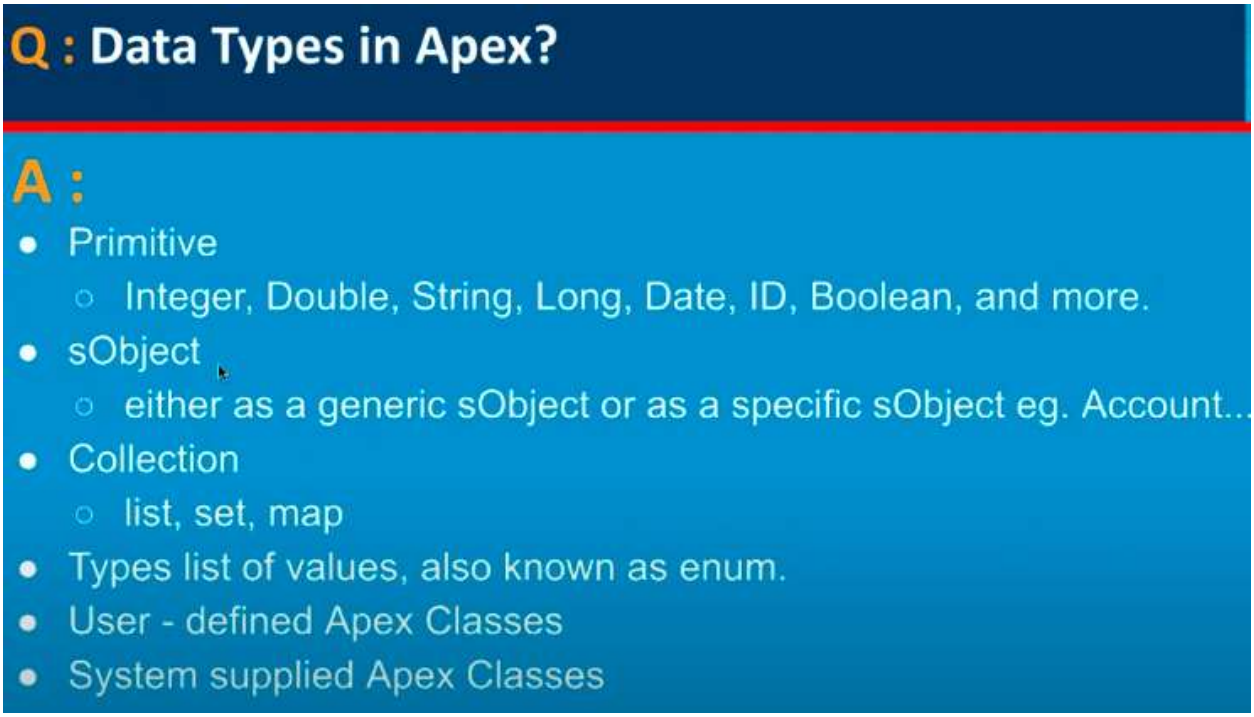
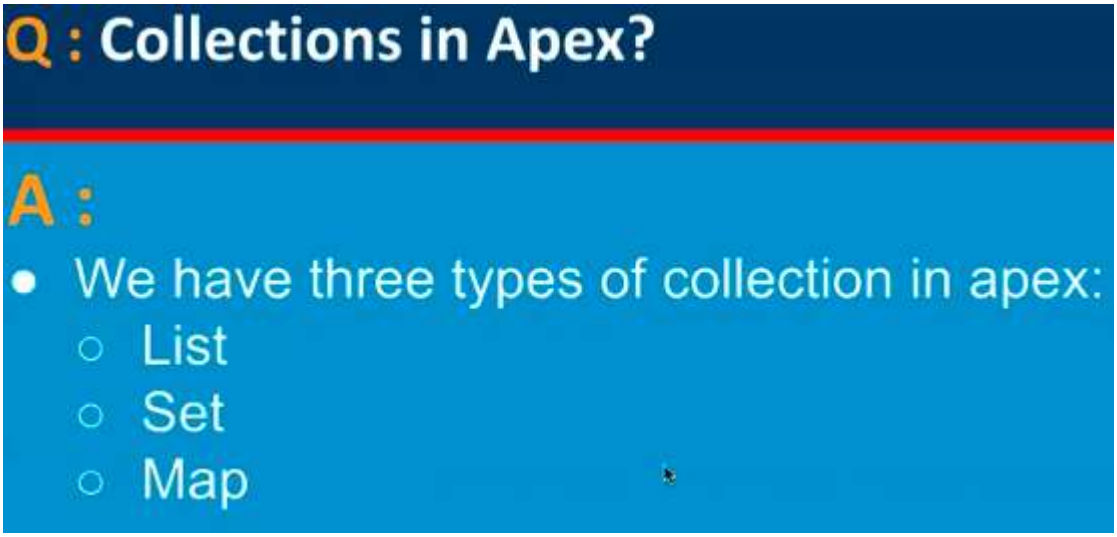
1.


Q : What is Apex?
A :
• Apex is Object Oriented Programming Language used in Salesforce.
• Strongly typed as it validates references to objects at compile time.
• Integrated with the database
• Enable developers to add business logics.
• We can call apex through web service requests and triggers on objects.

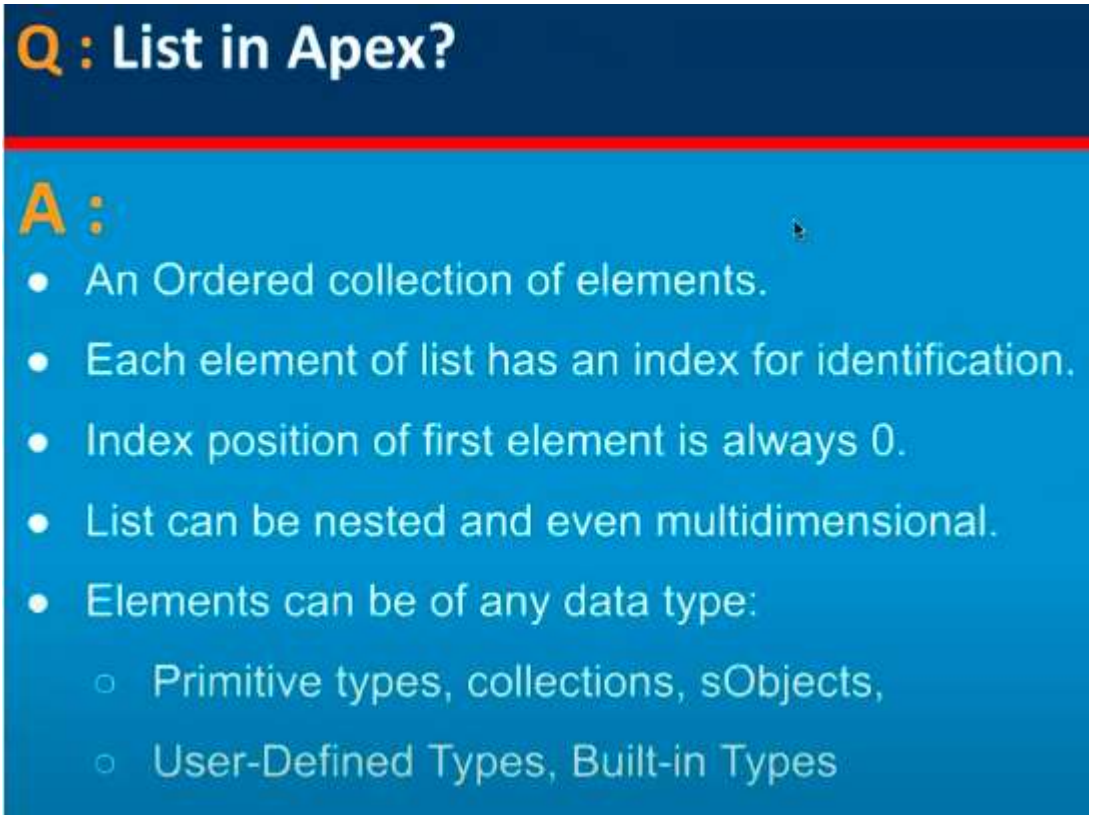
2.


