Trigger Scenarios (Practice Questions)

1. Create a trigger that updates the "Last Modified Date" field of an Account record whenever a related Contact record is updated.

2. Develop a trigger that prevents the deletion of an Opportunity record if its Stage is set to "Closed Won."

3. Write a trigger that automatically converts a Lead to a Contact and an Account when the Lead is marked as "Qualified."

4. Implement a trigger that sends an email notification to the Opportunity owner when the Opportunity amount exceeds $10,000.

5. Design a trigger that calculates the total amount of all related Opportunity records and updates a custom field on the Account record accordingly.

6. Develop a trigger that assigns a default Owner to a newly created Case record based on certain criteria.

7. Create a trigger that updates a custom field on the Contact record with the number of related Opportunity records.

8. Develop a trigger that prevents the creation of a Task record if the associated Account does not have an assigned Owner.

9. Write a trigger that automatically assigns a specific Record Type to newly created Opportunity records based on their Stage value.

10. Implement a trigger that updates the Close Date of an Opportunity to 30 days after the Created Date if it is not already set.

11. Create a trigger that updates a custom field on the Account object whenever a related Contact is inserted or updated.

12. Write a trigger that prevents users from deleting an Account if it has related Opportunities.

13. Develop a trigger that automatically sets the Close Date of an Opportunity to 30 days after the Created Date when the Stage is set to "Prospecting."

14. Implement a trigger that sends an email alert to the Opportunity Owner when an Opportunity reaches the "Closed Won" stage.

15. Create a trigger that assigns a default Owner to a Lead if no Owner is specified when the Lead is created.

16. Write a trigger that updates a custom field on the Case object based on the value of a related Account's Industry field.

17. Develop a trigger that calculates the total value of all related Opportunity Line Items and updates a custom field on the Opportunity accordingly.

18. Implement a trigger that sets the Account Rating to "Hot" if the total value of all related Opportunities is greater than $100,000.

19. Create a trigger that logs a custom object whenever a Contact's Mailing Address is updated.

20. Write a trigger that automatically creates a Follow-Up Task whenever a Task with a specific subject is completed.

21. Write a trigger to create task to the Account owner when an Case is Created related to an account.

22. Write a trigger to update Total Opportunity Amount associated with account when an opportunity is ‘Updated’ or ‘Deleted’ or ‘Undeleted’.

23. Write a trigger to update Total Opportunity Amount associated with account when an opportunity is “Inserted"

24. Write a trigger to create contact related to account with Same phone as account phone when an Create Contact checkbox is checked in Account record page.

25. Write a trigger when a lead is created create a task to the lead owner to “Follow Up the Lead”.

26. Write a trigger when an opportunity is “Closed Won”, create a task for opportunity owner to split the revenue among the team with high priority.

27. Write a trigger to Update Number of contacts filed in Account when Contact is Updated or Deleted or Undeleted.

28. Write a trigger when a contact is inserted update the count of contacts associated with an account.

29. Write a trigger when a contact description is updated then update its parent account description as same.

30. Write a trigger when an account inserts and CopyBillingToShipping (Custom Field) picklist is equal to Yes then automatically copy account billing address into account shipping address.

31. Write a trigger to “Update” related contact’s phone field when Account’s phone gets updated.

32. Write a trigger to update Account’s Name and Description when it is restored from recycle bin with Account name + Restored and for description Account has been restored.

33. Write a trigger to delete opportunity when account gets deleted.

34. Write a trigger to “Update” Opportunities Phone when Account phone is updated and set Opportunities phone as Account phone value.

35. Write a trigger to insert Contact when an Account is Inserted and populated Contact first name as Account name.

36. Write a trigger to prevent Account deletion, if the Account is Active (‘Yes’). If Account is not Active(‘No’) the deletion is done.

37. Write a trigger if account phone is updated, Update the description with old phone and new phone?

38. Write a trigger, when an account is inserted with type "Customer-Direct" Populate Customer priority as "High".

39. Write a trigger on Account, when an account is inserted, automatically account billing address should populate into the account shipping address.

