  
3 import java.sql.*;  
4 class InsertDataJDBC{  
5     public static void main(String[] args) {  
6         try{  
7             //Load Driver  
8             Class.forName("com.mysql.cj.jdbc.Driver");  
9  
10            //Create Connection  
11            String url = "jdbc:mysql://localhost:3306/jdbctest";  
12            String user = "root";  
13            String pass = "password";  
14            Connection conn = DriverManager.getConnection(url,user,  
15                pass);  
16  
17            //Create Query  
18            String q = "insert into table1(tname,tCity) values(?,?)";  
19  
20            //Get Prepared Statement Object  
21            PreparedStatement pstmt = conn.prepareStatement(q);  
22  
23            //Set Values to Query  
24            pstmt.setString(1,"Naruto");  
25            pstmt.setString(2,"Hidden Leaf");  
26            pstmt.executeUpdate();  
27            System.out.println("Inserted");  
28        }catch(Exception e){  
29            e.printStackTrace();  
30        }  
31    }  
32 }
```

```
PS C:\Users\USER\OneDrive\Desktop\jdbc> javac .\InsertingDynamically.java  
PS C:\Users\USER\OneDrive\Desktop\jdbc> java .\InsertingDynamically.java  
Enter Name :  
Shikamaru  
Enter City :  
Leaf  
Inserted  
PS C:\Users\USER\OneDrive\Desktop\jdbc> |
```

```
mysql> select * from table1;  
+-----+  
| tid | tname      | tCity      |  
+-----+  
| 1   | Naruto    | Hidden Leaf |  
+-----+  
1 row in set (0.00 sec)  
  
mysql>  
  
mysql> select * from table1;  
+-----+  
| tid | tname      | tCity      |  
+-----+  
| 1   | Naruto    | Hidden Leaf |  
| 2   | Kakashi   | Konoha     |  
| 3   | Sasuke    | Hidden Leaf |  
| 4   | Gaara     | Hidden Sand |  
+-----+  
4 rows in set (0.00 sec)  
  
mysql>
```

Inserting Image to DB using Java

```
13  
14 //Create a query  
15 // String q = "create table table1(tid int(20)  
16 primary key auto_increment, tname varchar(200)  
17 not null, tCity varchar(400))";  
18  
19 String q = "create table images(id int primary  
20 key auto_increment, pic blob)";  
21  
22 //Create a Statement  
23 Statement stmt = conn.createStatement();  
24 stmt.executeUpdate(q);  
25 System.out.println("Table Created");  
26  
27 //Close Connection  
28 conn.close();  
29  
30 //Create Query  
31 String q = "insert into images(pic) values(?)";  
32  
33 //Get Prepared Statement Object  
34 PreparedStatement pstmt = conn.prepareStatement(  
35     q);  
36  
37 //Set Values to Query  
38 FileInputStream fis = new FileInputStream("strike.jpg");  
39 pstmt.setBinaryStream(1,fis,fis.available());  
40 pstmt.executeUpdate();  
41 System.out.println("Image Inserted");  
42  
43 //Close Connection  
44 conn.close();  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

```
mysql> desc images;  
+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+  
| id    | int           | NO   | PRI | NULL    | auto_increment |  
| pic   | longblob      | YES  |     | NULL    |               |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql>
```

Creating table and inserting Image to DB using Java

```
1 import java.sql.*;  
2 import java.io.*;  
3 class InsertImage{  
4     public static void main(String[] args) {  
5         try{  
6             Class.forName("com.mysql.cj.jdbc.Driver");  
7  
8             Connection conn = DriverManager.getConnection("jdbc:mysql://localhost:3306/jdbctest","root","password");  
9  
10            String q = "create table images(id int primary key  
11                auto_increment, pic longblob)";  
12            Statement stmt = conn.createStatement();  
13            stmt.executeUpdate(q);  
14            System.out.println("Table Created");  
15  
16            String q2 = "insert into images(pic) values(?)";  
17            PreparedStatement pstmt = conn.prepareStatement(q2);  
18            FileInputStream fis = new FileInputStream("strike.jpg");  
19            pstmt.setBinaryStream(1,fis,fis.available());  
20            pstmt.executeUpdate();  
21            System.out.println("Image Inserted");  
22            conn.close();  
23        }catch(Exception e){  
24            e.printStackTrace();  
25        }  
26    }  
27 }  
28  
29  
30  
31  
32  
33  
34  
35  
36  
37  
38  
39  
40  
41  
42  
43  
44  
45  
46  
47  
48  
49  
50  
51  
52  
53  
54  
55  
56  
57  
58  
59  
60  
61  
62  
63  
64  
65  
66  
67  
68  
69  
70  
71  
72  
73  
74  
75  
76  
77  
78  
79  
80  
81  
82  
83  
84  
85  
86  
87  
88  
89  
90  
91  
92  
93  
94  
95  
96  
97  
98  
99  
100
```

```
mysql> desc images;  
+-----+  
| Field | Type          | Null | Key | Default | Extra |  
+-----+  
| id    | int           | NO   | PRI | NULL    | auto_increment |  
| pic   | longblob      | YES  |     | NULL    |               |  
+-----+  
2 rows in set (0.00 sec)  
  
mysql>
```