Answers to the Analytical Questions

Staging models

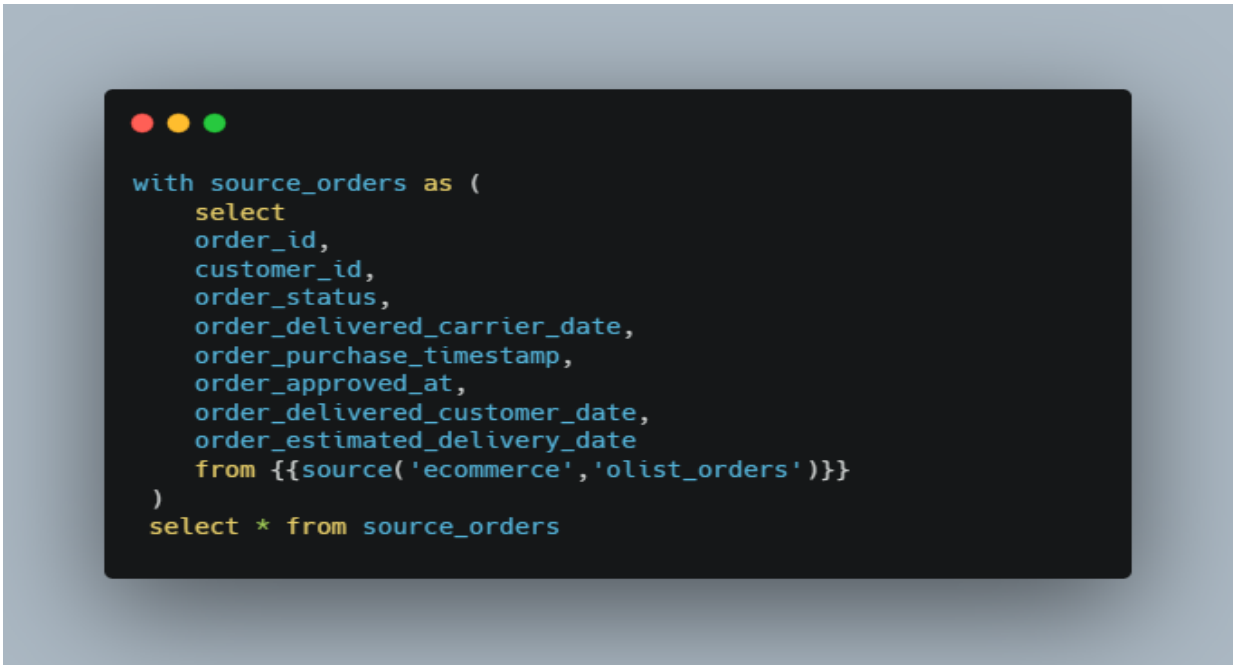
The staging models are

- stg_customers.sql
- stg_orders.sql
- stg_order_items.sql
- stg_products.sql

stg_customers.sql

```
{{ config(materialized ='view')}}
with source_customer as (
    select
    customer_id,
    customer_zip_code_prefix,
    customer_city,
    customer_state

    from {{source('ecommerce','olist_customers')}}
 )
 select * from source_customer
```

stg_orders.sql



```sql
with source_orders as (
    select
    order_id,
    customer_id,
    order_status,
    order_delivered_carrier_date,
    order_purchase_timestamp,
    order_approved_at,
    order_delivered_customer_date,
    order_estimated_delivery_date
    from {{source('ecommerce','olist_orders')}}
 )
 select * from source_orders
```

Stg_order_items.sql

```sql
with source as (
    select
    order_id,
    order_item_id,
    product_id,
    seller_id,
    shipping_limit_date,
    price,
    freight_value
    from {{source('ecommerce','olist_order_items')}}
 )
 select * from source
```

stg_products.sql

```sql
with source as (
    select
    product_id,
    product_category_name

    from {{source('ecommerce','olist_products')}}
 )
 select * from source
```

Staging models are intermediary data models used in data processing and ETL (Extract, Transform, Load) pipelines. They serve as temporary storage areas where raw data is cleaned, transformed, and prepared before being moved to more refined models or the final data warehouse.

