

Create a salesforce custom app on Recruit app with at least 2 objects(Recruiter/Manager, Recruitee/Candidate).Each object should have at least 4 fields with required validation wherever applicable and each object should hold at least 5 records in it. Relate the objects with Master relationship.Also create trigger to assign candidates automatically under a particular Recruiter/Specialization/Domain after insertion.

Create an application having the following custom objects

recruiter :

The screenshot shows the Salesforce Object Manager interface for the 'Recruiter' object. On the left, a sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main area is titled 'Fields & Relationships' and displays six items, sorted by Field Label. The table has columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The data is as follows:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Domain	Domain_c	Picklist		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
pno	pno_c	Phone		
Recuriter Name	Name	Text(80)		

Add the same domain in the picklist the you are going to add it in job

Ex.

WD

AD

AL

ML

DSA

Job :

The screenshot shows the Salesforce Setup interface with the following details:

- Tab Bar:** LP_2_Chips_(1)(1).pdf, yash | Salesforce, Jobb | Salesforce, Developer Console, New Tab.
- Header:** Search Setup.
- Page:** SETUP > OBJECT MANAGER.
- Left Sidebar:** Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules.
- Table:** Fields & Relationships (5 items, Sorted by Field Label).

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Description	Description__c	Text Area(255)		
Jobb Name	Name	Text(80)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		
- Bottom:** 79°F Clear, Search, File Explorer, Taskbar icons (including Microsoft Edge, Google Chrome, Visual Studio Code, etc.), Date: 25-05-2023, Time: 01:28.

Candidate :

The screenshot shows the Salesforce Setup interface with the following details:

- Tab Bar:** LP_2_Chips_(1)(1).pdf, yash | Salesforce, Candidate | Salesforce, Developer Console, New Tab.
- Header:** Search Setup.
- Page:** SETUP > OBJECT MANAGER.
- Left Sidebar:** Details, Fields & Relationships (selected), Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, Compact Layouts, Field Sets, Object Limits, Record Types, Related Lookup Filters, Search Layouts, List View Button Layout, Restriction Rules, Scoping Rules.
- Table:** Fields & Relationships (7 items, Sorted by Field Label).

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Candidate Name	Name	Text(80)		
Created By	CreatedById	Lookup(User)		
Email	Email__c	Email		
Jobb	Jobb__c	Master-Detail(Jobb)		
Last Modified By	LastModifiedById	Lookup(User)		
PhoneNo	PhoneNo__c	Phone		
Recruiter name	Recruiter_name__c	Text(30)		
- Bottom:** 79°F Clear, Search, File Explorer, Taskbar icons (including Microsoft Edge, Google Chrome, Visual Studio Code, etc.), Date: 25-05-2023, Time: 01:27.

Trigger code :

```
trigger CandidateTrigger on Candidate__c (before insert) {

    //getting job id of candidate
    //here candidate.Job__c is Master relationship
    Id job_id;
    for(Candidate__c candidate : trigger.new){
        job_id = candidate.Job__c;
    }

    //getting job name using job id
    String job_name;
    for(Job__c job : [ SELECT Id,Name FROM Job__c]){
        if(job.Id == job_id){
            job_name = job.Name;
        }
    }

    //now assigning the name of recruiter to the candidate
    //by cheacking is the recruiter has same domain as the candidates job
    for(Recruiter__c recruiter : [ SELECT Id,Name ,Domain__c FROM
Recruiter__c]){
        if(recruiter.Domain__c == job_name){

            //if we found the recruiter with same domain as the candidates
            job then
                //we assign his name to name of Recruiter filed of candidates
                for(Candidate__c cand : trigger.new){
                    cand.Recruiter_name__c = recruiter.Name;
                }
        }
    }
}
```

Now add job and recruiters :

The screenshot shows a Microsoft Edge browser window with the address bar displaying `mmcoe70-dev-ed.lightning.force.com/lightning/o/Jobb_c/list?filterName=Recent`. The main content area is titled "Jobb Recently Viewed" and lists six items under "Job Name": AI, WD, AD, ML, DA, and DSA. The browser interface includes a top navigation bar with tabs for "Recently Viewed | Jobb | Salesfor", "Candidate | Salesforce", and "Developer Console". A search bar and various browser controls are visible at the top.

