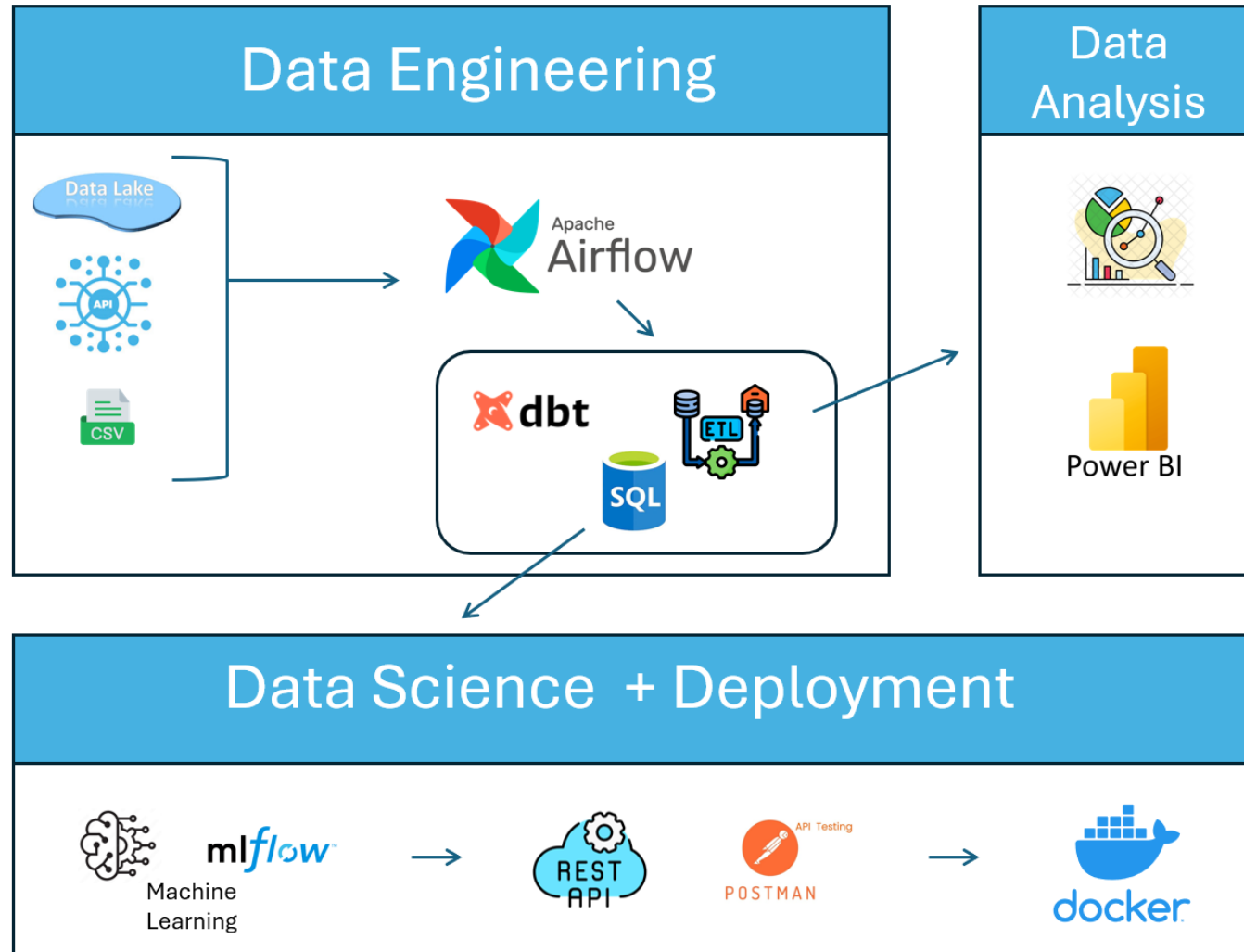


Sales Data Full Project - Workflow:

Data Engineering + Data Analysis + Data Science




1. Apache Airflow


← → ↺ 📍 localhost:8080/home 🔍 ☆ 📁 | 🔴 A ⋮

























































🧩 | 🏠 Scholarship Search... 🔗 https://de.thefreedi... 📺 Tutorial on Univaria... 🐍 Python LSTM (Long... 🌐 https://www.geeksf... 📖 https://uni-tuebing... 📄 https://www.javatpo... 🌐 https://www.w3sch...

