40

Apex Scenarios

(Test Your Apex Skills)

Find Solutions

@

▶/SanjayGuptaTechSchool

Or

https://studysalesforce.com

Helping Trailblazers learn Salesforce & build career in Salesforce Ecosystem

Apex

Implement Solutions for below Use Cases either by directly typing code in Anonymous Window or by creating Methods in Apex Class.

1. Query 5 Account records into a list. Now iterate on a list and create a set of account Ids and display values of the set.
2. Query 5 Account records into a list. Now iterate on a list and create a map where key will be account Id and value will Account sObject (using loop) and display values of map.
3. Query 5 Account records into a list. Now iterate on a list and create a map where key will be account Id and value will Account sObject (without using loop) and display values of map.
4. Query 5 Account records into a list. Now create a set of Account ID and Map <Id, Account> using a list. Now iterate on set and check whether set values match with map key, if yes then display map value.
5. Query This/Last Week created Accounts with related contacts.
   [Hint: Parent-Child SOQL]
6. Query This/Last Week created contacts with related account Name and Phone.
   [Hint: Child-Parent SOQL]
7. Query Account (Id, Name, Phone) with Related Opportunities (Name, CloseDate, Stage) where Account Phone != null. Sort the records in both Asc or Desc Order based on CreatedDate.
8. Query Account along with related contacts where Account phone!=null. Now display results through system.debug( ) in following ways:

   Account Name has **#ofcontacts** related.

9. Query Account along with related opportunities where Account phone!=null. Now display results through system.debug( ) in following ways:

   Account Name has **TotalAmountOnRelatedOpp** worth Opportunities.

10. Create 2 Account records with two separate insert statements. Then create 2 Account records with a single insert DML statement using List.

11. Insert 5 Accounts by populating Name, Phone, Type, Annual Revenue and Industry Field Values. **[Hint: Use Loop]**

12. Insert 5 Opportunities with different Opportunity Names, CloseDate = Today’s Date and Stage = ‘Prospecting’. **[Hint: Use Loop]**

13. Insert an Account along with 1 related Contact.

14. Insert an Account along with 1 related Contact and 1 related Opportunity.

15. Insert 5 Accounts with at least 1 related Opportunity. **[Hint: Use Loop and List]**

16. Insert 5 Accounts and a Contact & a Opportunity related to each account. **[Hint: Use Loop and List]**

17. Query Accounts where Phone =’123456’ and update Phone to ’654321’.

18. Query Contacts where Email is null and update their email with ‘welcome@studysalesforce.com.

19. Query Accounts where Phone =654321 and update Phone to 123456. Also create 5 new account records. **[Hint : Use Upsert Statement]**

20. Query Opportunities where created date is last month and updated their stage to closed won. Also create 5 new opportunities. **[Hint: Use Upsert Statement]**

21. Delete account records where Phone =’123456’. Verify whether records are deleted through Recycle Bin.

22. Undelete account records deleted in previous step. Check recycle bin is not having those records and they will be under the accounts tab in the app.

23. Insert 200 Account records and your code should not hit the governor limit. Once records are successfully created then just delete those 200 records only.

24. Delete opportunities which are closed lost and created last month.

25. Query Accounts created last week. Now Insert one opportunity under each account.

26. Query Account created in the last 7 days. Now update their phone number with ‘1234567890’.

27. Query Opportunities created in the last 7 days. If the opportunity stage is closed won then update description as ‘Opportunity is Closed Won’, in case of closed lost update description as ‘Opportunity is Closed Lost’, otherwise ‘Opportunity is Open’.

28. Query Accounts created in the last one month. Populated ’Total Contacts’ Field on Account by counting number of contacts associated with Account. If no contact then populate 0.

29. Query Contacts created last week and if Mobile phone is not populated on contact then copy Phone of Related Account. **[Hint: Use Child-Parent SOQL]**

30. Create a Custom Object named “Employee”. Create Name, Salary, Phone, Lookup (to Account) fields on Employee Object. Create a Total Salary field on Account. Now Query Accounts created this year along with related Employees. Populate total salary of all related employees on account. **[Hint: Use Parent-Child SOQL]**

31. Query Employee records created this year and if the Phone field is not populated on the Employee record then copy the Phone of Related Account record. **[Hint: Use Child-Parent SOQL]**

32. Create a lookup to Employee on Contact object. Query Contacts created last week and if Mobile phone is not populated on contact then copy Phone of Related Employee. **[Hint: Employee is Parent and Contact is Child, Use Child-Parent SOQL]**

Asynchronous Apex

1. Create a future method to count the number of contacts associated with an account. List of Account Ids will be passed to the future method. Ensure proper code coverage as well.

2. Create a future method to count how many cases are created through Phone, Web, Email. You need to put the counted value on account in 3 different fields. Ensure proper code coverage as well.

3. Write a batch apex to populate Account’s Billing Address on related Contact’s Mailing address. Ensure Proper code coverage as well. [Note: In test class of batch apex you cannot pass batch size because it runs for one batch only.]

4. Create a batch apex that updates all Cases where Priority is blank to Low. Ensure proper code coverage as well.

5. Populate Parent Account on accounts where Billing State = ‘NY’ using Queueable Apex. Ensure proper code coverage as well.

6. Create an Queueable Apex class that inserts the same Contact record for each account Industry = ‘Media’. Ensure proper code coverage as well.

7. Create an apex class that uses schedule apex to create a reminder task on opportunity where opportunity is open and close date is less than today. Ensure proper code coverage as well.

8. Create an apex class that uses scheduled apex to schedule a batch that updates Contact’s Mailing address with Account’s Billing Address. Ensure proper code coverage as well.