The screenshot shows a Microsoft Edge browser window with the address bar displaying `mmcoe70-dev-ed.lightning.force.com/lightning/o/Recruiter_c/list?filterName=Recent`. The main content area is titled "Recruiters Recently Viewed" and lists four items under "Recruiter Name": A, B, D, and C. The browser interface is identical to the previous screenshot, featuring a top navigation bar with tabs for "Recently Viewed | Recruiters | Salesfor", "Candidate | Salesforce", and "Developer Console". A search bar and various browser controls are visible at the top.

Now at the time of adding candidate keep the recruiter name field empty :

The screenshot shows the Salesforce Candidate Management interface. A new candidate record is being created. The 'Candidate Name' field contains 'yash'. The 'PhoneNo' field contains '1234567890'. The 'Jobb' field contains 'AI'. The 'Email' field contains 'yj@gmail.com'. The 'Recruiter name' field is empty. The status bar at the bottom shows it's 01:27 on 25-05-2023.

It will get added automatically :

The screenshot shows the Salesforce Candidate Management interface after the candidate has been saved. A green success message says 'Candidate "yash" was created.' The candidate details page is displayed, showing the 'Details' tab. The 'Candidate Name' field shows 'yash'. The 'PhoneNo' field shows '1234567890'. The 'Jobb' field shows 'AI'. The 'Recruiter name' field now displays 'B'. The 'Created By' field shows 'Yash Jadhav, 25/05/2023, 1:26 am'. The 'Last Modified By' field also shows 'Yash Jadhav, 25/05/2023, 1:26 am'. The status bar at the bottom shows it's 01:27 on 25-05-2023.

Congratulations on wasting your life in engineering 😊

Create a salesforce custom app on Project management app with at least 2-3 objects. Each object should have at least 4 fields with required validation wherever applicable and each object should hold at least 5 records in it. Relate the objects with Master relationship. Also create trigger to assign projects automatically under a particular team after insertion.

Create an application having the following custom objects

domain :

The screenshot shows the Salesforce Object Manager interface. The left sidebar is collapsed, and the main area displays the 'Fields & Relationships' section for the 'Domain' object. The table lists five fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Description	Description__c	Text Area(255)		
Domain Name	Name	Text(80)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		

team :

The screenshot shows the Salesforce Setup interface for the 'Team' object. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Field Sets. The main content area is titled 'Fields & Relationships' and displays a table of fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Domain	Domain_c	Master-Detail(Domain)		✓
Head	Head_c	Text(30)		
Last Modified By	LastModifiedBy	Lookup(User)		
No of team members	No_of_team_members_c	Number(18, 0)		
Team Name	Name	Text(80)		✓

Project :

The screenshot shows the Salesforce Setup interface for the 'Project' object. The left sidebar lists various setup categories. The main content area is titled 'Fields & Relationships' and displays a table of fields:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedBy	Lookup(User)		
Description	Description_c	Text Area(255)		
Domain	Domain_c	Master-Detail(Domain)		✓
End Date	End_Date_c	Date		
Last Modified By	LastModifiedBy	Lookup(User)		
Project Name	Name	Text(80)		✓
Team name	Team_name_c	Text(30)		

Trigger code :

```
trigger ProjectTrigger on Project__c (before insert) {  
  
    //getting project id of candidate  
    //here project.Domain__c is Master relationship  
    Id domain_id;  
    for(Project__c project : trigger.new){  
        domain_id = project.Domain__c;  
    }  
  
    //now assigning the team to the project  
    //by cheacking is the team has same domain as the project  
    for(Team__c team : [ SELECT Id,Name ,Domain__c FROM Team__c ]){  
        if(team.Domain__c == domain_id){  
  
            //if we found the team with same domain as the project then  
            //we assign teams name to name of team in project  
            for(Project__c project : trigger.new){  
                project.Team_name__c = team.Name;  
            }  
        }  
    }  
}
```

