Phase 3: Data Modeling & Relationships

This phase focuses on designing data structures, relationships, and layouts to manage weather information efficiently in Salesforce.

1. Standard & Custom Objects

Overview:

Salesforce provides Standard Objects (Account, Contact, User, etc.) and allows creation of Custom Objects to handle project-specific data.

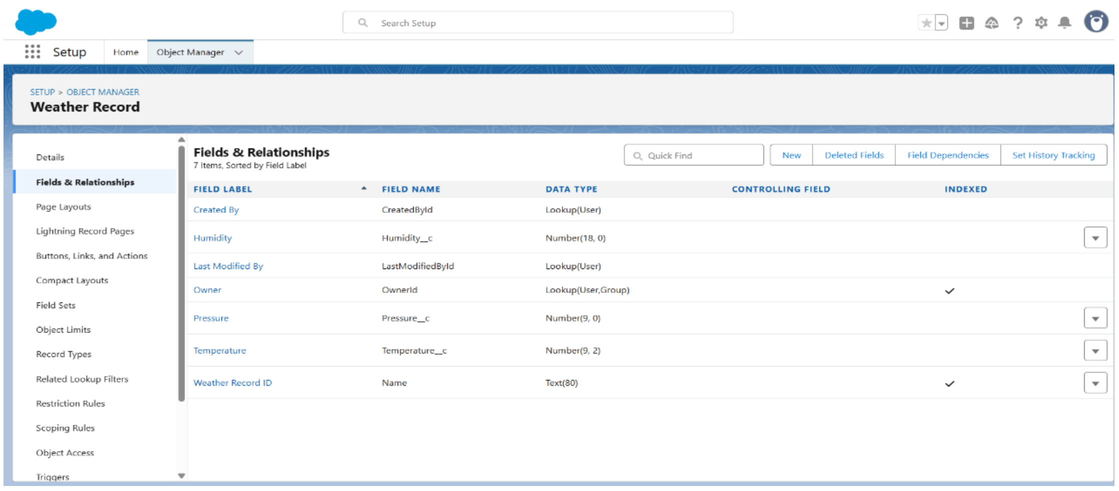
For this project, all weather-related objects are custom.

Objects Implemented:

Object Name Type Description
City__c Custom Stores city information including coordinates
Weather_Record__c Custom Stores weather data (Temperature, Humidity, Pressure, linked City)
Weather_Alert__c Custom Stores alerts for extreme weather conditions
Weather_Alert_City__c Custom Junction object to enable many-to-many relationship between Alerts and Cities
Optional: External_Weather_Data__x External Connects to OpenWeatherMap API to fetch live weather data

Implementation Notes:

All objects include standard audit fields: Created By, Last Modified By, Owner



Weather Record

Fields & Relationships
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
Created By	CreatedById	Lookup(User)		
Humidity	Humidity__c	Number(18, 0)		
Last Modified By	LastModifiedById	Lookup(User)		
Owner	OwnerId	Lookup(User,Group)		✓
Pressure	Pressure__c	Number(9, 0)		
Temperature	Temperature__c	Number(9, 2)		
Weather Record ID	Name	Text(80)		✓

