**50 Salesforce Interview Questions And Answers**

**1. Can two users have the same profile? Can two profiles be assigned to the same user?**

Profiles determine the level of access a user can have in a **Salesforce org.**

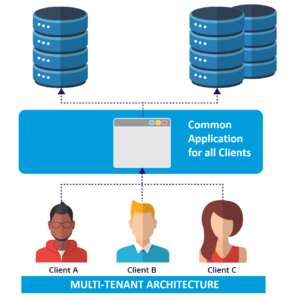
As far as the first part of the question is concerned, Yes. One profile can be assigned to any number of users. Take the example of a Sales or Service team in a company. The entire team will be assigned the same profile. The admin can create one profile: *Sales Profile,* which will have access to the Leads, Opportunities, Campaigns, Contacts and other objects deemed necessary by the company.

In this way, many users can be assigned the same profile. In case the team lead or manager need access to additional records/ objects then it can be done by assigning **permission sets** only for those users.

Answering the second part of the question, each user can only be assigned 1 profile.

**2. What are Governor Limits in Salesforce?**

In Salesforce, it is the **Governor Limits** which controls how much data or how many records you can store in the shared databases. Why? Because Salesforce is based on the concept of multi-tenant architecture. In simpler words, Salesforce uses a single database to store the data of multiple clients/ customers. The below image will help you relate to this concept.



To make sure no single client monopolizes the shared resources, Salesforce introduced the concept of **Governor Limits, which** is strictly enforced by the Apex run-time engine.

Governor Limits are a Salesforce developer’s biggest challenge. That is because if the Apex code ever exceeds the limit, the expected governor issues a run-time exception that cannot be handled. Hence as a Salesforce developer, you have to be very careful while developing your application.

Different Governor Limits in Salesforce are:

Per-Transaction Apex Limits

Force.com Platform Apex Limits

Static Apex Limits

Size-Specific Apex Limits

Miscellaneous Apex Limits

Email Limits

Push Notification Limits

**3. What is a sandbox org? What are the different types of sandboxes in Salesforce?**

A sandbox is a copy of the production environment/ org, used for testing and development purposes. It’s useful because it allows development on Apex programming without disturbing the production environment.

When can you use it?   
You can use it when you want to test a newly developed Force.com application or Visualforce page. You can develop and test it in the Sandbox org instead of doing it directly in production.

This way, you can develop the application without any hassle and then migrate the metadata and data (if applicable) to the production environment. Doing this in a non-production environment allows developers to freely test and experiment applications end to end.

Types of Sandboxes are:

Developer

Developer Pro

Partial Copy

Full

**4. Can you edit an apex trigger/ apex class in production environment? Can you edit a Visualforce page in production environment?**

No, it is not possible to edit apex classes and triggers directly in production environment.

It needs to be done first in Developer edition or testing org or in Sandbox org. Then, to deploy it in production, a user with Author Apex permission must deploy the triggers and classes using deployment tools.

However, Visualforce pages can be created and edited in both sandbox and in production.

Only if the page has to do something unique (different values), it would have to be developed via Sandbox.

