1. **Describe OFFSET and give example and why we use?**

Ans- OFFSET is a clause which is used in SOQL query. When using offset only first batch of records are returned for a given query. If you want to retrieve next batch you will need to re-execute query with the highest OFFSET Value.

Example- A query like- [select id, name from Account limit 10 OFFSET 10]

Then it will return record from 11 to 20

2. **Can we convert lookup in to master-detail relationship and how?**

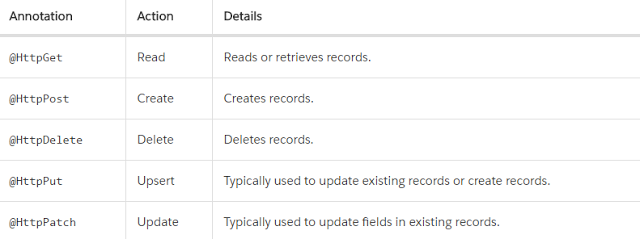
Ans- Yes , we can convert lookup to master but When we are converting lookup in to master then any of the lookup fields should not be blank, if there will be blank then populate values in lookup fields then covert.

3. **How to find that which record is inserted/ updated in batch class?**

Ans- Use SaveResult class like Database.SaveResult[] srLst=d atabase.update(lstData, false); and there is isSuccess() method which finds the successfully record inserted and getError() method used to get error from the failure record and by using getId() method we can get failure records Id in class.

How to use rest Api in class?

REST API  
  
The Force.com REST API lets you integrate with Force.com applications using simple HTTP methods, in either XML or JSON formats, making this an ideal API for developing mobile applications or external clients. Force.com also supports Apex REST, which lets you create Web services on Force.com using Apex

[](https://4.bp.blogspot.com/-M8nTWrNyssc/Vw5mvogT3II/AAAAAAAAAXY/mCtEbPMCuK8wO_xqb9jIBBIhBU0kjzoBACLcB/s1600/Untitled.png)

As you can see, the class is annotated with @RestResource(urlMapping='/api/Account/\*).

The base endpoint for Apex REST is

https://instance.salesforce.com/services/apexrest/.

The URL mapping is appended to the base endpoint to form the endpoint for your REST service.

For example, in the class example, the REST endpoint is

https://instance.salesforce.com/services/apexrest/api/Account/.

@RestResource(urlMapping='/api/Account/\*')  
global with sharing class MyFirstRestAPIClass  
{  
 @HttpGet  
 global static Account doGet()   
 {  
 RestRequest req = RestContext.request;  
 RestResponse res = RestContext.response;  
 String AccNumber = req.requestURI.substring(req.requestURI.lastIndexOf('/')+1);  
 Account result = [SELECT Id, Name, Phone, Website FROM Account WHERE AccountNumber = :AccNumber ];  
 return result;  
 }  
  
 @HttpDelete  
 global static void doDelete()   
 {  
 RestRequest req = RestContext.request;  
 RestResponse res = RestContext.response;  
 String AccNumber = req.requestURI.substring(req.requestURI.lastIndexOf('/')+1);  
 Account result = [SELECT Id, Name, Phone, Website FROM Account WHERE AccountNumber = :AccNumber ];  
 delete result;  
 }  
  
 @HttpPost  
 global static String doPost(String name,String phone,String AccountNumber )   
 {  
 Account acc = new Account();  
 acc.name= name;  
 acc.phone=phone;  
 acc.AccountNumber =AccountNumber ;  
 insert acc;  
   
 return acc.id;  
 }  
  
}

**Execute your Apex REST class in workbench**

**Step 1:-** Open <https://workbench.developerforce.com/>  
**Step 2:-** Select Environment as Production and select the checkbox to agree the terms and condition then select Login with Salesforce.

**Step 3:-** In the Workbench tool select **Utilities > REST Explorer**

**Step 4:-** In the REST Explorer window paste the following url in the box  
  
**Create Account Call**  
**Method :-** Post  
**URL :-** /services/apexrest/api/Account  
**Body:-**  
{  
  "name" : "Test Rest",  
  "phone" : "123456789",  
  "AccountNumber" : "12345"  
  
}

**Get Account Call**  
  
**Method :-** Get  
**URL :-** /services/apexrest/api/Account/12345  
**Body:- NA**

**What is custom metadata and how do we use custom metadata?**

By using custom metadata , we can create our framework for partners, teams, customers etc. Metadata is the data that describe other data.It is used to retrieve,deploy,create,update and delete customization information such as custom object definition and page layout for our organization. It is used to describe objects, their field and their properties.