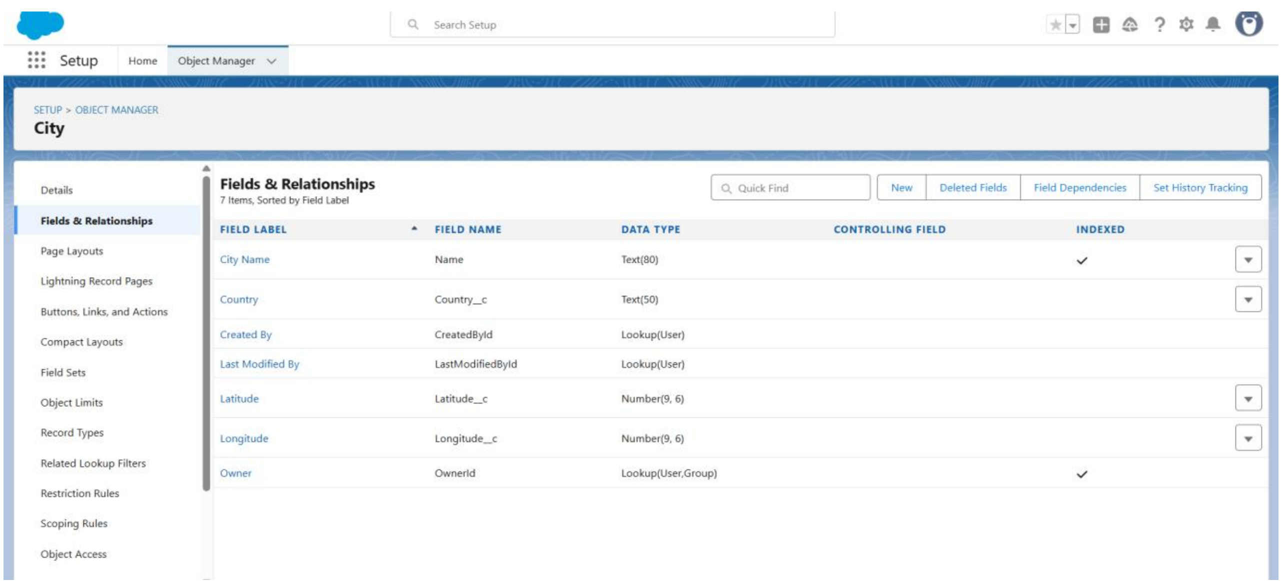
2. Fields

City__c Fields:

Field Name Type Description Real-Time Use Case
City Name Text Name of the city Identify city in Weather Records
Country__c Text Country of the city For geographical reporting
Latitude__c Number Latitude coordinate For mapping and external API calls

Longitude__c Number Longitude coordinate For mapping and external API calls

Weather_Record__c Fields:



Fields & Relationships
7 Items, Sorted by Field Label

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIELD	INDEXED
City Name	Name	Text(80)		✓
Country	Country__c	Text(50)		
Created By	CreatedById	Lookup(User)		
Last Modified By	LastModifiedById	Lookup(User)		
Latitude	Latitude__c	Number(9, 6)		
Longitude	Longitude__c	Number(9, 6)		
Owner	OwnerId	Lookup(User,Group)		✓

Key Field Considerations

1. **Essential Data Capture** – The fields are carefully designed to store vital weather and city information such as City Name, Temperature, Humidity, Pressure, and Coordinates, ensuring the system can manage both geographical and climate-related data effectively.
2. **Practical Usage** – Every field supports a real-time business use case. For example, Temperature and Humidity fields drive dashboard insights, Precipitation fields help in flood alerts, and Timezone ensures weather records are logged accurately for each city.
3. **Data Accuracy & Integrity** – Relationships like Lookup and Master-Detail guarantee that every Weather Record is tied to the correct City, maintaining consistency and preventing duplicate or orphaned records.
4. **Scalability & Analytics** – Advanced fields such as Wind Speed, Visibility, and Condition are included to allow deeper analysis, support predictive weather models, and ensure the system can easily adapt to future requirements without major redesign

3.Record Types

Currently using default record type for all objects.Optional future enhancement: Different alert types (e.g., Rain Alert, Heatwave Alert) can use Record Types to control page layouts and picklist values.

- All objects (City__c, Weather_Record__c, Weather_Alert__c) currently use the **default record type**, ensuring consistent layouts and access to fields.

- Record Types allow different **page layouts and picklist values** based on record type, supporting tailored business processes.

- **Future enhancements** may include specialized alert types (e.g., Rain Alert, Heatwave Alert) with custom layouts and picklists to improve usability and reporting.

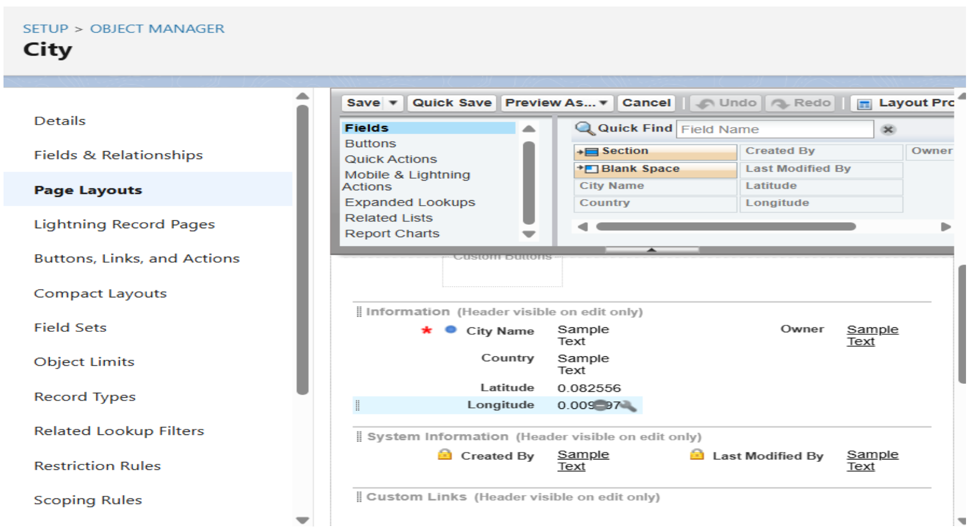
- Benefits include **better user experience**, reduced data entry errors, and **scalable reporting** for different alert categories.

