

**Dt : 15/10/2024(Day-16)**

**\*imp**

**Batch Processing in JDBC:**

**faq:**

**define Batch Processing?**

**=>The process of collecting multiple queries as a batch and executed at-a-time on database product is known as Batch Processing.**

**=>we use the following methods to perform batch processing:**

**(a)addBatch()**

**(b)executeBatch()**

**(c)clearBatch()**

**(a)addBatch():**

**=>addBatch()-method is used to add query to a batch.**

**Method Signature:**

**public abstract void addBatch(java.lang.String) throws java.sql.SQLException;**

**syntax:**

**stm.addBatch("NonSelect-Query");**

**(b)executeBatch():**

**=>executeBatch()-method is used to execute all queries from the batch at-a-time.**

**Method Signature:**

**public abstract int[] executeBatch() throws java.sql.SQLException;**

**syntax:**

**int k[] = stm.executeBatch();**

**(c)clearBatch():**

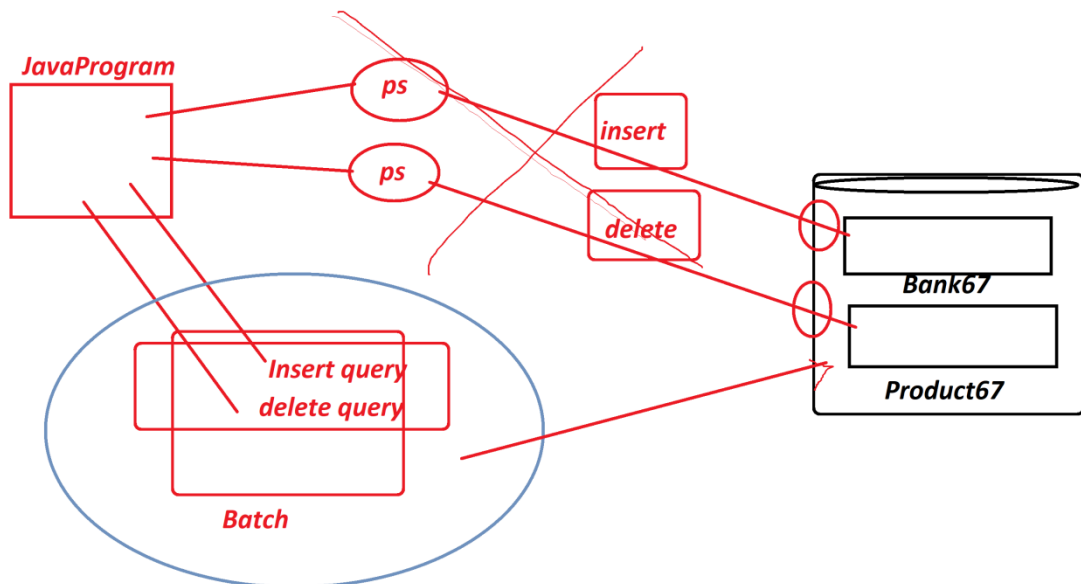
=>clearBatch()-method is used to delete all queries from the batch and destroy the batch.

**Method Signature:**

**public abstract void clearBatch() throws java.sql.SQLException;**

**syntax:**

**stm.clearBatch();**



**Ex:**

**Program : DBCon11.java**

```
package test;
import java.util.*;
import java.sql.*;
public class DBCon12 {
    public static void main(String[] args) {
        Scanner s = new Scanner(System.in);
        try(s){
            Class.forName("oracle.jdbc.driver.OracleDriver");
            Connection con = DriverManager.getConnection
```

```

("jdbc:oracle:thin:@localhost:1521:xe", "system", "tiger");
Statement stm = con.createStatement();
System.out.println("-----Add Customer to Bank67-----");
System.out.println("Enter the Cust-accNo:");
Long accNo = Long.parseLong(s.nextLine());
System.out.println("Enter the Cust-Name:");
String name = s.nextLine();
System.out.println("Enter the Cust-Bal:");
float bal = Float.parseFloat(s.nextLine());
System.out.println("Enter the Cust-accType:");
String accType = s.nextLine();
stm.addBatch
("insert into Bank67
values("+accNo+", '"+name+"', "+bal+", '"+accType+"'");

System.out.println("-----Delete product details based on
code(Product67)----");
System.out.println("Enter the Product code to delete product
details:");
String pCode = s.nextLine();
stm.addBatch
("delete from Product67 where code='"+pCode+"'");

int k[] = stm.executeBatch();
for(int i : k)
{
    System.out.println("query executed : "+i);
} //end of loop
stm.clearBatch();
con.close();
} catch (Exception e) {
    //System.out.println(e.toString());
    e.printStackTrace();
}
}
}
}