Q : Data Types in Apex?
A :
• Primitive
◦ Integer, Double, String, Long, Date, ID, Boolean, and more.
• sObject
◦ either as a generic sObject or as a specific sObject eg. Account...
• Collection
◦ list, set, map
• Types list of values, also known as enum.
• User - defined Apex Classes
• System supplied Apex Classes

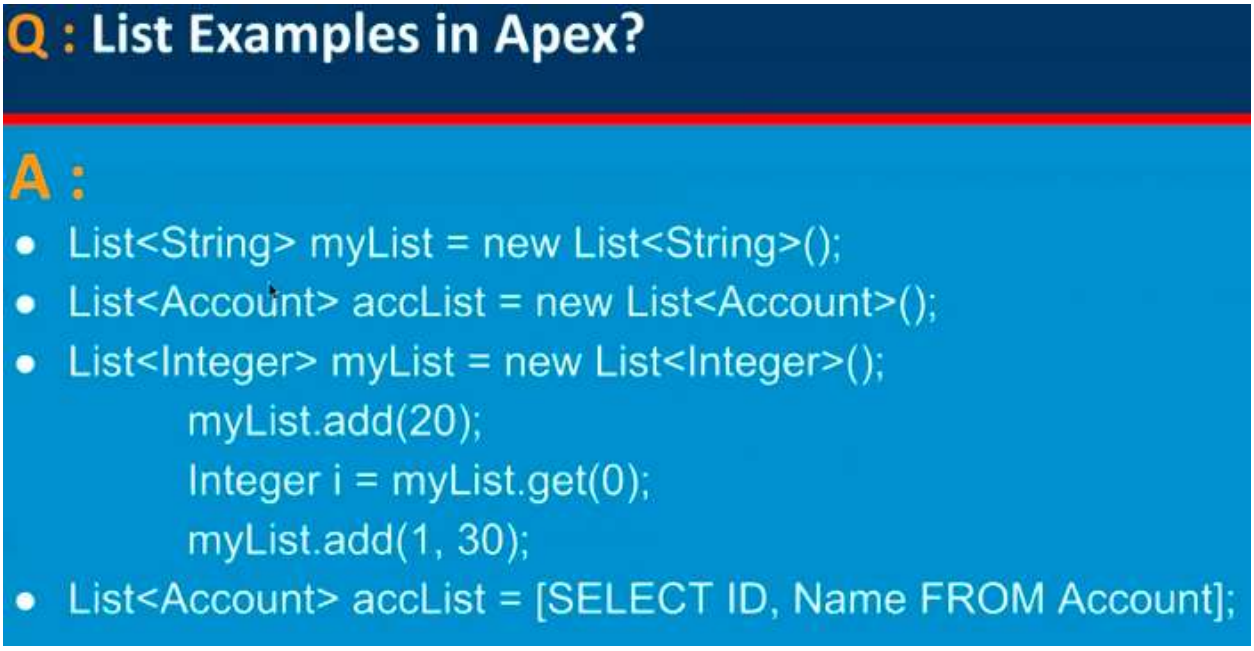
3.


Q : Collections in Apex?
A :
We have three types of collection in apex:
List
Set
Map

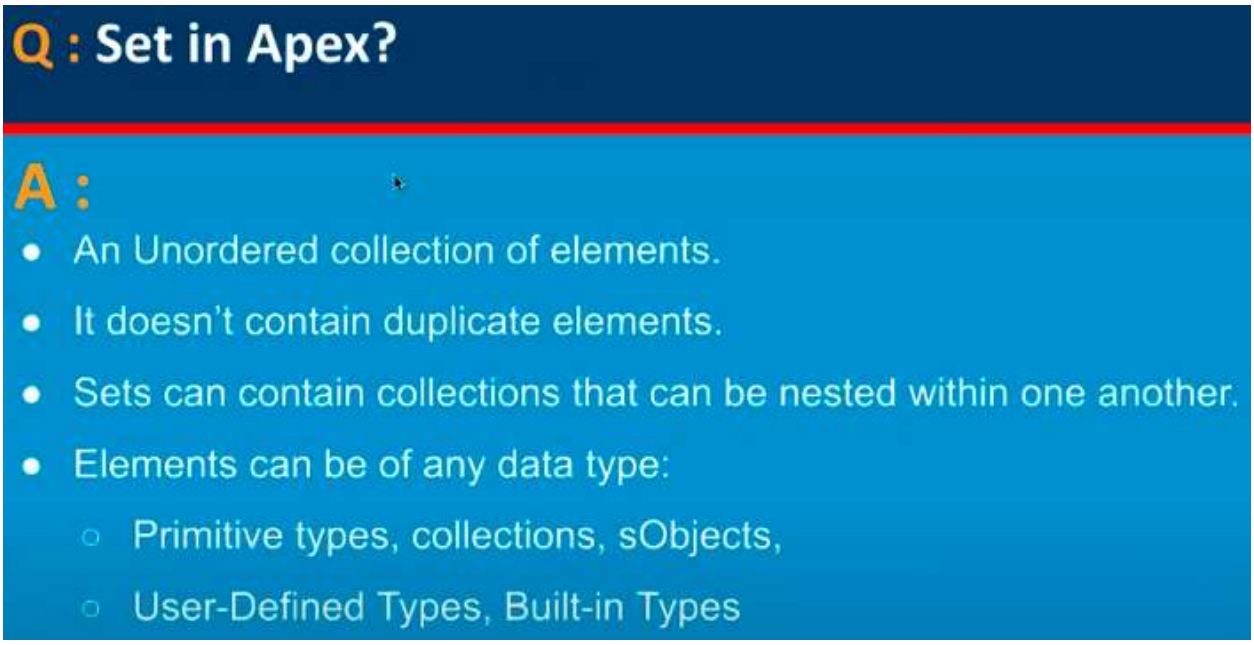
4.


Q : List in Apex?
A :
An Ordered collection of elements.
Each element of list has an index for identification.
Index position of first element is always 0.
List can be nested and even multidimensional.
Elements can be of any data type:
Primitive types, collections, sObjects,
User-Defined Types, Built-in Types

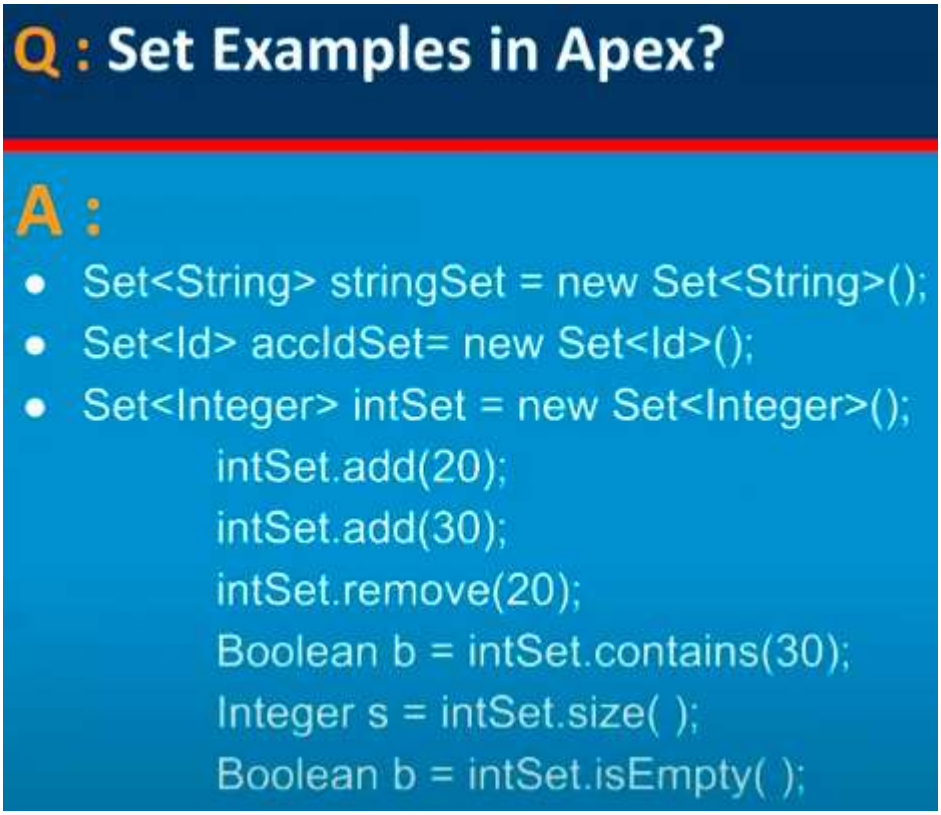
5.