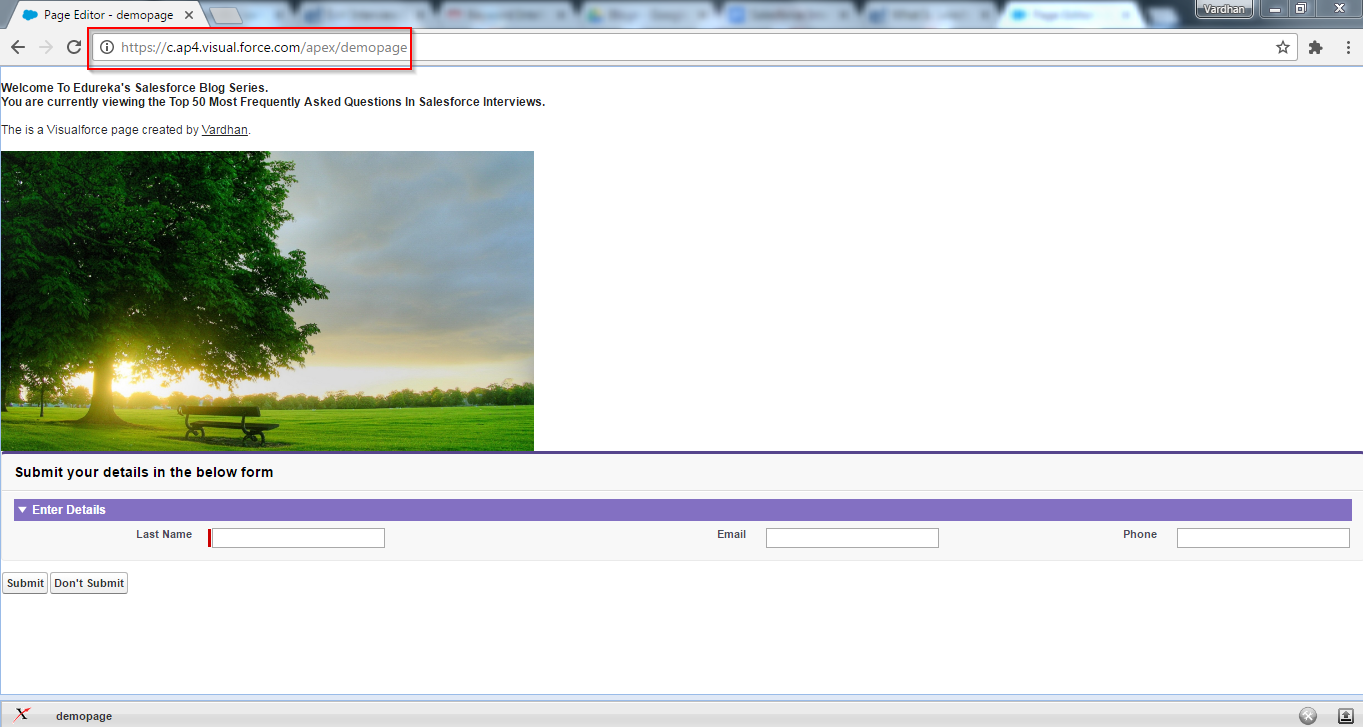
**5. What are the different data types that a standard field record name can have?**

A standard field record name can have data type of either **auto number** or **text field** with a limit of 80 chars.

For generating auto numbers, the format needs to be specified while defining the field and after that for every record that is added, the number will get auto generated. For example:-   
**Sr No-{1}**   
**Sr No-{2}**   
**Sr No-{3}**

**6. Why are Visualforce pages served from a different domain?**

Visualforce pages are served from a different domain to improve security standards and block cross site scripting. Take a look at the highlighted portion in the below Visualforce page:-



**B. Declarative Features – Salesforce Interview Questions**

**7. What is WhoId and WhatId in activities?**

*WhoID* refers to people. Typically: contacts or leads. Example: LeadID, ContactID

*WhatID* refers to objects. Example: AccountID, OpportunityID

**8. What is the use of writing sharing rules? Can you use sharing rules to restrict data access?**

Sharing rules are written to give **edit access** (public read and write) or **public read only access** to certain individuals in Salesforce org. A classic example is when:- only your managers or superiors need to be given extra credentials to your records in objects as compared to your peers.

By default, all users in your organization will have **organization-wide-default** sharing settings of either **Public Read Only** or **Private** .   
To give access to more records, which users do not own, we write sharing rules.   
Example: Sharing rules are used to extend sharing access to users in public groups or roles. Hence, sharing rules are not as strict as organization-wide default settings. They allow greater access for those users.

As far as the second part of the question is concerned, the answer is no. We cannot use sharing rules to restrict data access. It is only used for allowing greater access to records.