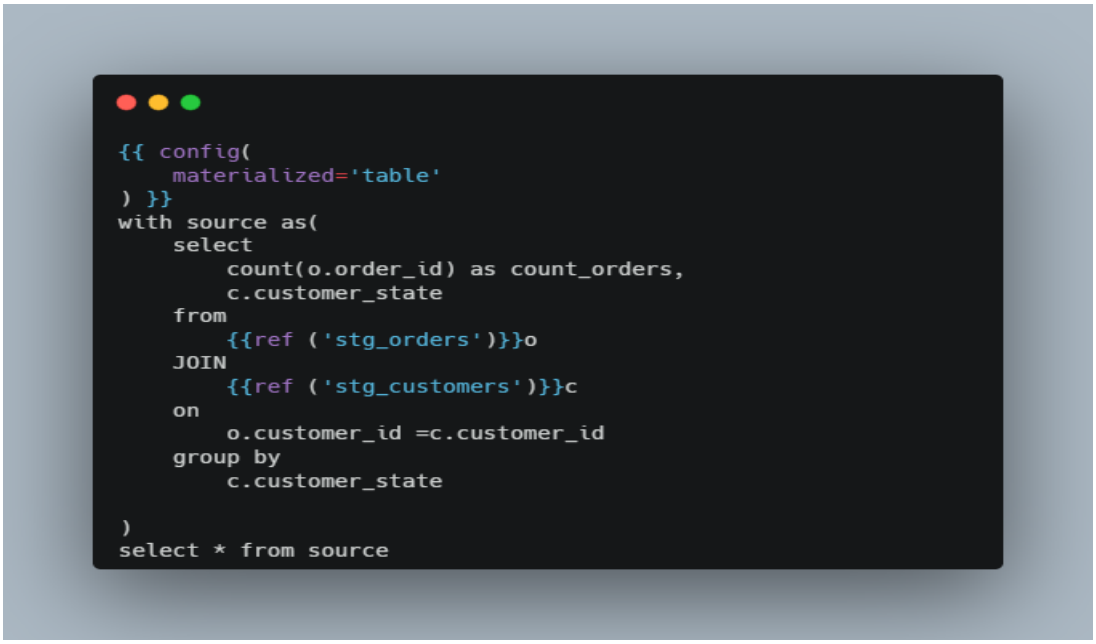
In this stage, the tables needed for the transformation were the four tables and the columns needed for the modelling were selected as well.

Intermediate Models

Int_sales_by_category.sql

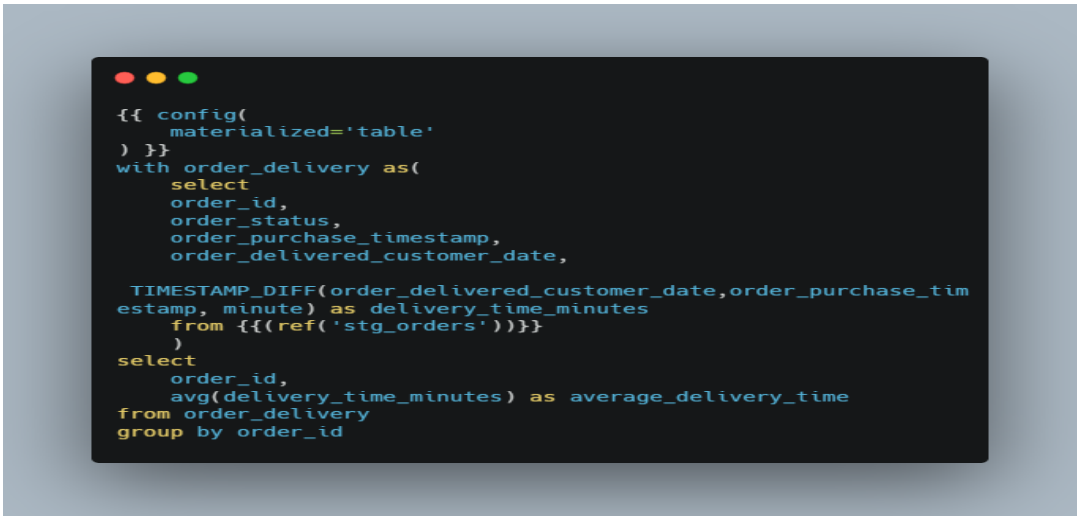
```sql
{{ config(
    materialized='table'
) }}
WITH product_data AS (
    SELECT
        p.product_category_name,
        SUM(o.price) AS total_price
    FROM
        {{ ref('stg_products') }} p
    JOIN
        {{ ref('stg_order_items') }} o
    ON
        p.product_id = o.product_id
    JOIN
        {{ ref('stg_orders') }} oo
    ON
        oo.order_id = o.order_id
    GROUP BY
        p.product_category_name
)

SELECT * FROM product_data
```

