

Admin and Developement scenario based questions

1.when ever a record is inserted to the account automatically inserted to the contact

```
trigger SCENARIO1 on Account (after insert) {
```

```
    list<contact> c=new list<contact>();
```

```
    for(account a:trigger.new)
```

```
    {
```

```
        contact b=new contact();
```

```
        b.LastName=a.Name;
```

```
        b.AccountId=a.Id;
```

```
        c.add(b);
```

```
    }
```

```
    insert c;
```

```
}
```

```
=====
```

2.when ever a record is inserted to the contact automatically inserted to the account

```
trigger scenario2 on Contact (after insert) {
```

```
    if(Recursive.flag)
```

```
    {
```

```
        Recursive.flag=false;
```

```
        list<account>a=new list<account>();
```

```
        for(contact c:trigger.new)
```

```
        {
```

```
            account a1=new account();
```

```
            a1.Phone=c.Phone;
```

```

        a1.Name=c.LastName;

        a.add(a1);

    }

    insert a;

}

```

Recursive trigger fire:

3.avoid recursive trigger

```

public class Recursive {

    public static boolean flag=true;

}

```

```

=====
=====

```

3.when ever a create opportunity object record updated total opportunies and total amount in account object

```

trigger scenario24 on Opportunity (after insert) {

    set<id>ids=new set<id>();

    for(Opportunity op:trigger.new)

    {

        ids.add(op.accountid);

    }

    list<account>ac=[select Total_opportunities__c,Total_Amount__c,(select id,Amount from
    Opportunities ) from account where id=:ids];

    for(account a:ac)

    {

        a.Total_opportunities__c=a.opportunities.size();

        decimal sum=0;

```

```

for(opportunity p:a.opportunities)
{
    sum=sum+p.amount;
}
a.Total_Amount__c=sum;

}

update ac;
}

=====
=====

```

4.contact object when ever department equal to cse automatically before inserted email field

trigger scenario4 on Contact (before insert) {

```

for(contact c:trigger.new)
{
    if(c.Department=='CSE')
    {
        c.Email='naveengorentla1@gmail.com';
    }
}

}

=====
=====

```

5.when ever we modify inputout object doctorname automatically update droppoff object text field
no relation ship

trigger SCENARIO32 on Inputout__c (after update) {

```

list<Dropoff1__c>d=[select id,name,Text__c from Dropoff1__c where Text__c='naveen'];

```

```

string name;
for(Inputout__c c:trigger.new)
{
    name=c.Doctor_Name__c;
}
for(Dropoff1__c dp:d)
{
    dp.Text__c=name;
}
update d;

}

```

```

=====
=====

```

6.limit reached the records

```

trigger SCENARIO6 on Account (before insert,before update) {
    integer count=0;

    list<account>a=[select id,name from account where createddate=today or
lastmodifieddate=today];

    for(account ac:trigger.new)
    {
        count=a.size();
        ac.NumberofLocations__c=count;
        if(count>2)
        {
            ac.adderror('reached limit today');
        }
    }
}

```

```
}
```

```
=====
```

7.can not inser/update/delete that user account object records

trigger scenario30 on Account (before insert,before update,before delete) {

```
    user u=[select id,name from user where username='naveensfdc98@gmail.com'];
```

```
    if(u.id==userinfo.getUserId())
```

```
    {
```

```
        if(trigger.isdelete)
```

```
        {
```

```
            for(account a:trigger.old)
```

```
            {
```

```
                a.adderror('cant delete record');
```

```
            }
```

```
        }
```

```
        if(trigger.isupdate)
```

```
        {
```

```
            for(account b:trigger.new)
```

```
            {
```

```
                b.adderror('can not update');
```

```
            }
```

```
        }
```

```
        if(trigger.isinsert)
```

```
        {
```

```
            for(account c:trigger.new)
```

```
            {
```

```
                c.adderror('can not insert');
```

```
            }
```

```

    }
}
}

=====
=====

```

8.already existing records display error message

trigger scenario8 on Contact (before insert) {

```
list<string>st=new list<string>();
```

```
for(contact c:trigger.new)
```

```
{
```

```
list<contact>a=[select id,name,Email,lastname from contact where Email=:c.Email];
```

```
if(a.size()>0)
```

```
{
```

```
c.Email.adderror('already existing');
```

```
}
```

```
}
```

```
}
```

for loop without query

```
=====
```

trigger duplicatettrigger on Inputout__c (before insert) {

```
set<string>s=new set<string>();
```

```
for(Inputout__c op:trigger.new)
```

```
{
```

```
s.add(op.Doctor_Name__c);
```

```
}
```

```
list<Inputout__c>d=[select id,Doctor_Name__c from Inputout__c where Doctor_Name__c=:s];
```

```

set<string>dupids=new set<string>();

for(Inputout__c don:d)
{
    dupids.add(don.Doctor_Name__c);
}

for(Inputout__c c:trigger.new)
{
    if(c.Doctor_Name__c!=null)
    {
        if(dupids.contains(c.Doctor_Name__c))
        {
            c.Doctor_Name__c.adderror('already existing record');
        }
    }
}
}

```

```

=====
=====

```

9. count of related contacts and accounts field display size

```

public class rollupsummery {

    public static void increment(list<contact>con)
    {
        set<id>ids=new set<id>();
        for(contact c:con)
        {
            ids.add(c.accountid);
        }
    }
}

```

```

    }

    list<account>a=[select id,name,NumberOfEmployees,(select id,lastname from contacts) from
account where id=:ids];

    for(account ac:a)
    {
        ac.NumberOfEmployees=ac.contacts.size();

    }

    update a;

}

}

```

trigger:

=====

```

trigger scenario11 on Contact (after insert) {
    rollupsummery.increment(trigger.new);

```

```

}

```

=====

10. when ever opportunity stagename =closedwon automatically update the account field
rating=hot

```

trigger scenario12 on Opportunity (after insert,after update) {

```

```

    set<id>ids=new set<id>();

```

```

    list<account>ac=new list<account>();

```

```

    for(opportunity op:trigger.new)

```

```

    {

```

```

        ids.add(op.AccountId);

```

```

    ac=[select id,name,rating from account where id=:ids];

```



```

if(op.StageName=='Closed won')
{

    for(account a:ac)
    {
        a.Rating='hot';
    }
    update ac;
}
}

