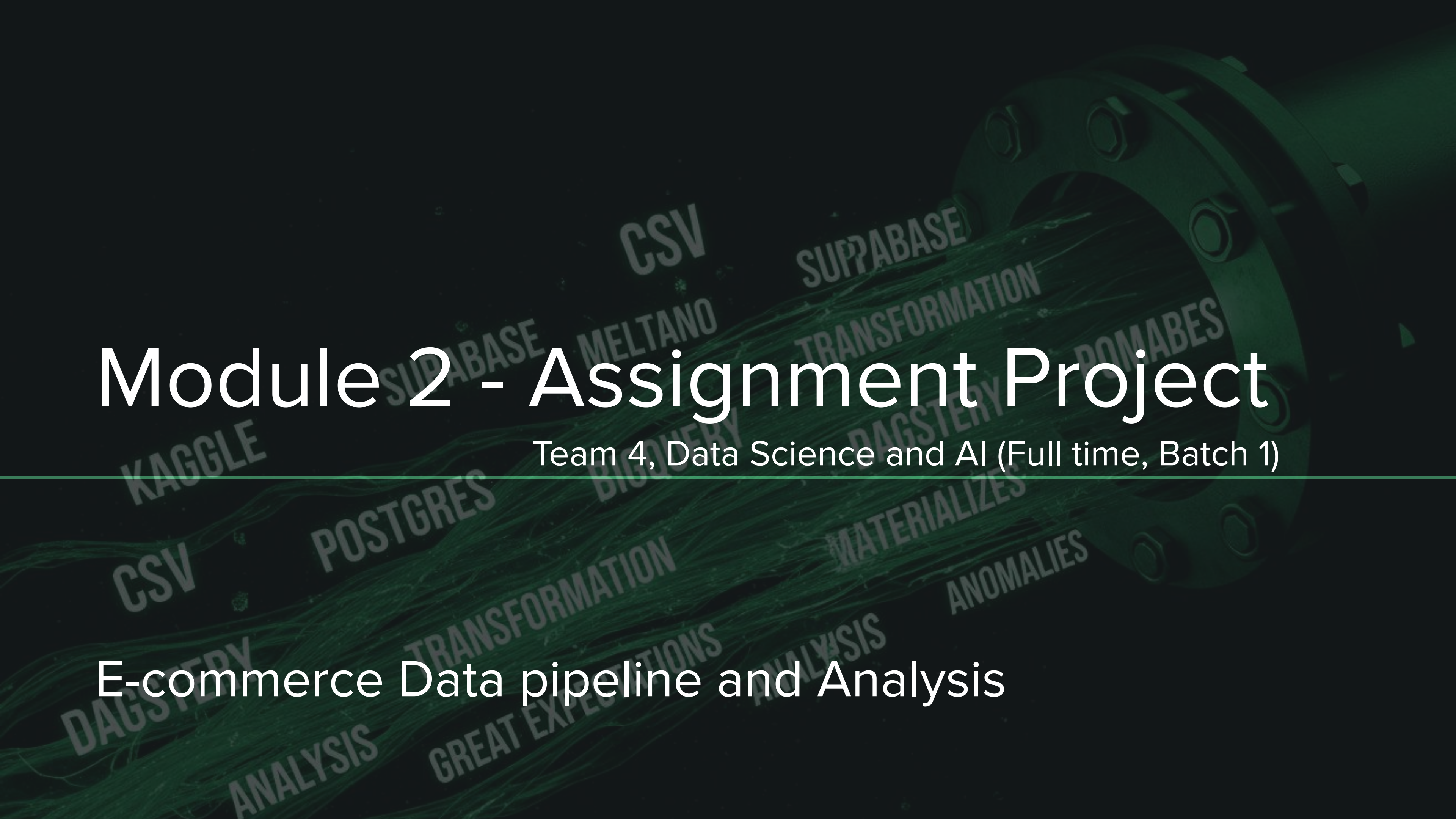
Module 2 - Assignment Project

Team 4, Data Science and AI (Full time, Batch 1)

