APPLICATION DOCUMENTATION

This application uses:

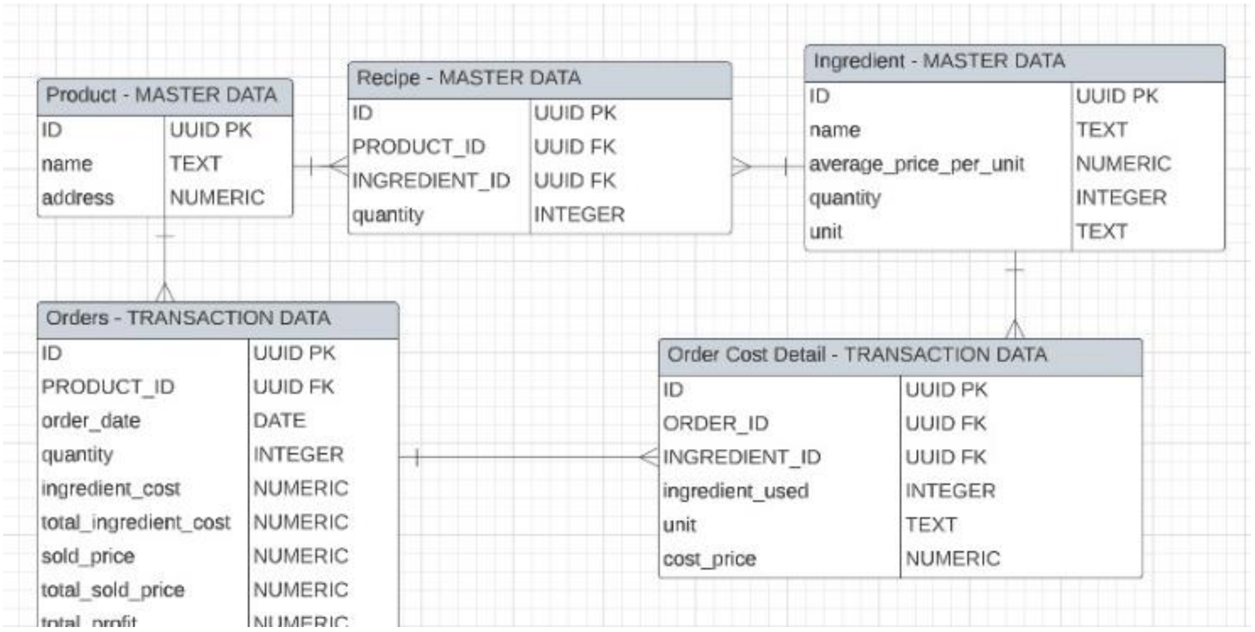
- PostgreSQL as the permanent storage database. The reason for this choice is that the transactional feature is crucial when recording order transactions.
- Elastic Search as the search engine. The reason for this choice is that ES can perform aggregation queries efficiently.

Note:

Although there are several development best practices that I did not implement in this project, it's important to keep them in mind for future projects. For example:
- Avoid returning database entities as API responses; instead, return DTO models.
- Throw specific error exceptions to improve error handling.
- Consider refactoring the code to make it more organized and maintainable.
- Implement a re-index API for Elastic Search to ensure that search results remain up-to-date.

Here I attached some documentation :

- ERD - This diagram shows the entity relationship diagram (ERD) that illustrates the relationships between all tables in the database.



Product - MASTER DATA	
ID	UUID PK
name	TEXT
address	NUMERIC

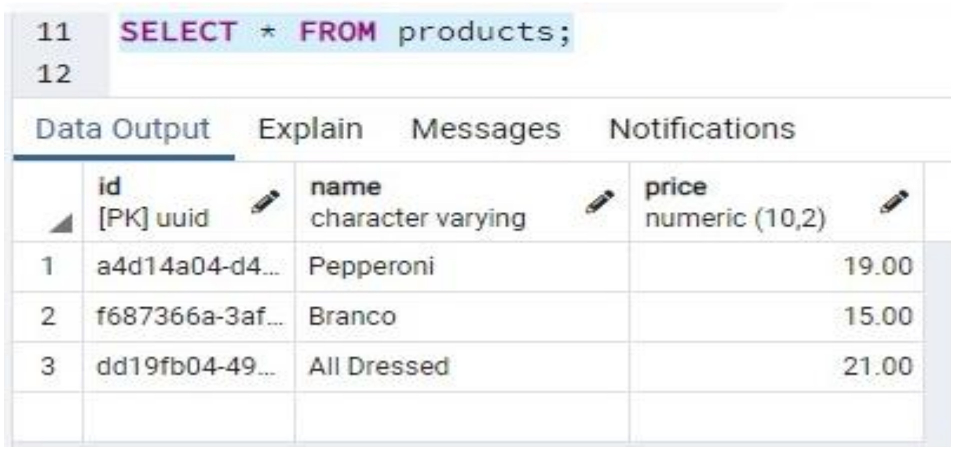
Recipe - MASTER DATA	
ID	UUID PK
PRODUCT_ID	UUID FK
INGREDIENT_ID	UUID FK
quantity	INTEGER