Q : List Examples in Apex?
A :
List<String> myList = new List<String>();
List<Account> accList = new List<Account>();
List<Integer> myList = new List<Integer>();
myList.add(20);
Integer i = myList.get(0);
myList.add(1, 30);
List<Account> accList = [SELECT ID, Name FROM Account];

6.


Q : Set in Apex?
A :
An Unordered collection of elements.
It doesn't contain duplicate elements.
Sets can contain collections that can be nested within one another.
Elements can be of any data type:
Primitive types, collections, sObjects,
User-Defined Types, Built-in Types

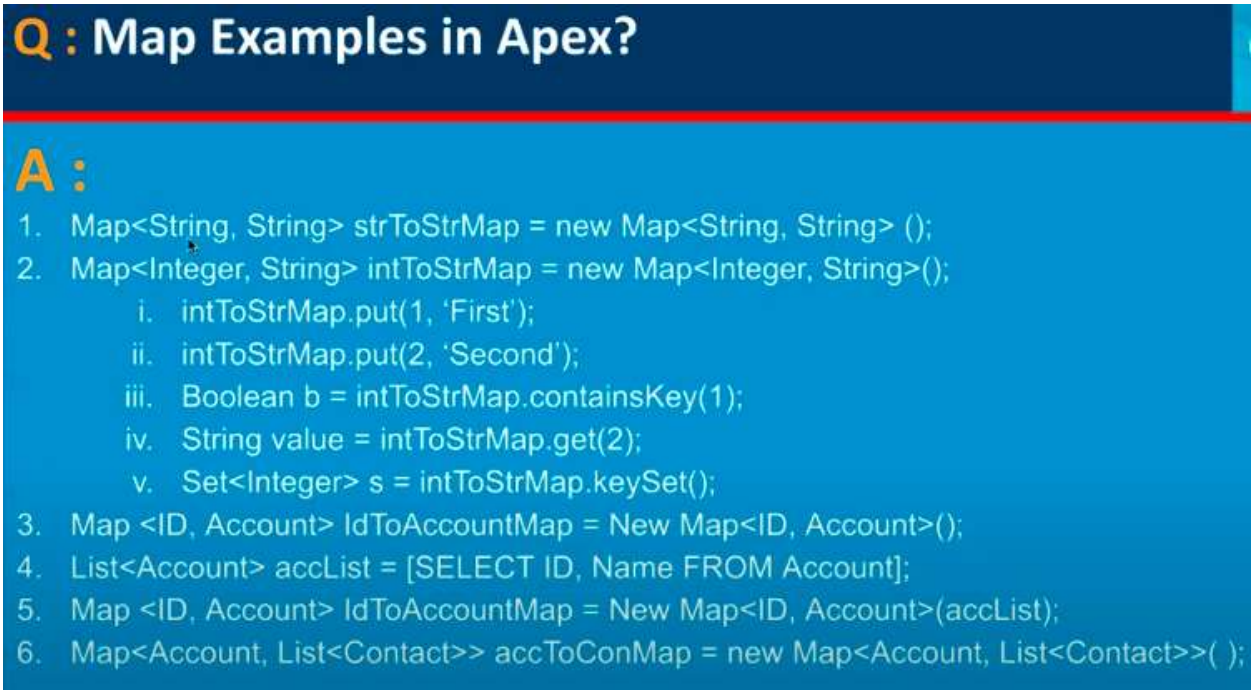
7.


Q : Set Examples in Apex?
A :
Set<String> stringSet = new Set<String>();
Set<Id> accIdSet= new Set<Id>();
Set<Integer> intSet = new Set<Integer>();
intSet.add(20);
intSet.add(30);
intSet.remove(20);
Boolean b = intSet.contains(30);
Integer s = intSet.size( );
Boolean b = intSet.isEmpty( );

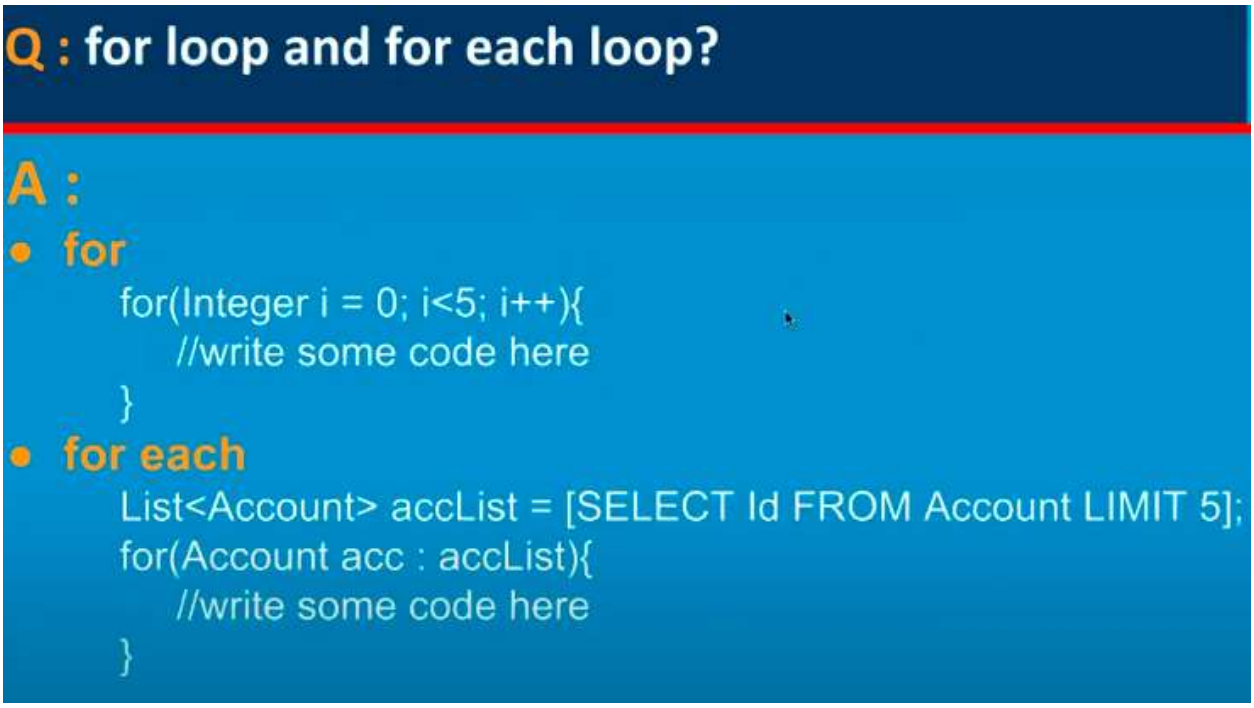
8.


