APEX - DATABASE METHODS

Database class methods is another way of working with DML statements which are more flexible than DML Statements like insert, update etc.

Differences between Database Methods and DML Statements

DML Statements	Database Methods
Partial Update is not allowed. For Example, if you have 20 records in list, then either all the records will be updated or none.	Partial update is allowed. You could specify the Parameter in Database method as true or false, true to allow the partial update and false for not allowing the same.
You cannot get the list of success and failed records.	You can get the list of success and failed records as we have seen in the example.
Example: insert listName;	Example: Database.insert*listName, False*, where false indicate that partial update is not allowed.

Insert Operation

Inserting new records via database methods is also quite simple and flexible. Let's take the previous scenario in which we have inserted new records using DML statements. We will be inserting the same using Database methods.

Example:

```
//Insert Operation Using Database methods
//Insert Customer Records First using simple DML Statement. This Customer Record will be
used when we will create Invoice Records
APEX_Customer__c objCust = new APEX_Customer__C();
objCust.Name = 'Test';
insert objCust;//Inserting the Customer Records

//Insert Operation Using Database methods
APEX_Invoice__c objNewInvoice = new APEX_Invoice__c();
List<apex_invoice__c> InvoiceListToInsert = new List<apex_invoice__c>();
objNewInvoice.APEX_Status__c = 'Pending';
objNewInvoice.APEX_Customer__c = objCust.id;
objNewInvoice.APEX_Amount_Paid__c = 1000;
InvoiceListToInsert.add(objNewInvoice);
Database.SaveResult[] srList = Database.insert(InvoiceListToInsert, false);//Database
method to insert the records in List

// Iterate through each returned result by the method
for (Database.SaveResult sr : srList) {
 if (sr.isSuccess()) {
    // This condition will be executed for successful records and will fetch the ids of
successful records
 System.debug('Successfully inserted Invoice. Invoice ID: ' + sr.getId());//Get the
invoice id of inserted Account
 }
else {
    // This condition will be executed for failed records
 for(Database.Error objErr : sr.getErrors()) {
  System.debug('The following error has occurred.');  //Printing error message in Debug
log
  System.debug(objErr.getStatusCode() + ': ' + objErr.getMessage());
  System.debug('Invoice oject field which are affected by the error: ' +
```

```
objErr.getFields());
 }
}
}
```

Update Operation

Let's have a look at our business case example using the database methods. Suppose we would like to update the status field of Invoice object but at the same time, we wanted to have information like status of records, failed record ids, success count, etc. This is not possible by using DML Statements, hence we must use Database methods to get the status of our operation.

Example:

We will be updating the Invoice's 'Status' field if it is in status 'Pending' and created date is today.

Below code is updating the Invoice records using the Database.update method. Also, create an Invoice record before executing this code.

```
//Code to update the records using the Database methods
List<apex_invoice__c> invoiceList = [SELECT id, Name, APEX_Status__c, createdDate FROM
APEX_Invoice__c WHERE createdDate = today];//fetch the invoice created today
List<apex_invoice__c> updatedInvoiceList = new List<apex_invoice__c>();
for (APEX_Invoice__c objInvoice: invoiceList) {
 if (objInvoice.APEX_Status__c == 'Pending') {
  objInvoice.APEX_Status__c = 'Paid';
  updatedInvoiceList.add(objInvoice);//Adding records to the list
 }

}
Database.SaveResult[] srList = Database.update(updatedInvoiceList, false);//Database method
to update the records in List

// Iterate through each returned result by the method
for (Database.SaveResult sr : srList) {
if (sr.isSuccess()) {
    // This condition will be executed for successful records and will fetch the ids of
successful records
    System.debug('Successfully updated Invoice. Invoice ID is : ' + sr.getId());
}
else {
    // This condition will be executed for failed records
 for(Database.Error objErr : sr.getErrors()) {
  System.debug('The following error has occurred.');//Printing error message in Debug
log
  System.debug(objErr.getStatusCode() + ': ' + objErr.getMessage());
  System.debug('Invoice oject field which are affected by the error: ' +
objErr.getFields());
 }
}
}
```

We will be looking at only the Insert and Update operations in this tutorial. The other operations are quite similar to these operations and what we did in last chapter.