int_orders_by_state.sql



```sql
{{ config(
    materialized='table'
) }}
with source as(
    select
        count(o.order_id) as count_orders,
        c.customer_state
    from
        {{ref ('stg_orders')}}o
    JOIN
        {{ref ('stg_customers')}}c
    on
        o.customer_id =c.customer_id
    group by
        c.customer_state

)
select * from source
```

Int_avg_delivery_time.sql



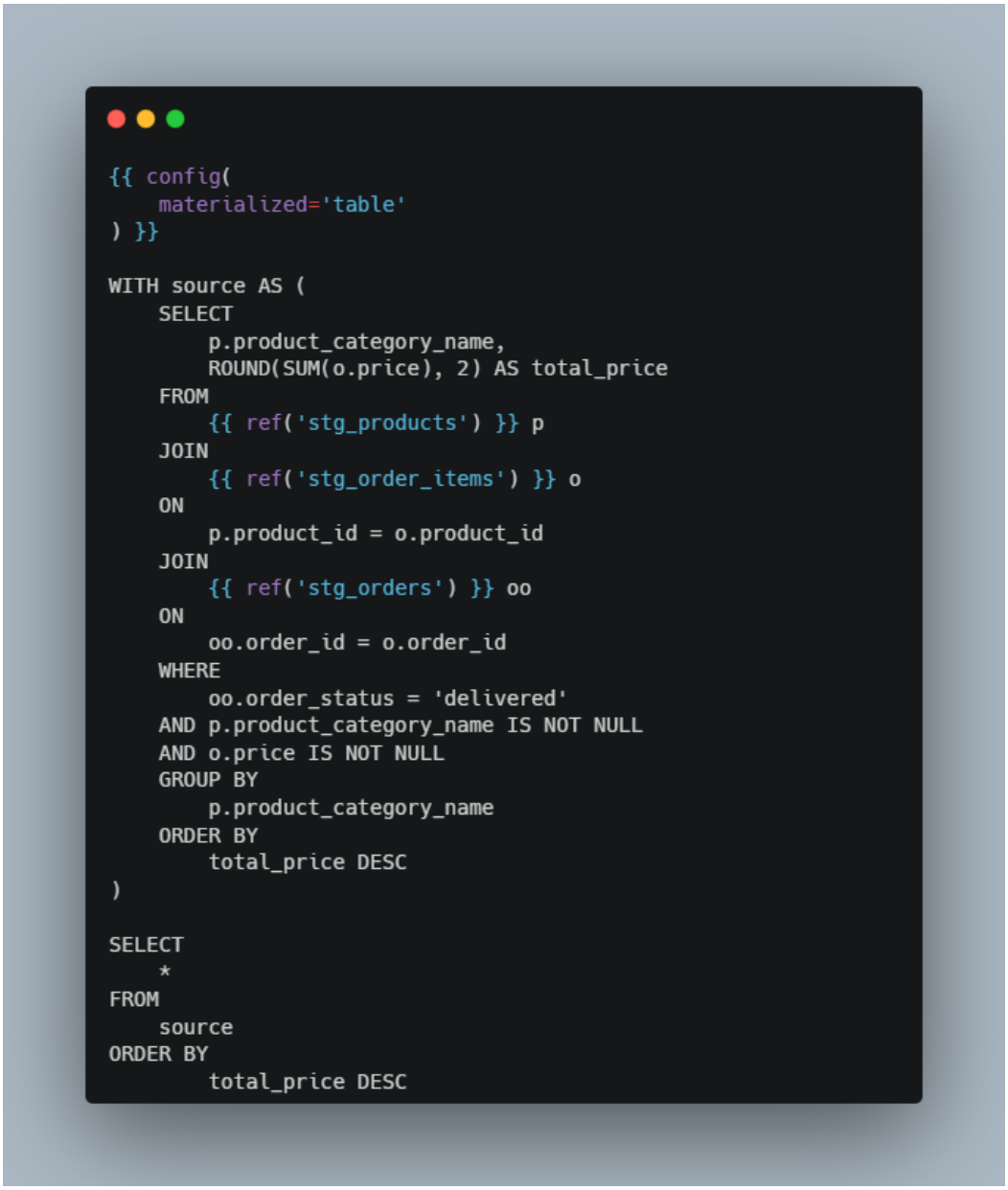
```
{{ config(
    materialized='table'
) }}
with order_delivery as(
    select
    order_id,
    order_status,
    order_purchase_timestamp,
    order_delivered_customer_date,

 TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_tim
estamp, minute) as delivery_time_minutes
    from {{(ref('stg_orders'))}}
    )
select
    order_id,
    avg(delivery_time_minutes) as average_delivery_time
from order_delivery
group by order_id
```