40. Write a trigger on create the account with the same name i.e Preventing the users to create Duplicate Accounts.

41. Write a trigger on if the current user profile is system admin then only the user can insert the record

42. Write a trigger on prevent account from deletion if account having contact more than 2 contacts

43. Write a trigger on prevent account from deletion if account having contacts write a trigger

44. Write a trigger on opportunity if stage is closed own then u cannot update that update

45. Write a trigger on Prevent account from deletion if account having parent account.

46. Whenever a case is created with origin as email then set status as new and Priority as Medium.

47. Whenever Lead is created with Lead Source as Web then give rating as cold otherwise hot.

48. Whenever New Account Record is created then we need to create associated Contact Record automatically.

49. Write a trigger, to have the count of total number of contacts associated to an account.

50. Write a trigger, whenever a new active User having profile “System Administrator” is inserted, add the user to the public group “Admins”.

51. Apex trigger that validates the Phone field of a Lead record to ensure it follows a specific format.

52. Apex trigger that validates the Email field of a Lead record to ensure it is in a valid format.

53. Write a trigger on contact, throw the error when the related account have the type = industry.

Error - Select Different type.

54. Write a trigger that updates the Total Amount field on the Account object to reflect the sum of the Amount field of all related Opportunities.

55. Develop a trigger that calculates and updates the Total Number of Cases custom field on the Account object whenever a new Case related to the Account is created or deleted.

56. Create a trigger that updates a custom field on the Account object to reflect the count of related Contacts.

57. Implement a trigger that calculates and updates the Total Revenue custom field on the Account object based on the sum of the Amount field of related Opportunity Line Items.

58. Write a trigger that updates the Total Number of Contacts custom field on the Opportunity object to reflect the count of related Contacts.

59. Develop a trigger that updates a custom field on the Contact object to indicate the total number of Opportunities related to that Contact.

60. Create a trigger that calculates and updates the Total Amount of Opportunities custom field on the Account object, considering only Opportunities in the "Closed Won" stage.

61. Implement a trigger that updates the Total Number of Open Cases custom field on the Account object whenever a Case related to the Account is closed or reopened.

62. Write a trigger that updates a custom field on the Opportunity object to reflect the count of related Opportunity Line Items.

63. Develop a trigger that updates a custom field on the Account object to indicate the total value of Opportunities with a Stage of "Proposal" or "Negotiation."

64. When we are trying to insert new record into object. If there is any record existing with same account name it should prevent duplicate record.

65. Write a trigger to prefix Account Name with ‘Mr’ when new record is inserted.

66. Whenever a new record is created into account object . Before this new record is inserted into Account, delete all the contacts records with this account name.

67. Whenever a new transaction is performed successfully then update the customer object balance field based on

If Transaction Type=Deposit, Balance= balance +amount ; withdraw balance= balance-amount; Note: Customers and Transaction has lookup Detail Relation.

68. Whenever a new contact is created for account update the corresponding account phone with the new contact phone field.

69. Whenever customer record is updated, before updating the record create new record in test object with old values of customer record.

70. To update the owner of a case based on the values selected within a picklist and populate the owner field with the created by field data. When we have selected any Field name=Status Picklist Values= priced-(Initial) Priced-(Re-priced) Price(File Loaded)

71. Write a trigger that will prevent a user from creating a lead that already exists as a contact. We will use the lead /contact email address to detect duplicates.

Lead is created or updated.

1. Lead has an email address.

2. Try to find a matching contact based on email address.(Using SOQL)

3. If a match is found give an error

4. If a match is not found do nothing.

72. When we are trying to delete customer record delete all the corresponding child records from test object, where customer is lookup field in the object.

73. When we create the opportunity with probability =50% then the opportunity owner will be automatically added to Account Team of the associated account for the opportunity.

74. Invoking the apex class from the trigger. Whenevr new customers record is created

/updated then income tax should be calculated based on salary and should be updated to field income tax(Currency field).

