

TECHNICAL TEST - DATA ENGINEER

TRIPUTRA AGRO PERSADA

1. Buat akun Snowflake Free Trial

2. Download dataset dari Kaggle

<https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce>

Tables yang digunakan:

- orders.csv
- order_items.csv
- customers.csv
- products.csv
- payments.csv

I

3. Upload CSV ke Snowflake stage

4. Create raw tables:

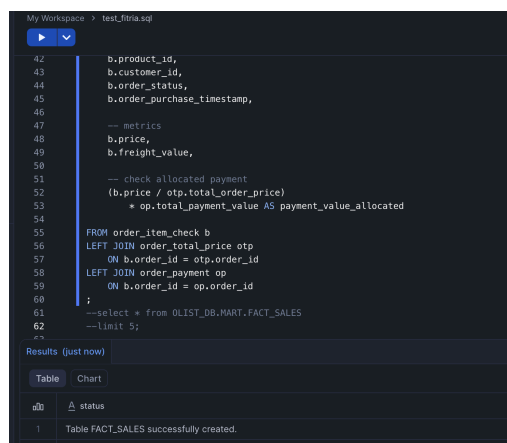
- raw_orders
- raw_order_items
- raw_customers
- raw_products
- raw_payments

Task 1

Buat:

- 1 fact table: fact_sales

Jawab : Create Table Fact_sales



```
My Workspace > test_facta.sql
42      b.product_id,
43      b.customer_id,
44      b.order_status,
45      b.order_purchase_timestamp,
46
47      -- metrics
48      b.price,
49      b.freight_value,
50
51      -- check allocated payment
52      (b.price / otp.total_order_price)
53      * op.total_payment_value AS payment_value_allocated
54
55 FROM order_item_check b
56 LEFT JOIN order_total_price otp
57     ON b.order_id = otp.order_id
58 LEFT JOIN order_payment op
59     ON b.order_id = op.order_id
60 ;
61 --select * from OLIST_DB.MART.FACT_SALES
62 --limit 5;
63

Results (just now)
Table  Chart
all status
1 Table FACT_SALES successfully created.
```

limit 5 fact tables

ORDER_ID	ORDER_ITEM_ID	PRODUCT_ID	CUSTOMER_ID	ORDER_STATUS	ORDER_PURCHASE_TIMESTAMP
00010242e8c5a6f0a20d792cb16214	1	4244733e06e7ecb4870a8e2683c13e81	3ce436f183e68e078776285a838db11a	delivered	2017-09-13 08:59:02.000
00010187727020c557190d7a144b0d3	1	e5f2d52b08218ee858865ca93a83a8f	f6dd3ec061db4e3987629fe6b28e5cce	delivered	2017-04-26 10:53:06.000
000229ec398224efc6a657da4fc703e	1	c777355d18b72b67ab0eef9d1441d0d	6489a5e433f3693df5ad4372ab6d3	delivered	2018-01-14 14:33:31.000
00024acbcdf0a6dsafe931b038114c75	1	7634da152a4610f1595efc32174722fc	d4eb9395c8c0431ee92fce09860c5a06	delivered	2018-08-08 10:00:35.000
00042b26cf59d7ce69dabb4e55b4fd9	1	ac6c3623068f309e03045865e4e10089	58dbd0d2d70206b040e62cd34e84d795	delivered	2017-02-04 13:57:51.000

penjelasan : fact_sale merepresentasikan aktivitas penjualan (sales) pada level order item, di mana 1 baris = 1 produk yang terjual dalam satu order.

Fact table ini menjadi sumber utama untuk analisis revenue, volume penjualan, dan performa customer maupun produk.

Sumber yang saya gunakan meliputi :

- RAW_ORDER_ITEM (pondasi), karena terdapat relasi order terhadap product, terdapat harga per item beserta biaya pengirimannya.
- RAW_ORDER (informasi order), karena terdapat relasi terhadap customer, status order dan waktu pembelian
- RAW Customer (informasi customer), table ini digunakan untuk memberikan detail customer yang melakukan pemesanan dan akan menjadi relasi untuk pembuatan dim_customer
- RAW PAYMENT (informasi finansial), karena payment berada di level order (1 row per order), sedangkan fact_sale berada di level order item (1 row per product per order), dilakukan pendekatan.

