Complete JDBC Notes

1. Introduction to JDBC

JDBC (Java Database Connectivity) is an API in Java to connect and interact with relational databases like MySQL, Oracle, PostgreSQL, etc.

Features:

- Platform-independent
- Works with any database supporting JDBC driver
- Supports transactions, batch processing, and stored procedures

Core Interfaces:

- Driver
- Connection
- Statement / PreparedStatement / CallableStatement
- ResultSet

2. Setting Up a Java Project with JDBC (Non-Maven)

Steps to Set Up in Eclipse:

1. Create a new Java project.
2. Download the JDBC driver (e.g., mysql-connector-java-8.0.xx.jar).
3. Right-click project > Build Path > Configure Build Path > Libraries > Add External JARs.
4. Select and add the MySQL JAR file.
5. Use the JDBC driver class: com.mysql.cj.jdbc.Driver

Sample Code to Test Connection:

```
Class.forName("com.mysql.cj.jdbc.Driver");
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "root", "root123");
System.out.println("Connected!");
```

3. JDBC Architecture

Complete JDBC Notes

1. Application initiates a request using JDBC API.
2. DriverManager loads the suitable JDBC driver.
3. Connection to the DB is established.
4. SQL commands are executed using Statement.
5. Results are returned via ResultSet.

JDBC Flow:

Application JDBC API DriverManager JDBC Driver Database

4. Driver, Connection, Statement Explained

Driver: Interface to load the database driver.

Example:

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

Connection: Represents a session with the database.

Example:

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "root", "pass");
```

Statement: Executes static SQL statements.

```
Statement stmt = con.createStatement();
stmt.executeUpdate("INSERT INTO students VALUES (1, 'Alice', 20)");
```

5. ResultSet Handling & Metadata

ResultSet: Represents data returned from SELECT queries.

```
while(rs.next()) {
  System.out.println(rs.getString("name"));
}
```

Cursor Navigation:

- next(), previous(), first(), last(), absolute(n)

Scrollable ResultSet:

```
Statement stmt = con.createStatement(ResultSet.TYPE_SCROLL_INSENSITIVE, ResultSet.CONCUR_READ_ONLY);
```

ResultSetMetaData: Gives info about columns.

```
ResultSetMetaData rsmd = rs.getMetaData();
int cols = rsmd.getColumnCount();
for (int i = 1; i <= cols; i++) {
  System.out.println(rsmd.getColumnName(i));
}
```

6. Transactions & Batch Processing

Transaction: Ensures all DB operations are executed as a single unit.

```
con.setAutoCommit(false);
stmt.executeUpdate(...);
con.commit(); // or con.rollback();
```

Batch Processing:

```
PreparedStatement ps = con.prepareStatement("INSERT INTO students VALUES (?, ?, ?)");
ps.setInt(1, 1);
ps.setString(2, "John");
ps.setInt(3, 22);
ps.addBatch();

ps.setInt(1, 2);
ps.setString(2, "Mary");
```

```
ps.setInt(3, 24);
ps.addBatch();

ps.executeBatch(); // Executes all
```

7. CallableStatement (Stored Procedures)

Used to call precompiled stored procedures in the database.

```
SQL (MySQL):
CREATE PROCEDURE getStudent(IN sid INT)
BEGIN
  SELECT * FROM students WHERE id = sid;
END;

Java:
CallableStatement cs = con.prepareCall("{call getStudent(?)}");
cs.setInt(1, 1);
ResultSet rs = cs.executeQuery();
while(rs.next()) {
  System.out.println(rs.getString("name"));
}
```

OUT Parameter Example:

```
CREATE PROCEDURE getAge(IN sid INT, OUT age INT)
BEGIN
  SELECT students.age INTO age FROM students WHERE id = sid;
END;

CallableStatement cs = con.prepareCall("{call getAge(?, ?)}");
cs.setInt(1, 1);
```

```
cs.registerOutParameter(2, Types.INTEGER);
cs.execute();
int age = cs.getInt(2);
```

8. Creating Databases and Tables via JDBC

Creating Database:

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/", "root", "pass");
Statement stmt = con.createStatement();
stmt.executeUpdate("CREATE DATABASE IF NOT EXISTS testdb");
```

Creating Table:

```
Connection con = DriverManager.getConnection("jdbc:mysql://localhost:3306/testdb", "root", "pass");
Statement stmt = con.createStatement();
stmt.executeUpdate("CREATE TABLE students (id INT, name VARCHAR(50), age INT)");
```