4.Page Layouts

City Page Layout:

- **Fields included:**
    - City Name
    - Country
    - Latitude
    - Longitude

Layout Design:

- Sections organized for **Quick View** and **System Info**.
- Includes **Related Lists** (Weather Alerts) for easy navigation.
    - 

Information (Header visible on edit only)

City Name	Sample Text	Owner	Sample Text
Country	Sample Text		
Latitude	0.082556		
Longitude	0.009 97		

System Information (Header visible on edit only)

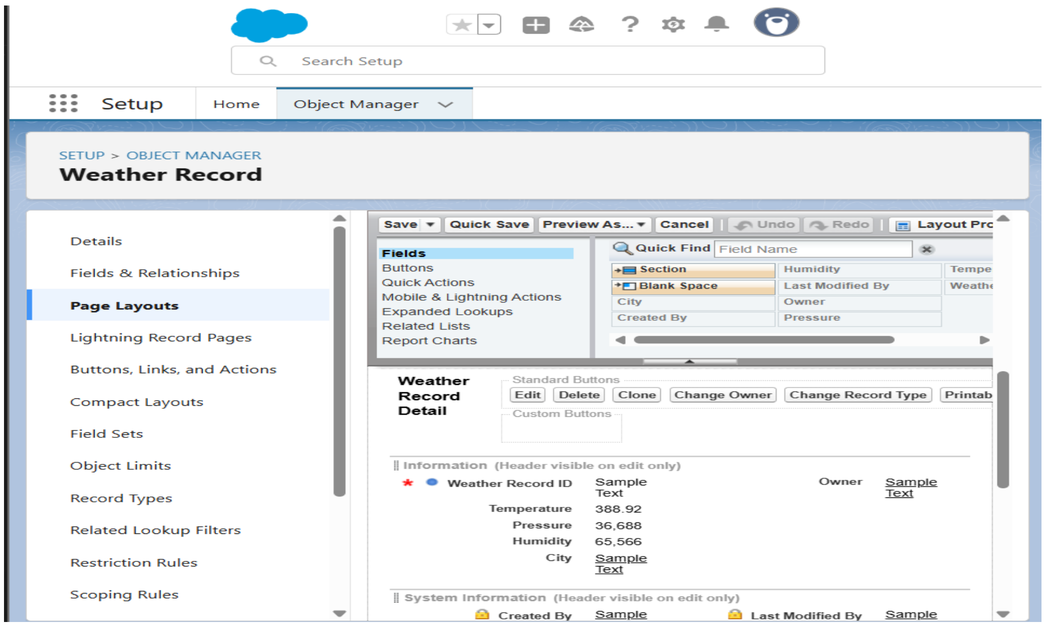
Created By	Sample Text	Last Modified By	Sample Text

Custom Links (Header visible on edit only)

    -

Weather Record Page Layout:

- **Fields included:**
    - City
    - Temperature
    - Humidity
    - Pressure
    - Weather_Record_ID
- **Layout Design:**
    - Sections organized for **Quick View** and **System Info**.
    - Includes **Related Lists** (Weather Alerts) for easy navigation.