» | 📁 All Bookmarks

 Airflow DAGs Cluster Activity Datasets Security - Browse - Admin - Docs - ⚙️ 16:13 UTC - AA - ▲

DAGs

All 7 Active 7 Paused 0 Running 0 Failed 0 Auto-refresh 

DAG ↕	Owner ↕	Runs ↕	Schedule	Last Run ↕	Next Run ↕	Recent Tasks	Actions	Links
 dbt_etl_pipeline_mart	azizjon	 1 	None	2025-04-19, 16:05:27		 2 	   ...	
 dbt_etl_pipeline_seed	azizjon	 1 	None	2025-04-19, 16:04:46		 1 	   ...	
 dbt_etl_pipeline_staging	azizjon	 1 	None	2025-04-19, 16:05:14		 1 	   ...	
 dbt_etl_pipeline_test	azizjon	 1 	@daily	2025-04-19, 16:06:27	2025-04-19, 00:00:00	 1 	   ...	
 get_dbt_docs	azizjon	 1 	None	2025-04-19, 16:06:40		 1 	   ...	
 load_new_order_items	azizjon	 2 	5*** Mon-Sat	2025-04-19, 16:05:59	2025-04-19, 16:05:00	 1 	   ...	
 load_new_orders	azizjon	 2 	0*** Mon-Sat	2025-04-19, 16:05:51	2025-04-19, 16:00:00	 1 	   ...	

⏪ < 1 > ⏩

Showing 1-7 of 7 DAGs

Version: v2.10.5
Git Hash: release-h03e3d6b1644b08d0b145ac7d05b58ff2c-rhf

2. DBT Tool and its Documentation

The screenshot displays the DBT documentation interface in a web browser. The address bar shows the URL `localhost:8001/#/model/model.commerce_dbt.product_sales#code`. The interface features a sidebar on the left with a tree view of the project structure. The main content area shows the 'product_sales' table documentation, with the 'Code' tab selected. The SQL code is displayed in a monospace font with syntax highlighting. A 'copy to clipboard' button is visible in the top right corner of the code editor area.

dbt

Search for models...

Project Database

Group

Sources

- commerce

Projects

- commerce_dbt
 - models
 - marts
 - business
 - customer_orders
 - product_sales
 - core
 - dim_customers
 - dim_products
 - dim_stores
 - fct_orders
 - ml_analytics
 - staging
 - stg_customers
 - stg_order_items
 - stg_orders
 - stg_product_categories

product_sales table

Details Description Columns Referenced By Depends On Code

Code

Source Compiled [copy to clipboard](#)

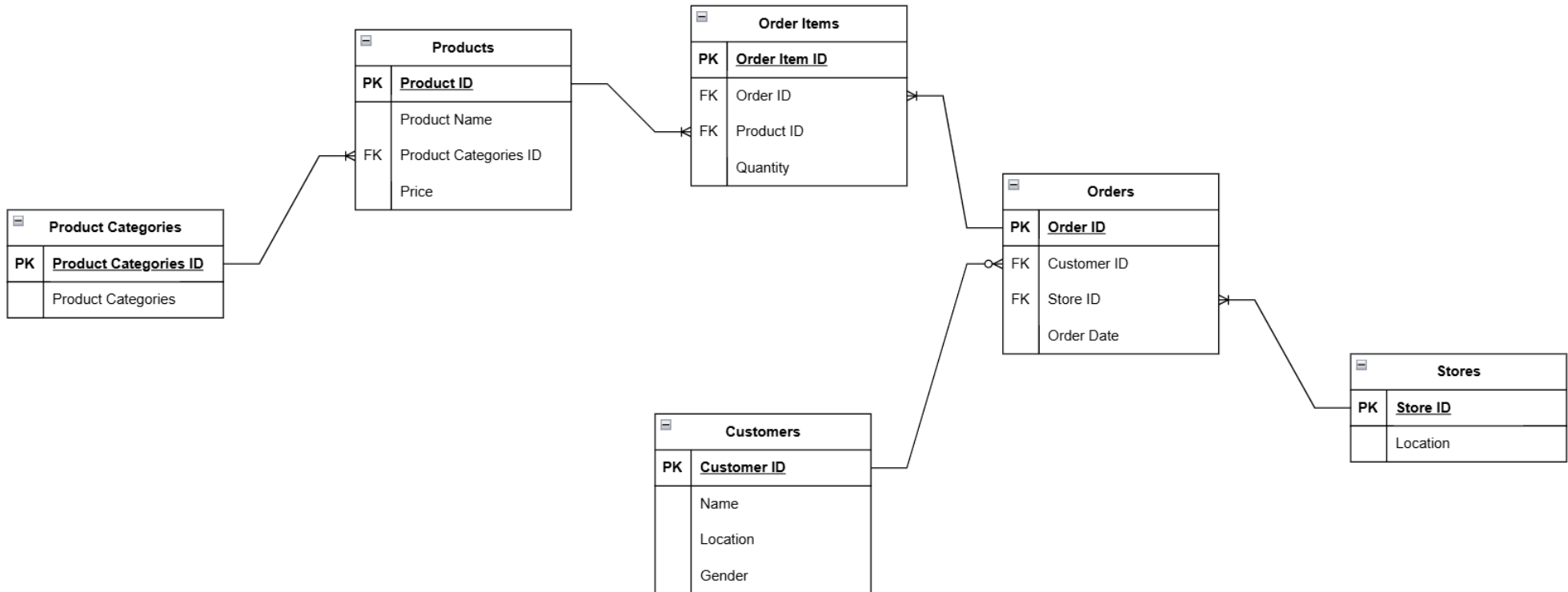
```
1 with orders as (  
2   select * from {{ ref('fct_orders') }}  
3 ),  
4 products as (  
5   select * from {{ ref('dim_products') }}  
6 )  
7  
8 select  
9   p.product_id,  
10  p.product_name,  
11  p.product_categories,  
12  p.price_segment,  
13  sum(o.quantity) as total_units_sold,  
14  sum(o.total_price) as total_revenue,  
15  avg(o.unit_price) as avg_selling_price,  
16  count(distinct o.order_id) as order_count  
17 from products p  
18 left join orders o on p.product_id = o.product_id  
19 group by 1, 2, 3, 4
```