75. Before we insert a new record in the customer object calculate the income tax field value based on salary field value and then insert.

If we delete any of the existing customer record then first create new test object record with customer record values then delete customer record.

76. Whenever we try to update the phone of account record then update the related contact phone number with the new Account phone number before account record is updated.

When we delete the account record then delete the corresponding contact records.

77. Suppose there is scenario where one trigger perform update operation, which results in invocation of second trigger and the update operation in second trigger acts as triggering criteria for trigger one.

78. Write a trigger which shows recursive trigger error.

79. Create 'Sales Rep' field with datatype (Text) on the account object. When we create account record, the account owner will be automatically added to Sales Rep field. When we update the Account owner of the record, then also the Sales Rep Will be automatically updated.

80. Create the field Called 'Contact Relationship' checkbox on the contact object and create the object called "Contact Relationship" which is related list to the contact.(Lookup Relationship).

Now logic is when we create Contact by checking Contact Relationship checkbox then contact relationship will be created automatically for that contact.

81. When we change the owner of the Contact Relationship, then the owner name will be automatically populated in the Contact Relationship name field.

82. Create the field Called 'Contact Relationship' checkbox on the contact object and create the object called "Contact Relationship" which is related list to the contact.(Lookup Relationship).

When we delete the contact then contact Relationship will be deleted automatically.

83. Create the field Called 'Contact Relationship' checkbox on the contact object and create the object called "Contact Relationship" which is related list to the contact.(Lookup Relationship).

When we undelete the contact then contact Relationship will be Undeleted automatically.

84. Create field Called 'Count of Contacts' on Account object. When we add the contacts for that Account then count will populate in the field on Account details page. When we delete the contacts for that account then count will update automatically.

85. Write a trigger that updates a custom field on Object A whenever a new record is created on Object B, and the related record on Object C meets specific criteria.

86. Develop a trigger that creates a new record on Object C when a record on Object A is updated, and the related record on Object B meets certain conditions.

87. Create a trigger that updates a field on Object A based on the sum of a numeric field from related records on Object B, and the related records on Object C meet specific criteria.

88. Write a trigger that updates a field on Object B whenever a new record is created on Object A, and the related record on Object C is in a particular status.

89. Develop a trigger that deletes related records on Object C when records on Object A and Object B are deleted simultaneously.

90. Implement a trigger that updates a field on Object C based on the values of fields from related records on Object A and Object B.

91. Create a trigger that sends an email notification when a record on Object A is updated, and related records on Object B and Object C meet certain conditions.

92. Write a trigger that updates a field on Object A based on the average value of a numeric field from related records on Object B, and the related records on Object C match specific criteria.

93. Develop a trigger that creates a new record on Object B whenever a record is deleted from Object A, and related records on Object C meet certain conditions.

94. Implement a trigger that updates a field on Object B when records on Object A and Object C are updated simultaneously and meet specific criteria.

95. Write a trigger that updates a custom field on the Account object based on the sum of a numeric field on related Opportunity Line Items.

96. Develop a trigger that calculates and updates a custom field on the Contact object to reflect the total value of related Opportunities that are in the "Closed Won" stage.

97. Create a trigger that assigns a default Owner to a Lead based on the Lead's country and the region specified in a custom object.

98. Implement a trigger that updates a custom field on the Case object to indicate the total number of related Tasks and Events.

99. Write a trigger that updates a custom field on the Opportunity object to reflect the count of related Opportunity Line Items.

100. Develop a trigger that calculates and updates a custom field on the Account object to indicate the total value of Opportunities related to that Account.

101. Create a trigger that updates a custom field on the Contact object based on the count of related Cases and the status of those Cases.

102. Implement a trigger that calculates and updates a custom field on the Account object to indicate the total number of Contacts associated with that Account.

103. Write a trigger that updates a custom field on the Opportunity object to reflect the count of related Contacts.

104. Develop a trigger that updates a custom field on the Account object based on the count of related Opportunities in specific stages.