APEX - DML

In Salesforce, we could perform all the database modification functionalities by two ways:

DML Statements

DML are the actions which are performed in order to perform insert, update, delete, upsert, restoring records, merging records, or converting leads operation.

DML is one of the most important part in Apex as almost every business case involves the changes and modifications to database.

Database Methods

All operations which you could perform using DML statements can be performed using Database methods as well. Database methods are the system methods which you could use to perform DML operations. Database methods provide more flexibility as compared to DML Statements.

In this chapter, we will be looking at the first approach using DML Statements. We will look at Database Methods in the next chapter.

DML Statements

Let's take an example of our Chemical supplier company again. Our Invoice records have fields as Status, Amount Paid, Amount Remaining, Next Pay Date and Invoice Number. Invoices which have been created today and have their status as 'Pending', should be updated to 'Paid'.

Insert Operation

Insert operation is used to create new records in Database. You could create records of any Standard or Custom object using the Insert DML statement.

Example:

We would like to create some new records in APEX_Invoice__c object as new invoices are being generated for new customer orders every day. We will create a Customer record first and then we could create an Invoice record for that new Customer record.

```
//fetch the invoices created today, Note, you must have at least one invoice created
today
List<apex_invoice__c> invoiceList = [SELECT id, Name, APEX_Status__c, createdDate FROM
APEX_Invoice__c WHERE createdDate = today];

//create List to hold the updated invoice records
List<apex_invoice__c> updatedInvoiceList = new List<apex_invoice__c>();
APEX_Customer__c objCust = new APEX_Customer__C();
objCust.Name = 'Test ABC';

//DML for Inserting the new Customer Records
insert objCust;
for (APEX_Invoice__c objInvoice: invoiceList) {
  if (objInvoice.APEX_Status__c == 'Pending') {
   objInvoice.APEX_Status__c = 'Paid';
   updatedInvoiceList.add(objInvoice);
  }

}

//DML Statement to update the invoice status
update updatedInvoiceList;

//Prints the value of updated invoices
System.debug('List has been updated and updated values are'+updatedInvoiceList);
```

```
//Inserting the New Records using insert DML statement
APEX_Invoice__c objNewInvoice = new APEX_Invoice__c();
objNewInvoice.APEX_Status__c = 'Pending';
objNewInvoice.APEX_Amount_Paid__c = 1000;
objNewInvoice.APEX_Customer__c = objCust.id;

//DML which is creating the new Invoice record which will be linked with newly created
Customer record
insert objNewInvoice;
System.debug('New Invoice Id is '+objNewInvoice.id+' and the Invoice Number is
'+objNewInvoice.Name);
```

Update Operation

Update operation is to perform the update on existing records. In this example, we would be updating the Status field of an existing Invoice record to 'Paid'.

Example:

```
//Update Statement Example for updating the invoice status. You have to create and
Invoice records before executing this code. This program is updating the record which is
at index 0th position of the List.
//First, fetch the invoice created today
List<apex_invoice__c> invoiceList = [SELECT id, Name, APEX_Status__c, createdDate FROM
APEX_Invoice__c];
List<apex_invoice__c> updatedInvoiceList = new List<apex_invoice__c>();

//Update the first record in the List
invoiceList[0].APEX_Status__c = 'Pending';
updatedInvoiceList.add(invoiceList[0]);

//DML Statement to update the invoice status
update updatedInvoiceList;

//Prints the value of updated invoices
System.debug('List has been updated and updated values of records
are'+updatedInvoiceList[0]);
```

Upsert Operation

Upsert operation is used to perform an update operation and if the records to be updated are not present in database, then create new records as well.

Example:

Suppose, we would like to update the customer records in Customer object. But we would like to update the existing Customer records if it is already present, else create a new one. This will be based on the value of field APEX_External_Id__c. This field will be our field to identify if records is already present or not.

Note: Before executing this code, please create a record in Customer object with the external Id field value as '12341' and then execute the below code:

```
//Example for upserting the Customer records
List<apex_customer__c> CustomerList = new List<apex_customer__c>();
for (Integer i=0; i< 10; i++) {
        apex_customer__c objcust=new apex_customer__c(name='Test' +i,
apex_external_id__c='1234' +i);
        customerlist.add(objcust);
        } //Upserting the Customer Records

upsert CustomerList;

System.debug('Code iterated for 10 times and created 9 records as one record with
External Id 12341 is already present');

for (APEX_Customer__c objCustomer: CustomerList) {
 if (objCustomer.APEX_External_Id__c == '12341') {
  system.debug('The Record which is already present is '+objCustomer);
 }
}
```

Delete Operation

You could perform the delete operation using the Delete DML.