$\text{allocated_payment} = (\text{item_price} / \text{total_order_price}) \times \text{total_payment_value}$

o Dimension tables:

§ dim_customers

Jawab :

```

80
81 CREATE OR REPLACE TABLE OLIST_DB.MART.DIM_CUSTOMER AS
82 SELECT DISTINCT
83   TRIM(UPPER(customer_id)) AS customer_id,
84   customer_unique_id,
85   customer_zip_code_prefix,
86   INITCAP(customer_city) AS customer_city,
87   UPPER(customer_state) AS customer_state
88 FROM OLIST_DB.RAW.RAW_CUSTOMER;
89
90
91

```

Results (just now)

Table Chart

status

1 Table DIM_CUSTOMER successfully created.

Penjelasan : pada dim_customer, dilakukan data cleansing pada script diatas dan dilakukan pengecekan jumlah terhadap fact_sales untuk

menghindari data yang tidak sesuai

```

13  -- OLS1_DB_MART_DIM_CUSTOMER
14  SELECT COUNT(*)
15  FROM MART_FACT_SALES f
16  LEFT JOIN MART_DIM_CUSTOMER c
17    ON f.customer_id = c.customer_id
18  WHERE c.customer_id IS NULL;
19
20  +-----+
21  | Add to Chat | Explain | Quick edit | Format |
22  +-----+
23
24  /CREATE OR REPLACE TABLE OLS1_DB_MART_DIM_CUSTOMER AS
25  SELECT DISTINCT
26    TRIM(SUBSTR(customer_id, 1, 10)) AS customer_id,
27    customer_unique_id,
28    customer_zip_code_prefix,
29    INITCAP(customer_city) AS customer_city,
30    MD5(SUBSTR(customer_id || customer_city, 1, 30))
31  FROM MART_FACT_SALES f
32  WHERE customer_id IS NULL;
33
34  results (just now)
35
36  Table Chart
37
38  # COUNT(*)
39
40  1 row 0 241ms

```

§ dim_products

```
90 CREATE OR REPLACE TABLE OLIST_DB.MART.DIM_PRODUCT AS
91 SELECT DISTINCT
92     TRIM(UPPER(product_id))           AS product_id,
93     INITCAP(product_category_name)    AS product_category_name,
94     product_name_length,
95     product_description_length,
96     product_photos_qty,
97     product_weight_g,
98     product_length_cm,
99     product_height_cm,
00     product_width_cm
01 FROM OLIST_DB.RAW.RAW_PRODUCT;
02
03
```

Results (just now)

Table Chart

status

Table DIM_PRODUCT successfully created.

Penjelasan : Dilakukan cleansing data dan cross check agak tidak ada duplicat

```
103
104 SELECT
105     COUNT(*) AS total_rows,
106     COUNT(DISTINCT product_id) AS unique_product
107 FROM OLIST_DB.MART.DIM_PRODUCT;
108
```

Results (just now)

Table	Chart	1 row	183ms	Download	Refresh
# TOTAL_ROWS	# UNIQUE_PRODUCT	1	32951	32951	

§ dim date

jawab :

```

151 CREATE OR REPLACE TABLE OLIST_DB.MART.DIM_DATE AS
152 SELECT DISTINCT
153     CAST(order_purchase_timestamp AS DATE) AS date_id,
154     DAY(order_purchase_timestamp) AS day,
155     MONTH(order_purchase_timestamp) AS month,
156     INITCAP(TO_CHAR(order_purchase_timestamp,'Month')) AS month_name,
157     'Q' || TO_CHAR(order_purchase_timestamp,'0') AS quarter,
158     YEAR(order_purchase_timestamp) AS year,
159     DAYOFWEEKISO(order_purchase_timestamp) AS day_of_week,
160     CASE WHEN DAYOFWEEKISO(order_purchase_timestamp) IN (6,7) THEN TRUE ELSE FALSE END AS is_weekend,
161     INITCAP(TO_CHAR(order_purchase_timestamp,'Day')) AS day_name,
162     WEEKISO(order_purchase_timestamp) AS week_of_year
163 FROM OLIST_DB.RAW.RAW_ORDERS;
164
165 | select * from OLIST_DB.MART.dim_date
166 | limit 10