```

=====

11.when ever account name is naveen automatically update the contact all lastnames

trigger scenario13 on Account (after update) {

```

    string names;
    list<contact>c=[select id,lastname,firstname from contact where lastname=:names ];
    for(account a:trigger.new)
    {
        names=a.name;
    }
    for(contact con:c)
    {
        con.lastname=names;
    }
    update c;

```

```
}
```

```
=====
```

12.when ever a opportunity created record amount field is calculated by account total field

trigger scenario21 on Opportunity (after insert,after update,after delete) {

```
    set<id>ids=new set<id>();
```

```
    map<id,opportunity>opp=new map<id,opportunity>();
```

```
    Decimal oldVal;
```

```
    Decimal newVal;
```

```
    if(trigger.isinsert)
```

```
    {
```

```
        for(opportunity op:trigger.new)
```

```
        {
```

```
            ids.add(op.AccountId);
```

```
            opp.put(op.AccountId, op);
```

```
        }
```

```
        list<account> acc=[select id,Total_Amount__c from account where id=:ids];
```

```
        for(account a:acc)
```

```
        {
```

```
            if(a.Total_Amount__c==null )
```

```
            {
```

```
                a.Total_Amount__c=opp.get(a.Id).amount;
```

```
            }
```

```
            else
```

```
            {
```

```
                a.Total_Amount__c= a.Total_Amount__c+opp.get(a.Id).amount;
```

```
            }
```

```

    }
    update acc;
}
if(trigger.isUpdate)
{
for(opportunity op:trigger.new)

{
    ids.add(op.AccountId);
    opp.put(op.AccountId, op);
    newVal=op.Amount;
}
for(Opportunity ops:trigger.old){
    oldVal=ops.Amount;
}
list<account> acc=[select id,Total_Amount__c from account where id=:ids];
for(account a:acc)
{
    if(a.Total_Amount__c==null )
    {
        a.Total_Amount__c=opp.get(a.Id).amount;
    }
    else
    {
        a.Total_Amount__c= a.Total_Amount__c+opp.get(a.Id).amount-oldVal;
    }
}
update acc;
}
}

```

```
}
```

```
=====
```

13.when ever a create a lead object automatically converted account ,contact,opportunity

```
trigger scenario19 on Lead (after insert) {
    list<account>acc=new list<account>();
    list<contact>con=new list<contact>();
    list<opportunity>op=new list<opportunity>();
    for(lead l:trigger.new)
    {
        account a=new account();
        a.Name=l.lastname;
        a.Phone=l.Phone;
        acc.add(a);
        contact c=new contact();
        c.LastName=l.Name;

        con.add(c);
        opportunity o=new opportunity();
        o.Amount=l.AnnualRevenue;
        o.CloseDate=system.today();
        o.StageName='closed won';
        op.add(o);
    }
    insert acc;
    insert con;
    insert op;
```

```
}
```

```
=====
```

14. when ever create a contact automatically update opportunity fields

trigger scenario17 on Contact (after insert) {

```
list<opportunity>op=[select id,name,stagename,Description,amount from opportunity limit 50];
```

```
for(contact c:trigger.new){
```

```
    for(opportunity o:op)
```

```
    {
```

```
        if(o.amount<5000 || o.Amount==null)
```

```
        {
```

```
            o.amount=5000;
```

```
            o.Name=o.Name+'Mr';
```

```
            o.StageName='prospecting';
```

```
        }
```

```
    else{
```

```
        o.Amount=o.Amount+1000;
```

```
        o.Name=o.Name+'Dr';
```

```
    }
```

```
    update o;
```

```
    }
```

```
}
```

```
}
```

```
=====
```

```
=====
```

15. Action poller:

```

public class actionpoller1 {

    public datetime dateandtime{get;set;}

    public void datetimemethod()

    {

        dateandtime=system.now();

    }

}

```

Visualforcepage:

=====

```

<apex:page controller="actionpoller1">

    <apex:form>

        <apex:pageBlock id="pb">

            <apex:actionPoller action="{!datetimemethod}" reRender="pb" interval="5" />

            time:{!dateandtime}

            refresh 5 mins

        </apex:pageBlock>

    </apex:form>

</apex:page>

```

=====

16.Action Status:

=====

```

public class actionstatus {

```

```

Integer count = 0;

```

```

public PageReference incrementCounter() {

```

```
count++;
```

```
return null;
```

```
}
```

```
public Integer getCount() {
```

```
return count;
```

```
}
```

```
}
```

visualforce page:

```
=====
```

```
<apex:page controller="actionstatus">
```

```
<apex:form >
```

```
<apex:outputpanel id="counter">
```

```
<apex:outputText value="Click Me!: {!count}"/>
```

```
<apex:actionSupport event="onclick" action="{!incrementCounter}" rerender="counter"
status="counterStatus"/>
```

```
</apex:outputpanel>
```

```
<apex:actionStatus id="counterStatus" startText=" (processing...)" stopText="(completed)"/>
```

</apex:form>

</apex:page>

=====

17.Aggregate functions

```
public class aggregatefunctions {  
    public list<account>ac{get;set;}  
    public integer count{get;set;}  
    public decimal sum{get;set;}  
    public decimal min{get;set;}  
    public decimal max{get;set;}  
    public aggregatefunctions()  
    {  
        ac= [select id,name,AnnualRevenue from account where name=: 'naveen'];  
        count=[select count() from account];  
        aggregateresult res=[select  
sum(AnnualRevenue)sumt,min(AnnualRevenue)mint,max(AnnualRevenue)maxt from account ];  
        sum=(decimal)res.get('sumt');  
        min=(decimal)res.get('mint');  
        max=(decimal)res.get('maxt');  
  
    }  
}
```

=====VISUALFORCE PAGE AGGREGATE
FUNCTIONS=====

<apex:page controller="aggregatefunctions" >

<apex:form >

<apex:pageBlock >


```

<apex:pageBlockSection >
    <apex:pageBlockTable value="{!ac}" var="n">
        <apex:column value="{!n.name}"/>
        <apex:column value="{!n.AnnualRevenue}"/>
    </apex:pageBlockTable>
</apex:pageBlockSection>

    {!sum}

    {!min}

    {!max}

</apex:pageBlock>
</apex:form>
</apex:page>
=====
=====

```

18.sending email outbound email message

```

public class emailprogramme1 {
    public void myemails()
    {
        messaging.SingleEmailMessage m1=new messaging.SingleEmailMessage();
        string[] toadd=new string[]{'naveengorentla1@gmail.com'};
        string[] tocc=new string[]{'naveen123sfdc@gmail.com'};
        m1.setToAddresses(toadd);
        m1.setCcAddresses(tocc);
        m1.setSubject('accenture');
        m1.setPlainTextBody('this is interview call letter');
        messaging.email[] m2=new messaging.Email[]{m1};
        messaging.sendEmail(m2);
    }
}