Ingredient - MASTER DATA	
ID	UUID PK
name	TEXT
average_price_per_unit	NUMERIC
quantity	INTEGER
unit	TEXT

Orders - TRANSACTION DATA	
ID	UUID PK
PRODUCT_ID	UUID FK
order_date	DATE
quantity	INTEGER
ingredient_cost	NUMERIC
total_ingredient_cost	NUMERIC
sold_price	NUMERIC
total_sold_price	NUMERIC
total_profit	NUMERIC

Order Cost Detail - TRANSACTION DATA	
ID	UUID PK
ORDER_ID	UUID FK
INGREDIENT_ID	UUID FK
ingredient_used	INTEGER
unit	TEXT
cost_price	NUMERIC

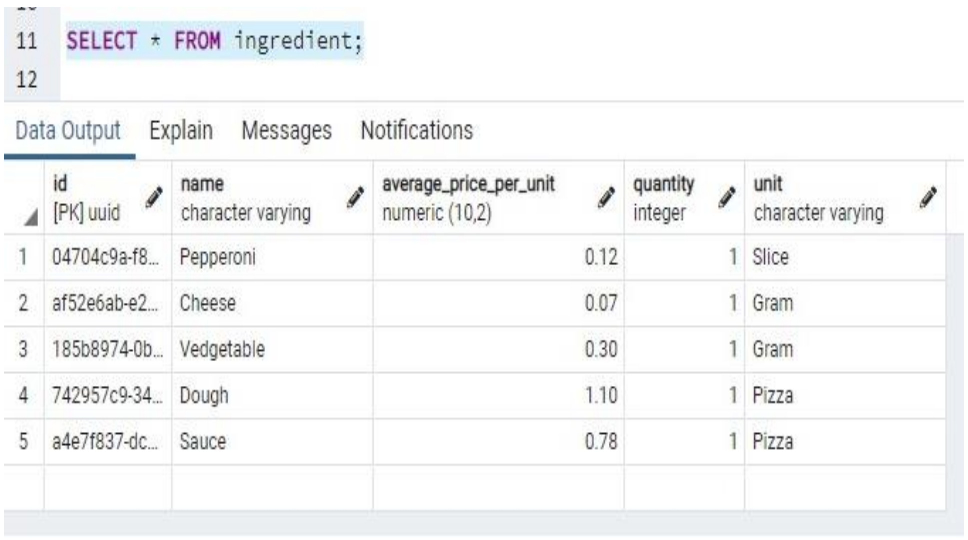
- Sample Data in PostgreSQL - These screenshots show sample data in the PostgreSQL database.



```sql
SELECT * FROM products;
```