```

Results (just now)

Table Chart

10 rows 126ms

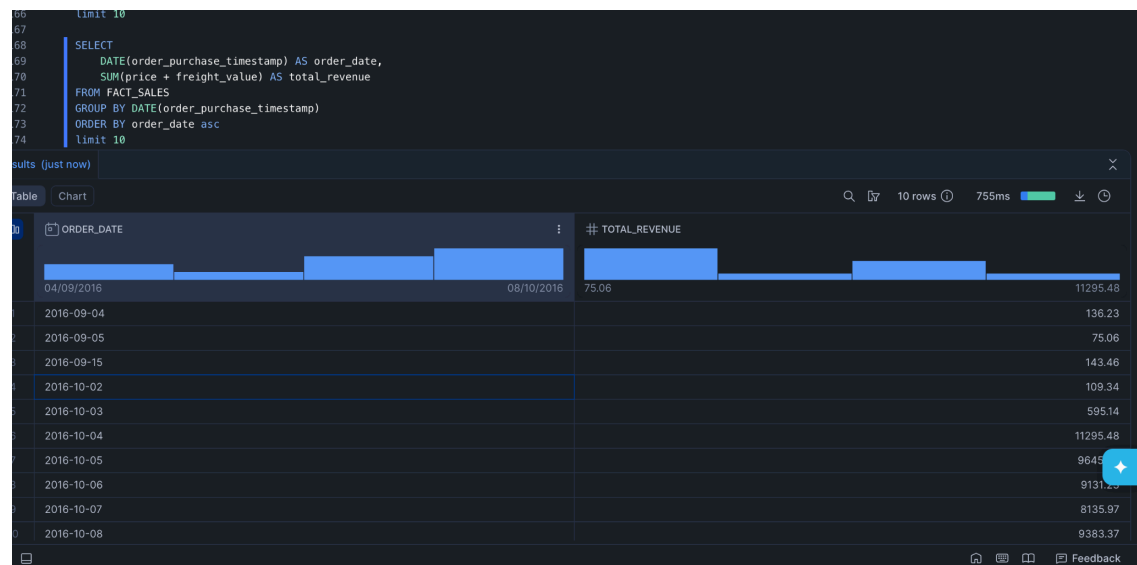
	DATE_ID	# DAY	# MONTH	MONTH-NAME	QUARTER	# YEAR	# DAY-OF-WEEK	IS_WEEKEND	DAY-NAME	# WEEK-OF-YEAR
				Augth 20.0% Febth 20.0% +5 more	QQ 100.0%			false 100.0%	Day 100.0%	
1	2017-07-13	13	7	Julth	QQ	2017	4	false	Day	28
2	2018-06-07	7	6	Junth	QQ	2018	4	FALSE	Day	23
3	2018-03-15	15	3	Marth	QQ	2018	4	FALSE	Day	11
4	2018-01-08	8	1	Janth	QQ	2018	1	FALSE	Day	

Task 2

Buat SQL query untuk menjawab:

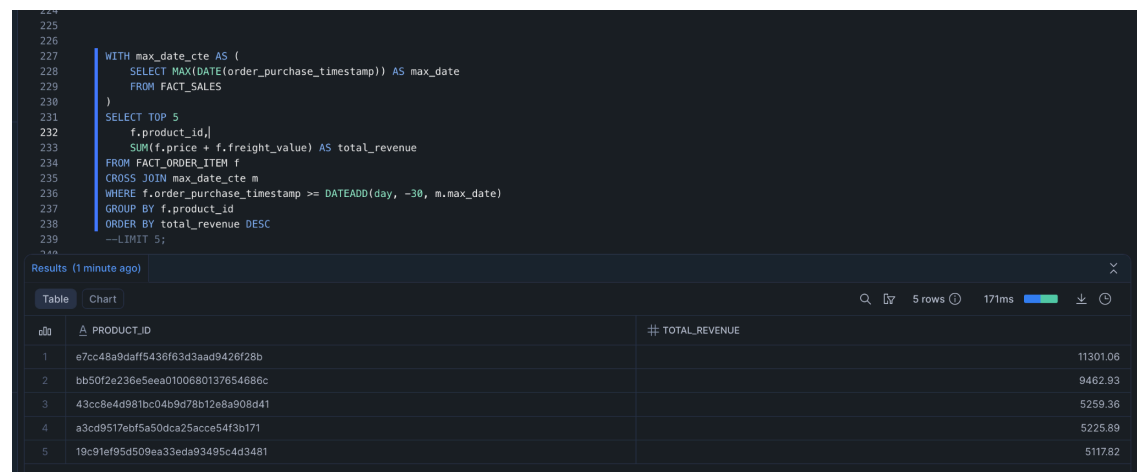
1. Total revenue per day

Jawab :



2. Top 5 products by revenue in last 30 days

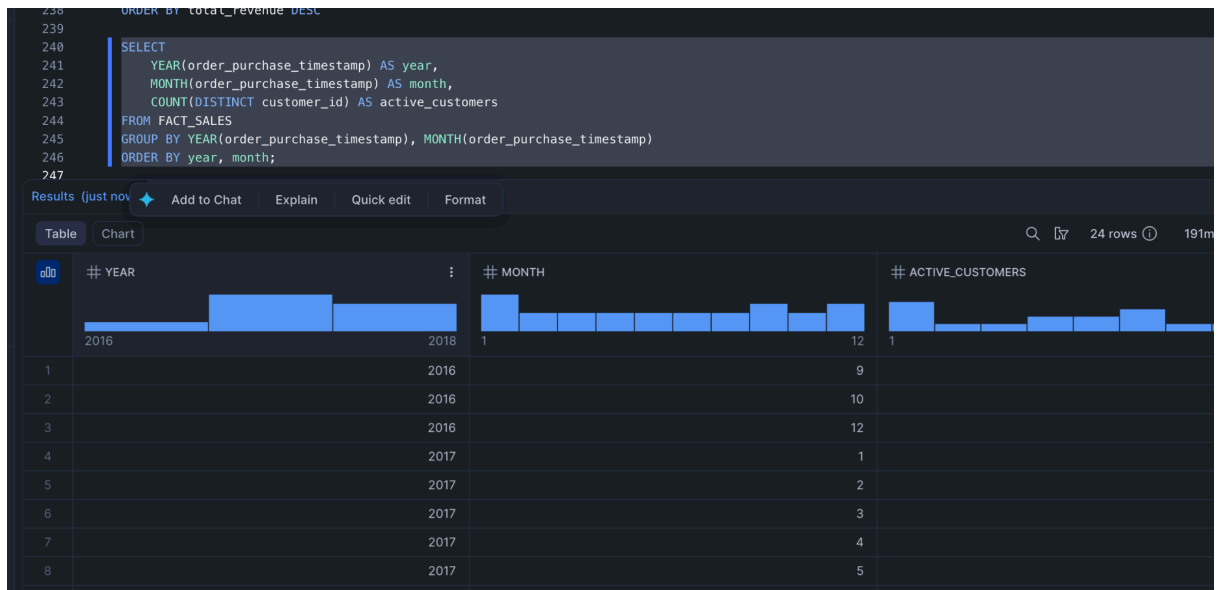
Jawab :



Penjelasan : disini untuk revenue 30 hari terhari, menggunakan 30 hari terakhir dari Maxdate