```

execution:

=====

```
emailprogramme1 v=new emailprogramme1();
```

```
v.myemails();
```

=====

18.Outbound message pdf file

```
public class emailprogramme2 {
```

```
    public void emailbody()
```

```
    {
```

```
        messaging.SingleEmailMessage m1=new messaging.SingleEmailMessage();
```

```
        string[] toadd=new string[]{'naveengorentla1@gmail.com'};
```

```
        m1.setToAddresses(toadd);
```

```
        m1.setSubject('Pdf file');
```

```
        m1.setPlainTextBody('THIS IS BILL OF TELEPHONE');
```

```
        messaging.EmailFileAttachment m2=new messaging.EmailFileAttachment();
```

```
        pagereference p=page.page1;
```

```
        blob body=p.getContentAsPDF();
```

```
        m2.setBody(body);
```

```
        m2.setFileName('jan-feb-march');
```

```
        messaging.EmailFileAttachment[] eft1=new messaging.EmailFileAttachment[]{m2};
```

```
        m1.setFileAttachments(eft1);
```

```
        messaging.Email[] m3=new messaging.Email[]{m1};
```

```
        messaging.sendEmail(m3);
```

```
    }
```

```
}
```

output:

```
=====
```

```
emailprogramme2 v=new emailprogramme2();
```

```
v.emailbody();
```

```
=====
```

19.sending email template to outbound message

```
public class emailprogramme3 {
```

```
    public void emailmethod()
```

```
    {
```

```
        messaging.SingleEmailMessage m1=new messaging.SingleEmailMessage();
```

```
        emailtemplate et=[select id from emailtemplate where name='doctor' ];
```

```
        m1.setTemplateId(et.Id);
```

```
        contact c=[select id,lastname,phone from contact where phone='999'];
```

```
        m1.setTargetObjectId(c.id);
```

```
        Inputout__c D=[select id,Doctor_Name__c from Inputout__c limit 1];
```

```
        m1.setWhatId(D.Id);
```

```
        messaging.Email[] m2=new messaging.Email[]{m1};
```

```
        messaging.sendEmail(m2);
```

```
    }
```

```
}
```

output:

```
emailprogramme3 v=new emailprogramme3();
```

```
v.emailmethod();
```

=====

20.sending email trigger

```
trigger sendingmailtrigger on Inputout__c (before insert) {
    for(Inputout__c i:trigger.new)
    {
        if(i.Check_box__c==true)
        {
            messaging.SingleEmailMessage m1=new messaging.SingleEmailMessage();
            string[] toadd =new string[]{'naveengorentla1@gmail.com'};
            m1.setToAddresses(toadd);
            m1.setSubject('accenture');
            m1.setPlainTextBody('this is interview call letter');
            messaging.Email[] mail1=new messaging.Email[]{m1};
            messaging.sendEmail(mail1);

        }
    }
}
```

=====

21.APEX sharing rules mandatory things (Philips interview question)

opportunity share

particular user details

opportunity id

row cause

opportunity accesslevel

user or groupid

=====

13. Best practices for triggers

1) One Trigger Per Object

A single Apex Trigger is all you need for one particular object. If you develop multiple Triggers for a single object, you have no way of controlling the order of execution if those Triggers can run in the same contexts

2) Logic-less Triggers

If you write methods in your Triggers, those can't be exposed for test purposes. You also can't expose logic to be re-used anywhere else in your org.

3) Context-Specific Handler Methods

Create context-specific handler methods in Trigger handlers

4) Bulkify your Code

Bulkifying Apex code refers to the concept of making sure the code properly handles more than one record at a time.

5) Avoid SOQL Queries or DML statements inside FOR Loops

An individual Apex request gets a maximum of 100 SOQL queries before exceeding that governor limit. So if this trigger is invoked by a batch of more than 100 Account records, the governor limit will throw a runtime exception

6) Using Collections, Streamlining Queries, and Efficient For Loops

It is important to use Apex Collections to efficiently query data and store the data in memory. A combination of using collections and streamlining SOQL queries can substantially help writing efficient Apex code and avoid governor limits

7) Querying Large Data Sets

The total number of records that can be returned by SOQL queries in a request is 50,000. If returning a large set of queries causes you to exceed your heap limit, then a SOQL query for loop must be used instead. It can process multiple batches of records through the use of internal calls to query and queryMore

8) Use @future Appropriately

It is critical to write your Apex code to efficiently handle bulk or many records at a time. This is also true for asynchronous Apex methods (those annotated with the @future keyword). The differences between synchronous and asynchronous Apex can be found

9) Avoid Hardcoding IDs

When deploying Apex code between sandbox and production environments, or installing Force.com AppExchange packages, it is essential to avoid hardcoding IDs in the Apex code. By doing so, if the record IDs change between environments, the logic can dynamically identify the proper data to operate against and not fail.

Few more Best Practices for Triggers

There should only be one trigger for each object.

Avoid complex logic in triggers. To simplify testing and reuse, triggers should delegate to Apex classes which contain the actual execution logic. See Mike Leach's excellent trigger template for more info.

Bulkify any "helper" classes and/or methods.

Triggers should be "bulkified" and be able to process up to 200 records for each call.

Execute DML statements using collections instead of individual records per DML statement.

Use Collections in SOQL "WHERE" clauses to retrieve all records back in single query

Use a consistent naming convention including the object name (e.g., AccountTrigger)

=====

Dynamic approval process:

* Dynamic approval process is used to route approval requests to users listed in lookup fields on the record requiring approval. In this, approver names are defined dynamically from an object.

```
*approval.processsubmitrequest req1=new approval.processsubmitrequest();
```

==>this class create the request object which need to be submit approved

```
*req1.setobjectid(a.id);
```

=====>it will set id of the record which need to be submitted for approval

```
*req1.setsubmitterid(user1.id);
```

=====>it is id of the user who need to be submit on behalf of whom the record need to be submitted

```
approval.processresult result=approval.process(req1);
```

=====

Order of excution of triggers

=====

==>fetch the data in to database

==>system validations

==>before triggers

==>custom validations

===>save the record to s object

===>after triggers

===>assignment rules

===>auto response rules

===>workflow rules | |

| |==>fieldupdate==>before triggers /after triggers

====>escalation rule

====>rollup summery

====>criteria based sharing

====>commit record

====>email actions

=====

How can we controll excution of a trigger for a specific user or profile?