Data Output | Explain | Messages | Notifications

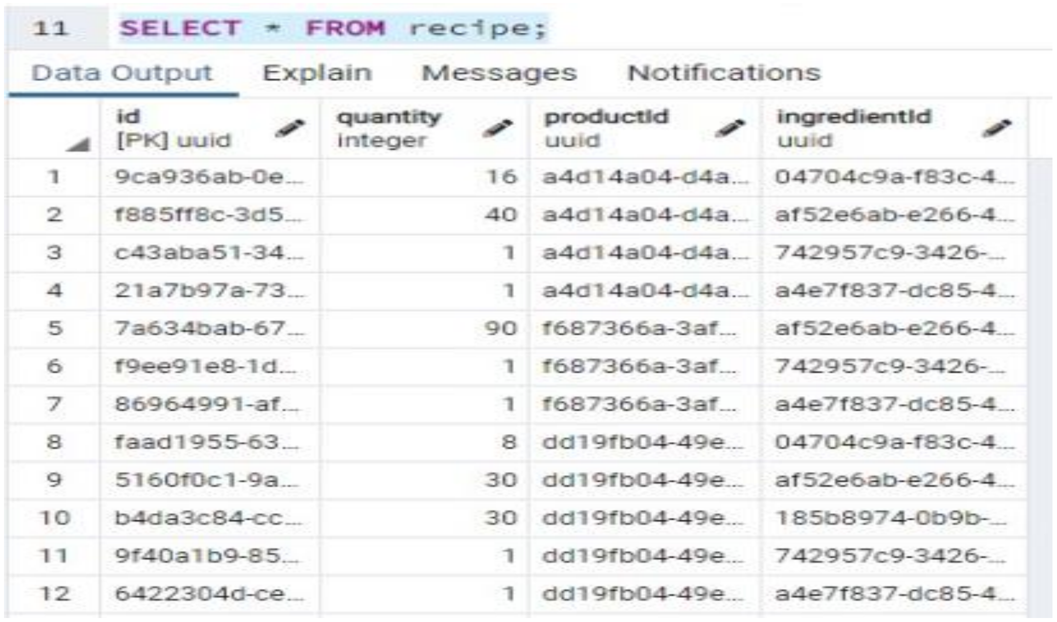
	id [PK] uuid	name character varying	price numeric (10,2)
1	a4d14a04-d4...	Pepperoni	19.00
2	f687366a-3af...	Branco	15.00
3	dd19fb04-49...	All Dressed	21.00



```sql
SELECT * FROM ingredient;
```

Data Output | Explain | Messages | Notifications

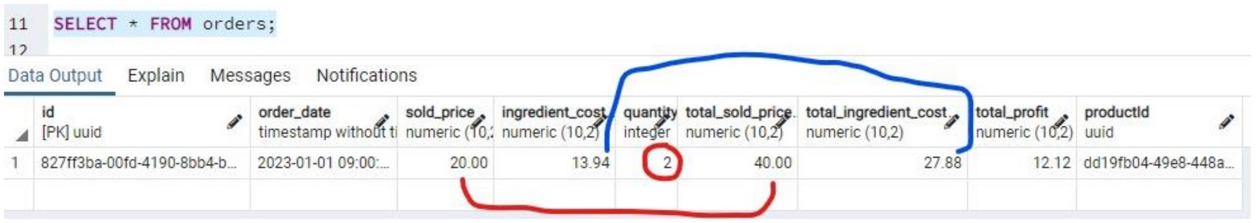
	id [PK] uuid	name character varying	average_price_per_unit numeric (10,2)	quantity integer	unit character varying
1	04704c9a-f8...	Pepperoni	0.12	1	Slice
2	af52e6ab-e2...	Cheese	0.07	1	Gram
3	185b8974-0b...	Vedgetable	0.30	1	Gram
4	742957c9-34...	Dough	1.10	1	Pizza
5	a4e7f837-dc...	Sauce	0.78	1	Pizza



```sql
SELECT * FROM recipe;
```

Data Output | Explain | Messages | Notifications

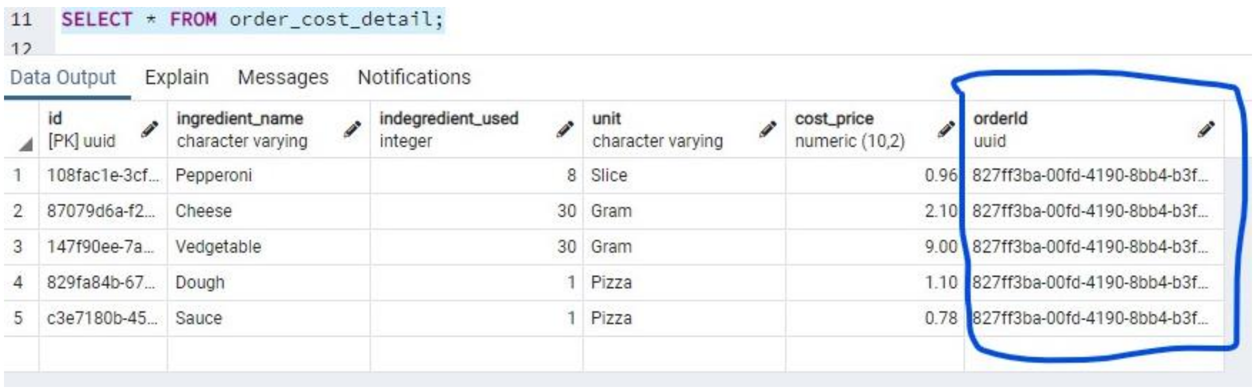
	id [PK] uuid	quantity integer	productId uuid	ingredientId uuid
1	9ca936ab-0e...	16	a4d14a04-d4a...	04704c9a-f83c-4...
2	f885ff8c-3d5...	40	a4d14a04-d4a...	af52e6ab-e266-4...
3	c43aba51-34...	1	a4d14a04-d4a...	742957c9-3426-...
4	21a7b97a-73...	1	a4d14a04-d4a...	a4e7f837-dc85-4...
5	7a634bab-67...	90	f687366a-3af...	af52e6ab-e266-4...
6	f9ee91e8-1d...	1	f687366a-3af...	742957c9-3426-...
7	86964991-af...	1	f687366a-3af...	a4e7f837-dc85-4...
8	faad1955-63...	8	dd19fb04-49e...	04704c9a-f83c-4...
9	5160f0c1-9a...	30	dd19fb04-49e...	af52e6ab-e266-4...
10	b4da3c84-cc...	30	dd19fb04-49e...	185b8974-0b9b-...
11	9f40a1b9-85...	1	dd19fb04-49e...	742957c9-3426-...
12	6422304d-ce...	1	dd19fb04-49e...	a4e7f837-dc85-4...



```sql
SELECT * FROM orders;
```