Q : Map in Apex?
A :
A Map is a collection of key - value pairs.
Keys are always unique having a value associated.
Values can be duplicate.
Adding a map entry with an existing key overrides the existing entry with new.
Map key and values can contain any collection and can contain nested collections.
Keys and values can be of any data type:
Primitive types, collections, sObjects, User-Defined Types, Built-in Types

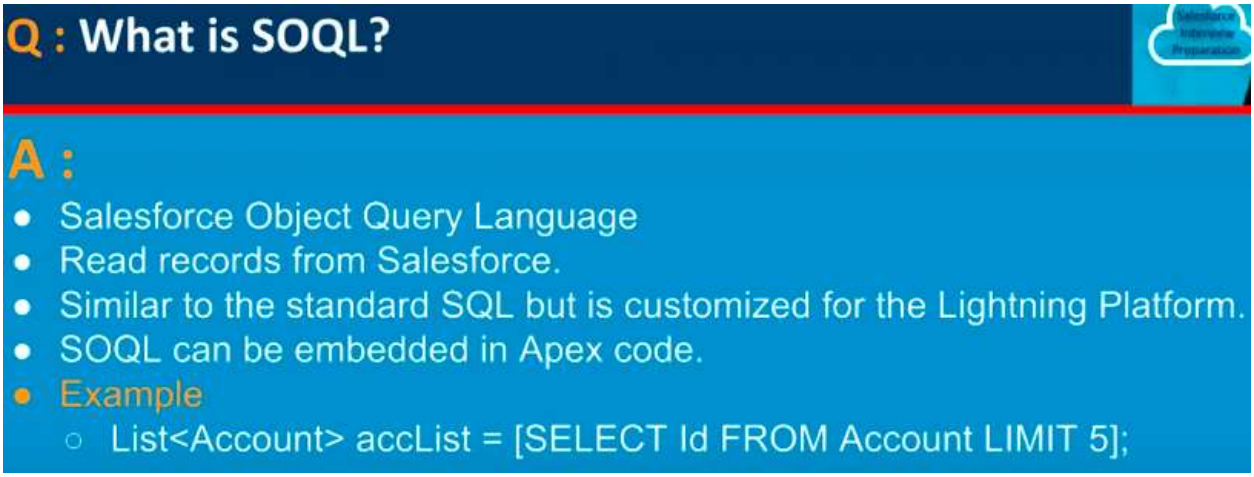
9.


Q : Map Examples in Apex?
A :
1. Map<String, String> strToStrMap = new Map<String, String> ();
2. Map<Integer, String> intToStrMap = new Map<Integer, String>();
i. intToStrMap.put(1, 'First');
ii. intToStrMap.put(2, 'Second');
iii. Boolean b = intToStrMap.containsKey(1);
iv. String value = intToStrMap.get(2);
v. Set<Integer> s = intToStrMap.keySet();
3. Map <ID, Account> IdToAccountMap = New Map<ID, Account>();
4. List<Account> accList = [SELECT ID, Name FROM Account];
5. Map <ID, Account> IdToAccountMap = New Map<ID, Account>(accList);
6. Map<Account, List<Contact>> accToConMap = new Map<Account, List<Contact>>( );

10.


Q : for loop and for each loop?
A :
• for
for(Integer i = 0; i<5; i++){
//write some code here
}
• for each
List<Account> accList = [SELECT Id FROM Account LIMIT 5];
for(Account acc : accList){
//write some code here
}

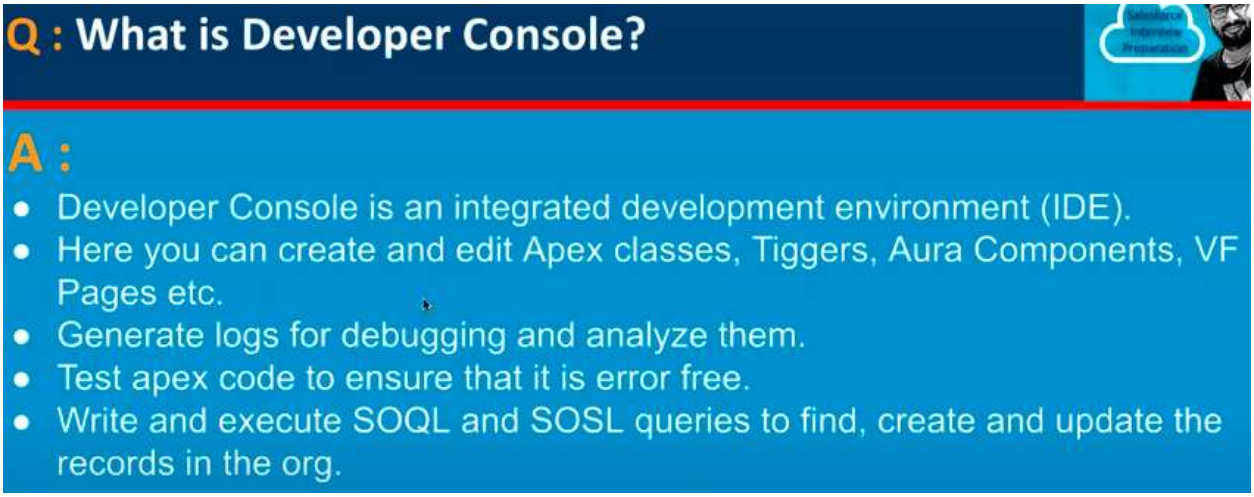
11.


Q : What is SOQL?
A :
Salesforce Object Query Language
Read records from Salesforce.
Similar to the standard SQL but is customized for the Lightning Platform.
SOQL can be embedded in Apex code.
Example
List<Account> accList = [SELECT Id FROM Account LIMIT 5];

12.


Q : What is SOSL?
A :
Salesforce Object Search Language (SOSL)
Used to perform text search in records.
Use SOSL to search fields across multiple sObjects records.
Use SOQL to retrieve records for a single object whereas use SOSL to search fields across multiple objects.
Syntax
Find 'SearchQuery' [IN SearchGroup] [RETURNING ObjectsAndFields];

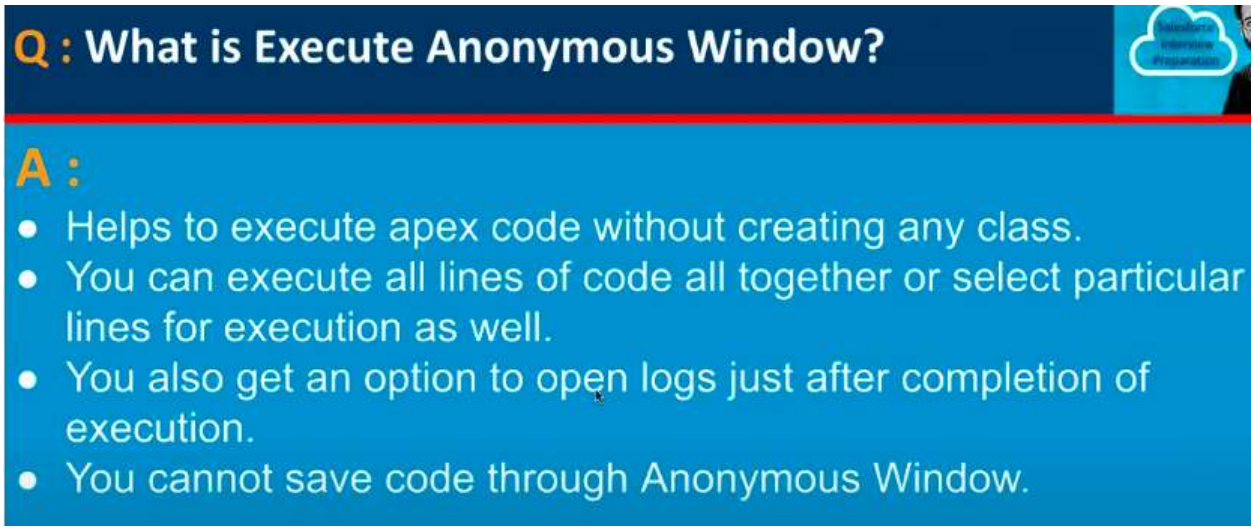
13.