==>using custom settings(hierachy)

=====>what are the errors you faced in triggers=====

* system.limit exception (soql query 101 exception)

*system.nullpointer exception dereference null object

*recursive trigger

*missing the required field

=====

Best Practices for Test classes

Please follow below salesforce Best Practice for Test Classes :-

1. Test class must start with @isTest annotation if class version is more than 25
 2. Test environment support @testVisible , @testSetUp as well
 3. Unit test is to test particular piece of code working properly or not .
 4. Unit test method takes no argument ,commit no data to database ,send no email ,flagged with testMethod keyword .
 5. To deploy to production at-least 75% code coverage is required
 6. System.debug statement are not counted as a part of apex code limit.
 7. Test method and test classes are not counted as a part of code limit
 9. We should not focus on the percentage of code coverage ,we should make sure that every use case should covered including positive, negative,bulk and single record .
- Single Action -To verify that the the single record produces the correct an expected result .
- Bulk action -Any apex record trigger ,class or extension must be invoked for 1-200 records .

Positive behavior : Test every expected behavior occurs through every expected permutation , i.e user filled out every correctly data and not go past the limit .

Negative Testcase :-Not to add future date , Not to specify negative amount.

Restricted User :-Test whether a user with restricted access used in your code .

10. Test class should be annotated with `@isTest` .

11 . `@isTest` annotation with `testmethod` is equivalent to `testMethod` keyword .

12. Test method should static and no void return type .

13. Test class and method default access is private ,no matter to add access specifier .

14. classes with `@isTest` annotation can't be a interface or enum .

15. Test method code can't be invoked by non test request .

16. Starting with salesforce API 28.0 test method can not reside inside non test classes .

17. `@Testvisible` annotation to make visible private methods inside test classes.

18. Test method can not be used to test web-service call out . Please use call out mock .

19. You can't send email from test method.

20. User, profile, organization, AsyncApexjob, Corntrigger, RecordType, ApexClass, ApexComponent ,ApexPage we can access without (`seeAllData=true`) .

21. `SeeAllData=true` will not work for API 23 version earlier .

22. Accessing static resource test records in test class e.g `List<Account> accList=Test.loadData(Account,SubjectType,'ResourceName')`.

23. Create TestFactory class with @isTest annotation to exclude from organization code size limit .

24. @testSetup to create test records once in a method and use in every test method in the test class .

25. We can run unit test by using Salesforce Standard UI,Force.com IDE ,Console ,API.

26. Maximum number of test classes run per 24 hour of period is not greater than 500 or 10 multiplication of test classes of your organization.

27. As apex runs in system mode so the permission and record sharing are not taken into account .
So we need to use system.runAs to enforce record sharing .

28. System.runAs will not enforce user permission or field level permission .

29. Every test to runAs count against the total number of DML issued in the process .

=====

Best practice for visualforce page

=====

1.Accessing component IDs

2.Page block components

3.Controllers and controller extensions

4.Improving Visualforce's performance

==>Static resources

==>Rendering PDFs

==>Using component facets

5.viewstate error we can avoid transient key word/action region/javascript

Order of execution of visualforce page

=====

1.invoking the controller

2.invoking the constructor

3.invoking the methods

4.invoking the getter and setter methods

=====

Quable and future method

=====

quable || future

1.asynchronous same

2.non primitive also 2.primitive data types
supported

3.job id no job id

=====what is view state in visualforce?=====

state of the data is maintained to the visual force page view state size is 135 kb

decrease view state size we can use transient key word

=====

processbuilder || workflow

8 actions 4 actions

parent to child || only master detail relationship child update parent only update

child to parent

(lookup and master)

update possible

apex class methods || no apex classes

record created || task created

field update || field update

email sending || email sending

post chatter

launch a wizard

out bound message not possible || possible

submit the records || yes

call

=====

profile

role

profile is a collection of

role is a record level security

settings and permissions

what user can do what user can

perform the operations

profile controlles object level,

record

field level,record type,page layout

loggin history

profile is mandatory to the user

role is not mandatory

=====

what is permission set?

==> when we want to give extra permissions to the perticuler user then we go to permission set

example:

i have 2 users

u1 u2

===Profile is same===all permission applied two users i want only one user extra permission we use permission set

=====

* Annotations in Salesforce*

=====

=====

*@ istest:

@future:

@deprecated:Use the deprecated annotation to identify methods, classes, exceptions, enums, interfaces, or variables that can no longer be referenced in subsequent releases of the managed package in which they reside. This is useful when you are refactoring code in managed packages as the requirements evolve. New subscribers cannot see the deprecated elements, while the elements continue to function for existing subscribers and API integrations.

@readonly:he @ReadOnly annotation allows you to perform unrestricted queries against the Force.com database. All other limits still apply. It's important to note that this annotation

@remoteaction:The RemoteAction annotation provides support for Apex methods used in Visualforce to be called via JavaScript. This process is often referred to as JavaScript remoting.

@aura enabled:The @AuraEnabled annotation enables client- and server-side access to an Apex controller method. Providing this annotation makes your methods available to your Lightning components.

@testsetup:Methods defined with the @testSetup annotation are used for creating common test records that are available for all test methods in the class.

Apex REST Annotations

Six new annotations have been added that enable you to expose an Apex class as a RESTful Web service.

`@RestResource(urlMapping='/yourUrl')`

`@HttpDelete`

`@HttpGet`

`@HttpPatch`

`@HttpPost`

`@HttpPut`

=====

Batch apex:

==> Large amount of data divided into no of batches. every batch processed separately is called batch apex

why we use batch apex:

over come the governor limits that's why we are using batch apex

batch apex supports 50 million records

batch apex have 3 methods

1.start method

2.execute method

3.finish method

batch apex have default batch size 200

minimum batch size 1

maximum batch size 2000

can we call the batch apex to another batch apex?

yes we can call the batch apex to another batch apex by declare finish method

Batch apex:

=====

```
global class classname implements database.batchable<subject>
```

```
{
```

```
    global database.querylocator start(database.batchablecontext bc)
```

```
{
```

```
    return database.getquerylocator('Query');
```

```
}
```

```
    global void execute(database.batchablecontext bc,scope)
```

```
{
```

```
    ==logic==
```

```
    global void finish(database.batchablecontext bc)
```

```
{
```

==post operations====

}}

can we call webservice callouts from batch apex?