Now add domain and team :

Domains

Recently Viewed

5 items • Updated a few seconds ago

Domain Name
1 DA
2 ML
3 WD
4 AD
5 AI

Teams

Recently Viewed

4 items • Updated a few seconds ago

Team Name
1 T4
2 T3
3 T2
4 T1

Now at the time of adding project keep the team name field empty :

The screenshot shows the Salesforce Project Management interface. A 'New Project' dialog box is open, displaying fields for 'Project Name' (chatGPT), 'Description' (the god), 'End Date' (27/05/2023), 'Team name' (empty), and 'Domain' (AI). The 'Team name' field is highlighted with a yellow background.

It will get added automatically :

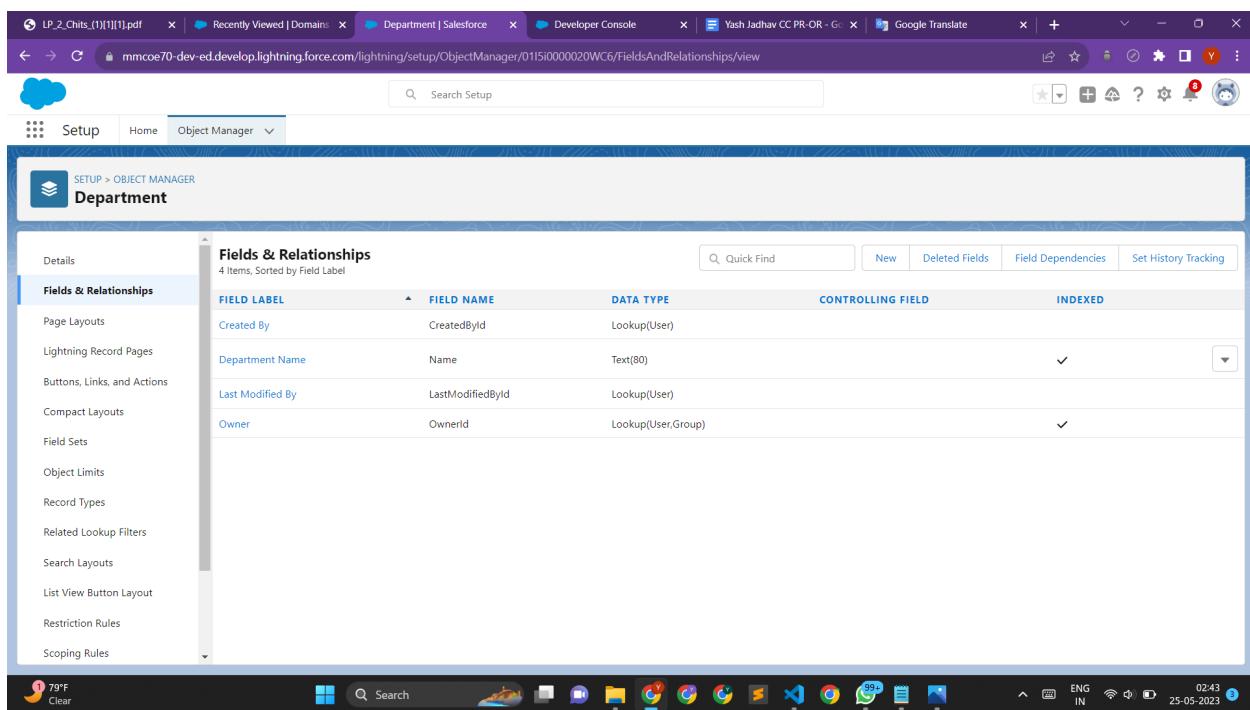
The screenshot shows the details of the newly created project 'chatGPT'. The 'Team name' field is now populated with 'T1'. A green success message at the top of the page states 'Project [chatGPT] was created.'