3. SQL Database - DBeaver

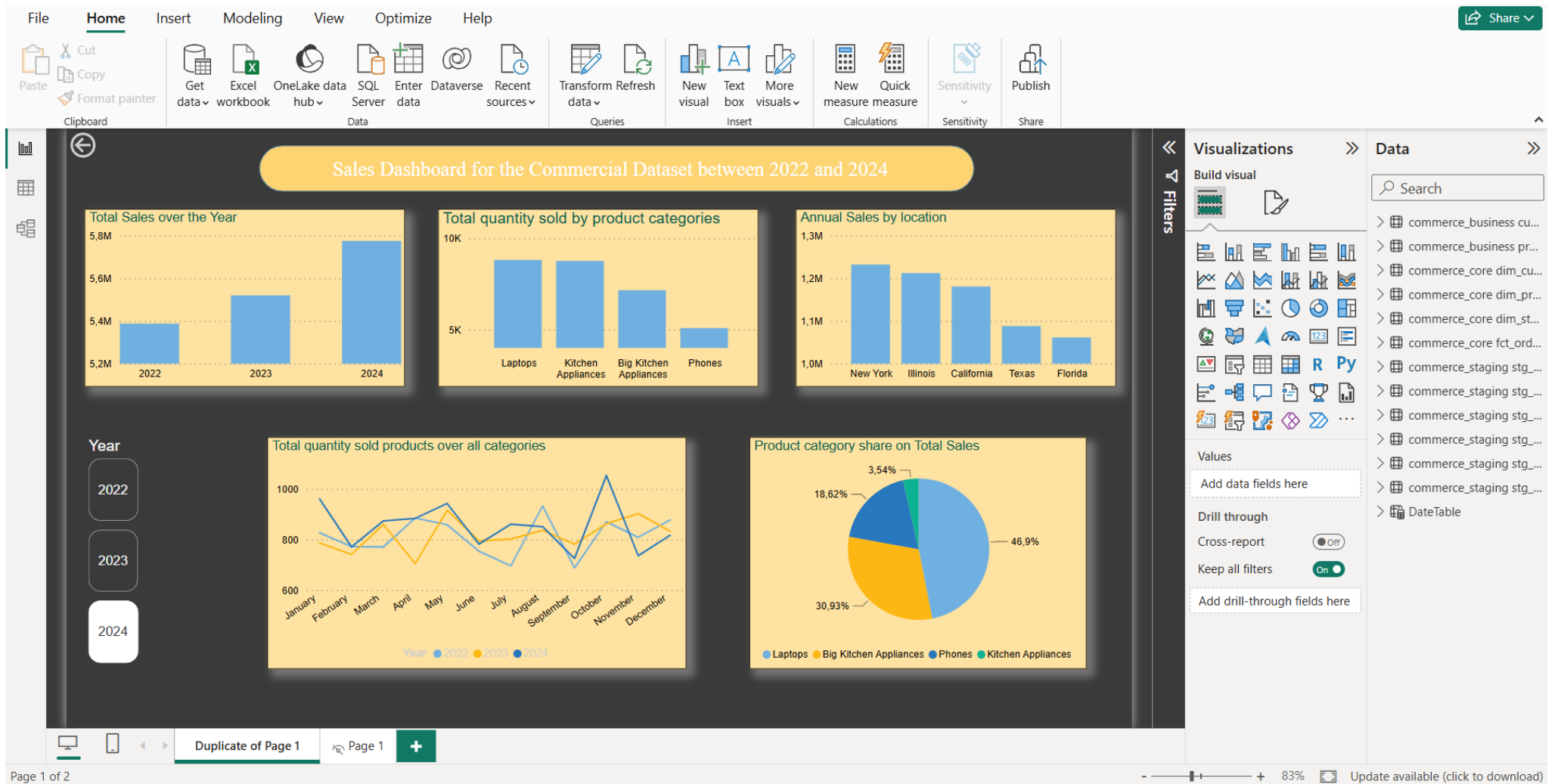
The screenshot displays the DBeaver 24.3.3 interface. The top menu bar includes File, Edit, Navigate, Search, SQL Editor, Database, Window, and Help. The left sidebar shows the Database Navigator with a tree view of the PostgreSQL database structure, including schemas (commerce_business, commerce_core, commerce_raw, commerce_staging, public), event triggers, extensions, storage, system info, roles, administrator, and system info. The main SQL Editor window shows a query: `select name, customer_segment, order_count, total_spend, last_order_date from commerce_business.customer_orders co order by name;`. The bottom panel displays the results of the query in a table grid. The table has columns: name, customer_segment, order_count, total_spend, and last_order_date. The results are sorted by name. A right-hand panel shows the 'Value' of the selected cell, 'Aaron Carlson'.