**9. What are the different types of email templates that can be created in Salesforce?**

The different types of Email templates are listed in the below table:-

|  |  |
| --- | --- |
| **Text** | All users can create or change this template |
| **HTML with letterhead** | Only Administrators and users having “Edit HTML Templates” permissions can create this template based on a letterhead. |
| **Custom HTML** | Administrators and users having “Edit HTML Templates” permissions can create this template without the need of a letterhead |
| **Visualforce** | Only administrators and developers can create this template. Advanced functionalities like merging data from multiple records is available only in this template |

**C. Audit & Reporting Features – Salesforce Interview Questions**

**10. What is a bucket field in reports?**

A bucket field lets you group related records together by ranges and segments, without the use of complex formulas and custom fields. Bucketing can thus be used to **group** , **filter** , or **arrange** report data. When you create a bucket field, you need to define multiple categories (buckets) that are used to group report values.

The advantage is that earlier, we had to create custom fields to group or segment certain data.

**11. What are dynamic dashboards? Can dynamic dashboards be scheduled?**

Before we understand dynamic dashboards, let us first understand static dashboards. Static dashboards are the basic dashboard types that will be visible to any user who has made a report out of his data. An example of this is what a Sales manager/ Marketing manager would be able to see on his Salesforce org. In other words, a normal dashboard shows data only from a single user’s perspective. Now comes the concept of dynamic dashboards.

**Dynamic dashboards** are used to display information which is tailored to a specific user. Let us consider the same example as above. In case the Sales manager wants to view the report generated specific to only one of his team members, then he can use dynamic dashboards.

You can use dynamic dashboards when you want to show user-specific data of a particular user, such as their personal quotas and sales, or number of case closures, or leads converted etc.   
You can also use a normal/ static dashboard when you want to show regional or organization-wide data to a set of users, such as a particular region’s sales number, or a particular support team’s performance on case closures.

As far as the second part of the question is concerned, no we cannot schedule a dynamic dashboard. That is because whenever we open the dashboard, it will show the data generated in real-time.

**12. What are the different types of reports available in Salesforce? Can we mass delete reports in Salesforce?**

**Salesforce Report Types**

|  |  |
| --- | --- |
| **1. Tabular reports** | Simple Excel type tables which provide a list of items with the grand total |
| **2. Summary reports** | Similar to Tabular reports, but also have functionality of grouping rows, viewing subtotals & creating charts |
| **3. Matrix reports** | Two-dimensional reports which allow you to group records both by row and column |
| **4. Joined reports** | Multiple blocks showing data from different reports based on same or different report types |

Another important point to note here is that, only Summary reports and Matrix reports can be fed as data source for dashboards. Tabular and Joined reports cannot be used as data source for dashboards.

Can we mass delete reports in Salesforce? Of Course we can mass delete reports in Salesforce. The option to mass delete reports can be found under **Data Management** in **Setup** .

**D. Data Modelling & Data Management – Salesforce Interview Questions**

**13. What are the different types of object relations in salesforce? How can you create them?**

No list of Salesforce interview questions is complete without involving relationships between objects in Salesforce. Relationships in Salesforce can be used to establish links between two or more objects.

The different types of object relationships in Salesforce are:

**Master-Detail Relationship (1:n)** :- It is a parent-child relationship in which the master object controls the behavior of the dependent child object. It is a 1:n relationship, in which there can be only one parent, but many children. The main concept you need to be know is that, being the controlling object, the master field cannot be empty. If a record/ field in master object is deleted, the corresponding fields in the dependent object are also deleted. This is called a cascade delete. Dependent fields will inherit the owner, sharing and security settings from its master.

You can define master-detail relationships between two custom objects, or between a custom object and standard object as long as the standard object is the master in the relationship.

**Lookup Relationship (1:n)** :-   
Lookup relationships are used when you want to create a link between two objects, but without the dependency on the parent object. Similar to Master-Detail relationship, you can think of this as a form of parent-child relationship where there is only one parent, but many children i.e. 1:n relationship. The difference here is that despite being controlling field, deleting a record will not result in automatic deletion of the lookup field in the child object. Thus the records in the child object will not be affected and there is no cascade delete here. Neither will the child fields inherit the owner, sharing or security settings of its parent.