Congratulations on wasting your life in engineering 😊

Create a salesforce custom app on the School database with at least 2-3 objects(Teacher, student,Batch).Each object should have at least 4 fields with required validation wherever applicable and each object should hold at least 5 records in it. Relate the objects with Master relationship.Also create trigger to assign students automatically under a particular batch after insertion

Create an application having the following custom objects

Department :



The screenshot shows the Salesforce Object Manager interface for the 'Department' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, Buttons, Links, and Actions, etc. The main area displays the 'Fields & Relationships' section with the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Department Name	Name	Text(80)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓

Batch :

The screenshot shows the Salesforce Setup interface for the 'Batch' object. The left sidebar lists various setup categories like Page Layouts, Lightning Record Pages, and Field Sets. The main content area displays the 'Fields & Relationships' section for the 'Batch' object. It includes a table with columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table contains the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Batch Name	Name	Text(80)		✓
Created By	CreatedById	Lookup(User)		
Department	Department__c	Master-Detail(Department)		✓
From	From__c	Number(18, 0)		
Last Modified By	LastModifiedById	Lookup(User)		
To	To__c	Number(18, 0)		
Year	Year__c	Number(18, 0)		

Students :

The screenshot shows the Salesforce Setup interface for the 'Student' object. The left sidebar lists various setup categories. The main content area displays the 'Fields & Relationships' section for the 'Student' object. It includes a table with columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table contains the following data:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Batch	Batch__c	Text(30)		
Created By	CreatedById	Lookup(User)		
Department	Department__c	Master-Detail(Department)		✓
Last Modified By	LastModifiedById	Lookup(User)		
Roll no	Roll_no__c	Number(18, 0)		
student Name	Name	Text(80)		✓
Year	Year__c	Number(18, 0)		

Trigger code :

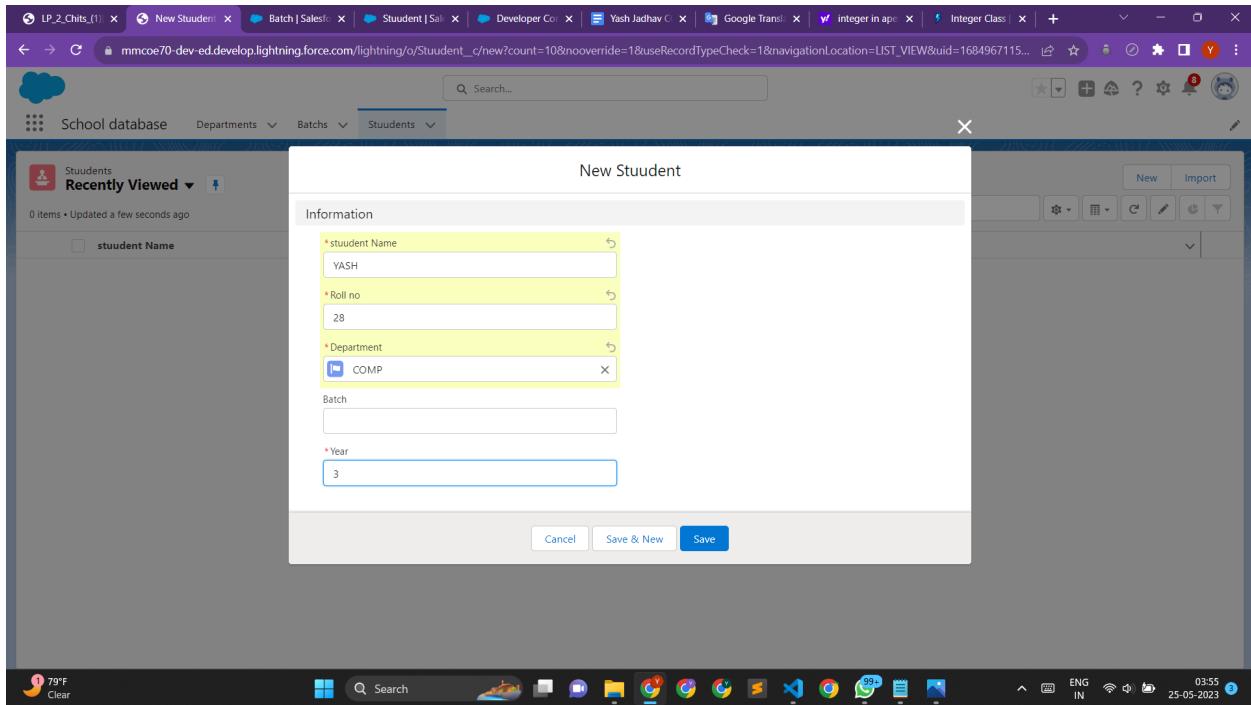
```
trigger StudentTrigger on Stuudent__c (before insert) {  
  
    Id department_id;  
    Decimal year;  
    Decimal rno;  
  
    for(Stuudent__c stud : trigger.new) {  
  
        department_id = stud.Department__c;  
        year = stud.Year__c;  
        rno = stud.Roll_no__c;  
  
    }  
  
    for(Batch__c batch : [ SELECT Id,Name ,From__c, To__c,  
Year__c,Department__c FROM Batch__c]) {  
  
        if(batch.Department__c == department_id && batch.Year__c == year  
&& ( (rno <= batch.To__c) && (rno >= batch.From__c) ) ){  
  
            for(Stuudent__c stud : trigger.new){  
                stud.Batch__c = batch.Name ;  
            }  
        }  
  
    }  
}
```