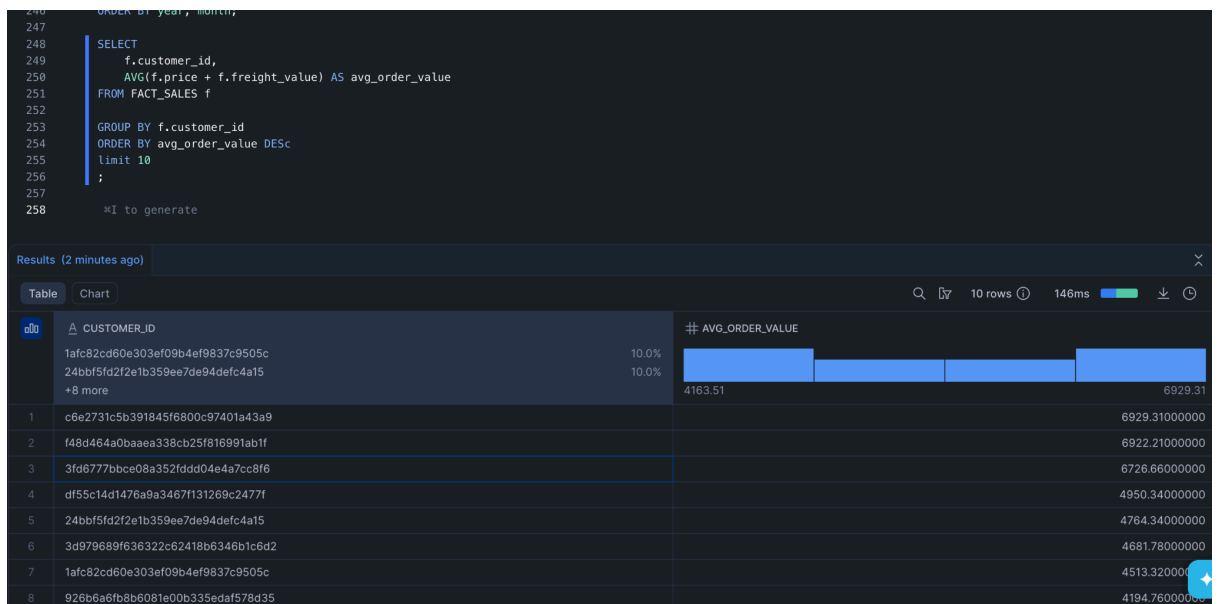
3. Monthly active customers

Jawab



4. Average order value per customer

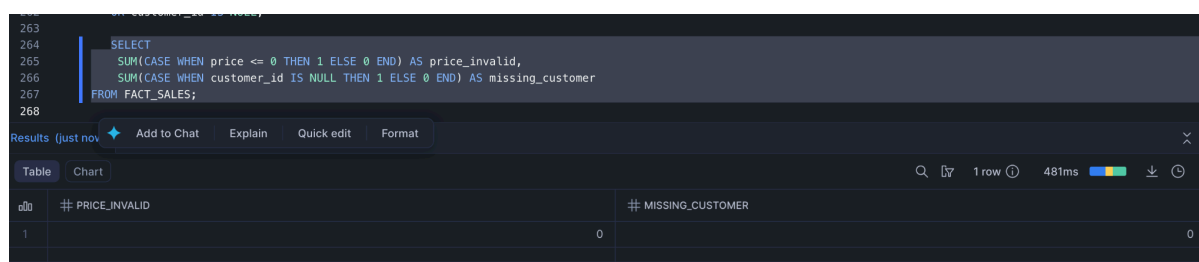
Jawab :



5. Data anomaly:

- price <= 0
- quantity <= 0
- missing customer_id

Jawab :