=====

yes we can call up to 10

database.allowcallouts=true

what is the use of iterable?

=====

the iterable is a return type of start method which is used to write custom logics on multiple objects
fetch the records

by using iterable 50,000 records

=====

database.stateful:>>>state of the data is not maintain to the execute method .if maintain state we
use database.stateful interface we use

=====

Schedule apex:

=====

particular time we can schedule in batch that is schedule apex

public class scheduleapex implements schedulable

{

public void execute(database.schedulablecontext sc)

{

==logic==

```
}  
}
```

schedulable interface which is used in the schedule apex

what is cron trigger: any job which is scheduled in the salesforce is first register with cron trigger

=====

cron trigger likes job is scheduled

if you want to trace the status of the job we use a soql query on cron trigger

cron expression:

=====

secs mins hours daymonth monthyear dayweek year

how to stop the schedule jobs using apex?

=====

system.abortjob(job id)

it has a method getjobid()

=====

Quable apex:

=====

public class quableclass implements queaeable

{

public void execute()

{

//logic//

```
}}
```

```
ID jobId = System.enqueueJob(new AsyncExecutionExample());
```

```
=====
=====
```

Wrapper class:

A wrapper or container class is a class, data structure, or an abstract data type whose instances are a collections of other objects

```
=====
=====Custom
Lable=====
```

====>>custom lable is like custom text values .that values can be accessed by apex classes,visual force pages,custom label values can be translated in to any language salesforce supports.Custom labels enable developers to create multilingual applications by automatically presenting information (for example, help text or error messages) in a user's native language.

in apex class==>system.label.labelname

visualforce page==>{!\$label.labelname}

```
=====
=====
```

Reports:

Reports are nothing but analysis of the data

reports are 4 types

1.tabuler report:

=====>>>this will give you list of records based on the filter criteria with out any sub totals

the tabuler reports are used when you want simple list of items with a grand total

ex:list of all accounts,list of all contacts

2.summery report:

=====

this type of reports provide listing of data with groupings and summarize the records row wise is called summery report

we can following operations

bucketing the field

conditional hilighting

adding a chart

3.matrix report:this type of reports allow you to group records both row and column wise

=====

matrix reports are used when you want to see data by the different dimensions that arent related such as date and product that time we are using mr

summarize opportunities by month vertically and by account horizontally

4.joint report:we can join two or more report types in to a single report we call it as joined reports

=====

we can create 5 blocks for joined report

can we store standard or custom reports in single folder?

=====

====>we cant store both in single folder

*what is difference b/w running user and viewing user?

user who clicks down the run button to generate the report is called running user

user who is accessing the report from the folder is called viewing user

====>in which format report are exported?

.csv or xl format

====>how many records we can exported?

50,000 records we can export

====>what is bucketing field?

bucketing fields in reports in salesforce is used to group values to the name we specify

we can group values

1.picklist

2.number

3.text

====>what is conditional highlighting?

conditional highlighting in reports in salesforce is used to highlighting the field values on summary or matrix reports based on the ranges colours you specify

====>what is dashboard?

dashboard is the graphical representation of the data generated by report or multiple reports

====>what are the different dashboard components?

charts, tables, gauges, metrics, other components we can create by using visual force page

metric: which is used when you have only one key value to display

=====

Sales cloud: "Sales Cloud" refers to the "sales" module in salesforce.com. It includes Leads, Accounts, Contacts, Contracts, Opportunities, Products, Pricebooks, Quotes, and Campaigns (limits apply). It includes features such as Web-to-lead to support online lead capture, with auto-response rules. It is designed to be a start-to-end setup for the entire sales process; you use this to help generate revenue.

Service Cloud: "Service Cloud" refers to the "service" (as in "customer service") module in salesforce.com. It includes Accounts, Contacts, Cases, and Solutions. It also encompasses features such as the Public Knowledge Base, Web-to-case, Call Center, and the Self-Service Portal, as well as customer service automation (e.g. escalation rules, assignment rules). It is designed to allow you to support past, current, and future clients' requests for assistance with a product, service, billing, etcetra; you use this to help make people happy.

Sales Cloud Implements Sales and Marketing for business development while Service cloud implements Salesforce Knowledge.

Sales Cloud is a great solution for small and mid-sized sales groups that want to rapidly increase revenue and cost effectively deploy Salesforce While Service Cloud provides Customer Support to the Clients and giving you the platform to provide a better customer experience for your clients.

Sales Cloud gives you the ability to open Cases and relate them to Accounts, Contacts; etc. While The Service Cloud is a super set of Sales Cloud, meaning you get everything that is in Sales Cloud PLUS some other features.

When we develop product in force.com for sales then it comes in Sales Cloud Ex: – Account, Contacts, and Lead. While when we want to provide some facility and also provides support to the clients then it comes in Service Cloud. Ex: – create cases is the example of Service Cloud in which client write his problem into cases instead of call.

=====

what is setup audittrail in salesforce?

The setup audit trail history helps you track the recent setup changes that you and other administrators have made to your organization. This can be especially useful in organizations with multiple administrators.

=====

what is system log?

=>The Salesforce.com System Log (now the Developer Console) is a valueable tool for any administrator or developer. It can be used to watch requests come into Salesforce.com in real-time. The Salesforce.com Developer Console also allows you to execute anonymous Apex code in real-time.

what is debuglog?

=>A debug log records data base operations, system process, and errors that occur when executing a transaction or while running unit tests. We can monitor specific users in Debug log by adding them to list of Monitored Users.

