

Salesforce Trigger Interview Scenarios

1) Account trigger to set account Rating as 'Hot' if Account Industry is 'Banking' or 'HealthCare'

```
trigger AccountTriggerHot on Account (before insert)
{
    if(Trigger.isBefore && Trigger.isInsert)
    {
        for(Account acc: Trigger.New)
        {
            if(acc.Industry == 'Banking' || acc.Industry == 'HealthCare')
            {
                acc.Rating = 'Hot';
            }
        }
    }
}
```

2) trigger on Contact which will Create Account record Whenever contact is created without an Account

```
trigger ContactTriggerAccountIdNull on Contact (after insert)
{
    List<Account> acclist = new list<Account>();
    if(Trigger.isAfter && Trigger.isInsert)
    {
        for(Contact con: Trigger.New)
        {
            if(con.AccountId == null)
            {
                Account acc = new Account();
                acc.Name = con.LastName;
                acc.Phone = con.Phone;
                acclist.add(acc);
            }
        }
    }
    if(!acclist.isEmpty())
    {
        insert acclist;
    }
}
```

```

    }
  }
}

```

3) Create a trigger on Contact Object, to make the email and phone fields should be mandatory upon creating or updating a contact record

```

trigger ContactTriggerErrorMessage on Contact (before insert,before Update)
{
  if(Trigger.isBefore && Trigger.isInsert || Trigger.isUpdate)
  {
    for(Contact con:Trigger.New)
    {
      if(con.Email==Null || con.Email =="")
      {
        con.Email.addError('please enter the email ');
      }
      if(con.Phone==Null || con.Phone =="")
      {
        con.Phone.addError('please enter the PhoneNumber');
      }
    }
  }
}

```

4) Create a Trigger in Account Object,To Assign the AnnualRevenue value as Below

Industry Name AnnualRevenue

Banking -----> 50000;
Finance ----->40000;
Insurance----->35000;
HealthCare----->25000;
Else ----->5000;

```

trigger AccountTriggerAnnualRevenue on Account (before insert,before Update)
{
  if(Trigger.isBefore && Trigger.isInsert || Trigger.isUpdate)
  {
    for(Account acc:Trigger.New)

```

```

{
    if(acc.Industry == 'Banking')
    {
        acc.AnnualRevenue = 50000;
    }

    else if(acc.Industry == 'Finance')
    {
        acc.AnnualRevenue = 40000;
    }
    else if(acc.Industry == 'Insurance')
    {
        acc.AnnualRevenue = 35000;
    }
    else if(acc.Industry == 'HealthCare')
    {
        acc.AnnualRevenue = 30000;
    }
    else
    {
        acc.AnnualRevenue = 5000;
    }
}
}
}

```

5) Create a trigger to Make Sure Each Contact Should be get Associated with an Account

```

trigger ContactTriggerWithAccount on Contact (before insert, before update)
{
    if(Trigger.isBefore && Trigger.isInsert || Trigger.isUpdate)
    {
        for(Contact con:Trigger.New)
        {
            if(con.AccountId == Null || con.AccountId == "")
            {
                con.addError('please select to the Account Id Field');
            }
        }
    }
}

```

```
}
```

6) Create a Trigger to prevent the Deletion an Activate Account Record

```
trigger AccountTriggerActiveDelete on Account (before Delete)
{
    if(Trigger.isBefore && Trigger.isDelete)
    {
        for(Account acc:Trigger.old)
        {
            if(acc.Active__c =='Yes')
            {
                acc.addError('you can't delete to the Active Account');
            }
        }
    }
}
```

7) Write an ApexTrigger To Prevent The Duplicate Account Records Inside The Object

```
trigger AccountTriggerDuplicateRecord on Account (before insert,before Update)
{
    if(Trigger.isBefore && Trigger.isInsert || Trigger.isUpdate)
    {
        for(Account acc: Trigger.New)
        {
            Integer Recordcount =[select count() From Account Where
Name=:acc.Name];
            if(Recordcount>0)
            {
                acc.addError('This Record Is Already insert in Account object');
            }
        }
    }
}
```

8) Write a Trigger to Prevent The Deletion of the Associated Contacts Upon Remove The AccountRecord

```
trigger AccountTriggerAccountIdDelete on Account (before Delete)
{
    if (Trigger.isBefore && Trigger.isDelete)
    {
        List<Contact> conlist=[Select AccountId From Contact Where AccountId
IN:trigger.oldMap.keySet()];
        if(!conlist.isEmpty())
        {
            for(Contact con:conlist)
            {
                con.AccountId=null;
            }
        }
        update conlist;
    }
}
```

9) Write an apexTrigger to Synchronize the Contact Records phone and Fax value with the Associated Account Record Phone And Fax Numbers

```
trigger AccountTriggerSynchronizeFields on Account (after update)
{
    if (Trigger.isAfter && Trigger.isupdate)
    {
        List<Account> acclist = [Select id,name,Phone,Fax,(Select
Id,LastName,Phone,Fax From Contacts) From Account Where Id
IN:Trigger.NewMap.KeySet()];
        List<Contact> conlist = new List<contact>();

        if(!acclist.isEmpty())
        {
            for(Account acc:acclist)
            {
                if(!acc.contacts.isEmpty())
                {
                    for(Contact con:acc.contacts)
                    {
                        con.phone=acc.phone;
                        con.Fax=acc.Fax;
                    }
                }
            }
        }
    }
}
```

```

        conlist.add(con);
    }
}
}
}
if(!conlist.isEmpty())
{
    Update conlist;
}
}
}

```

10) Upon Account Creation if Industry Is Not Null And Having Values As 'Insurance' Then populated Rating as Hot

```

trigger AccountTriggerIndusrtyNull on Account (before insert,before Update)
{
    if(Trigger.isBefore && Trigger.isInsert || Trigger.isUpdate)
    {
        for(Account acc :Trigger.New)
        {
            if(acc.Industry !=Null && acc.Industry=='Insurance')
            {
                acc.Rating = 'Hot';
            }
        }
    }
}

```

11)Upon Opportunity Creation IF Amount is Not Null And Is GreaterThan 100000 Then populated 'HotOpportunity' In Description Field

```

trigger OpportunityTriggerAmountFiled on Opportunity (before insert)
{
    if(Trigger.isBefore && Trigger.isInsert)
    {
        for(Opportunity opp:Trigger.New)

```

```
{
  if(opp.Amount!=NULL && opp.Amount>100000)
  {
    opp.Description = 'HotOpportunity';
  }
}
}
```