Q : What is Developer Console?
A :
• Developer Console is an integrated development environment (IDE).
• Here you can create and edit Apex classes, Tiggers, Aura Components, VF Pages etc.
• Generate logs for debugging and analyze them.
• Test apex code to ensure that it is error free.
• Write and execute SOQL and SOSL queries to find, create and update the records in the org.

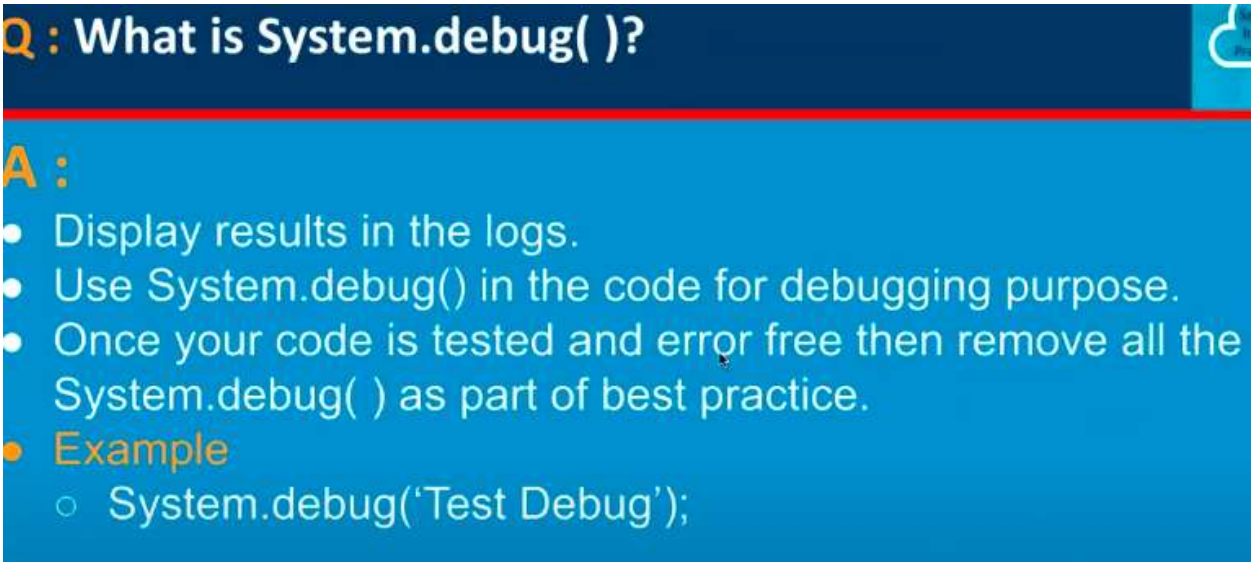
14.


Q : What is Query Editor in Developer Console?
A :
• Helps to execute SOQL and SOSL.

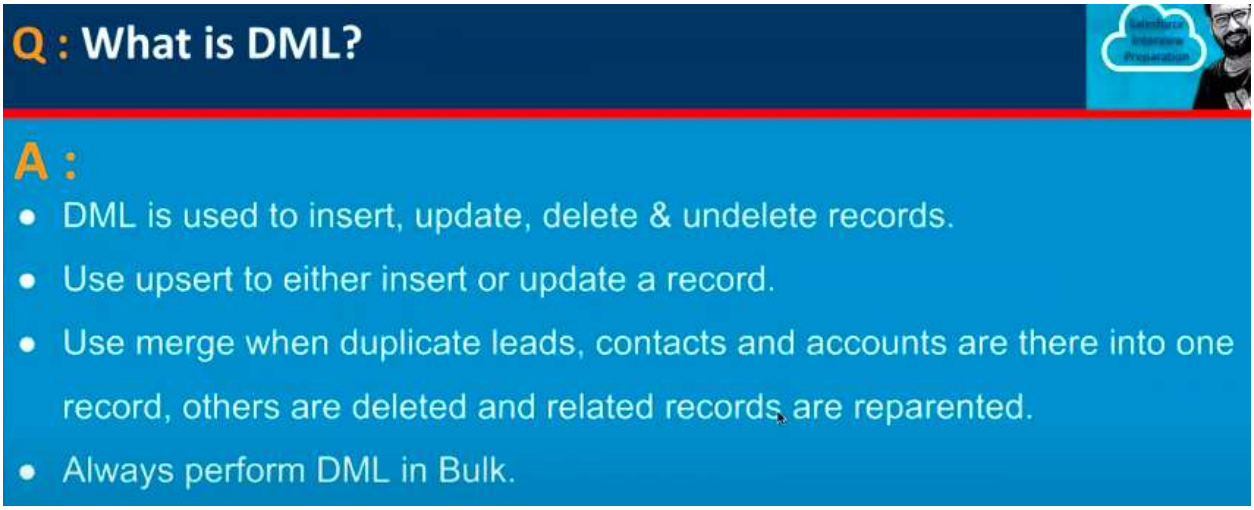
15.


Q : What is Execute Anonymous Window?
A :
• Helps to execute apex code without creating any class.
• You can execute all lines of code all together or select particular lines for execution as well.
• You also get an option to open logs just after completion of execution.
• You cannot save code through Anonymous Window.

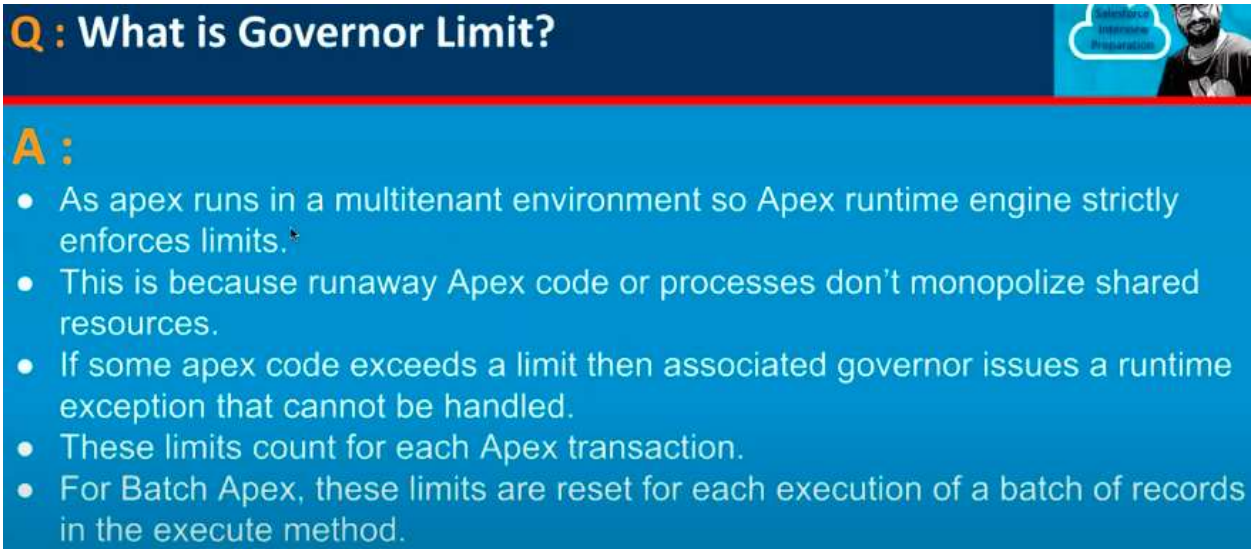
16.