=====

difference between lookup and masterdetail relation ship?

lookup		master
--------	--	--------

1.look up relation ship is	1.same
----------------------------	--------

one to many relation ship

2.look up create for one object	2.2per one object
---------------------------------	-------------------

40 relation ships

3.look up is not required field	3.required field
---------------------------------	------------------

4.parent deleted child will not be	4.deleted
------------------------------------	-----------

deleted

5.can not support rollup summery object	5.we can create rollup summery in parent
---	--

6.it create standard object also	6.custom object
----------------------------------	-----------------

7.already existing data in object we	7.not possible
--------------------------------------	----------------

can create look up

=====

can we convert lookup to master?

yes possible

can we convert master to lookup?

yes possible

=====

Environments: ==environments in the salesforce is nothing but force.com instance

these are three types of environment

1.developer

2.testing environment

3.production:production is the instance of force.com environments where live data that is actively used to run your business logic is stored

Sandbox: sandbox is a identical environment or copy of production metadata another data where we can perform development and testing

sandboxes are 4 types

=====>1.developer sandbox:this is identical copy of production that are intended for coding and testing by individual developer

data size:200 mb

file size:200 mb

type of data:only metadata

refreshrate:once in a day

actions:development&testing

=====>2.developer pro:this sandbox is a copy of productions which is intended for coding and testing in isolated environment

datasize:1 gb

file size:1gb

refresh rate:1 time in a day

typeof data:only metadata

actions:Development and testing data loading

=====>3.partial copy:this is copy of production which are intended for testing environment these are used for quality assurance & UAT

Data size:5 gb

file size:5 gb

refresh rate:1 time in 5 days

typeof data:only metadata+10000 records for each table

actions:UAT testing quality assurance testing and training

4.full copy:it is exact copy of production with complete metadata & data

Data size:Production size

file size:Production size

refresh rate:29 days once

typeof data:only metadata+complete data

actions:Load testing & performance testing there is also called as staging environments

=====

=====>>documents:when we want to upload any images or supporting files from desktop to the salesforce then we use the concept of documents

Note:if you want to upload logo for application on size should not be more than 20 kb

Static Resource==>Static resources allow you to upload content that you can reference in a Visualforce page, including archives (such as .zip and .jar files), images, style sheets, JavaScript, and other files

=====

To reference a stand-alone file, use `$Resource.<resource_name>` as a merge field, where `<resource_name>` is the name you specified when you uploaded the resource

`<apex:image url="{!$Resource.TestImage}" width="50" height="50"/>`

=====

Standard profiles:

System administrator

standard platform user

authenticated website

chatter free user

readonly

=====

system fields: these are the fields which are created by the salesforce and updated by salesforce time to time

==>id

==>isdeleted

===>created by id

==>created date

==>lastmodified date

===>last modified id

===>system mod

=====

what is pagelayout in salesforce?

specifies how an object should be displayed to the profile

it controls list of fields need to displayed

which buttons need to be displayed

it controls which related lists need to be displayed

it controls the order fields need to be displayed

record types

=====

record types are used to assign more than one pagelayout to the same profile on the same object

using record types we cal also control the options that need to be displayed picklist field

=====

Difference between soql and sosl

=====

SOQL

||

SOSL

1.salesforce object query language

1.salesforce object search language

2.per single transaction we can write queries

2.per single transaction we can write 20 soql

100 soql queries

3.per single transaction retriving the

3.per single transaction retriving data 2000 records

data 50,000 records

4.return type list<subject>

4.return type list<list<subject>>

5.select key word

5.find key word

6.fetching the data in single object

6.fetching the data in multiple objects

or related object data

=====

Soql Query

=====

Parent to child query

=====

list<register>r=[select id,name,phone,(select id,pen,rubber from students__r) from register];

list<account>a=[select id,name,(select lastname,firstname from contacts) from account];

child to parent

=====

list<student__c>s=[select id,pen__c,rubber__c,register__r.name,register__r.phone__c from student];

list<contact> c=[select id,lastname,firstname,account.name,account.phone from contact];

Sosl Query

=====

List<list<subject>>r=[find 'naveen' in all fields Returning
Account(name,industry),contact(lastname,firstname,phone);

we can write where conditions limits,offsets with in sosl

=====

What is vLOOKUP?

vlookup is actually a function in salesforce which is used to bring relevant value to that record from another record automatically

=====

Custom setting Object

Custom Object

=====

=====

1.stores data in application cache

1.stores data in database

memory

2.no need to use soql query to fetch the
from the database

2.we need to use soql query to fetch the records

records

3.limited data types

3.all the data types available

4.we can not create validation rules and apex

4.we can create

triggers on list custom settings

5. we can not create tab for list custom

5. we can create

setting object

=====

Pageblock table: standard salesforce look and like feel displayed

=====

datatable: without standard salesforce look and like feel

=====

Datalist: by using datalist we can display the records in the ordered

=====

we can display only 1000 records in visual force page more than 1000 records we use readonly=true
we can display more than records

what is the use of immediate attribute?

=====

when ever we click on back or cancel button on a vf page if there are mandatory fields then we will see the error messages

=> to bypass validation upon clicking on a button / link we can use immediate attribute

action function: a component that provides support for invoking controller action methods directly from java script code using ajax request

=====

actionpoller:a timer that sends an ajax update request to the server according to a time interval that you specify the update request can then result in a full or partial page update

=====

action region:an area of visualforce page that a democrates which components should be processed by the force.com server when an ajax request is generated

=====

action status:a component that displays the status of an ajax update request an ajax request can either be in progress or complete

=====

action support:a component that adds ajax support to another components ,allowing the component to be refreshed asynchronously by the server

=====

apex:message:a message for a specific component such as a warning or error

=====

apex:messages:all the messages that were generated for all components to the current page

=====

with sharing and with out sharing:

===== =====

With sharing:The with sharing keyword allows you to specify that the sharing rules for the current user be taken into account for a class.

=====

without sharing:Use the without sharing keywords when declaring a class to ensure that the sharing rules for the current user are not enforced.

=====

====>what are the app exchange in salesforce?

=====

answer:the developed custom applications can be uploaded in to the app exchange so that the other person can share the application

what is model view controller?

=====

the main aim of the mvc architecture is to separate the business logic and application data from the presentation data to the user

model:the model object knows about all the data that need to be displayed

=====

view:the view represens the presentation of the application

=====

controller:Actual business logic of vf is present here

=====

what are the controllers available in salesforce?

=====

3 types of controllers

standard controller:used for both custom and standard objects

=====

Custom controller:apex class that implements all the logic for a page apex class funtionalty invoke to the visualforce page

=====

Extension Controller:apex class adds funtionalty to existing standard and custom controller

=====

what are the differences between render,render, and render as?

=====

render:is an attribute used in vf to hide or show certain components in visualforce page

=====

Render as:used to convert visualforce page in to "PDF"

=====

ReRender:used to refresh a part of a page in visualforce page

=====

Content type:we can convert visualforce page in to msword/excel

=====

Email Services

=====

=>what is email service in salesforce?

=====

the email services are automated process that use apex classes to process the contents,headers and attachments of inbound email

example:we can create an email services that automatically creates a contact records based on contact information message

=>what are the different types of email services?

=====

1.outbound email service

2.inbound email service

what is outbound email service?

