**Stock Market Predictions**

**📌 Phase 1: Problem Understanding & Industry Analysis**

**1. Requirement Gathering**

**The goal is to help investors and analysts manage stock data inside Salesforce.**

* **Store stock details (company name, ticker, sector, price).**
* **Allow investors to track their portfolios (buy/sell transactions).**
* **Show predictions (future price trends).**
* **Provide reports and dashboards (gains/losses, prediction accuracy).**
* **Send alerts when stock crosses certain thresholds.**

**2. Stakeholder Analysis**

* Investors (End Users) → Want to see their portfolio and stock predictions.
* Financial Analysts → Need dashboards to compare stock performance.
* Admin/Developer → Setup objects, flows, automation, and integrations.
* Executives → Want summary dashboards for decision making.

**3. Business Process Mapping**

* Current Process (outside Salesforce):
* Investors use multiple apps like Yahoo Finance, Excel, TradingView.
* Predictions are either manual or from third-party websites.
* No single system to combine portfolio + real-time data + predictions.
* Proposed Process (inside Salesforce):
* Salesforce fetches stock data from an external API.
* Data is stored in custom objects (Stock, Portfolio, Transaction, Prediction).
* Automation updates portfolio value and prediction records.
* Reports and dashboards show insights (Top gainers, Loss %).
* Alerts sent to investors when price moves significantly.

4. Industry-Specific Use Case Analysis

* Retail Investors → Manage their personal stock portfolios inside Salesforce.
* Investment Firms → Track multiple client portfolios in one platform.
* Wealth Management Companies → Give AI-based recommendations using predictions.
* 📌 Example: If an investor has 10 shares of Company, Salesforce shows:
* Current Value = Quantity × Price.
* Predicted Next Price = +5%.
* System sends an alert: “Your Company holdings may rise 5% tomorrow.”

5. AppExchange Exploration

* Existing apps provide stock tickers or financial dashboards.
* But there is no complete prediction-focused app on AppExchange.
* This project fills that gap by combining portfolio tracking + prediction + alerts.

**📌 Phase 2: Org Setup & Configuration**

**1. Salesforce Edition**

* Use Developer Edition → It’s free and provides all the features required (custom objects, automation, API integration).
* Suitable for learning and building a POC (Proof of Concept).

**2. Company Profile Setup**

* Fiscal Year → Set from January to December (to align with global stock reporting).
* Default Currency → Indian Rupee (₹) or USD ($), depending on stock market focus.
* Timezone → Align with the stock exchange (e.g., GMT+5:30 for India).

**3. Business Hours & Holidays**

* Set Business Hours → 9:30 AM – 3:30 PM (Indian NSE/BSE market).
* Configure Holidays → Official stock market holidays (e.g., Diwali, Independence Day, Republic Day).
* Purpose → Helps with SLA calculations and alerts only during market hours.

**4. User Setup & Licenses**

* Users:

Admin User → Full control (manages configurations).

* Investor User → Restricted access (views only their portfolio).

Assign Salesforce Platform License to Investor user (cost-effective in real scenario).

**5. Profiles**

* Admin Profile → CRUD (Create, Read, Update, Delete) on all objects.
* Investor Profile → Read-only access to Stocks & Predictions, Read/Write to their Portfolio.

**6. Roles**

* Admin Role → Higher in hierarchy, can see all data.
* Investor Role → Lower in hierarchy, can only see their own records.

**7. Permission Sets**

* Create "API Access Permission Set" → Grants API access for stock data integration.
* Assign this only to Admin.

**8. Org-Wide Defaults (OWD)**

* Portfolio & Transactions → Private (only owner can see).
* Stock → Public Read-Only (everyone can see stock prices).
* Prediction → Controlled by Parent (linked to Stock).

**9. Sharing Rules**

* Share Portfolios only with the specific investor.
* Admin can see everything, but one investor cannot see another’s portfolio.

**10. Login Access Policies**

* Enable IP Restrictions for Admin login (e.g., office network only).
* Allow investors to log in from anywhere.

**11. Dev Org Setup**

* Create Developer Org (from Salesforce.com → Free Signup).
* Install sample financial datasets (CSV imports).

**12. Sandbox Usage**

Use Developer Sandbox for:

* Testing API integrations (stock price API).
* Testing automation (flows, triggers).
* Deploy to production only after validation.