Task 3

Buat dbt project dengan struktur:

models/

staging/

stg_orders.sql

stg_customers.sql

stg_products.sql

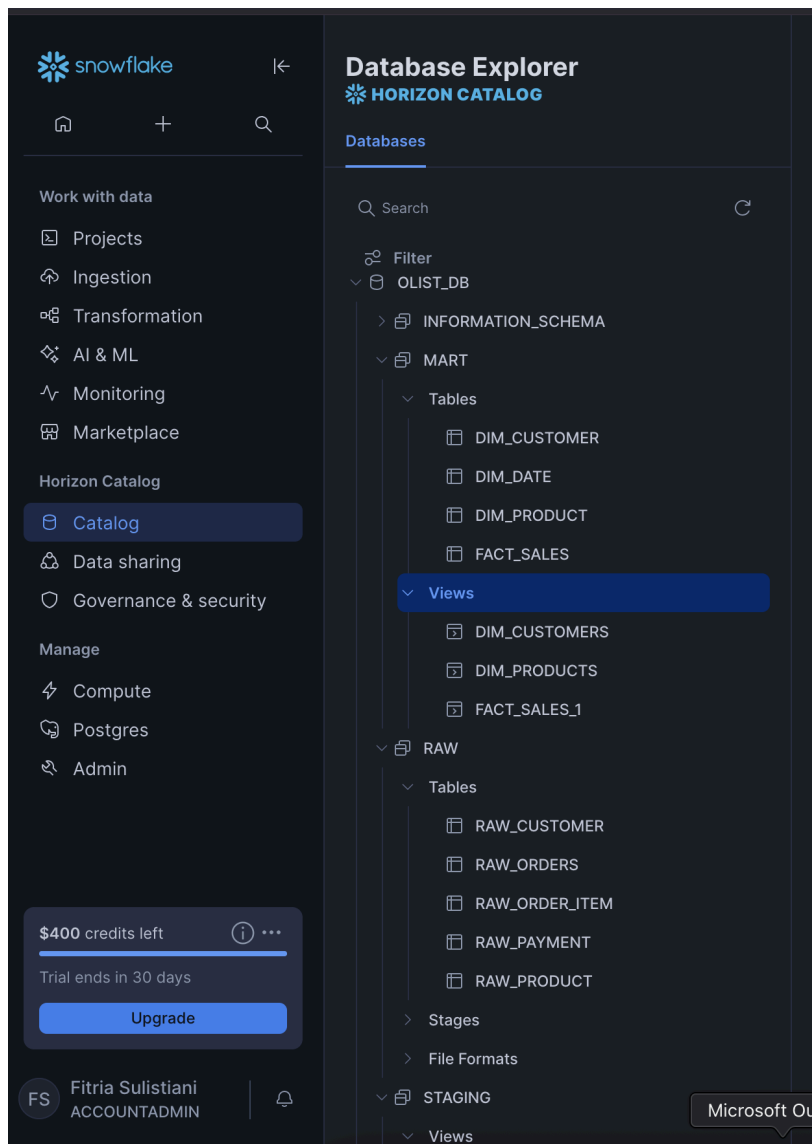
marts/

dim_customers.sql

dim_products.sql

fact_sales.sql

jawab



penjelasan: untuk dbt saya gagal dalam koneksi dilocal, sehingga saya coba bikin alternatif disnowflake

Task 4

Buat file schema.yml berisi tests berikut:

- not_null pada primary key
- unique pada order_id
- accepted_values pada order_status
- relationships:
 - fact_sales.customer_id → dim_customers.customer_id
 - fact_sales.product_id → dim_products.product_id

Jawab:

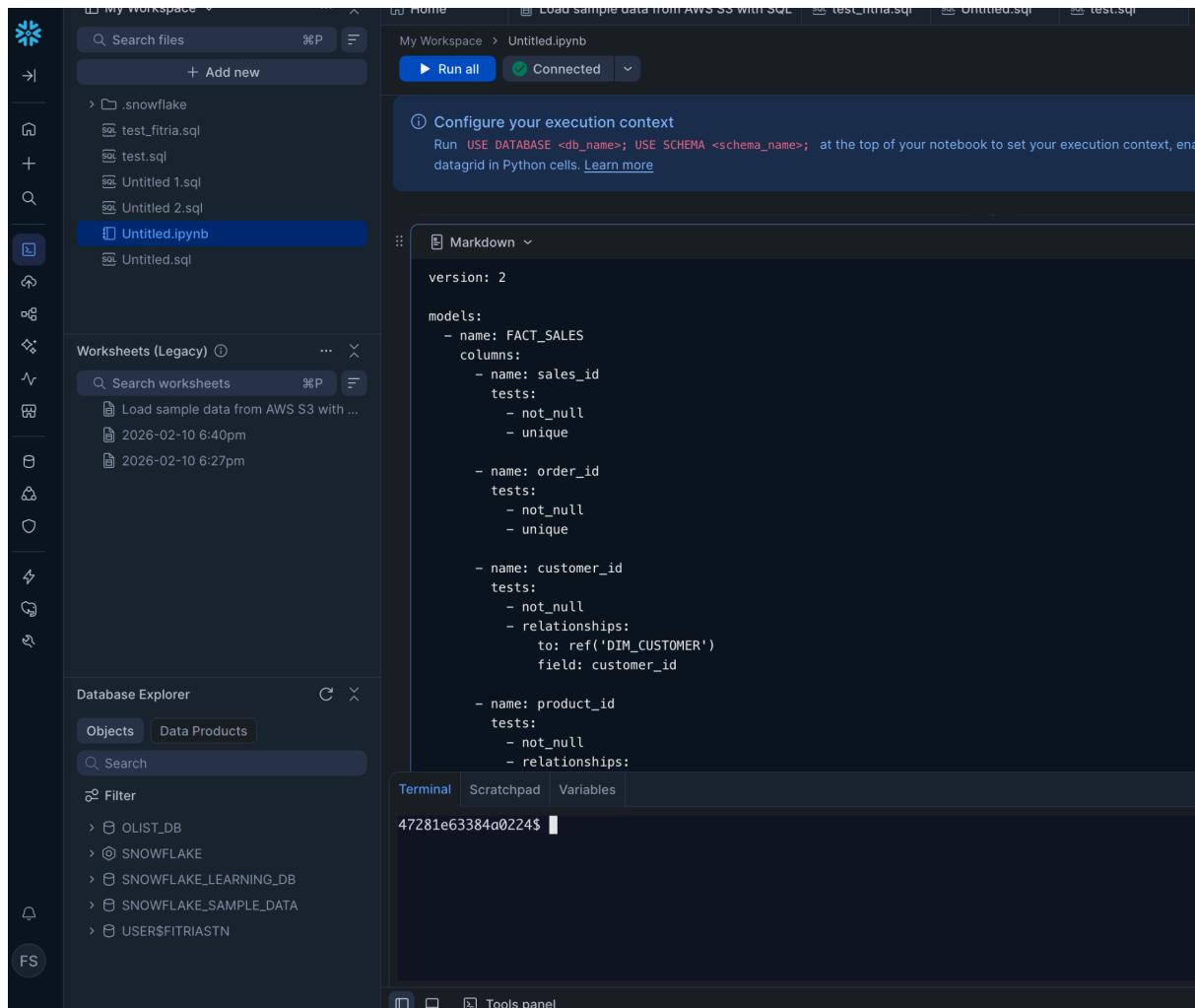
models:

- name: FACT_SALES
columns:
 - name: sales_id
tests:
 - not_null
 - unique
- name: order_id
tests:
 - not_null
 - unique
- name: customer_id
tests:
 - not_null
 - relationships:
 - to: ref('DIM_CUSTOMER')
 - field: customer_id
- name: product_id
tests:
 - not_null
 - relationships:
 - to: ref('DIM_PRODUCT')
 - field: product_id

- name: order_status
 - tests:
 - not_null
 - accepted_values:
 - values: ['pending', 'completed', 'canceled', 'returned']

- name: DIM_CUSTOMER
 - columns:
 - name: customer_id
 - tests:
 - not_null
 - unique

- name: DIM_PRODUCT
 - columns:
 - name: product_id
 - tests:
 - not_null
 - unique



penjelasan : lanjutan dari task 3, saya masih gagal dalam mebuat koneksi

DELIVERABLE

GitHub repository berisi:

1. SQL scripts
2. dbt project
3. Screenshot :
 - o dbt run
 - o dbt test
 - o Snowflake tables