Now add department and batch :

The screenshot shows a web-based application interface for managing school data. At the top, there's a navigation bar with tabs for 'School database', 'Departments', 'Batches', and 'Students'. Below the navigation is a search bar with placeholder text 'Search...'. A toolbar on the right includes icons for 'New', 'Import', and 'Change Owner'. The main content area displays a 'Recently Viewed' list titled 'Departments'. The list shows four items: 1. COMP, 2. MECH, 3. ENTC, and 4. IT. Each item has a small checkbox icon to its left. A status message at the bottom left says '4 items • Updated a few seconds ago'. A toolbar at the bottom of the list includes 'Search this list...', 'New', 'Import', and 'Change Owner' buttons.

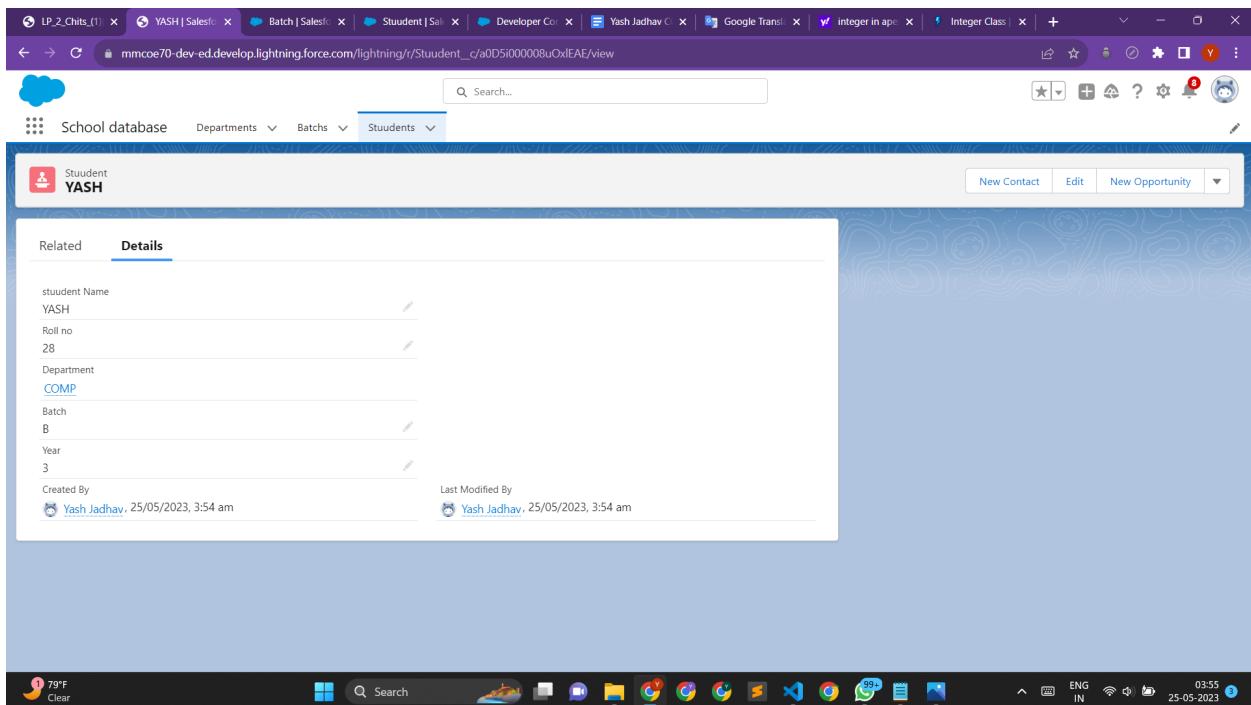
This screenshot shows the same application interface as the previous one, but the 'Batches' tab is active. The main content area displays a 'Recently Viewed' list titled 'Batches'. The list shows three items: 1. C, 2. B, and 3. A. Each item has a small checkbox icon to its left. A status message at the bottom left says '3 items • Updated a few seconds ago'. A toolbar at the bottom of the list includes 'Search this list...', 'New', 'Import', and 'Change Owner' buttons.

Now at the time of adding a student keep the batch field empty :



The screenshot shows the Salesforce Lightning interface for creating a new student. The 'Information' section contains fields for 'student Name' (YASH), 'Roll no' (28), 'Department' (COMP), 'Batch' (empty), and 'Year' (3). The 'Save' button is highlighted.

It will get added automatically :



The screenshot shows the Salesforce Lightning interface displaying the details of a student named YASH. The 'Details' tab is selected, showing fields for student Name (YASH), Roll no (28), Department (COMP), Batch (B), and Year (3). The 'Created By' field shows 'Yash Jadhav, 25/05/2023, 3:54 am'. The 'Last Modified By' field also shows 'Yash Jadhav, 25/05/2023, 3:54 am'.