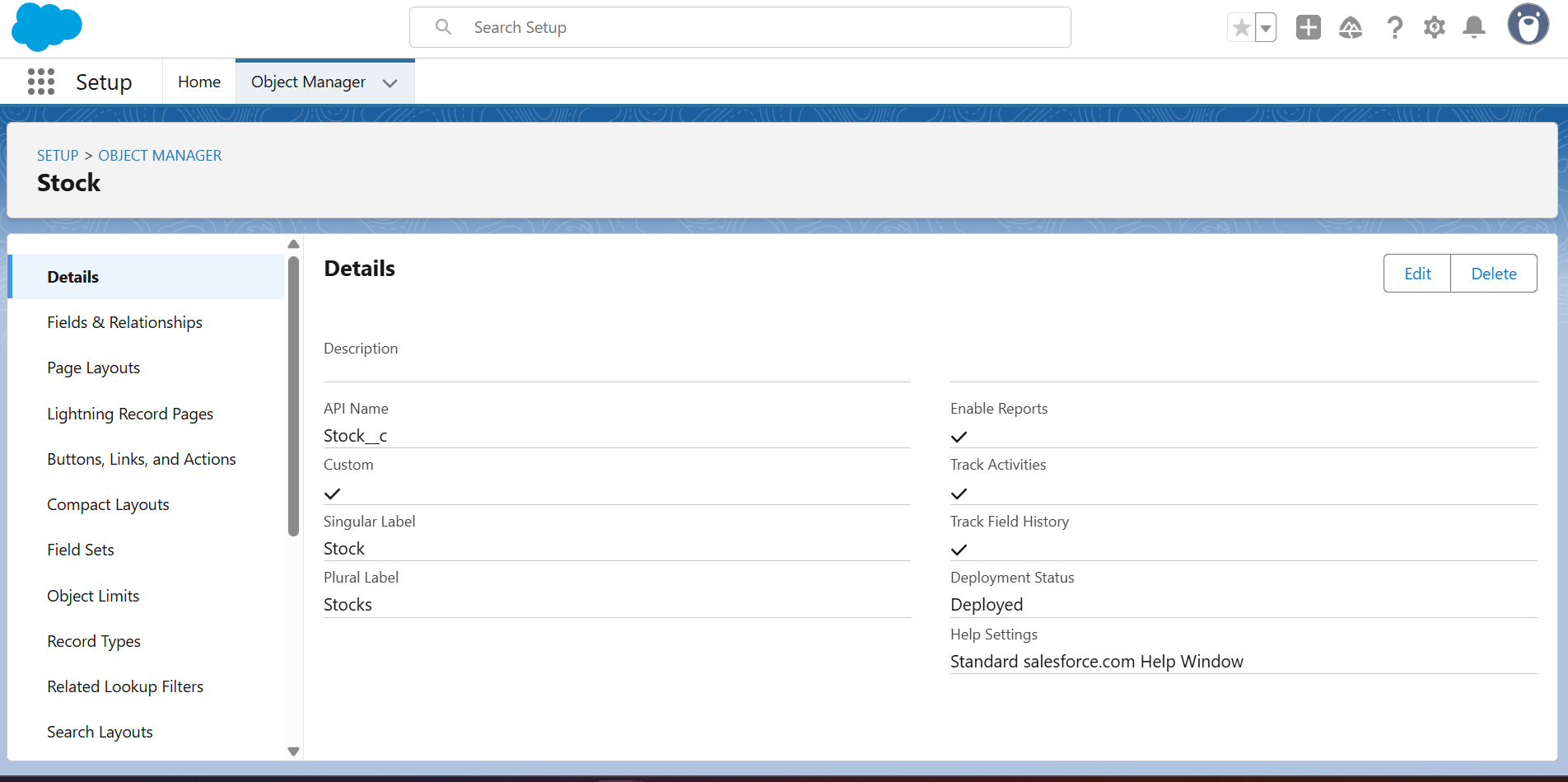
**13. Deployment Basics**

* Use **Change Sets** for deployment (simple beginner method).
* Advanced option: Use **VS Code + SFDX** for metadata deployment.