Data Output | Explain | Messages | Notifications

	id [PK] uuid	order_date timestamp without ti	sold_price numeric (10,2	ingredient_cost numeric (10,2)	quantity integer	total_sold_price numeric (10,2)	total_ingredient_cost numeric (10,2)	total_profit numeric (10,2)	productId uuid
1	827ff3ba-00fd-4190-8bb4-b...	2023-01-01 09:00:...	20.00	13.94	2	40.00	27.88	12.12	dd19fb04-49e8-448a...

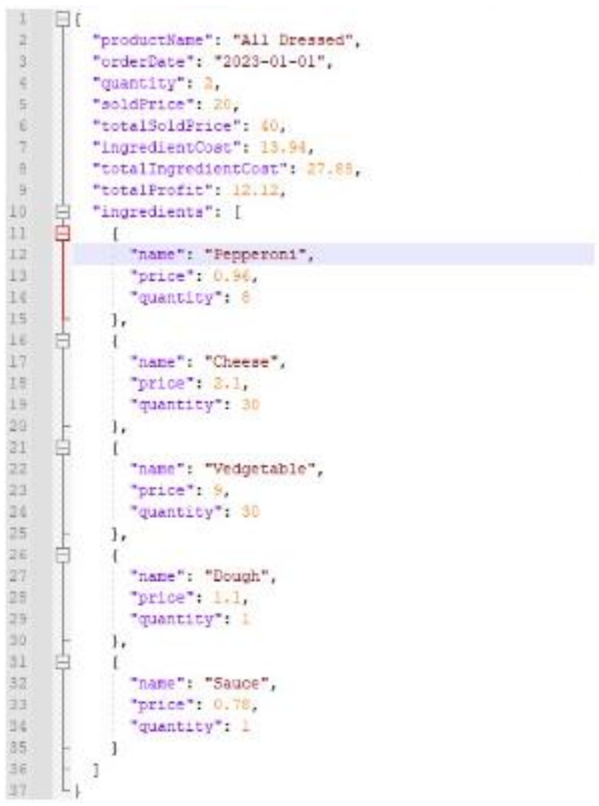


```sql
SELECT * FROM order_cost_detail;
```