**Junction Relationship (Many-To-Many):-**   
This kind of a relationship can exist when there is a need to create two master-detail relationships. Two master-detail relationships can be created by linking 3 custom objects. Here, two objects will be master objects and the third object will be dependent on both the objects. In simpler words, it will be a child object for both the master objects.

**14. What happens to detail record when a master record is deleted? What happens to child record when a parent record is deleted?**

In a Master-Detail relationship, when a master record is deleted, the detail record is deleted automatically (Cascade delete).

In a Lookup relationship, even if the parent record is deleted, the child record will not be deleted.

**15. Can you have a roll up summary field in case of Master-Detail relationship?**

Yes. You can have a roll-up summary in case of a master-detail relationship. But not in case of a lookup relationship.

A roll-up summary field is used to display a value in a master record based on the values of a set of fields in a detail record. The detail record must be related to the master through a master-detail relationship.

There are 4 calculations that you can do using roll-up summary field. You can count the number of detail records related to a master record. Or, you can calculate the sum, minimum value, or maximum value of a field in the detail records.

**16. Explain the term “Data Skew” in Salesforce.**

“ **Data skew** ” is a condition which you will encounter when working for a big client where there are over 10,000 records. When one single user owns that many records we call that condition ‘ownership data skew’.

When such users perform updates, performance issues will be encountered because of “data skew”. This happens when a single user/ members of a single role own most of the records for a particular object.

**17. Explain skinny table. What are the considerations for Skinny Table?**

In Salesforce, skinny tables are used to access frequently used fields and to avoid joins. This largely improves performance. Skinny tables are highly effective, so much so that even when the source tables are modified, skinny tables will be in sync with source tables.

Considerations for skinny tables:

Skinny tables can contain a maximum of 100 columns.

Skinny tables cannot contain fields from other objects.

For full sandboxes: Skinny tables are copied to your Full sandbox organizations, as of the Summer ’15 release.

**18. Which fields are automatically Indexed in Salesforce?**

Only the following fields are automatically indexed in Salesforce:

**Primary keys** (Id, Name and Owner fields).

**Foreign keys** (lookup or master-detail relationship fields).

**Audit dates** (such as SystemModStamp).

**Custom fields** marked as an External ID or a unique field.

**19. How to handle comma within a field while uploading using Data Loader?**

In a Data Loader .CSV, if there is a comma in field content, you will have to enclose the contents within double quotation marks: **” “** .

**E. Logic & Process Automation – Salesforce Interview Questions**

**20. For which criteria in workflow “time dependent workflow action” cannot be created?**

Time dependent workflow action cannot be create for: “ **created, and every time it’s edited** ”.

**21. What are the types of custom settings in Salesforce? What is the advantage of using custom settings?**

There are two types of custom settings in Salesforce: **List Custom Settings** and **Hierarchy Custom Settings** .

List Custom Settings are a type of custom settings that provides a reusable set of static data that can be accessed across your organization irrespective of user/ profile.   
Hierarchy Custom Settings are another type of custom settings that uses built-in hierarchical logic for “personalizing” settings for specific profiles or users.

The advantage of using custom settings is that it allows developers to create a custom set of access rules for various users and profiles.

**22. How many active assignment rules can you have in a lead/ case?**

Only one rule can be active at a time.

**23. What are custom labels in Salesforce? What is the character limit of custom label?**

Custom labels are custom text values that can be accessed from Apex classes or Visualforce pages. The values here can be translated into any language supported by Salesforce.   
Their benefit is that they enable developers to create multilingual applications which automatically presents information in a user’s native language.

You can create up to 5,000 custom labels for your organization, and they can be up to 1,000 characters in length.

**24. What is the difference between a Role and Profile in Salesforce?**

As mentioned in one of the previous Salesforce interview questions, a profile will ultimately control access to which records a user has in a Salesforce org. No user can work on the Salesforce org without being assigned a profile. The **Profile** is therefore mandatory for every user.