**📌 Phase 3: Data Modeling & Relationships**

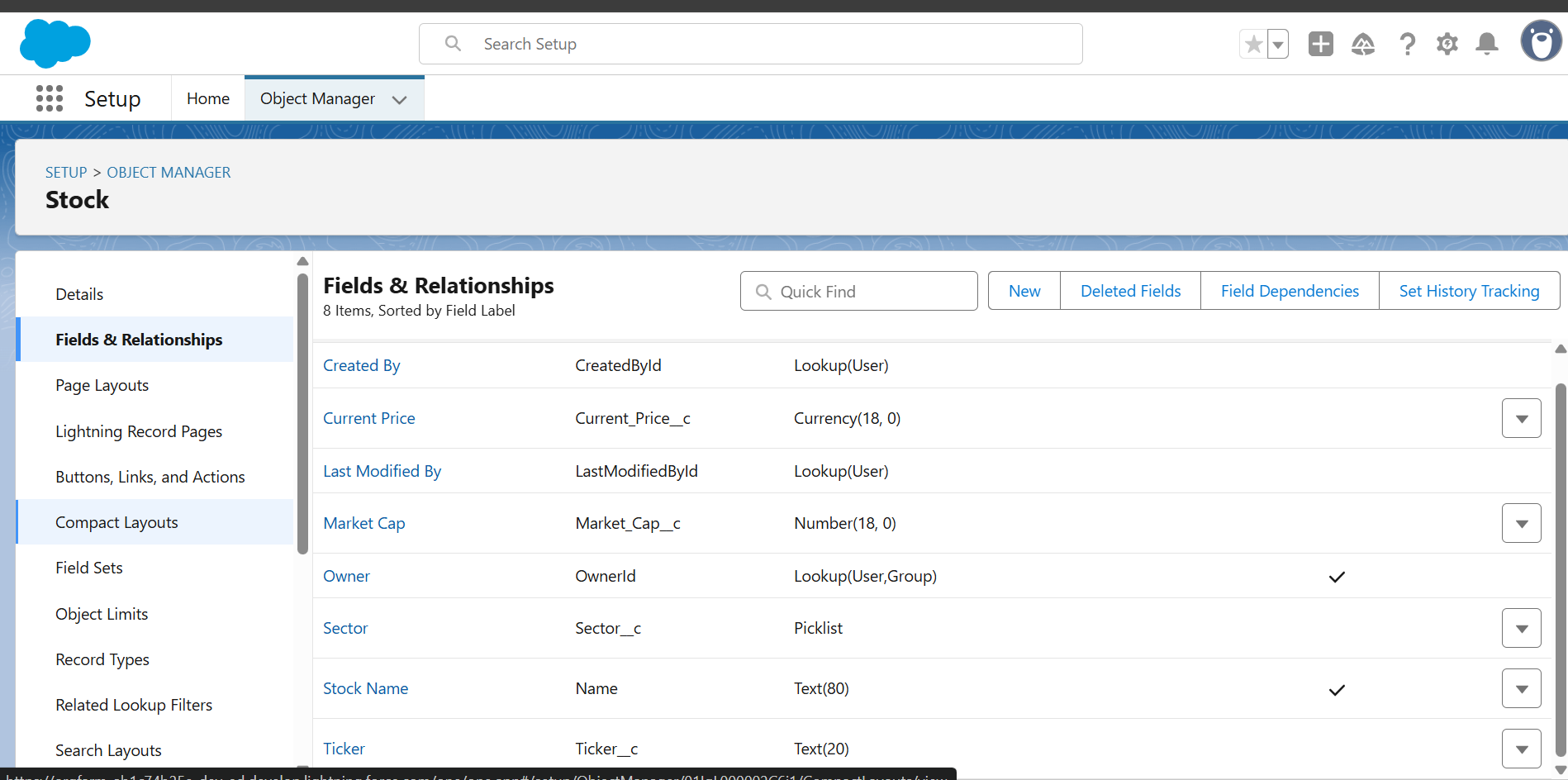
**1.Custom Objects & Key Fields**

* Stock\_\_c

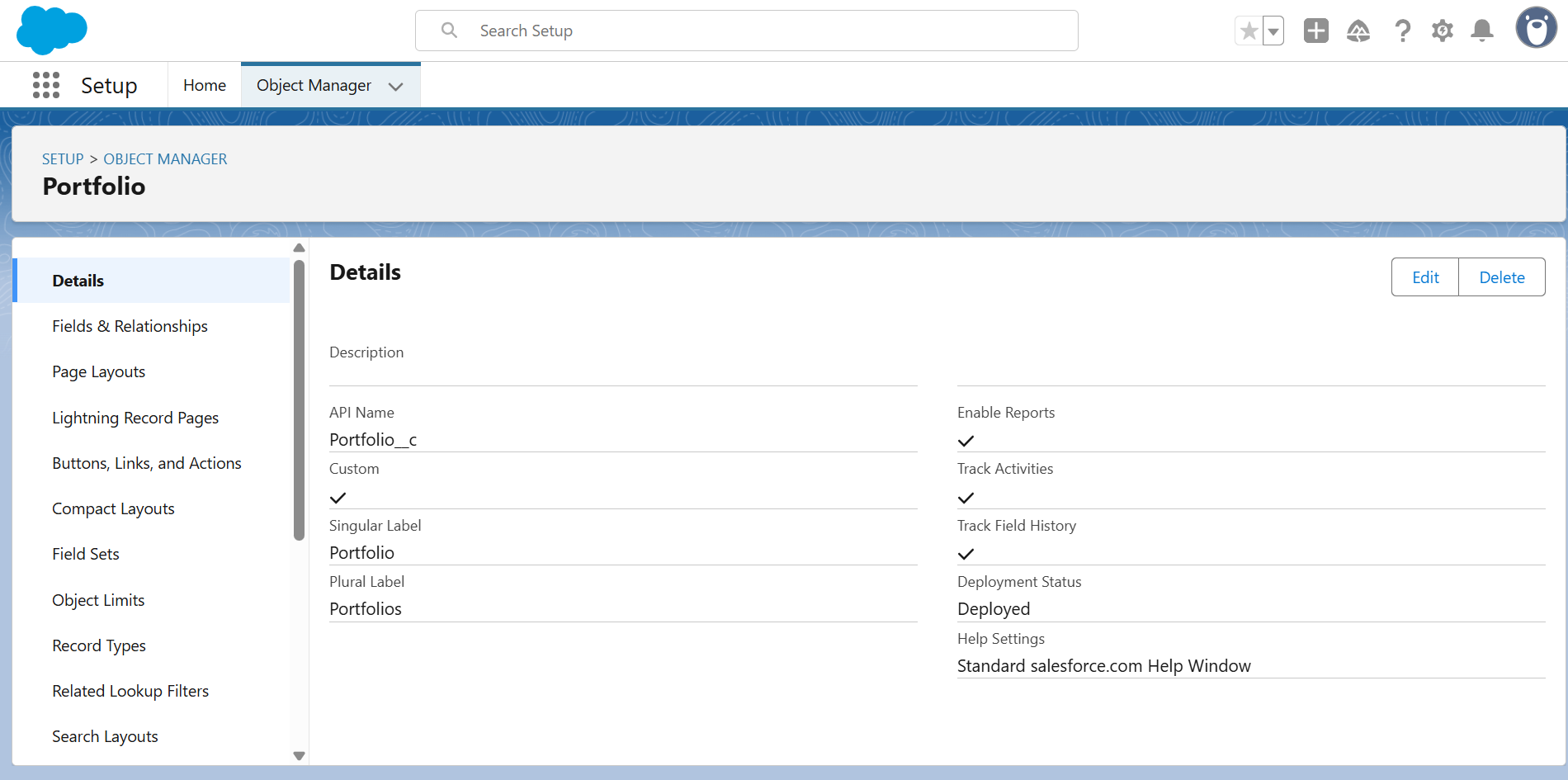


Fields:

* Ticker\_\_c (Text, e.g., TCS, INFY)
* Sector\_\_c (Picklist: IT, Pharma, Banking, etc.)
* Current\_Price\_\_c (Currency)
* Market\_Cap\_\_c (Number)

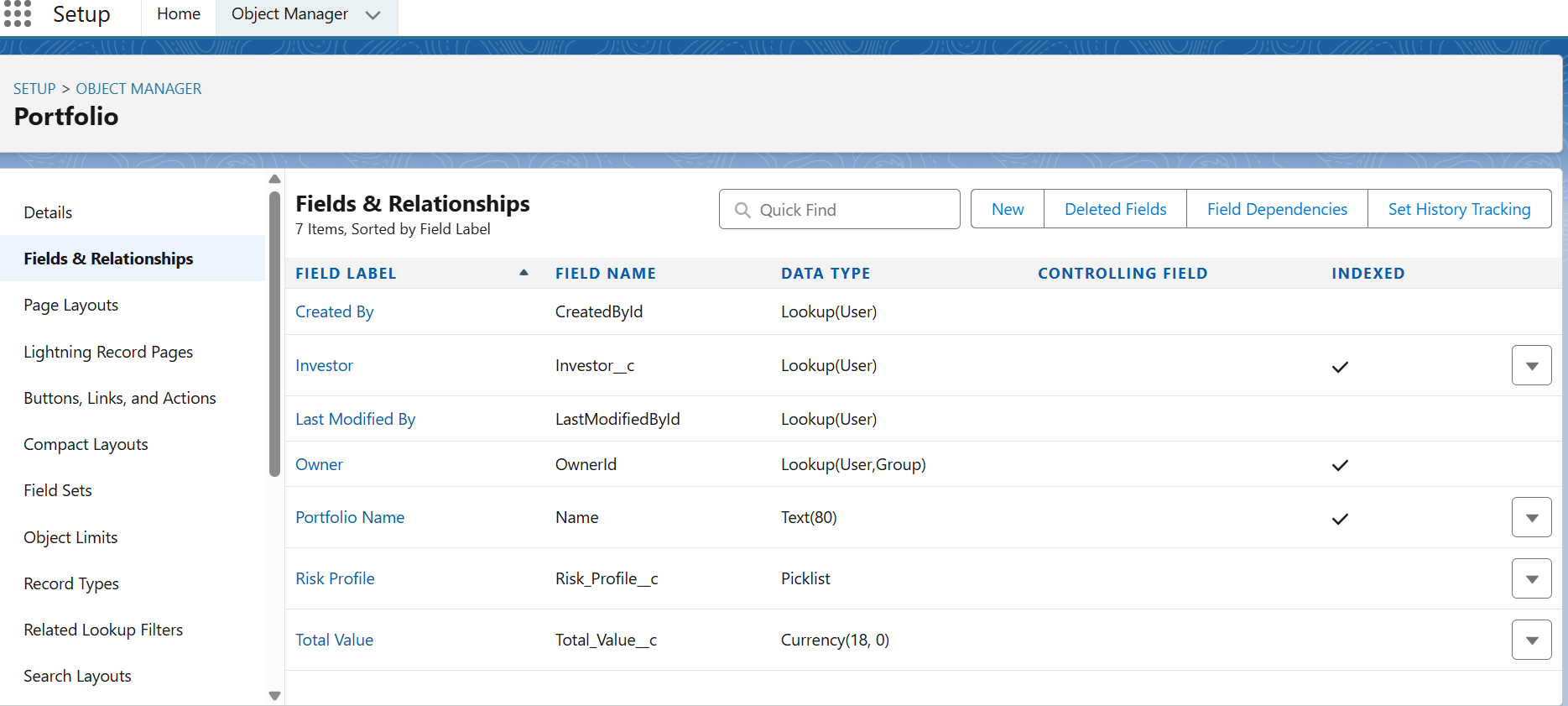