Weather Record Detail

Information (Header visible on edit only)

Weather Record ID: Sample Text
Owner: Sample Text
Temperature: 388.92
Pressure: 36,688
Humidity: 65,566
City: Sample Text

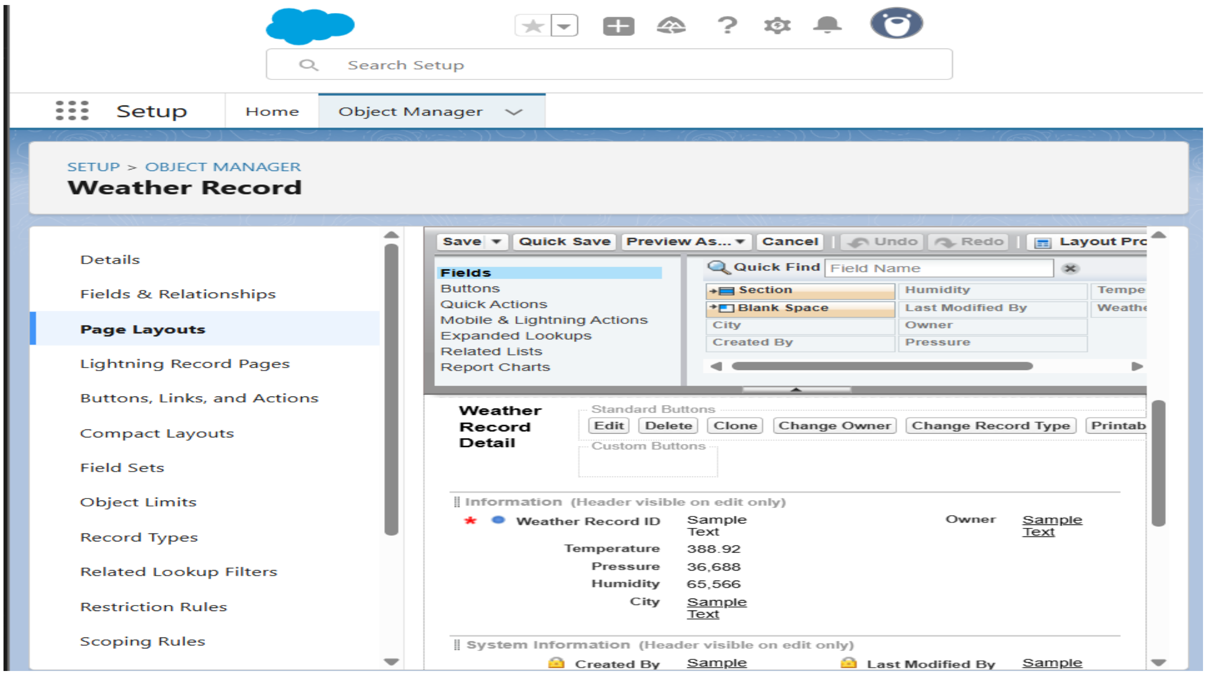
System Information (Header visible on edit only)

Created By: Sample
Last Modified By: Sample

5. Compact Layouts

- **Purpose:** Displays essential fields in the Lightning Highlights Panel for quick viewing.
- **City__c Fields Shown:** City Name, Country, Latitude, Longitude.
- **Weather_Record__c Fields Shown:** City, Temperature, Humidity, Weather_Record_ID.
- **Primary Assignment:** Each object has a primary compact layout assigned for consistent record previews.
- **User Benefit:** Enables quick access to critical information without opening the full record page.
- **Lightning Compatible:** Optimized for both desktop and mobile Lightning Experience.

Temperature



Information (Header visible on edit only)

Weather Record ID	Sample Text	Owner	Sample Text
Temperature	388.92		
Pressure	36,688		
Humidity	65,566		
City	Sample Text		

System Information (Header visible on edit only)

Created By	Sample	Last Modified By	Sample

6.Schema Builder

Purpose:

- Provides a **visual overview of all objects and their relationships** in the project.
- Helps admins and developers **understand how data flows** between objects.

Relationships Implemented:

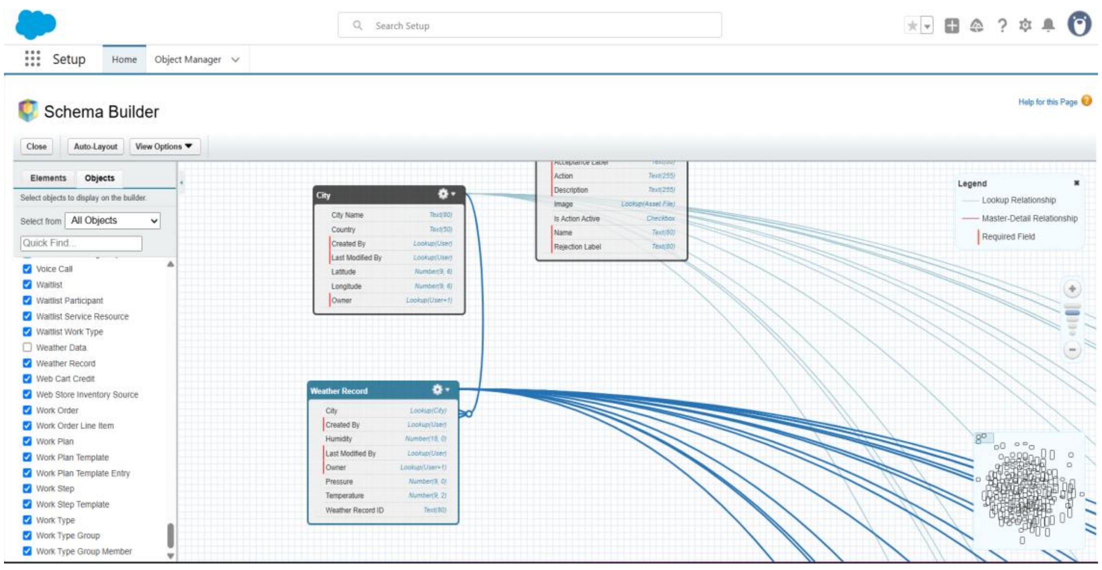
- **Weather_Record__c** → **City__c** (Lookup or Master-Detail) to link weather data with its city.
- **Weather_Alert_City__c** → **Weather_Alert__c** (Master-Detail) and → **City__c** (Master-Detail) to enable many-to-many alerts for multiple cities.

Layout & Organization:

- Objects are arranged for clarity: **City at the top**, Weather Records below, Alerts on the side.
- Drag-and-drop is used to organize the schema for **easy visualization and comprehension**.

Benefits:

- Simplifies understanding of **object relationships** and data dependencies.
- Supports **reporting, dashboards, and future enhancements**.
- Provides a clear reference for **new team members or admins**

- 

- ✓ Voice Call
- ✓ Waitlist
- ✓ Waitlist Participant
- ✓ Waitlist Service Resource
- ✓ Waitlist Work Type
- ☐ Weather Data
- ✓ Weather Record
- ✓ Web Cart Credit
- ✓ Web Store Inventory Source
- ✓ Work Order
- ✓ Work Order Line Item
- ✓ Work Plan
- ✓ Work Plan Template
- ✓ Work Plan Template Entry
- ✓ Work Step
- ✓ Work Step Template
- ✓ Work Type
- ✓ Work Type Group
- ✓ Work Type Group Member

City	
City Name	Text(80)
Country	Text(50)
Created By	Lookup(User)
Last Modified By	Lookup(User)
Latitude	Number(9, 6)
Longitude	Number(9, 6)
Owner	Lookup(User+1)

Acceptance Label	Text(80)
Action	Text(255)
Description	Text(255)
Image	Lookup(Asset File)
Is Action Active	Checkbox
Name	Text(80)
Rejection Label	Text(80)

Weather Record	
City	Lookup(City)
Created By	Lookup(User)
Humidity	Number(18, 0)
Last Modified By	Lookup(User)
Owner	Lookup(User+1)
Pressure	Number(9, 0)
Temperature	Number(9, 2)
Weather Record ID	Text(80)