**Role** however is not mandatory for every user. The primary function of the Role/ Role hierarchy is that it allows higher level users in hierarchy get access to records owned by lower level users in the hierarchy. An example of that is Sales Managers getting access to records owned by Sales Reps while their peers do not get access to it.

**25. What are the examples of non-deterministic Force.com formula fields?**

Before I mention some of the examples, let me give you an introduction to deterministic and non-deterministic formula fields. Formula fields whose value will be static are referred to as deterministic fields. Whereas, formula fields whose value will be changed dynamically or whose values will have to be calculated on the fly, they are referred to as non-deterministic formula fields. A classic example of that is a formula returning the current date and time.

Some examples of non-deterministic fields in Force.com are:

Lookup fields

Formula fields whose reference spans over other entities

Fields having dynamic date functions like:- TODAY() or NOW()

**F. Software Testing – Salesforce Interview Questions**

**26. Why do we need to write test classes? How to identify if a class is a test class?**

Software developers from around the world will unanimously agree that writing code in test classes makes debugging more efficient. Why? That is because test classes help in creating robust and error-free code be it Apex or any other programming language. Since Unit tests are powerful in their own right, Salesforce requires you to write test classes in Apex code.

Why are they so powerful? Because test classes and test methods verify whether a particular piece of code is working properly or not. If that piece of code fails, then developers/ testers can accurately locate the test class having the faulty bug.

Test classes can be determined easily because every test class will be annotated with **@isTest** keyword. In fact, if we do not annotate a test class with **@isTest** , then it cannot be defined as a test class. Similarly, any method within a class which has the keyword **testMethod** , is a test method.

**27. What is minimum test coverage required for trigger to deploy?**

In Salesforce, if you want to deploy your code to production, then you must make sure that at least **75% of your Apex code** is covered by unit tests. And all these tests must complete successfully.

**G. Debug & Deployment Tools – Salesforce Interview Questions**

**28. What are the different ways of deployment in Salesforce?**

You can deploy code in Salesforce using:

Change Sets

Eclipse with Force.com IDE

Force.com Migration Tool – ANT/Java based

Salesforce Package

**H. Integration – Salesforce Interview Questions**

**29. What is an external ID in Salesforce? Which all field data types can be used as external IDs?**

An external ID is a custom field which can be used as a unique identifier in a record. External IDs are mainly used while importing records/ data. When importing records, one among the many fields in those records need to be marked as an external ID (unique identifier).

An important point to note is that only custom fields can be used as External IDs. The fields that can be marked as external IDs are: **Text** , **Number** , **E-Mai** l and **Auto-Number** .

**30. How many callouts to external service can be made in a single Apex transaction?**

Governor limits will restrict a single Apex transaction to make a maximum of 100 callouts to an HTTP request or an API call.

**31. How can you expose an Apex class as a REST WebService in Salesforce?**

You can expose your Apex class and methods so that external applications can access your code and your application through the REST architecture. This is done by defining your Apex class with the **@RestResource** annotation to expose it as a REST resource. You can then use **global classes** and a **WebService callback method** .

Invoking a custom Apex REST Web service method always uses system context. Consequently, the current user’s credentials are not used, and any user who has access to these methods can use their full power, regardless of permissions, field-level security, or sharing rules.

Developers who expose methods using the Apex REST annotations should therefore take care that they are not inadvertently exposing any sensitive data. Look at the below piece of code for instance:-

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15  16  17  18  19  20 | global class AccountPlan {  webservice String area;  webservice String region;  //Define an object in apex that is exposed in apex web service  global class Plan {  webservice String name;  webservice Integer planNumber;  webservice Date planningPeriod;  webservice Id planId;  }  webservice static Plan createAccountPlan(Plan vPlan) {  //A plan maps to the Account object in salesforce.com.  //So need to map the Plan class object to Account standard object  Account acct = new Account();  acct.Name = vPlan.name;  acct.AccountNumber = String.valueOf(vPlan.planNumber);  insert acct;  vPlan.planId=acct.Id;  return vPlan;  } } |

**I. Programmatic Features – Salesforce Interview Questions**

**32. What is the difference between a standard controller and a custom controller?**

Standard controller in Apex, inherits all the standard object properties and standard button functionality directly. It contains the same functionality and logic that are used for standard Salesforce pages.