* Portfolio\_\_c

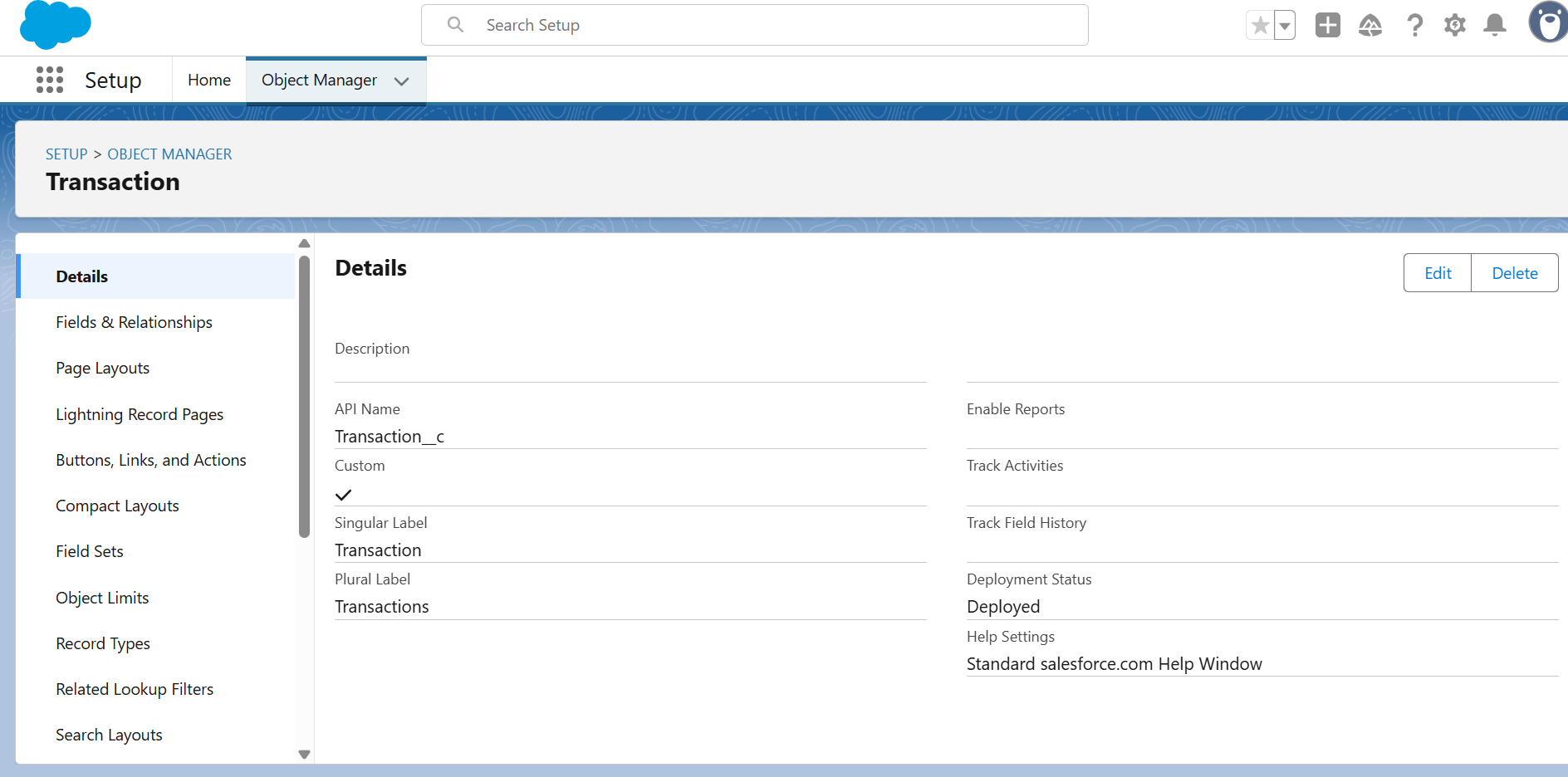


Fields:

* Investor\_\_c (Lookup → User)
* Total\_Value\_\_c (Currency, roll-up from Transactions)
* Risk\_Profile\_\_c (Picklist: Low, Medium, High)