	name	customer_segment	order_count	total_spend	last_order_date
1	Aaron Carlson	Loyal	6	22,883	2024-05-22
2	Aaron Jennings	Loyal	8	17,237	2024-07-05
3	Aaron Little	Regular	1	4,249	2022-10-12
4	Aaron Moore	Loyal	6	17,674	2024-07-03
5	Aaron Young	Loyal	6	31,689	2024-10-25
6	Abigail Baird	Loyal	5	14,341	2024-07-19
7	Adam Brooks	Loyal	3	21,923	2024-11-24
8	Adrian Dean	Loyal	6	17,585	2024-07-27
9	Alejandra Price	Regular	4	12,092	2024-03-24
10	Alejandro Deleon	Loyal	8	43,617	2023-05-22
11	Alejandro Martinez	Loyal	11	40,939	2024-11-02
12	Alexa Buck	Loyal	8	27,327	2024-11-01
13	Alexander Davis	Loyal	10	28,496	2024-09-14

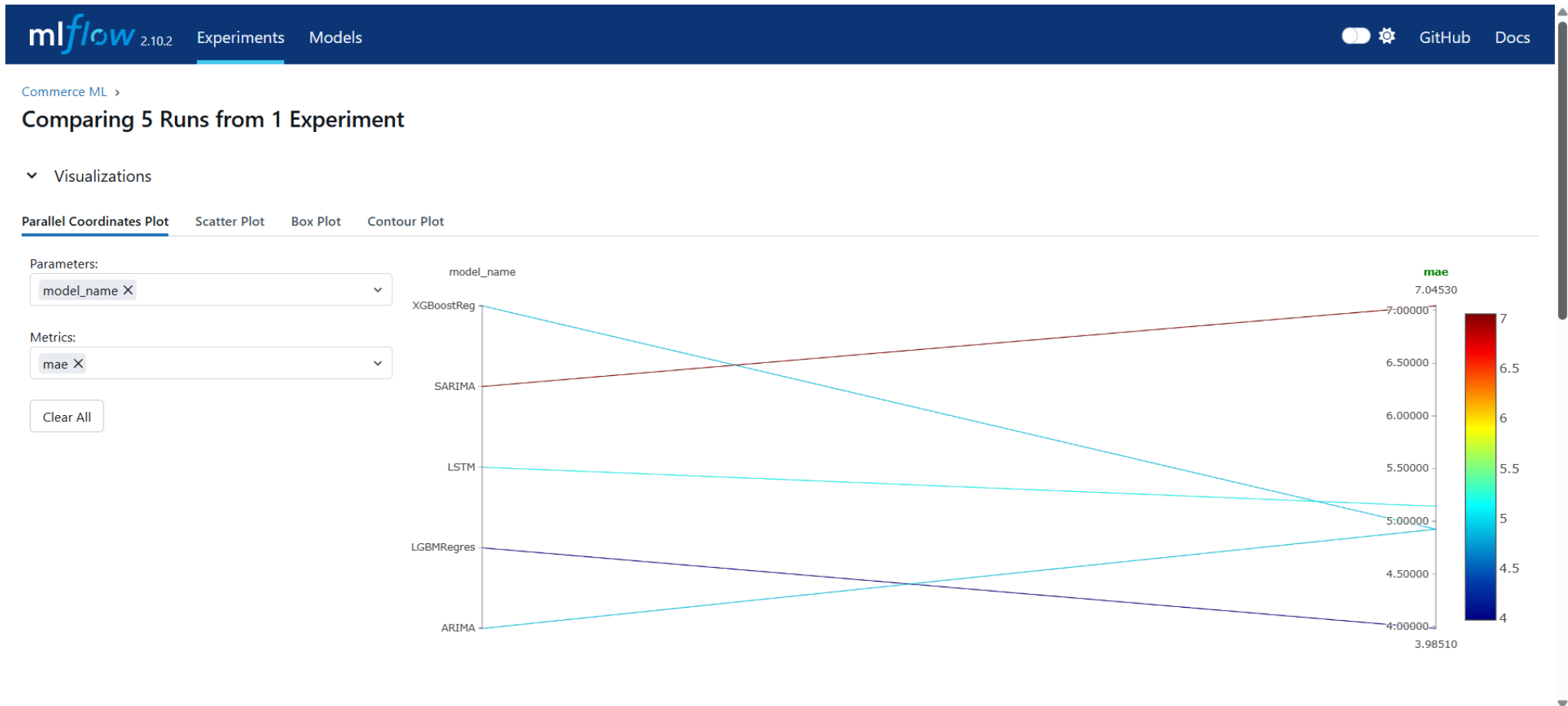
4. Entity Relationship between the tables



