

Database and Java Application Development

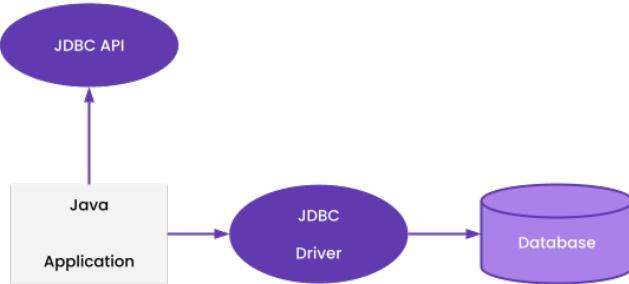
part 1

Assignment Solutions



1. What is JDBC?

Ans: JDBC is a Java API that is used to connect and execute the query to the database. JDBC API uses JDBC drivers to connect to the database. JDBC API can be used to access tabular data stored into any relational database.



2. What are the steps to connect to the database in java? Explain with an Example

Ans: The following steps are used in database connectivity.

Registering the driver class:

The `forName()` method of the `Class` class is used to register the driver class. This method is used to load the driver class dynamically. Consider the following example to register `OracleDriver` class.

```
Class.forName("oracle.jdbc.driver.OracleDriver");
```

Creating connection:

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

The syntax of the `getConnection()` method is given below.

- 1) `public static Connection getConnection(String url) throws SQLException`
- 2) `public static Connection getConnection(String url, String name, String password) throws SQLException`

Example to establish the connection with the Oracle database.

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

Creating the statement:

The `createStatement()` method of `Connection` interface is used to create the Statement. The object of the Statement is responsible for executing queries with the database.

```
public Statement createStatement() throws SQLException
```

Example to create the statement object

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

Executing the queries:

The `executeQuery()` method of Statement interface is used to execute queries to the database. This method returns the object of `ResultSet` that can be used to get all the records of a table.

Syntax of `executeQuery()` method is given below.

```
public ResultSet executeQuery(String sql) throws SQLException
```

Example to execute the query

```
ResultSet rs=stmt.executeQuery("select * from emp");
```

```
while(rs.next()) {
```

```
System.out.println(rs.getInt(1)+" "+rs.getString(2));
```

```
}
```

However, to perform the insert and update operations in the database, `executeUpdate()` method is used which returns the boolean value to indicate the successful completion of the operation.

Closing connection:

By closing connection, object statement and ResultSet will be closed automatically. The close() method of Connection interface is used to close the connection.

Syntax of close() method is given below.

```
public void close()throws SQLException
```

Example to close the connection.

```
con.close();
```

3. What is Connection Pooling in Database?

Ans: The Connection Pooling in the database is the process of storing the connection in Cache which can be reused later. It provides faster connections and easier to troubleshoot the issues.

4. What is the difference between a Generic Servlet and HTTP Servlet?

Generic Servlet	HTTP Servlet
Protocol Independent	Protocol Specific
Belongs to javax.servlet package	Belongs to javax.servlet.http package
supports only service() method	supports doGet(), doPost(), doHead() methods

5. What is the use of RequestDispatcher Interface?

Ans: The RequestDispatcher interface defines the object that receives the request from the client and dispatches it to the resources such as a servlet, JSP, HTML file. The RequestDispatcher interface has the following two methods:

1. `public void forward(ServletRequest request, ServletResponse response)`: Forwards request from one servlet to another resource like servlet, JSP, HTML etc.
2. `public void include(ServletRequest request, ServletResponse response)`: Includes the content of the resource such as a servlet, JSP, and HTML in the response.

6. How to build a Application using PreparedStatement in JDBC

refer: JDBCPreparedStatementApp [Link](#)