

OBJECT RELATIONSHIPS

Storing the whole data related to the business inside a single object is not viable due to the following reasons.

1. Number of columns limitation in the database object.
2. It increases the redundancy of data in the table.
3. It occupies more storage space to store the redundant data.
4. It causes the performance issues.
5. Managing the records will be difficult.

To avoid the above problems, we use "Normalization" to split the data from one single table to multiple tables without any data loss.

Normalization can be achieved with the help of "Relationships" in Salesforce.

Using relationships, we can map the association between the objects i.e., mapping the association between the records inside the objects.

While mapping the relationship between the objects, follow the below 2 steps.

Step 1: Identify the parent object and child objects.

Step 2: Map the relationship between the objects i.e., create a field inside the child object by selecting the relationship datatype which references the parent object.

Salesforce provides the below Relationship Datatypes.

1. Lookup Relationship
2. Master-Detail Relationship
3. External Lookup Relationship
4. Hierarchical Relationship

Lookup Relationship

Use Case: Configure the lookup relationship between "Hiring Manager" and "Position" objects and list out all the Observations.

Step 1: Identify the Parent Object and Child Object.

Hiring Manager ----> Parent Object.

Position Object----> Child Object.

Step 2: Map the Lookup Relationship between the Objects i.e., create a new field inside the Position Object with "Lookup Relationship" Datatype, which references the "Hiring Manager" object as the Parent.

Observations

1. Lookup Relationship provides One-Many association between the objects, i.e., a Parent record can be associated with one/more Child Records.
2. Lookup Relationship can be applicable even though both Parent and Child object contains the pre-existing records.
3. While creating the child record, lookup field is an optional field, i.e., child record may/may not be associated with the Parent Record.
4. Re-Parenting option is available by default in Lookup Relationship, i.e., once a child record has been created by selecting a parent, we can change the parent of the child at any point of time.
5. While removing the Parent record, only parent record will get removed from the object. All the related Childs will exist as it is. In Child record, lookup field value will get cleared out.
6. We can make the Lookup Field required by selecting "Required Checkbox", upon creating/editing the field settings.
7. If the Lookup Field is mandatory field, if we try to remove a parent record which is associated with few child records inside the object, Salesforce won't allow to remove the parent record from the object.

Sol 1: Remove all the child records and then remove the parent record.

Sol 2: Re-Parent the child records to another parent record and then remove the parent record from the object.

8. Sharing Settings and Security Settings of the child record is independent of the parent record.
9. An object can have maximum of 40 Lookup Relationship fields.
10. Roll-up Summary Fields cannot be applicable on the Lookup Relationship.
11. We can make the standard object as the child and custom object as the parent.

Hierarchical Relationship

Hierarchical association allows us to map the association of the object with itself, i.e., both parent records and child records exist in the same object.

Hierarchical Relationship will be applicable for only "User Object" in Salesforce, so that we can map the association of one user with another user based on the role.

Master-Detail Relationship:

Use Case: Create a Custom Object "Candidate" by adding the below fields.

1. Candidate Name : (Standard field)
2. Location : Picklist

(LOV's: Hyderabad, Bangalore, Chennai, Mumbai, Pune,etc.)

3. Qualification : Picklist

(LOV's: B.Sc, BCA, B.Com, M.Sc, MCA, B.E, B.Tech,....etc.)

4. Years Of Experience : Picklist

(LOV's: 0 - 20)

5. Current CTC : Currency (Max: 7)

7. Expected CTC : Currency (Max:7) (Mandatory)

8. Current Employer : Text

9. Contact Number : Phone (Mandatory)

10. Email ID : Email (Mandatory)

11. PAN Number : Text (Max: 10) (Mandatory)

12. Passport Number : Text (Max:10)

13. Technical Skills : Picklist-MultiSelect.

14. Reference ID : CID-900001. (Auto Number)

15. Notice Period : Picklist

(LOV's: Immediate, 1 Week, 15 Days, 1 Month, 2 Months, 3 Months)

Use Case: Configure the Master-Detail Relationship between "Position" and "Candidate" objects and list out all the Observations.

Step 1: Identify the Parent Object and Child Object.

Parent Object: Position Object

Child Object : Candidate Object.

Step 2: Map the Master-Detail Relationship between the Objects, i.e., create a "Master-Detail Relationship" field inside the Candidate Object, which references the "Position" object as the Parent Object.

Observations

1. Master-Detail Association provides One-Many Relationship between the objects by default, i.e., a Parent record can be associated with one/more child records.

2. Master-Detail Relationship can't be applicable between the Objects if the child object contains pre-existing Records.

Sol 1: Remove all the Child Records from the Child Object Permanently and then Map Master-Detail Relationship.

Sol 2: Map the Lookup Relationship between the Objects and make sure that each child record is associated with the Parent. Once all the Child Records are associated with the Parent then convert the Relationship datatype from "Lookup to Master-Detail".

3. While creating child record, Lookup Field is a required field, i.e., each child record should be associated with a Parent.

4. Re-Parenting option is not available by default. We must enable this option by selecting the Checkbox "Allow Re-Parenting" upon creating/editing the field settings.
5. While removing the Parent Record, all the related child records will be removed automatically, i.e., parent record contains "Owner Field" but child record won't have Owner field.
6. Sharing Settings and Security Settings of the child record is purely dependent on parent record.
7. Roll-up Summary fields can be applicable on "Master-Detail Relationship" objects.
8. We can't make a Standard Object as child with the Custom Object as the Parent.
9. An Object can have maximum of two Master-Detail relationship fields.

Converting Lookup to Master-Detail relationship

Converting Lookup to Master-Detail means, we have to Change the Datatype of the Relationship field from "Lookup Relationship to Master-Detail Relationship".