For the intermediate models:

- int_sales_by_category.sql
- int_avg_delivery_time.sql
- int_order_by_state.sql

Final Models

fct_int_sales_by_category.sql



```
{{ config(
    materialized='table'
) }}

WITH source AS (
    SELECT
        p.product_category_name,
        ROUND(SUM(o.price), 2) AS total_price
    FROM
        {{ ref('stg_products') }} p
    JOIN
        {{ ref('stg_order_items') }} o
    ON
        p.product_id = o.product_id
    JOIN
        {{ ref('stg_orders') }} oo
    ON
        oo.order_id = o.order_id
    WHERE
        oo.order_status = 'delivered'
    AND p.product_category_name IS NOT NULL
    AND o.price IS NOT NULL
    GROUP BY
        p.product_category_name
    ORDER BY
        total_price DESC
)

SELECT
    *
FROM
    source
ORDER BY
        total_price DESC
```

Fct_avg_delivery_time

```
{{ config(
    materialized='table'
) }}
with fct_order_delivery as(
    select
    order_id,
    order_status,
    order_purchase_timestamp,
    order_delivered_customer_date,

 TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_tim
estamp, minute) as delivery_time_minutes
    from {{(ref('stg_orders'))}}
    where order_status ='delivered'
    and
TIMESTAMP_DIFF(order_delivered_customer_date,order_purchase_time
stamp, minute) is not null
    )
select
    order_id,
    avg(delivery_time_minutes) as average_delivery_time
from fct_order_delivery
group by order_id
order by average_delivery_time desc
```

Fct_Orders_by_state

```
{{ config(
    materialized='table'
) }}

with source as(
    select
        count(o.order_id) as count_orders,
        c.customer_state
    from
        {{ref ('stg_orders')}}o
    JOIN
        {{ref ('stg_customers')}}c
    on
        o.customer_id =c.customer_ids
    where o.order_status ='delivered'
    group by
        c.customer_state

)
select * from source
order by count_orders desc
```