Q : What is System.debug( )?
A :
Display results in the logs.
Use System.debug() in the code for debugging purpose.
Once your code is tested and error free then remove all the System.debug( ) as part of best practice.
Example
System.debug('Test Debug');

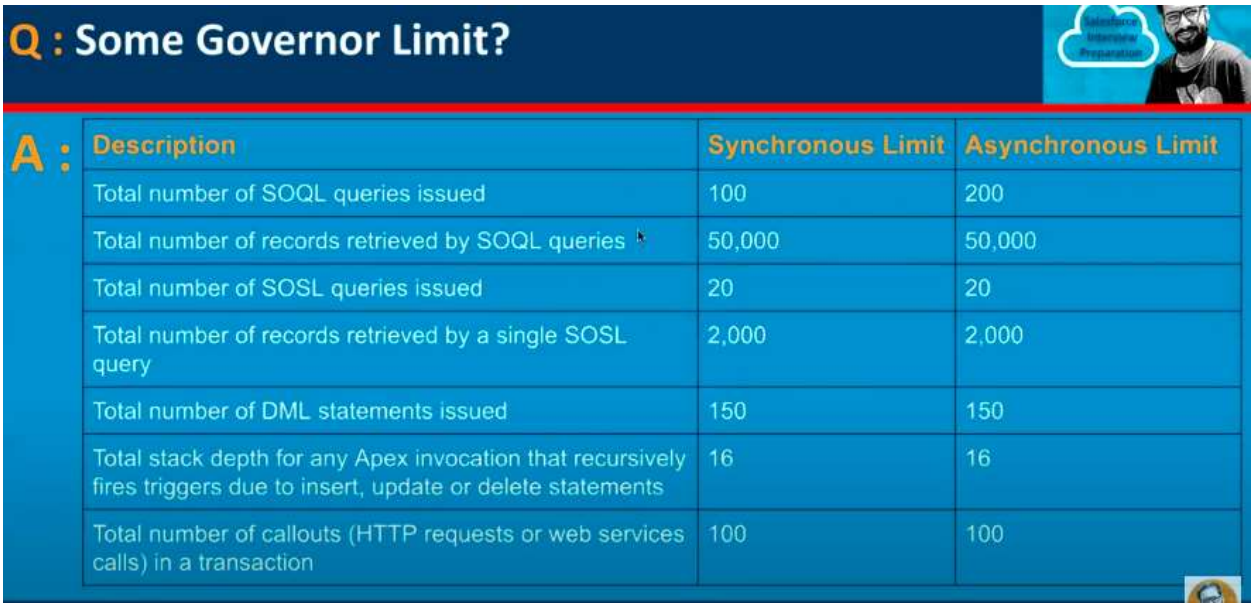
17.


Q : What is DML?
A :
DML is used to insert, update, delete & undelete records.
Use upsert to either insert or update a record.
Use merge when duplicate leads, contacts and accounts are there into one record, others are deleted and related records are reparented.
Always perform DML in Bulk.

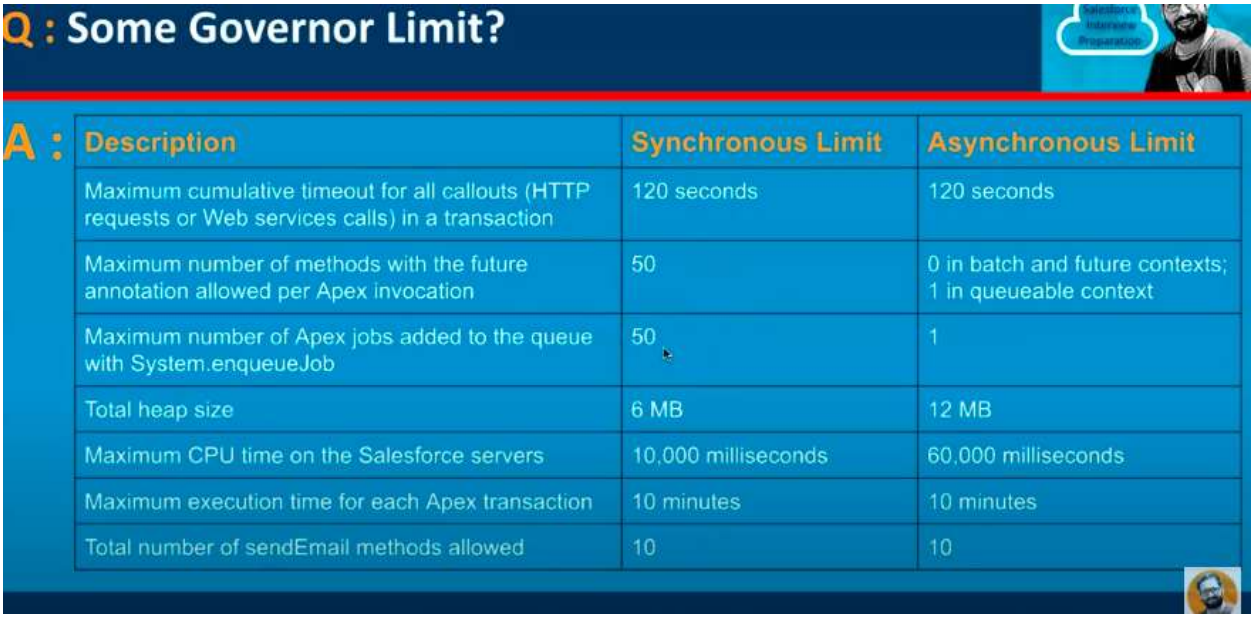
18.


Q : What is Governor Limit?
A :
• As apex runs in a multitenant environment so Apex runtime engine strictly enforces limits.
• This is because runaway Apex code or processes don't monopolize shared resources.
• If some apex code exceeds a limit then associated governor issues a runtime exception that cannot be handled.
• These limits count for each Apex transaction.
• For Batch Apex, these limits are reset for each execution of a batch of records in the execute method.

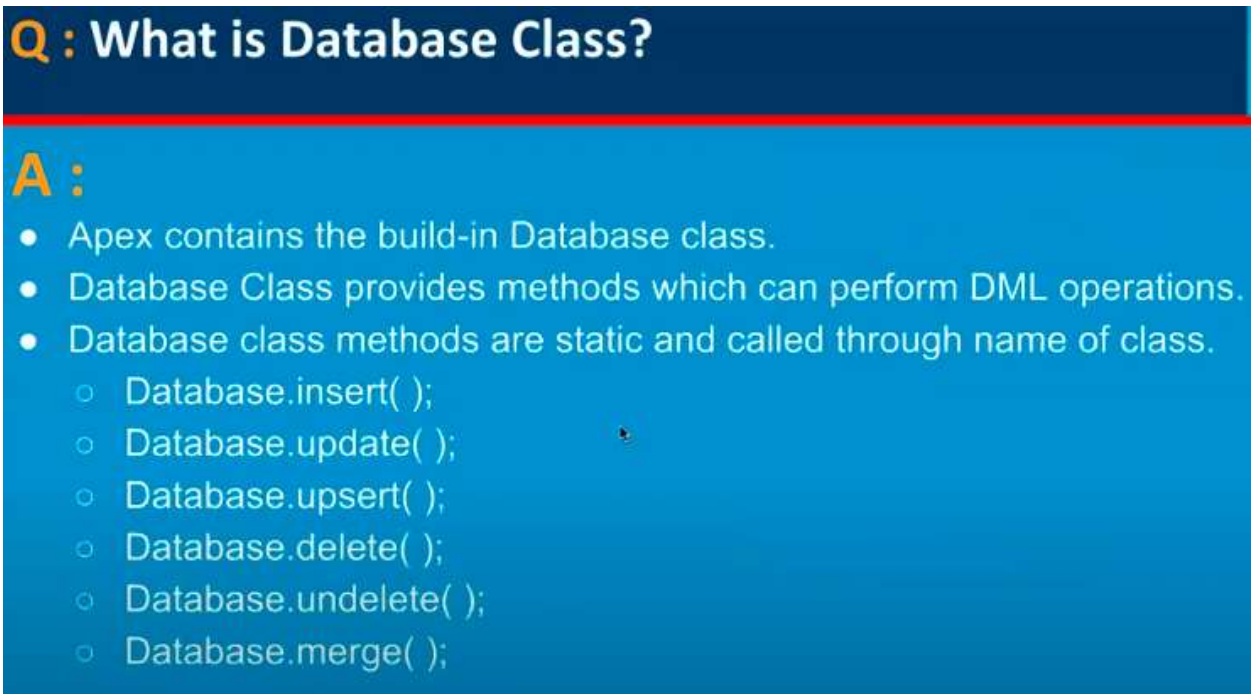
19.