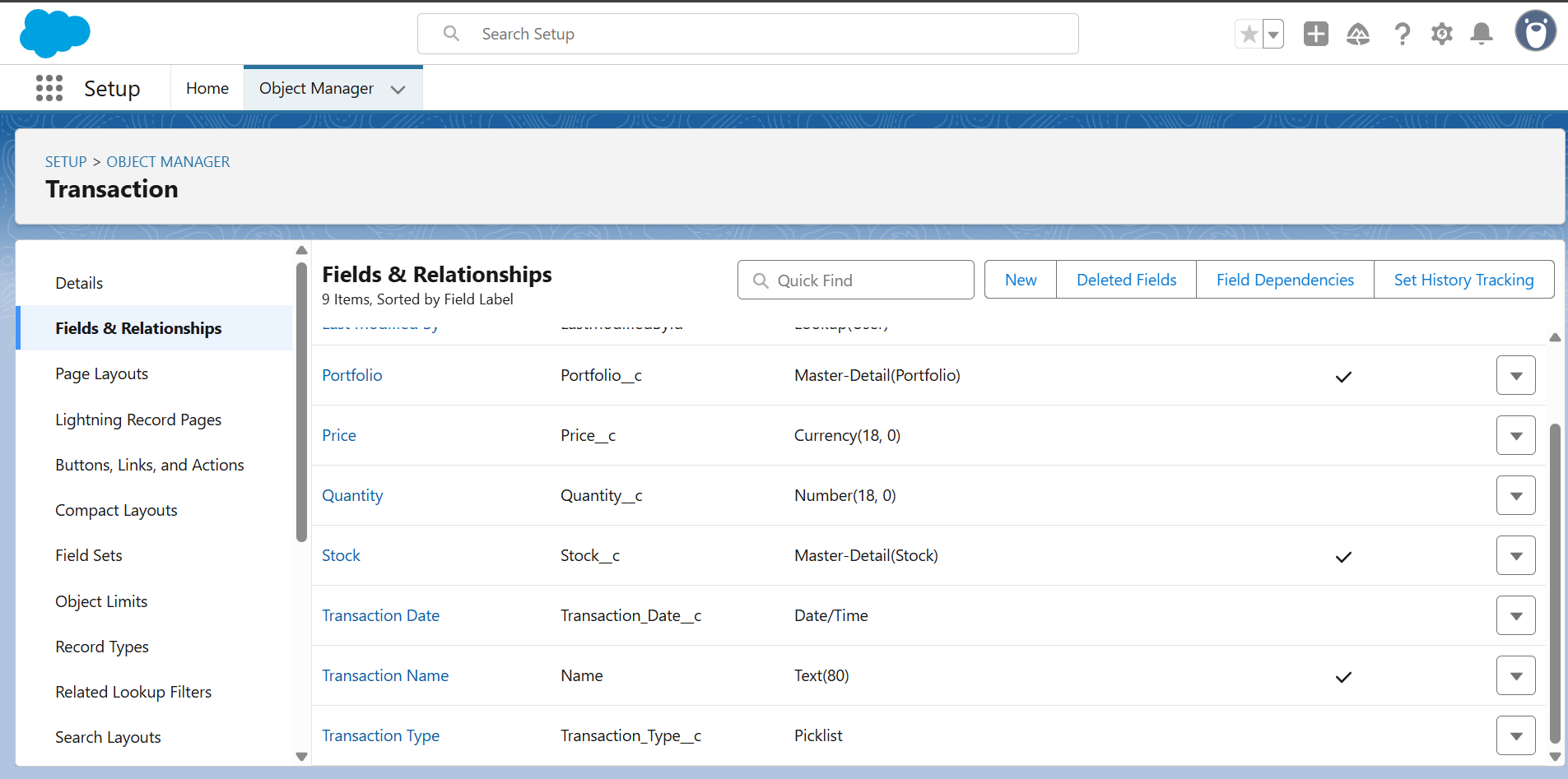


* Transaction\_\_c (Junction between Stock & Portfolio)

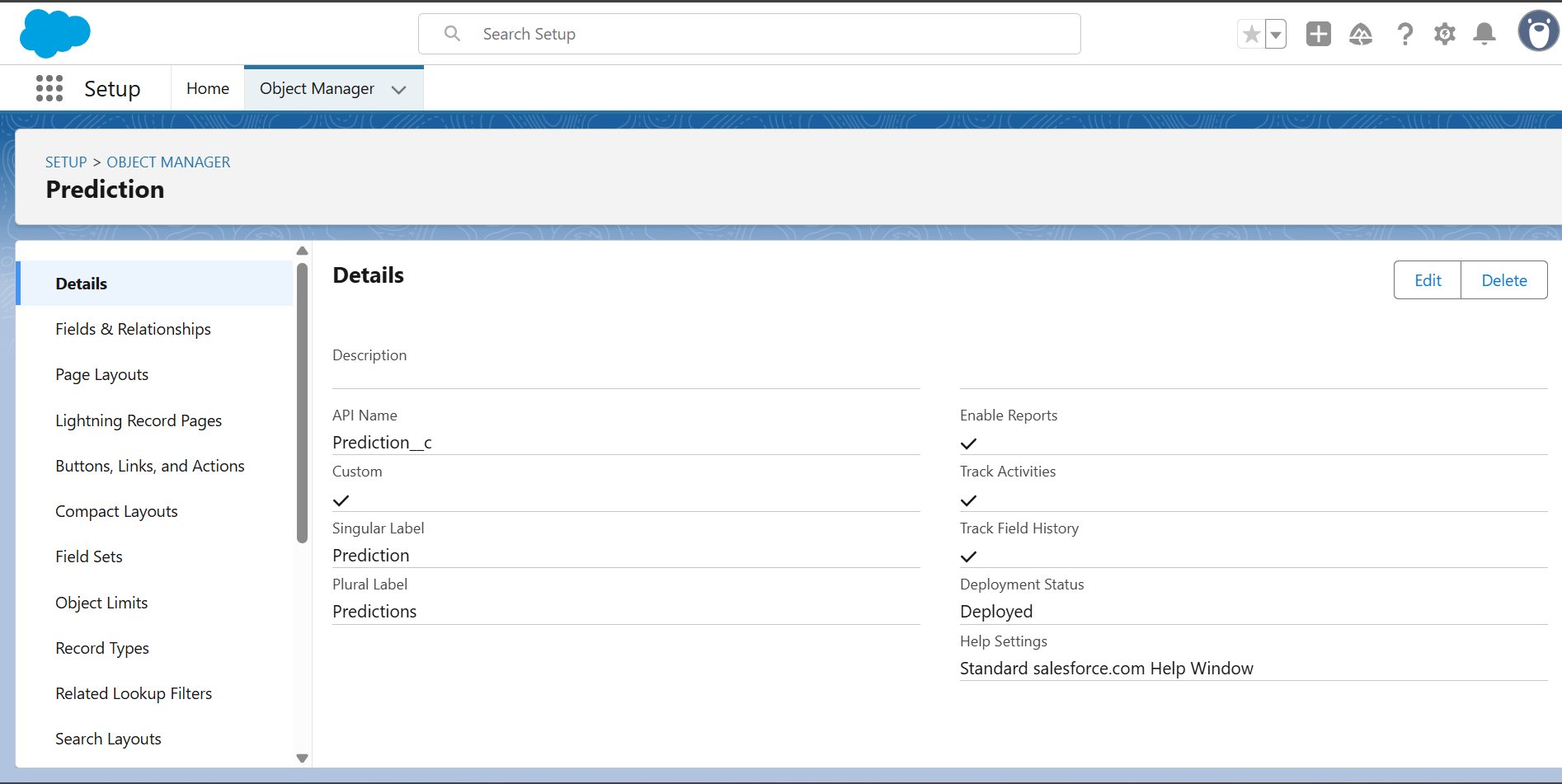


Fields:

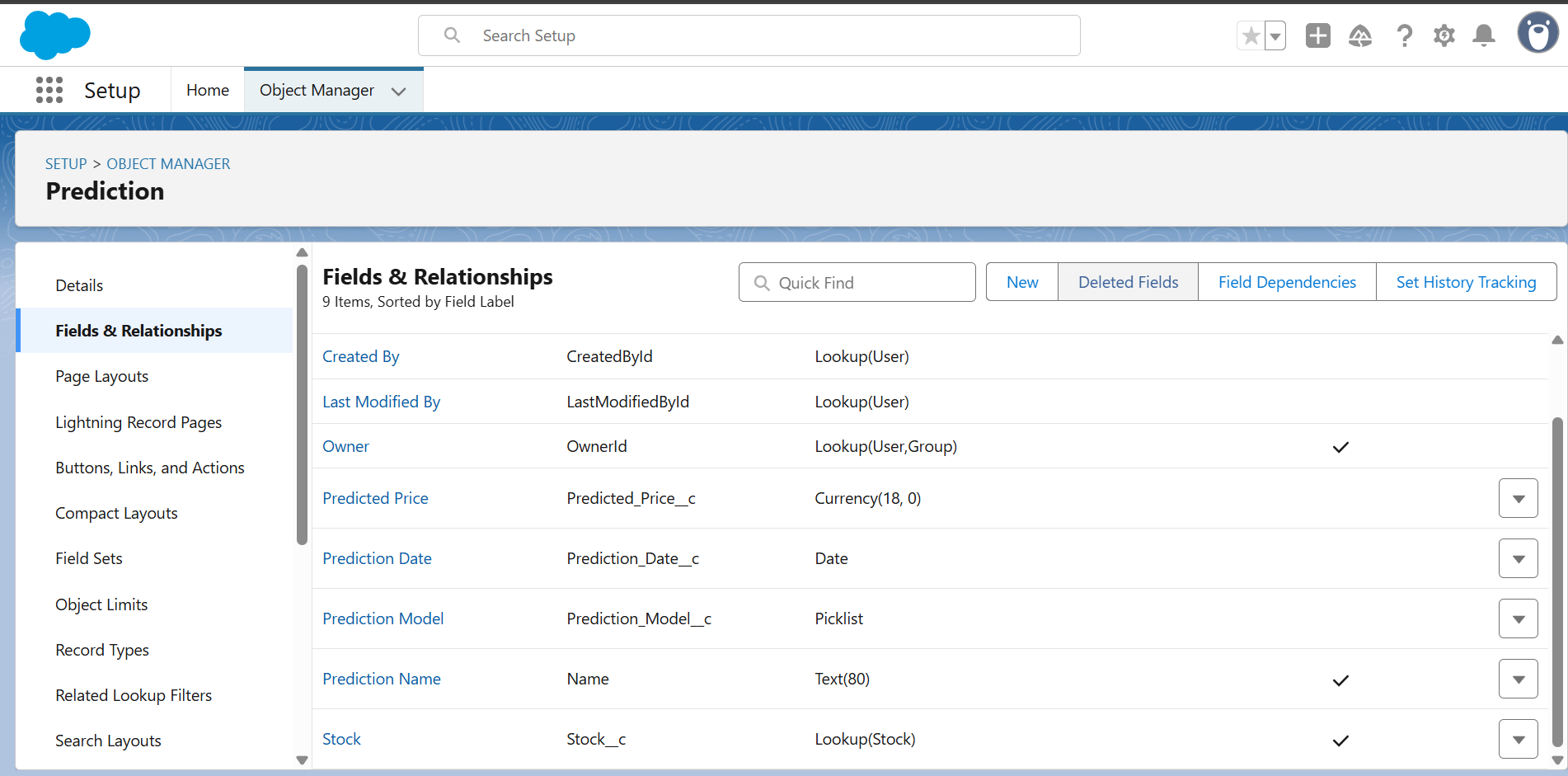
* Portfolio\_\_c (Master-Detail)
* Stock\_\_c (Master-Detail)
* Transaction\_Type\_\_c (Picklist: Buy, Sell)
* Quantity\_\_c (Number)
* Price\_\_c (Currency)
* Transaction\_Date\_\_c (Date/Time)



* Prediction\_\_c



* Fields:
* Stock\_\_c (Lookup)
* Predicted\_Price\_\_c (Currency)
* Confidence\_Score\_\_c (Percent)
* Prediction\_Date\_\_c (Date)
* Prediction\_Model\_\_c (Picklist: Moving Avg, ML, AI Service, etc.)



**2.Relationships**

* **Portfolio\_\_c → Transaction\_\_c → Stock\_\_c**  
  🔗 Many-to-Many via Transaction\_\_c. One portfolio can hold many stocks, and one stock can belong to many portfolios.
* **Stock\_\_c → Prediction\_\_c**  
  🔗 One-to-Many. A stock can have multiple predictions over time.
* **User (Investor)** → Portfolio\_\_c (One-to-Many).

**3.Schema Builder (Visualization)**

* **Investor (User)** → **Portfolio\_\_c** → **Transaction\_\_c** → **Stock\_\_c**
* **Stock\_\_c** → **Prediction\_\_c**

**📌** This way:

* Admins see all data.
* Investors see **only their portfolios & related transactions**.
* Predictions link back to **stocks** for insights.