Custom controller is an Apex class that implements all of the logic for a page without leveraging a standard controller. Custom Controllers are associated with Visualforce pages through the controller attribute.

**33. How can we implement pagination in Visualforce?**

To control the number of records displayed on each page, we use pagination. By default, a list controller returns 20 records on the page. To customize it, we can use a controller extension to set the **pageSize** . Take a look at the sample code below:-

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14  15 | <apex:page standardController="Account" recordSetvar="accounts">  <apex:pageBlock title="Viewing Accounts">  <apex:form id="theForm">  <apex:pageBlockSection >  <apex:dataList var="a" value="{!accounts}" type="1">  {!a.name}  </apex:dataList>  </apex:pageBlockSection>  <apex:panelGrid columns="2">  <apex:commandLink action="{!previous}">Previous</apex:commandlink>  <apex:commandLink action="{!next}">Next</apex:commandlink>  </apex:panelGrid>  </apex:form>  </apex:pageBlock>  </apex:page> |

**34. How can you call a controller method from JavaScript?**

To call a controller method (Apex function) from JavaScript, you need to use **actionfunction** .

Look at the below piece of code to understand how a controller method is called using actionfunction.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11 | <apex: page>  <apex:form id ="frm">  <apex:actionfunction name="callfromJS" action="{!controllerMethodName} reRender="frm"/>  </apex:form>  <script>  function JSmethodCallFromAnyAction()  {  callfromJS();  }  </apex:page> |

**35. How to get the UserID of all the currently logged in users using Apex code?**

You can get the ID’s of all the currently logged in users by using this global function: **UserInfo.getUserId()** .

**36. How many records can a select query return? How many records can a SOSL query return?**

The Governor Limits enforces the following:-

Maximum number of records that can be retrieved by SOQL command: 50,000.

Maximum number of records that can be retrieved by SOSL command: 2,000.

**37. What is an attribute tag? What is the syntax for including them?**

An attribute tag is a definition of an attribute of a custom component and it can only be a child of a component tag.

Note that you cannot define attributes with names like id or rendered. These attributes are automatically created for all custom component definitions. The below piece of code shows the syntax for including them:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | <apex:component>  <apex:attribute name="myValue" description="This is the value for the component." type="String" required="true"/>  <apex:attribute name="borderColor" description="This is color for the border." type="String" required="true"/>  <h1 style="border:{!borderColor}">  <apex:outputText value="{!myValue}"/>  </h1>  </apex:component> |

**38. What are the three types of bindings used in Visualforce? What does each refer to?**

There are three types of bindings used in Salesforce:-

**Data bindings** , which refer to the data set in the controller

**Action bindings** , which refer to action methods in the controller

**Component bindings** , which refer to other Visualforce components.

Data bindings and Action bindings are the most common and they will be used in every Visualforce page.

**39. What are the different types of collections in Apex? What are maps in Apex?**

Collections are the type of variables which can be used to store multiple number of records (data).

It is useful because Governor Limits restrict the number of records you can retrieve per transaction. Hence, collections can be used to store multiple records in a single variable defined as type collection and by retrieving data in the form of collections, Governor Limits will be in check. Collections are similar to how arrays work.

There are 3 collection types in Salesforce:

Lists

Maps

Sets

Maps are used to store data in the form of key-value pairs, where each unique key maps to a single value.   
Syntax: Map<String, String> country\_city = new Map<String, String>();

**40. How can you embed a Visualflow in a Visualforce page?**

Find the flow’s unique name.