Q : Some Governor Limit?
A :
Description
Synchronous Limit
Asynchronous Limit
Total number of SOQL queries issued
100
200
Total number of records retrieved by SOQL queries
50,000
50,000
Total number of SOSL queries issued
20
20
Total number of records retrieved by a single SOSL query
2,000
2,000
Total number of DML statements issued
150
150
Total stack depth for any Apex invocation that recursively fires triggers due to insert, update or delete statements
16
16
Total number of callouts (HTTP requests or web services calls) in a transaction
100
100

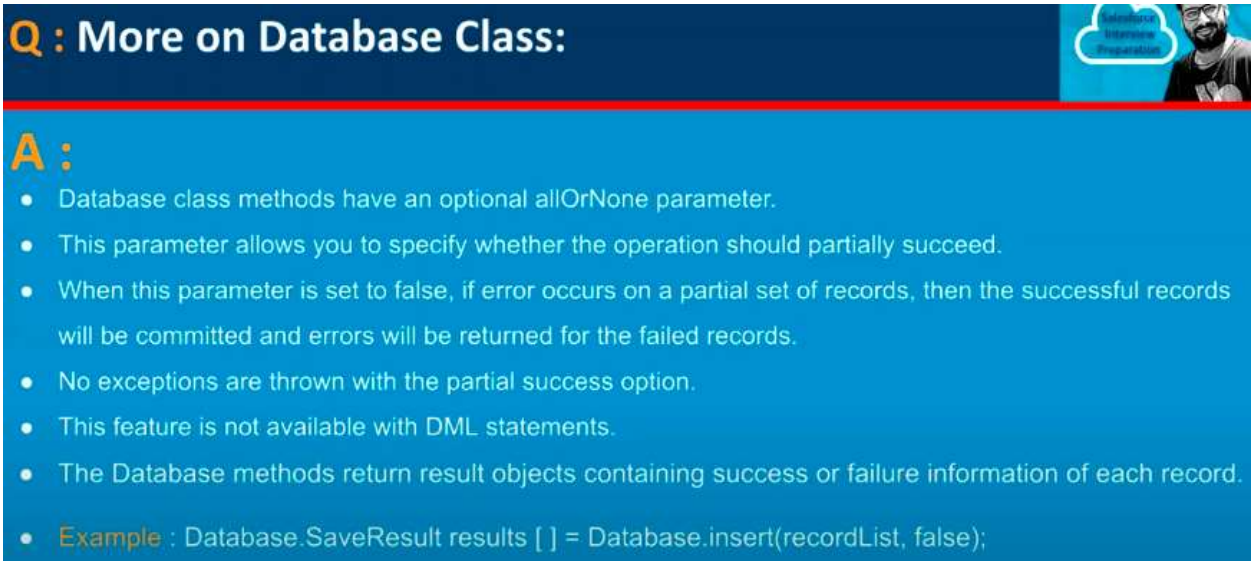
20.


Q : Some Governor Limit?
A :
Description
Synchronous Limit
Asynchronous Limit
Maximum cumulative timeout for all callouts (HTTP requests or Web services calls) in a transaction
120 seconds
120 seconds
Maximum number of methods with the future annotation allowed per Apex invocation
50
0 in batch and future contexts; 1 in queueable context
Maximum number of Apex jobs added to the queue with System.enqueueJob
50
1
Total heap size
6 MB
12 MB
Maximum CPU time on the Salesforce servers
10,000 milliseconds
60,000 milliseconds
Maximum execution time for each Apex transaction
10 minutes
10 minutes
Total number of sendEmail methods allowed
10
10

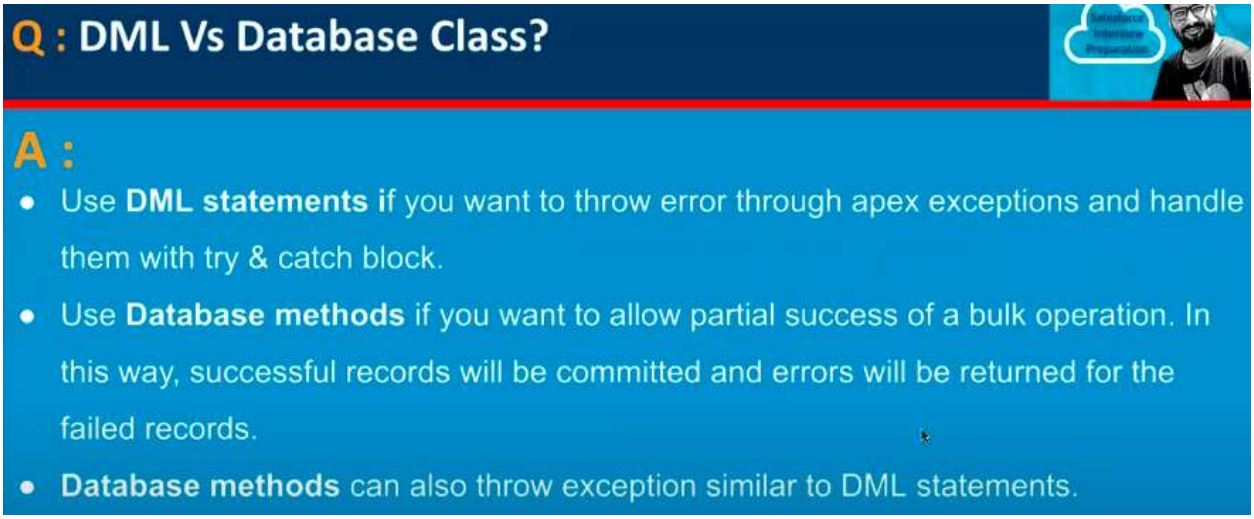
21.


Q : What is Database Class?
A :
Apex contains the build-in Database class.
Database Class provides methods which can perform DML operations.
Database class methods are static and called through name of class.
Database.insert( );
Database.update( );
Database.upsert( );
Database.delete( );
Database.undelete( );
Database.merge( );

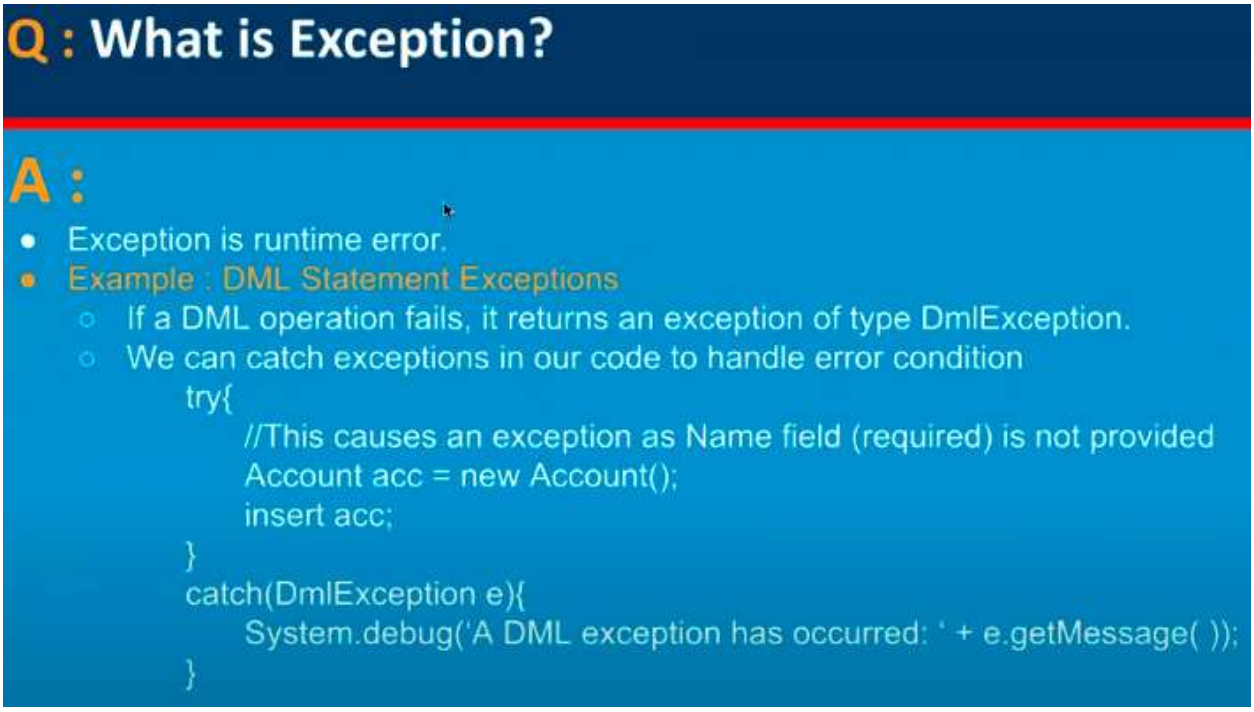
22.