=====

if we want to send an email from salesforce to external system using outbound message we can send

what is inbound email service?

=====

if we received any email from external system to salesforce then we call it as inbound email

what are the types of outbound email service?

=====

two types

1.single email message:single email message we can send message related to only single record

the single email message contains the methods and classes that are required to send an email these are defined `messaging.SingleEmailMessage` namespace

it contains the methods like

`ToAddresses`

`CCAddresses`

`BccAddresses`

`subject`

`plain text body`

`html body`

syntax:`messaging.SingleEmailMessage msg=new messaging.SingleEmailMessage;`

SetTargetobjectId:the id of the contact,lead or user to which the mail will be sent

we can send up to 100

=====

2.mass email message:by using mass email message we can send mails related to multiple records at a time[whatid and targetobjectId]

=====

messaging.massemailmessage

when we want to send a different email template to different set of recipients we use mass email message

we can up to 250 emails ,but what id can be contacts,case,opportunity or product

who ids group of users /contacts

what is inbound email message?

=====

if you want to receive a email from external system to salesforce we use inbound email message

based on the details or information available in the message

we can create account,contact,case

if we want to acheive the concept of inbound email that class has implemented an interface called

messaging.inboundemailresult:this method contains the datamember called success information,whose value is set to true if the email handled properly

messaging.inboundemail:this class contains the properties of the email which is received as in bound from name,from adress,subject,plaintextbody etc

messaging.inboundenvelope: this class contains to and from address of the inbound email which is received

messaging.inboundemailhandler: it is an interface defined in messaging name space

=====

Analytic Snapshots in Salesforce

=====

==> An Analytic Snapshot helps us to create report on historical data.

Analytic Snapshots comprises of three things:

- * A source Report of type Tabular or Summary

- * A Custom Object with fields matching the columns in the source Report

- * An Analytic Snapshot definition, which maps fields and schedules the snapshot runs

=====

=====DML
OPERATIONS:=====

DML Stnds for data manipulation language

which is used to insert, update, delete, upsert, undelete, merge the dt in to s object

- * what is atomic operations?

for example: if we are performing any operation of dml if any one of the record fails then entire operation will fail

* what are non atomic operations?

if any one of the operation fails only that record fails rest of the records will continue manually

* how to empty the recyclebin?

we can empty the recyclebin by using `database.emptyrecyclebin`

* how to avoid 151 exception?

by bulkifying the operation

bulkifying means by adding all the records to a list and calling the dml on the list

* how do you convert lead?

`database.convertlead()`

=====

===== difference between dataloader and import wizard =====

=====

dataloader		import wizard
------------	--	---------------

=====

1. dataloader is etl tool

1. importing the data in to salesforce

it support export and import the data

2.data loader supports all standard
account,contact,lead,solutions

import wizard supports

objects and custom objects

3.dataloader support 5 milion records

support 50000 records

4.duplicates values allowed

duplicate values not allowed

5.delete operation is available

not available

=====

What is External id:When importing custom objects, solutions, or person accounts, you can use external IDs to prevent duplicate records from being created as a result of the import operation. An external ID is a custom field that has the "External ID" attribute, meaning that it contains unique record identifiers from a system outside of Salesforce. When you select this option, the import wizard will detect existing records in Salesforce that have the same external ID. Note that this operation is not case-sensitive - for example, "ABC" will be matched with "abc". However, there is an exception: if the custom field has the separate "Unique" attribute and the case-sensitive option for that attribute is selected, uppercase and lowercase letters will not be considered identical.

=====

1. Number

2. Text

3. Email

During upsert operation,

1. If External Ids are matched, it will update the record.

2. If External Ids are not matched, it will create a new record.

3. If External Ids are matched more than once, it will throw an error.

=====

what is package?

=>package is a bundle /collection /container for list of components or related applications we can distribute these package to other salesforce organization and users there are two types of packages

1.unmanaged package

2.managed package

unmanaged package:unmanaged package is used to distribute open source applications to provide developers with basic functionality

=====

* we can edit unmanaged package components after installing the unmanged package in a salesforce organization

*the developers who created unmanaged package has no control on installed components can't edit,can't upgrade

managed package: managed package is used to distribute and sell application to customers by using the app exchange developers can sell and manage user based licence to the managed package application

managed packages are fully upgradable

*to ensure seamless upgrades certain destructive changes like removing objects or fields can not be performed

=====

=====

Deployment in Salesforce

=====

ANT:

- 1.installing the java in your local machine
- 2.install ant your local machine
- 3.after installation java and ant creating environment variables path
- 4.download jar file to your ant instalation directory lib

How to use migration tool to delpoy components in salesforce?

=>1.build.properties:this file contains your organization credentials inorder to run the tasks in build.xml file

username

password

server url

2.Build.xml file:this file specifies a series of commonds to be executed by ant with in the build.xml file are named targets that process a series of commonds

when you run ant with a target name

3.package.xml :this file is a project manifest that list all the components you want retrieve or deploy
in a single request you can retrieve or deploy in a single request you can retrieve or deploy only a single
single package at a time

How to run this script in command prompt?

=====

command prompt build.xml file

Eclipse:

=====

we can do deploy one organization to another organization

time is more

1.force.com==>right click deploy

what is zira tool?

Business analyst==>project tracking the status

==>ticket assignment

==>inprogress==>workstart/pending

==>prereview==>senior developer

==>Qa test==>tester checking==>ok

==>user acceptance testing

developer sandbox/qa sandbox/full sandbox

Custom settings

=====

custom setting is like custom object which we called as cache of the salesforce application

the data which is custom setting can be fetched directly using custom setting object with out writing soql query

two types of custom settings:

=====

1.list:this is use to store the list of values and access them using name field

2.heirachy custom setting:when you want to control the values application with respect profile or user

then we use hierachy custom setting

customsettingobject ref=customsettingobject.getinstance(user id/profileid);

=====

What are the Exception handling types in sfdc except try catch block?

Ans:1. AsyncException

2. CalloutException

3. DmlException

4.NullPointerException

5.XmlException

6.SecurityException

7.TypeException

8.StringException

9.SObjectException

10.SearchExceptionN

=====

what is salesforce.com?what are the benifits of salesforce?