From Setup, enter Flows in the Quick Find box, then select Flows.

Click the name of the flow.

Copy the unique name of the flow.

From Setup, enter Visualforce Pages in the Quick Find box, then select Visualforce Pages.

Define a new Visualforce page, or open an existing one.

Add the <flow:interview> component somewhere between the <apex:page> tags.

Set the name attribute to the unique name of the flow.

For example:

|  |  |
| --- | --- |
| 1  2  3 | </apex:page>  <flow:interview name="flowuniquename"/>  <apex:page> |

Click Save.

Restrict which users can access the Visualforce page.

Click Visualforce Pages.

Click Security next to your Visualforce page.

Move all the appropriate profiles from Available Profiles to Enabled Profiles by using the ‘add’ and ‘remove’ buttons.

Click Save.

Add the Visualforce page to your Force.com app by using a custom button, link, or Visualforce tab.

**41. What is the use of “@future” annotation?**

Future annotations are used to identify and execute methods asynchronously. If the method is annotated with “ **@future** ”, then it will be executed only when Salesforce has the available resources.

For example, you can use it while making an asynchronous web service callout to an external service. Whereas without using the annotation, the web service callout is made from the same thread that is executing the Apex code, and no additional processing will occur until that callout is complete (synchronous processing).

**42. What are the different methods of batch Apex class?**

**Database.Batchable** interface contains three methods that must be implemented:

**Start method** :   
global (Database.QueryLocator | Iterable<sObject>) start(Database.BatchableContext bc) {}

**Execute method** :   
global void execute(Database.BatchableContext BC, list<P>){}

**Finish method** :   
global void finish(Database.BatchableContext BC){}

**43. What is a Visualforce component?**

A Visualforce Component is either a predefined component (standard from component library) or a custom component that determines the user interface behavior. For example, if you want to send the text captured from the Visualforce page to an object in Salesforce, then you need to make use of Visualforce components. Example: **<apex:detail>**

**44. What is Trigger.new?**

Triger.new is a command which returns the list of records that have been added recently to the sObjects. To be more precise, those records will be returned which are yet to be saved to the database. Note that this sObject list is only available in insert and update triggers, and the records can only be modified in before triggers.

But just for your information, Trigger.old returns a list of the old versions of the sObject records. Note that this sObject list is only available in update and delete triggers.

**45. What all data types can a set store?**

Sets can have any of the following data types:

Primitive types

Collections

sObjects

User-defined types

Built-in Apex types

**46. What is an sObject type?**

An sObject is any object that can be stored in the Force.com platform database. Apex allows the use of generic sObject abstract type to represent any object.

For example, Vehicle is a generic type and Car, Motor Bike all are concrete types of Vehicle.   
In SFDC, sObject is generic and Account, Opportunity, CustomObject\_\_c are its concrete type.

**47. What is the difference between SOQL and SOSL?**

The differences are mentioned in the table below:

**SOQL vs SOSL**

|  |  |
| --- | --- |
| **SOQL (Salesforce Object Query Language)** | **SOSL (Salesforce Object Search Language)** |
| Only one object can be searched at a time | Many objects can be searched at a time |
| Can query any type of field | Can query only on email, text or phone |
| Can be used in classes and triggers | Can be used in classes, but not triggers |
| DML Operation can be performed on query results | DML Operation cannot be performed on search results |
| Returns records | Returns fields |

**48. What is an Apex transaction?**

An Apex transaction represents a set of operations that are executed as a single unit. The operations here include the DML operations which are responsible for querying records. All the DML operations in a transaction either complete successfully, or if an error occurs even in saving a single record, then the entire transaction is rolled back.

**49. What is the difference between public and global class in Apex?**

**Global class** is accessible across the Salesforce instance irrespective of namespaces.   
Whereas, **public classes** are accessible only in the corresponding namespaces.

**50. What are getter methods and setter methods?**

**Get (getter) method** is used to pass values from the controller to the VF page.   
Whereas, the **set (setter) method** is used to set the value back to controller variable.