Data Output | Explain | Messages | Notifications

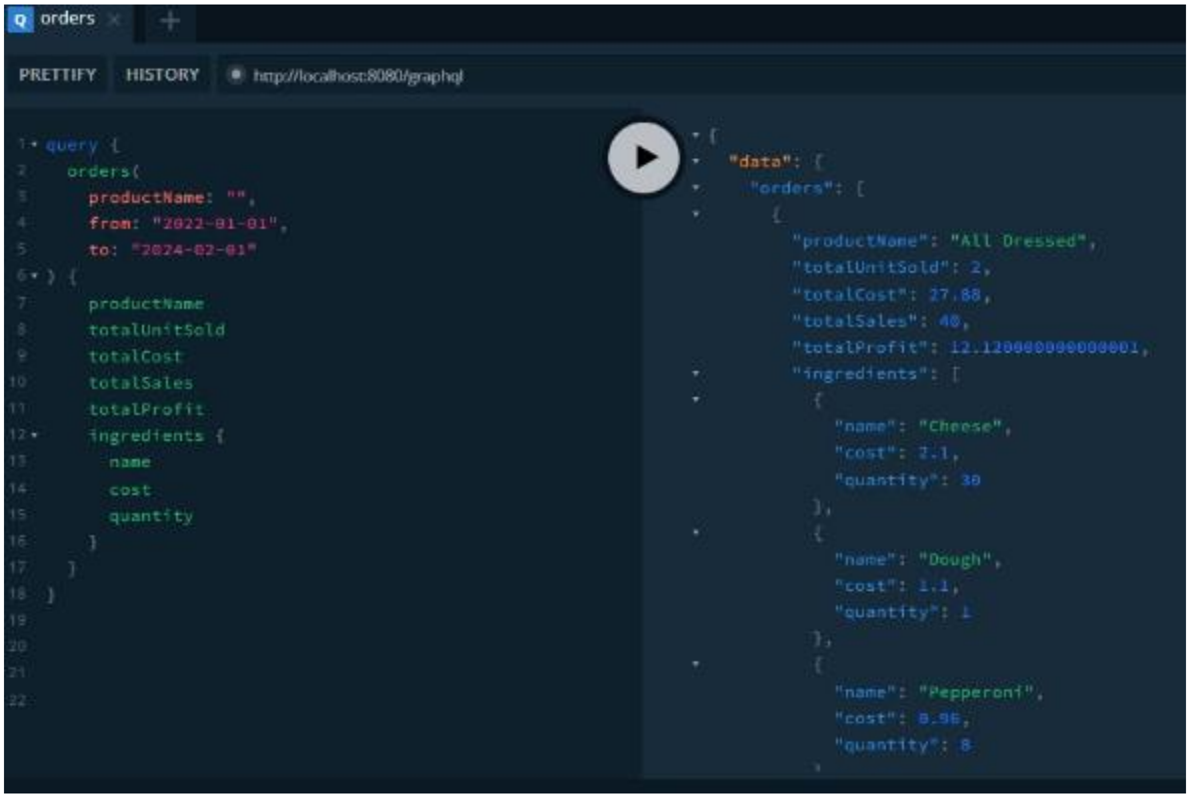
	id [PK] uuid	ingredient_name character varying	indegredient_used integer	unit character varying	cost_price numeric (10,2)	orderId uuid
1	108fac1e-3cf...	Pepperoni	8	Slice	0.96	827ff3ba-00fd-4190-8bb4-b3f...
2	87079d6a-f2...	Cheese	30	Gram	2.10	827ff3ba-00fd-4190-8bb4-b3f...
3	147f90ee-7a...	Vedgetable	30	Gram	9.00	827ff3ba-00fd-4190-8bb4-b3f...
4	829fa84b-67...	Dough	1	Pizza	1.10	827ff3ba-00fd-4190-8bb4-b3f...
5	c3e7180b-45...	Sauce	1	Pizza	0.78	827ff3ba-00fd-4190-8bb4-b3f...

- Sample Data in Elastic Search - This is a screenshot of a document stored in Elastic Search.



```json
{
   "productName": "All Dressed",
   "orderDate": "2023-01-01",
   "quantity": 2,
   "soldPrice": 20,
   "totalSoldPrice": 40,
   "ingredientCost": 13.94,
   "totalIngredientCost": 27.88,
   "totalProfit": 12.12,
   "ingredients": [
     {
       "name": "Pepperoni",
       "price": 0.96,
       "quantity": 8
     },
     {
       "name": "Cheese",
       "price": 2.1,
       "quantity": 30
     },
     {
       "name": "Vedgetable",
       "price": 9,
       "quantity": 30
     },
     {
       "name": "Dough",
       "price": 1.1,
       "quantity": 1
     },
     {
       "name": "Sauce",
       "price": 0.78,
       "quantity": 1
     }
   ]
}
```

- GraphQL - This is a screenshot of a GraphQL request and response.



```
query {
  orders(
    productName: "",
    from: "2022-01-01",
    to: "2024-02-01"
  ) {
    productName
    totalUnitSold
    totalCost
    totalSales
    totalProfit
    ingredients {
      name
      cost
      quantity
    }
  }
}
```

```
{
  "data": {
    "orders": [
      {
        "productName": "All Dressed",
        "totalUnitSold": 2,
        "totalCost": 27.88,
        "totalSales": 40,
        "totalProfit": 12.120000000000001,
        "ingredients": [
          {
            "name": "Cheese",
            "cost": 2.1,
            "quantity": 30
          },
          {
            "name": "Dough",
            "cost": 1.1,
            "quantity": 1
          },
          {
            "name": "Pepperoni",
            "cost": 0.96,
            "quantity": 8
```