Example:

In this case, we would like to delete the invoices which have been created for the testing purpose, that is the ones which contains name as 'Test'.

You could execute this snippet from Developer console as well without creating the class.

```
//fetch the invoice created today
List<apex_invoice__c> invoiceList = [SELECT id, Name, APEX_Status__c, createdDate FROM
APEX_Invoice__c WHERE createdDate = today];
List<apex_invoice__c> updatedInvoiceList = new List<apex_invoice__c>();
APEX_Customer__c objCust = new APEX_Customer__C();
objCust.Name = 'Test';
//Inserting the Customer Records
insert objCust;
for (APEX_Invoice__c objInvoice: invoiceList) {
    if (objInvoice.APEX_Status__c == 'Pending') {
     objInvoice.APEX_Status__c = 'Paid';
     updatedInvoiceList.add(objInvoice);
    }
}

//DML Statement to update the invoice status
update updatedInvoiceList;

//Prints the value of updated invoices
System.debug('List has been updated and updated values are'+updatedInvoiceList);

//Inserting the New Records using insert DML statement
APEX_Invoice__c objNewInvoice = new APEX_Invoice__c();
objNewInvoice.APEX_Status__c = 'Pending';
objNewInvoice.APEX_Amount_Paid__c = 1000;
objNewInvoice.APEX_Customer__c = objCust.id;

//DML which is creating the new record
insert objNewInvoice;
System.debug('New Invoice Id is '+objNewInvoice.id);

//Deleting the Test invoices from Database
//fetch the invoices which are created for Testing, Select name which Customer Name is
Test.
List<apex_invoice__c> invoiceListToDelete = [SELECT id FROM APEX_Invoice__c WHERE
APEX_Customer__r.Name = 'Test'];

//DML Statement to delete the Invoices
delete invoiceListToDelete;
System.debug('Success, '+invoiceListToDelete.size()+' Records has been deleted');
```

Undelete Operation

You could undelete the records which has been deleted and is present in Recycle bin. All the relationships which the deleted record has, will also be restored.

Example:

Suppose, you would like to restore the Records deleted in previous example. This can be achieved using the following example. We have modified our previous example and added some extra code here.

```
//fetch the invoice created today
List<apex_invoice__c> invoiceList = [SELECT id, Name, APEX_Status__c, createdDate FROM
APEX_Invoice__c WHERE createdDate = today];
List<apex_invoice__c> updatedInvoiceList = new List<apex_invoice__c>();
APEX_Customer__c objCust = new APEX_Customer__C();
objCust.Name = 'Test';


//Inserting the Customer Records
insert objCust;
for (APEX_Invoice__c objInvoice: invoiceList) {
 if (objInvoice.APEX_Status__c == 'Pending') {
  objInvoice.APEX_Status__c = 'Paid';
  updatedInvoiceList.add(objInvoice);
 }
}

//DML Statement to update the invoice status
update updatedInvoiceList;

//Prints the value of updated invoices
System.debug('List has been updated and updated values are'+updatedInvoiceList);

//Inserting the New Records using insert DML statemnt
APEX_Invoice__c objNewInvoice = new APEX_Invoice__c();
objNewInvoice.APEX_Status__c = 'Pending';
objNewInvoice.APEX_Amount_Paid__c = 1000;
objNewInvoice.APEX_Customer__c = objCust.id;

//DML which is creating the new record
insert objNewInvoice;
System.debug('New Invoice Id is '+objNewInvoice.id);

//Deleting the Test invoices from Database
//fetch the invoices which are created for Testing, Select name which Customer Name is
Test.
List<apex_invoice__c> invoiceListToDelete = [SELECT id FROM APEX_Invoice__c WHERE
APEX_Customer__r.Name = 'Test'];

//DML Statement to delete the Invoices
delete invoiceListToDelete;
system.debug('Deleted Record Count is '+invoiceListToDelete.size());
System.debug('Success, '+invoiceListToDelete.size()+'Records has been deleted');

//Restore the deleted records using undelete statement
undelete invoiceListToDelete;
System.debug('Undeleted Record count is '+invoiceListToDelete.size()+'. This should be
same as Deleted Record count');
```