Create a apex class **'PermissionSetClr'**to get current host url to find your org you are using and create permission for this org.

|  |
| --- |
| public class PermissionSetClr{     public String sHostUrl{get;set;}     public PermissionSetClr()     {         sHostUrl= ApexPages.currentPage().getHeaders().get('Host'); //Get current Host url.     } } |

Now create visualforce page for above controller and define Javascript function 'createPermissionSet()' in this create XML and provide contents like permission set activation when create in org like here I provide false and provide permission set name and provide page name and class name and set object permissions in permission set when create in org.

Now create XML Http Request to send data through API to create permission set. here data will be XML type which we create in 'sXML' variable and attach with metadata and send the request to create permission set.



After create visualforce page and apex class class successfully then preview visualforce page then permission set will create automatically in your org.

Now go to visualforce page access and Apex class access you will find visualforce page and class which you provide name in <pageAccess> and in <classAccess> and these page name and class name should exist in org.

**What is future method?**

Future method is the asynchronous method which runs in background and used to run processes in separate thread, when system resources become available.

We use @future annotation to define future method.

We can call future method for long-running processes, such as calling external web services or any operation you would like to run in its own thread.

We can also use future method to overcome Mixed DML Error.

Each future method is queued and executes when system resource available then execution of future method does not have to wait for the completion of long running processes.

1. Method with the future annotation must be static method with void return type.

2. Methods with the future annotation cannot take sObjects or objects as arguments.

Syntax :

**global class FutureMethodClass**

**{**

**@future**

**public static void processRecords(List<ID> recordIds)**

**{**

**// Get those records based on the IDs**

**List<Account> accts = [SELECT Name FROM Account WHERE Id IN :recordIds];**

**// Process your  records here**

**}**

**}**

**External Web service syntax:**

**global class FutureMethodClass**

**{**

**@future(callout=true)**

**public static void processRecords(List<ID> recordIds)**

**{**

**// Perform a callout to an external service**

**}**

**}**

**What is difference between insert and database.insert()?**

Insert

Insert is a DML statement which is same as Database.insert to insert the record in database. If we use insert DML statement in bulk operation if error occurs in this operation, then execution stop and rollback all the insertion from the database those has been inserted successfully. When error occurs then you can handle this error by try catch block.

Database.Insert

If we use DML database method (Database.insert) to insert/update record in bulk, then if error occurs the remaining records will be inserted/updated means partially operation will be done.

If you insert records using Database.insert(lstAccount,false) and any error comes in between record inserting like some records are getting insert due to required field blank then this statement does not throw error while insertion then insert the right record and do not insert error records.

If you insert records using Database.insert(lstAccount,true) then this statement behave as an insert DML statement. If error comes in between record inserting like some records are getting insert due to required field blank, then this statement throws error and does not insert any record i.e. rolled back whole process.

**Difference between with sharing and without sharing?**

Ans- "with sharing" keyword in apex class so that it will enforce only sharing rules of current user but not, object permissions, field level permissions

But “without sharing” will not enforce only sharing rules of current user . does it mean it will enforce current user object and field level permissions.

**What is upsert? How it works?**

Using the upsert operation, you can either insert or update an existing record in one call. To determine whether a record already exists, the upsert statement or Database method uses the record’s ID as the key to match records, a custom external ID field, or a standard field with the idLookup attribute set to true.

* If the key is not matched, then a new object record is created.
* If the key is matched once, then the existing object record is updated.
* If the key is matched multiple times, then an error is generated and the object record is neither inserted or updated.

Account[] acctsList = [SELECT Id, Name, BillingCity FROM Account WHERE BillingCity = 'Bombay'];

for (Account a : acctsList) {

a.BillingCity = 'Mumbai';

}

Account newAcct = new Account(Name = 'Acme', BillingCity = 'San Francisco');

acctsList.add(newAcct);

try {

upsert acctsList;

} catch (DmlException e) {

// Process exception here

}