Q : More on Database Class:
A :
Database class methods have an optional allOrNone parameter.
This parameter allows you to specify whether the operation should partially succeed.
When this parameter is set to false, if error occurs on a partial set of records, then the successful records will be committed and errors will be returned for the failed records.
No exceptions are thrown with the partial success option.
This feature is not available with DML statements.
The Database methods return result objects containing success or failure information of each record.
Example : Database.SaveResult results [ ] = Database.insert(recordList, false);

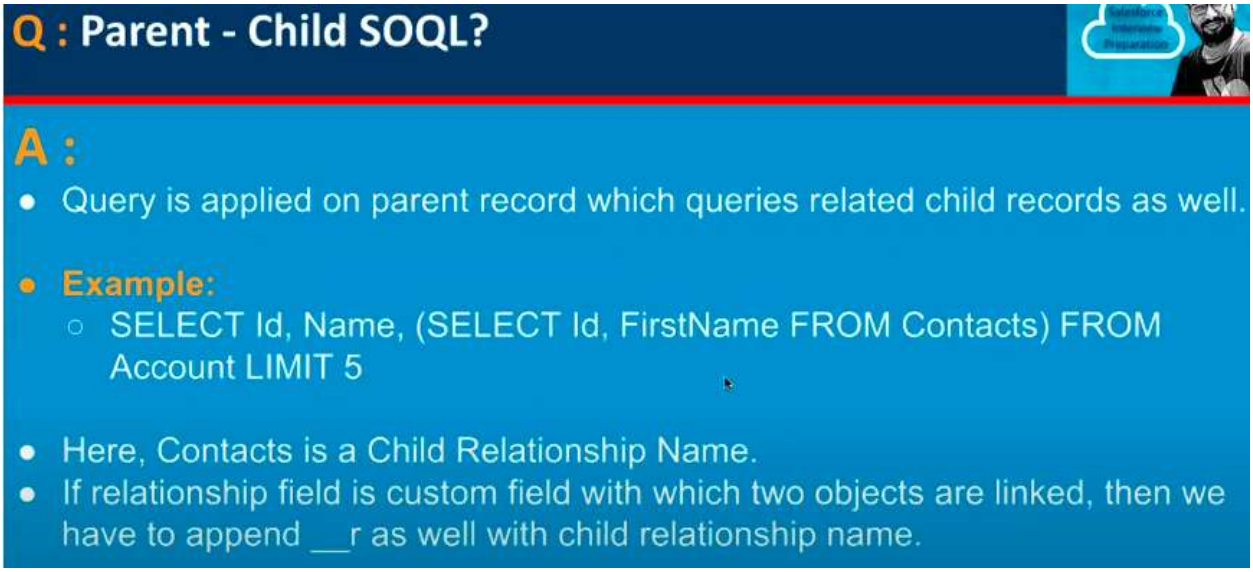
23.


Q : DML Vs Database Class?
A :
Use DML statements if you want to throw error through apex exceptions and handle them with try & catch block.
Use Database methods if you want to allow partial success of a bulk operation. In this way, successful records will be committed and errors will be returned for the failed records.
Database methods can also throw exception similar to DML statements.

24.


Q : What is Exception?
A :
Exception is runtime error.
Example : DML Statement Exceptions
If a DML operation fails, it returns an exception of type DmlException.
We can catch exceptions in our code to handle error condition
try{
//This causes an exception as Name field (required) is not provided
Account acc = new Account();
insert acc;
}
catch(DmlException e){
System.debug('A DML exception has occurred: ' + e.getMessage( ));
}

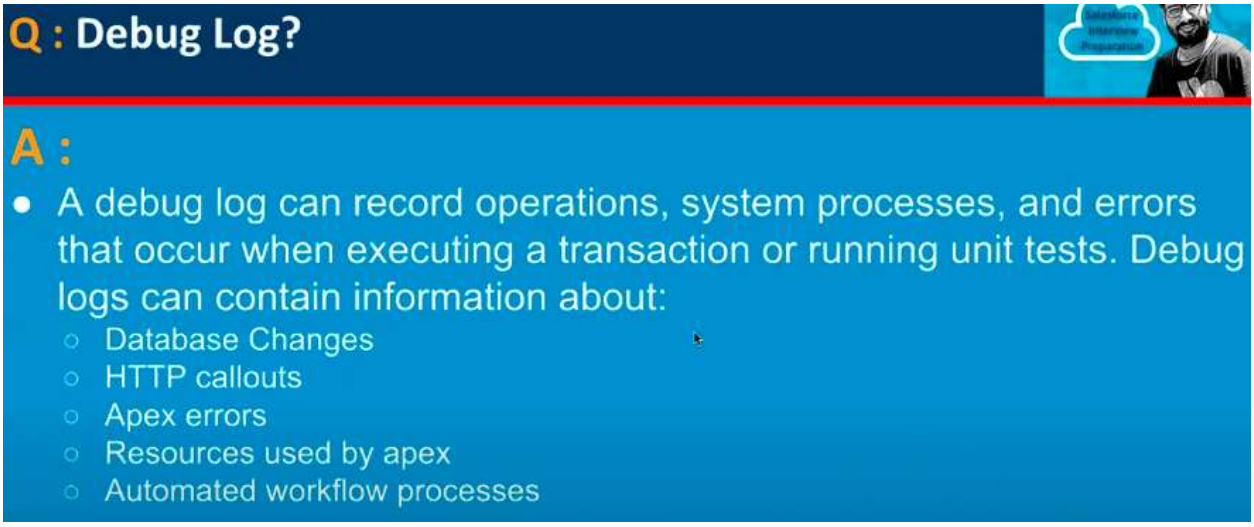
25.


Q : Parent - Child SOQL?
A :
Query is applied on parent record which queries related child records as well.
Example:
SELECT Id, Name, (SELECT Id, FirstName FROM Contacts) FROM Account LIMIT 5
Here, Contacts is a Child Relationship Name.
If relationship field is custom field with which two objects are linked, then we have to append __r as well with child relationship name.

26.


Q : Child - Parent SOQL?
A :
• Query is applied on child object which queries parent details as well.
• Example
○ SELECT Id, FirstName, Account.Name, Employee__r.Salary__c FROM Contact LIMIT 5
• Here, Account is representing standard relationship where as Employee is representing custom relationship. So in custom relationship append __r.

27.


Q : Debug Log?
A :
• A debug log can record operations, system processes, and errors that occur when executing a transaction or running unit tests. Debug logs can contain information about:
○ Database Changes
○ HTTP callouts
○ Apex errors
○ Resources used by apex
○ Automated workflow processes