5. Power BI Dashboards



6. MLFlow application



7. Machine Learning Models Evaluation

Model Evaluation Statistics					
product_name	mae_arima	mae_sarima	mae_lstm	mae_lgbm	mae_xgb
Acer	4,448	8,405	4,347	4,362	5,55
Coffee Maker	7,185	7,519	7,351	5,945	6,476
Dell	4,946	5,159	5,126	4,403	4,899
Dish Washer	4,064	6,977	4,606	3,698	4,936
Electric Kettle	4,573	3,499	3,591	4,425	3,647
HP	7,4	10,636	6,838	6,244	7,877
iPhone	6,753	8,851	6,728	6,789	7,515
MacBook	6,288	9,581	6,108	6,248	6,966
Microwave Oven	4,575	6,227	4,353	4,087	3,66
Mixer	5,674	8,876	5,805	5,308	5,173
Range Hood	3,741	5,226	4,404	4,194	4,667
Refrigerator	8,251	8,452	8,755	9,916	9,153
Samsung	7,601	9,775	7,872	5,889	6,352
Stove	4,589	5,563	4,375	3,462	4,6
Toaster	4,927	7,045	5,191	3,985	4,926
Xiaomi	6,518	6,798	6,925	6,3	5,387
Average_MAE	5,721	7,412	5,773	5,328	5,737

8. Postman API Testing

The screenshot displays the Postman interface for an API test. At the top, a tab shows the request method as **POST** to the URL `http://127.0.0.1:8000/weekly_predictions`. Below the tab, the request details are visible, including the method **POST** and the same URL. The **Body** tab is selected, showing a JSON payload:

```
1 {
2   "product_name": "Xiaomi",
3   "forward_week_curve": 3
4 }
5
```

On the right side of the request editor, there are buttons for **Save**, **Send**, **Cookies**, and **Beautify**. Below the request editor, the response section is visible, showing the **Body** tab selected. The response status is **200 OK**, with a time of **3.06 s** and a size of **273 B**. The response body is displayed in a **Pretty** JSON format:

```
1 {
2   "ts": "2024-12-30T00:00:00",
3   "prediction": 12
4 },
5 {
6   "ts": "2025-01-06T00:00:00",
7   "prediction": 15
8 },
9 {
10  "ts": "2025-01-13T00:00:00",
11  "prediction": 18
12 }
```

9. Fast API built-in web User Interface

FastAPI 0.1.0 QAS 3.1
[/openapi.json](#)

default

GET / Home

POST /weekly_predictions Get Weekly Predictions

Get a weekly forward demand prediction for a chosen product name. Number of forward curves can be also specified. By default it predicts for the next 1 week

Parameters

Try it out

No parameters

Request body required

application/json

Example Value | Schema

```
{
  "product_name": "string",
  "forward_week_curve": 1
}
```

Responses

Code	Description	Links
------	-------------	-------

200	Successful Response	No links
-----	---------------------	----------

Media type

application/json

Controls Accept header.

Example Value | Schema

```
"string"
```