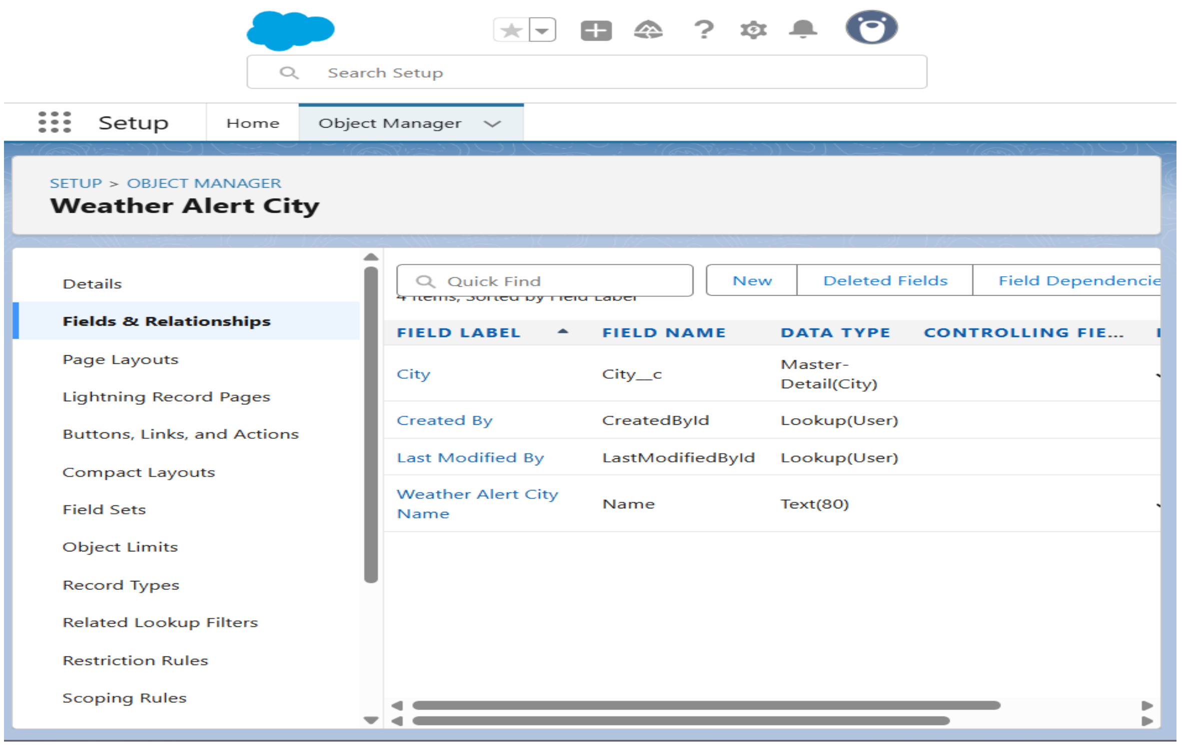
7.Lookup vs Master-Detail vs Hierarchical Relationships

- **Lookup Relationship:**
  - Example: Weather_Record__c → City__c
  - Weather record exists independently of the city; deleting a city does not delete linked weather records.
  - Provides flexibility and optional linking between objects.

- **Master-Detail Relationship:**
  - Example: Weather_Record__c → City__c
  - Strong dependency; deleting the parent (City) deletes all child records.
  - Supports roll-up summary fields (e.g., average temperature per city).

- **Master-Detail (Junction):**
  - Example: Weather_Alert_City__c → Weather_Alert__c / City__c
  - Enables **many-to-many relationships**, allowing multiple alerts for multiple cities.

- **Hierarchical Relationship:**
  - Example: User → Manager
  - Not used in the Weather Project but useful for **approval processes and organizational hierarchy**.

- 

SETUP > OBJECT MANAGER

Weather Alert City

FIELD LABEL	FIELD NAME	DATA TYPE	CONTROLLING FIE...
City	City__c	Master-Detail(City)	
Created By	CreatedById	Lookup(User)	
Last Modified By	LastModifiedById	Lookup(User)	
Weather Alert City Name	Name	Text(80)	

8.Junction Objects

- **Purpose:**
  - Enables **many-to-many relationships** between Weather Alerts and Cities.
  - Allows a single alert to be associated with multiple cities and vice versa.
- **Implementation:**
  - Custom object **Weather_Alert_City__c** acts as a junction.
  - Connected to **Weather_Alert__c** and **City__c** using Master-Detail relationships.
- **Real-Time Functionality:**
  - Multiple cities can be linked to a single alert.
  - Multiple alerts can be linked to a single city.
  - Ensures accurate tracking and reporting of alert coverage across locations.
- **Benefit:**
  - Supports **scalable alert management** across multiple cities.
  - Maintains **accurate relationships** between alerts and locations.
  - Facilitates **reporting, dashboards, and real-time monitoring** of alert coverage.

9. External Objects

External objects allow Salesforce to **access and display data from external systems** without storing it in Salesforce. In the Weather Project, the external object **External_Weather_Data__x** connects to a service like **OpenWeatherMap API** to fetch live weather data.

- Fields such as **Temperature, Humidity, and Pressure** are mapped from the external system to Salesforce.
- Data is displayed in **Lightning Record Pages or Dashboards**, providing **real-time insights**.
- This approach reduces **Salesforce storage usage** while keeping information **up-to-date**.

Key Benefits:

- Enables **dynamic and real-time monitoring** of weather conditions.
- Reduces internal storage requirements.
- Supports **analytics, dashboards, and decision-making** based on current weather data.