Pre-Condition: While converting the association to "Master-Detail", make sure each child record should be associated with a Parent Record.

Click on the Child Object's Tab.

1. Click on "Setup" menu.
2. Click on "Edit Object" link.
3. Click on "Fields & Relationships" link from Left Panel.
4. Click on "Lookup Relationship" field name.
5. Click on "Edit" button.
6. Click on "Change Field Type" button.
7. Select the datatype as "Master-Detail".
8. Click on "Next" button.
9. Click on "Save" button.

Observation: It will convert the association from Lookup to Master-Detail.

Converting Master-Detail to Lookup

Converting the relationship datatype from master-detail to lookup is straightforward.

Many-Many Relationship

Book Object (ISBN#, Book Title, Unit Price)

Author Object (Author Id, Author Name, Location)

Standard Associations

1. Account and Contact: Between Account and Contact Objects we have Lookup Relationship.

Account Object----> Parent Object

Contact Object----> Child Object

Relationship Field ----> Contact: AccountID (which contains the Account Record Id to which the contact record gets associated)

While removing the Account Record, all the related Contact Records will be removed automatically by default, which is an additional behavior given by Salesforce based on the "CRM Users Request".

We can break this additional behavior by using "Apex Triggers".

2. Account and Opportunity: Between Account and Opportunity objects, we have the Lookup Relationship.

Account Object-----> Parent Object

Opportunity Object-----> Child Object.

Relationship Field-----> Opportunity: AccountID (which contains the Account Record Id to which the Opportunity gets associated)

1. While removing the account record, all the associated opportunity records will get removed automatically by default.

2. Roll-up Summary fields can be applicable between Account and Opportunity Objects.

3. Account and Case: Between Account and Case Objects as have Lookup Relationship.

Account Object ----> Parent Object

Case Object ----> Child Object

Relationship Field ----> Case: AccountID (which contains the Account Record Id to which the Case Record gets associated)

While removing the account record, if the account contains any related case records, we cannot remove the account from the object.

Roll-up Summary Field

Roll-up Summary fields are the Read-only System-Generated fields, whose value will be generated by the Salesforce.

Roll-up Summary field will summarize the child records into the various smaller groups based on the Parent Record Id and will perform the operations on the groups and will represent the result on the Parent Record.

Always Roll-up Summary field can be applicable on the "Parent Object" in the Master-Detail Associated Objects.

Salesforce supports the below 4 Roll-up Types.

1. Count(): It will Count the number of records inside the Group.

2. Sum(<FieldName>): It will Sumup all the values inside the specified field.

Supported Input Fields: Number Type / Currency / Percent.

3. Min(<FieldName>): It will find out the least value inside the specified field.

Supported Input Fields: Number Type / Currency / Percent / Date /Time / Date Time.

4. Max(<FieldName>): It will find out the highest value inside the specified field.

Supported Input Fields: Number Type / Currency / Percent / Date /Time / Date Time.

Limit: An Object can have maximum of 25 Roll-up Summary fields.

Use Case: Configure a Roll-up Summary field to count the Number of Candidates applied for each Position Record.

Object Name : Position Object.

Summarized Object : Candidate Object.

Roll Up Type: Count.

Use Case: Configure a Roll-up Summary field on the Position object to Count the Immediate Joiners for each Position.

Object Name : Position Object.

Summarized Object: Candidate Object

Roll-up Type: Count.

Filter Condition: Candidate: NoticePeriod == 'Immediate'

Use Case: Configure a Roll-up Summary field on Account Object which represents the Sum of all the Related Opportunity Records Amount value.

Object Name : Account Object

Summarized Object : Opportunity

Roll-up Type : Sum (Amount)

One-One Relationship

In this relationship, parent record should be always associated with only one child record inside the child object.

One-One Association can be achieved with the help of

1. Roll-up Summary Field

2. Validation Rule

Use Case: Configure One-One association between "Position and Candidate Objects".

Step 1: Create a Roll-Up Summary field on the Position Object, which counts the number of Related Candidates.

Step 2: Create a Validation Rule on the Position Object, which restricts the "Roll-Up Summary Field Value" should not exceed "1".

Click on "Positions" Tab.

1. Click on "Setup" menu.
2. Click on "Edit Object" link.
3. Click on "Validation Rules" link from Left Panel.
4. Click on "New" button.
5. Enter the Validation Rule Name, and Label.
6. Select the Checkbox "Active", to make the Rule Active.
7. Enter the Validation Condition inside the "Formula Editor".

Ex: Number_Of_Candidates__c > 1

8. Enter the Error Message to be visible if the Conditions are Satisfied.
9. Select the Error Location as "Top of the Page".
10. Click on "Save" button.

Use Case: Configure an Automation Process on account to make sure an account can have maximum of 5 Opportunity Records.

Standard Junction Objects

1. Campaign and Lead Object: Between Campaign and Lead Objects we have Many-Many Relationship.

Parent Objects---> Campaign and Lead

Junction Object ---> Campaign Member (Child)

2. Lead and Contact Object: Between Lead and Contact Object we have Many-Many Relationship.

Parent Objects---> Lead and Contact

Junction Object ---> Campaign Member (Child)

3. Contact and Opportunity: Between Contact and Opportunity we have Many-Many Relationship.

Parent Objects---> Contact and Opportunity

Junction Object ---> OpportunityContactRole (Child)

4. Opportunity and Product: Between Opportunity and Product we have Many-Many relationship.

Parent Objects---> Opportunity and Product

Junction Object ---> OpportunityLineItem (Child)

5. User and Permission Set: Between User and Permission Set Objects, we have Many-Many Association.

Parent Objects---> User and Permission Set

Junction Object---> PermissionSetAssignment (Child)