**📌 Phase 4: Process Automation (Admin)**

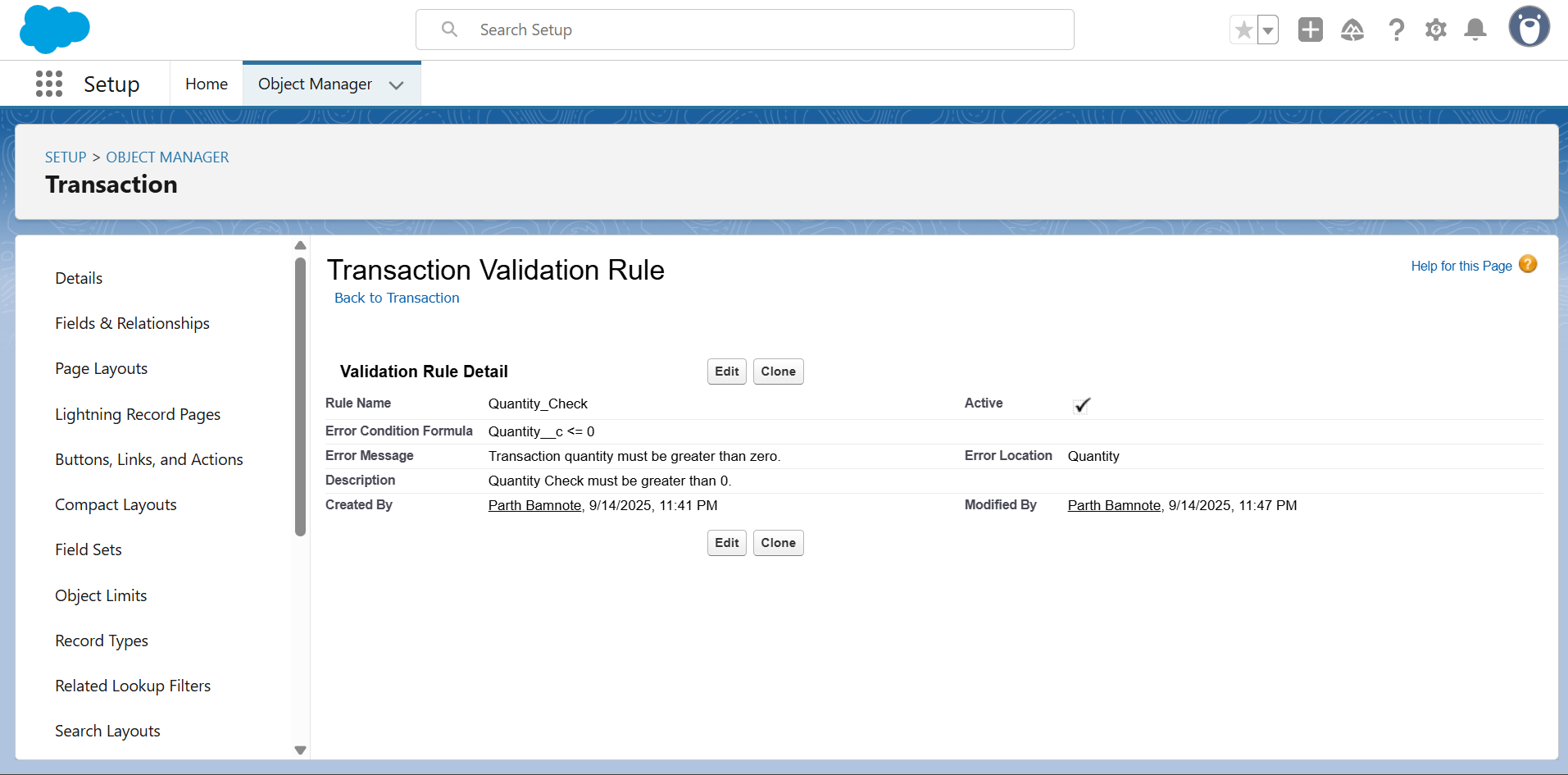
**1. Validation Rules**

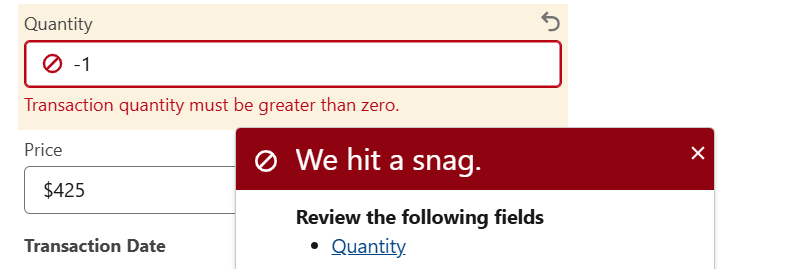
✅ Prevent bad data entry.

* Transaction\_\_c.Quantity\_\_c > 0
* Formula:

Quantity\_\_c <= 0

**→ Error Message: “Transaction quantity must be greater than zero.”**

****

****

**2. Flow (Record-Triggered)**

⚡ Type: After-Save Flow on Transaction\_\_c

Trigger: When a new transaction is inserted or updated.

Logic:

Get all transactions related to the portfolio.

Calculate :

**SUM(Quantity\_\_c \* Price\_\_c).**

Update :

**Portfolio\_\_c.Total\_Value\_\_c.**

💡 This keeps portfolio valuation auto-updated in real-time.