Congratulations on wasting your life in engineering 😊

Create a salesforce custom app on maharashtra tourism app with at least 2-3 objects. Each object should have at least 4 fields with required validation wherever applicable and each object should hold at least 5 records in it. Relate the objects with the Master relationship. Also create trigger to assign the place automatically under a particular category after insertion

Create an application having the following custom objects

City :

The screenshot shows the Salesforce Object Manager interface. The top navigation bar includes tabs for 'Setup', 'Home', and 'Object Manager'. The main content area is titled 'City' under 'SETUP > OBJECT MANAGER'. On the left, a sidebar lists various setup options like 'Page Layouts', 'Lightning Record Pages', and 'Field Sets'. The right side displays the 'Fields & Relationships' section for the 'City' object. This section contains a table with columns: FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED. The table rows are:

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Category	Category__c	Text(30)		
City Name	Name	Text(80)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		

Place :

The screenshot shows the Salesforce Object Manager interface for the 'Place' object. The left sidebar lists various setup options like Page Layouts, Lightning Record Pages, and Field Sets. The main content area displays the 'Fields & Relationships' section, which lists seven fields: Category_of_palce_c, City__c, Created By, Description__c, Last Modified By, Name, and Rating__c. The table includes columns for FIELD LABEL, FIELD NAME, DATA TYPE, CONTROLLING FIELD, and INDEXED status.

