

3. Write a trigger and test class when an account is created , insert an contact related to the account with account name as contact first name.

Object : Account

Trigger AccountTrigger on Account (After **Insert**)

```
{  
  
    if(trigger.isAfter && trigger.isInsert)  
    {  
        AccountTriggerHandler.handleAfterInsert(trigger.new);  
    }  
}
```

SWIPE THE PAGE TO SEE HANDLER CLASS....



```

Public class AccountTriggerHandler
{

    Public static void handleAfterInsert(list<account>Acclist)
    {
        list<contact>conlist = new list<contact>();

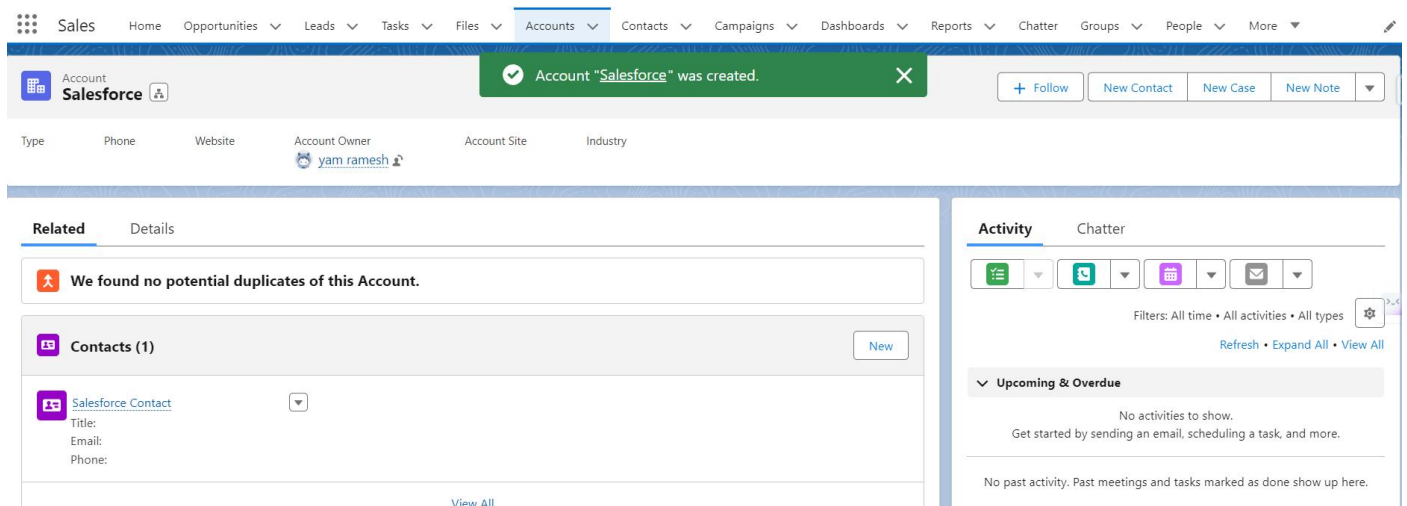
        for(Account A : Acclist)

        {
            Contact C = new Contact();
            C.FirstName = A.Name;
            C.LastName = 'Contact';
            C.AccountId = A.Id;
            conlist.add(C);
        }

        if(!conlist.isEmpty())
        {
            Insert conlist;
        }
    }
}

```

RESULT:



The screenshot displays the Salesforce user interface. At the top, a navigation bar includes tabs for Sales, Home, Opportunities, Leads, Tasks, Files, Accounts (selected), Contacts, Campaigns, Dashboards, Reports, Chatter, Groups, People, and More. Below the navigation bar, a green success message states: "Account 'Salesforce' was created." To the right of this message are buttons for "+ Follow", "New Contact", "New Case", and "New Note".

The main content area shows the details of the "Salesforce" account. It includes fields for Type, Phone, Website, Account Owner (yam ramesh), Account Site, and Industry. Below these fields, there are two tabs: "Related" and "Details". The "Related" tab is active, showing a message: "We found no potential duplicates of this Account." Below this message, there is a section titled "Contacts (1)" with a "New" button. Under "Contacts (1)", there is a single contact entry: "Salesforce Contact" with fields for Title, Email, and Phone.

On the right side of the page, there is an "Activity" section. It includes a "Chatter" tab and a "Filters" section with options for "All time", "All activities", and "All types". Below the filters, there is a section titled "Upcoming & Overdue" which states: "No activities to show. Get started by sending an email, scheduling a task, and more." At the bottom of the activity section, it says: "No past activity. Past meetings and tasks marked as done show up here."

TEST CLASS:

@isTest

Public class AccountTriggerTest

{

@isTest

public static void handleAfterInsert()

{

Account A = new Account();

A.Name = 'Test';

Test.startTest();

Insert A;

Test.stopTest();

list<Account>Accs = [select id ,name ,(select FirstName,LastName from
Contacts)from Account where Name ='Test'];

system.assertEquals(1,Accs[0].Contacts.size());

for(Account Ac :Accs)

{

for(Contact C : Ac.contacts)

{

system.assertEquals('Test',C.FirstName);

}

}

}

}

RESULT:

```
AccountTriggerHandler.apxc | AccountTrigger.apxt | AccountTriggerTest.apxc
Code Coverage: All Tests 100% | API Version: 60

1  Public class AccountTriggerHandler
2  {
3      Public static void handleBeforeInsert(list<account>Acclist)
4      {
13
14     Public static void handleAfterInsert(list<account>Acclist)
15     {
16         list<contact>conlist = new list<contact>();
17
18         for(Account A : Acclist)
19
20     {
21         Contact C = new Contact();
22         C.FirstName = A.Name;
23         C.LastName = 'Contact';
24         C.AccountId = A.Id;
25         conlist.add(C);
26     }
27
28     if(!conlist.isEmpty())
29     {
30         Insert conlist;
31     }
```

```
AccountTriggerHandler.apxc | AccountTrigger.apxt | AccountTriggerTest.apxc
Code Coverage: AccountTriggerTest.handleAfterInsert 100% | API Version: 60

1  Trigger AccountTrigger on Account (Before Insert,After Insert)
2  {
3      if(Trigger.isBefore && Trigger.isInsert)
4      {
5          AccountTriggerHandler.handleBeforeInsert(Trigger.New);
6      }
7
8      if(trigger.isAfter && trigger.isInsert)
9      {
10         AccountTriggerHandler.handleAfterInsert(trigger.new);
11     }
12 }
```