=====

salesforce is a cloud computing social enterprices as a provider

software as a service provider

salesforce is webtool

cloud computing: cloud computing is a group of unknown resources that are giving for a specfic purpose to the user

Advantages:

=====

it reduces the cost

increased storage

flexibility

we can access from any location

low maintenance

salesforce will have multitenant architecture

software as a service(SAAS):

=====

Any software application over the internet is called saas

paas:(Platform as a service):

=====

any programming language which application over the internet is called pass

laas:(Infrastructure as a service):

=====

any storage server,application server over the internet is called iaas

what is sand box:

=====

sand box is a identical environment or copy of production metadata another data we can perform development testing training

production:

=====

production is the instance of force.com environments where the live data that is actively used to run your business logic is stored

what is role:

=====

role specifies struture of your organization and where you stand in the hierachy of your organization

whom you have to report your work

role is record level security

role is not mandatory to the user

what is profile:

=====

profile is a collection of setting and pemiissions

profile will specify what user can do what user can perform with in the organization

profile controlles

object level

field level

record type

pagelayout

usersloginhours/ipranges

tab settings

app settings

apex class

visualforce pages

Standard profiles:

=====

system administrator

standard platform user

authenticated website

chatter free user

what is site.com in salesforce?

=====

Site.com is a license based tool from salesforce which allows you to build content rich websites with just drag and drop element type of functionality. Whereas, Force.com site is just a platform to display your visualforce pages to unauthenticated users (i.e non salesforce users)

what is community in salesforce?

=====

Community Cloud is an online social platform that enables companies to connect customers, partners, and employees with each other and the data and records they need to get work done.

what is the difference between freeze and deactivate:

=====

freeze:user cant login but licence is still with the user

deactivate:user cant login but licence return to the organization

what is object:

=====

object is nothing but table in the regular database

what is the tab:

=====

tabs are the interface between user and the application

fields:

=====

fields are nothing but columns in the regular database

formulafields:fields in the salesforce will give you the result in the form of

=====

check box

currency

date

date/time

text

percentage

number

What is Cross object formula:

=====

If you want to refer to the parent object data from the child object then we use cross object

what is validation rule?

=====

validation rule implement the business logic to validate the data ,this are used to verify whether the data what we have entered is it in the Acceptable format or not

Difference between ISBLANK AND ISNULL?

=====

ISBLANK:it will return true if the given value is blank(text/numerical)

ISNULL:it will return true if the given value is null(numerical)

what is field dependency?

=====

if the value in one field controll the value of another field we called it as field dependency

dependent field

=====

we can choose picklist field or multiselect picklist field as a dependent field

what is pagelayouts?

=====

specify which object need to be displayed to the profile

it controls list of fields need to be displayed

which buttons need to be displayed

which related list need to be displayed

what is record type in salesforce?

=====

record types are used to assign morethan one pagelayout to the same profile on the same object

using record types we can also controll the options that need to be displayed picklist fields

OWD(Organization wide defaults)

=====

This will specify which records in the object or visible to the user what type of operations performed by the user

owd is prive

public read only

public read and write

public read and write transfer

public full access

controlled by parent:two objects are connected by master detail relation ship owd assigned on the parent object is applied on the corresponding child object

Grant Access using hierachy:

=====

if you are enable this option who are above higher hierachy he can perform complete operations

difference between queue and public group

=====

queue:when you want to assign group of users as a owner on a record we use queue

workflow:

=====

OUTBOUND MESSAGE:WHEN EVER A workflow rule is met we want to transfer the record from salesforce to external system

for sending outbound message we need url of the server

Approval process:

=====

automated process

what is parallel approval process:

when the record is submitted to more than one user in a single step then we called as parallel approval process

two types:

first response approval:if any one of the user who has given first response decision will be taken as final approval

unanimous:all the users who has choosen approval in the given step has to approve the records to get final approval

what is live agent:

=====

Live Chat Software from Salesforce. ... With Service Cloud chat software, your agents can give customers personalised, real-time help based on how their engaging with your site. Service Cloud is the world's #1 customer service solution for keeping customers satisfied and loyal.

what is assignment rule in salesforce?

=====

Lead Assignment Rules—Specify how leads are assigned to users or queues as they are created manually, captured from the web, or imported via the lead import wizards.

=====

Case Assignment Rules—Determine how cases are assigned to users or put into queues as they are created manually, using Web-to-Case, Email-to-Case, On-Demand Email-to-Case, the Self-Service portal, the Customer Portal, Outlook, or Lotus Notes.

=====

Autoresponce rule in saqlsforce:

=====

An auto-response rule is a set of conditions for sending automatic email responses to lead or case submissions based on the attributes of the submitted record. Applicable leads include those captured through a Web-to-Lead form. Applicable cases include those submitted through a:

Self-Service portal

Customer Portal

Web-to-Case form

Email-to-Case message

On-Demand Email-to-Case message

=====

what is apex in salesforce?

=====

Apex is a development platform for building software as a service (SaaS) applications on top of

Salesforce.com's customer relationship management (CRM) functionality.

Apex allows developers to access Salesforce.com's back-end database and client-server interfaces to create third-party SaaS applications.

what is accessmodifiers in salesforce?

=====

Private: variables accessing the with in the apex classes

=====

public: this means the method or variable can be used by an apex in application

=====

global: this method or variable can be used by any apex code that has access to the class. this access modifier should be used for any method that need to be referenced outside of the application

=====

protected: this means that the method or variable is visible to any inner classes in the defining apex class that extend defined class

=====

what is inline editing in visualforce page?

=====

Inline editing lets users quickly edit field values, right on a record's detail page. ... To see the power of inline editing, create a page called inlineDetail with the following code: 1. <apex:page standardController= "Account" >

IMPQUESTION: If page is having multiple extensions and if two extensions have methods of same name. Then which method out of these two will be called upon calling from vf page ?

=====

=====

==

Ans: The one which is present in the controller defined on the left side will be called.

How can you call a controller method from java script ?

=====

Ans: Use action function component to call controller method from java script.

How can you sort a select SOQL query ?

=====

Ans: use order by clause in select query for sorting a list of records

How to get current logged in users id in apex ?

=====

Ans: Use `Userinfo.getUserId()` to get the current logged in user's id in apex.

How to convert a csv file browsed in visualforce page into a string.

=====

Ans: use `csvfilecontents.toString()` method to convert blob to string

Can you use dml statement in visualforce component controller ?

=====

Ans: To use dml in visualforce component you have to declare `allowdml=true` in visualforce component otherwise you will get an exception

"DML is currently not allowed"

How can you get all the keys of a map variable ?

=====

Ans: Use method `keyset()` for this

Example = `Set<id> idSet = mapname.keySet();`

=====