```

o/p:

-----Add Customer to Bank67-----

Enter the Cust-accNo:

232323

Enter the Cust-Name:

Rajj

**Enter the Cust-Bal:**

**8000**

**Enter the Cust-accType:**

**Savings**

**-----Delete product details based on code(Product67)----**

**Enter the Product code to delete product details:**

**A123**

**query executed : 1**

**query executed : 1**

-----

**faq:**

**wt is the advantage of Batch Processing?**

**=>In Batch Processing the execution-control reaches database product only once and execute all queries at-a-time,which saves the execution time and generate high performance of an application.**

**Note:**

**(i)Through Batch Processing,we can execute only NonSelect queries and which is also known as Batch Update processing.**

**(ii)'Statement' is preferable to perform batch processing,because we can perform operations on multiple database tables.**

=====

**faq:**

**define Meta Data?**

**=>The data which is holding information about other data is known as Meta data.**

**(Meta Data means data about data)**

=>According to JDBC,Meta data means one object holding information about other

Object known as Meta data object(Meta data component)

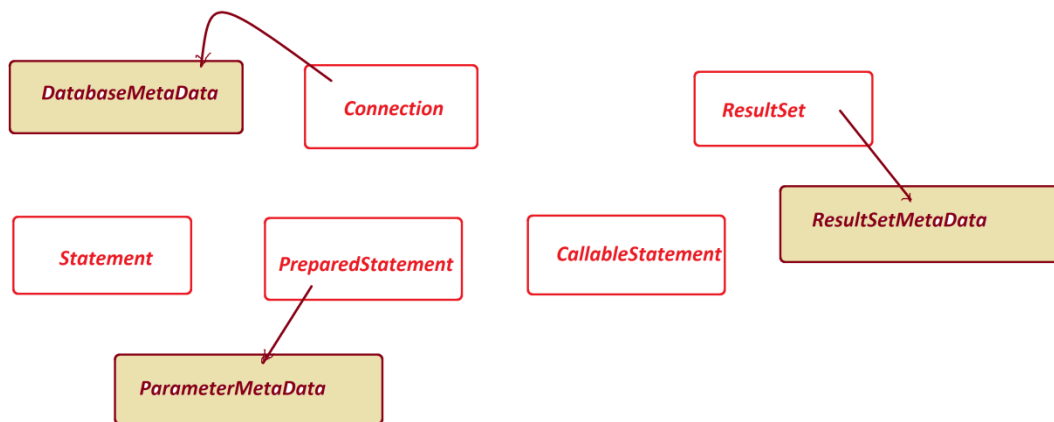
=>The following are some important Meta data components in JDBC:

1.DatabaseMetaData

2.ParameterMetaData

3.ResultSetMetaData

Diagram:



1.DatabaseMetaData:

=>DatabaseMetaData is an interface from java.sql package and which is instantiated using getMetaData()-method from 'Connection' interface.

=>This DatabaseMetaData-Object will hold information about Connection-Object.

syntax:

```
DatabaseMetaData dmd = con.getMetaData();
```

2.ParameterMetaData:

**=>ParameterMetaData is an interface from java.sql package and which is instantiated using getParameterMetaData()-method from 'PreparedStatement' interface.**

**=>This ParameterMetaData-Object will hold information about PreparedStatement Object.**

**syntax:**

**ParameterMetaData pmd = ps.getParameterMetaData();**

### **3.ResultSetMetaData:**

**=>ResultSetMetaData is an interface from java.sql package and which is instantiated using getMetaData()-method from ResultSet-Interface.**

**=>This ResultSetMetaData-object will hold information about ResultSet-Object.**

**syntax:**

**ResultSetMetaData rsmd = rs.getMetaData();**

---