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Category of palce	Category_of_palce_c	Text(30)		
City	City__c	Master-Detail(City)		✓
Created By	CreatedById	Lookup(User)		
Description	Description__c	Text(30)		
Last Modified By	LastModifiedById	Lookup(User)		
Place Name	Name	Text(80)		✓
Rating	Rating__c	Number(18, 0)		

Trigger code :

```
trigger PlaceTrigger on Place__c (before insert) {  
  
    Id city_id;  
    for(Place__c place : trigger.new){  
        city_id = place.City__c;  
    }  
  
    String category;  
    for(City__c city : [ SELECT Id,Category__c FROM City__c ]){  
        if(city.Id == city_id){  
            category = city.Category__c;  
        }  
    }  
  
    for(Place__c place : trigger.new){  
        place.Category_of_palce__c = category;  
    }  
}
```

Now add cities :

Screenshot of a web browser showing a Salesforce Lightning interface for managing cities. The page displays a 'Recently Viewed' list for the 'cities' category, showing three items: Lonavala, Aurangabad, and Mumbai. The browser tabs include 'LP_2_Chips_(1)(1).pdf', 'Place | Salesforce', 'Recently Viewed | cities | Salesforce', 'Developer Console', and 'list of cities - Yahoo India Search'. The system status bar at the bottom shows '90°F Sunny' and the date '25-05-2023'.

Now at the time of adding a place keep the category field empty :

Screenshot of a web browser showing a Salesforce Lightning interface for creating a new place. The 'New Place' form is open, showing fields for Place Name (Ajanta Caves), City (Aurangabad), Description (Ancient Buddhist caves), and Rating (4.7). The 'Category of place' field is empty. The browser tabs include 'LP_2_Chips_(1)(1).pdf', 'Place | Salesforce', 'New Place | Salesforce', 'Developer Console', and 'list of cities - Yahoo India Search'. The system status bar at the bottom shows '90°F Sunny' and the date '25-05-2023'.

It will get added automatically :

The screenshot shows a Salesforce Lightning interface for a 'Place' record. The top navigation bar includes tabs for 'LP_2_Chips.{1}|{1}.pdf', 'Place | Salesforce', 'Ajanta Caves | Salesforce', 'Developer Console', and 'list of cities - Yahoo India Search'. The main header displays 'Maharashtra Tourism' and navigation links for 'cities' and 'Places'. A search bar is present at the top right.

The main content area shows a green success message: 'Place "Ajanta Caves" was created.' Below this, the 'Details' tab is selected for the 'Ajanta Caves' place record. The record details are as follows:

- Place Name:** Ajanta Caves
- City:** Aurangabad
- Description:** Ancient Buddhist caves
- Rating:** 5
- Category of place:** Caves

On the left, there is a 'Related' section which is currently empty. On the right, there are buttons for 'New Contact', 'Edit', and 'New Opportunity'. At the bottom, it shows the 'Created By' information: Yash Jadhav on 25/05/2023, 12:02 pm, and the 'Last Modified By' information: Yash Jadhav on 25/05/2023, 12:02 pm.

Congratulations on wasting your life in engineering 😊

For validation of fields

- 1) Go to object manager
- 2) Select the object
- 3) Scroll down on left menu and select “validation rules”
- 4) Click on “new”
- 5) Give the name for rule
- 6) Now write validation rule in “**Error Condition Formula**”

Ex

For phone no:

OR(

 NOT(LEN(PhoneNo__c) = 10),
 ISBLANK(PhoneNo__c),
 NOT(ISNUMBER(PhoneNo__c))

)

For email :

NOT(REGEX(Email__c , '^([a-zA-Z0-9_\\-\\.]+) @ ([a-zA-Z0-9_\\-\\.]+) \\.([a-zA-Z]{2,5}) \$'))

- Change your filed name accordingly

- 7) Write the error message that should be displayed after the wrong value is entered
- 8) Select field and select the particular field where you want to display error message