E-commerce Data pipeline and Analysis



Members

Team 4

ian

siew wen

eve

selena

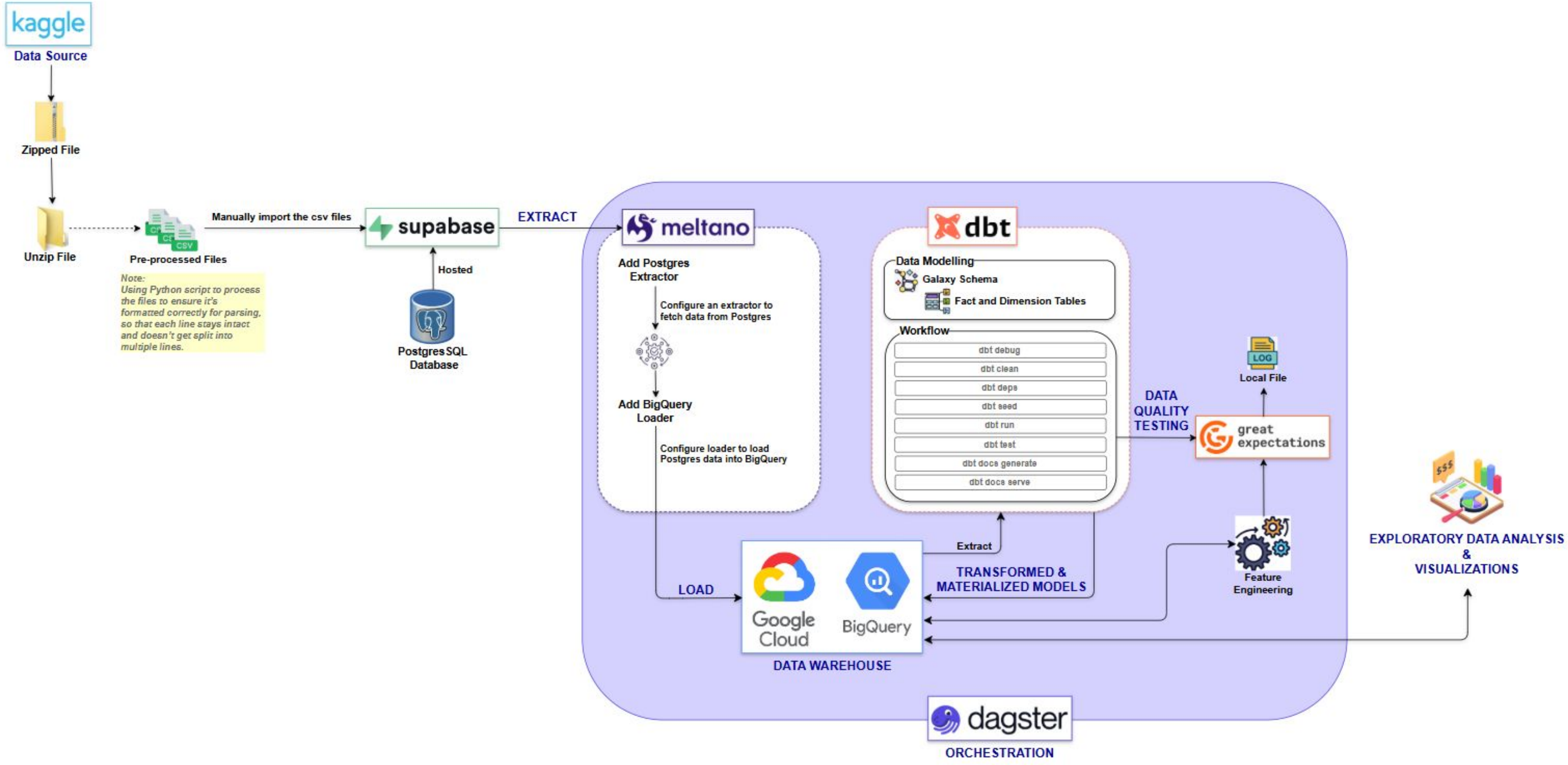
andrew

Project Overview

an end to end approach

SOURCE	kaggle
DATA	supabase
WAREHOUSE	bigquery
EXTRACT & LOAD	meltano
TRANSFORM	dbt
VALIDATE	great expectations
ANALYSIS	pandas
ORCHESTRATE	dagster

Data Pipeline Architecture



kaggle

Data Source

Zipped File

Unzip File

Pre-processed Files

Note:
Using Python script to process the files to ensure it's formatted correctly for parsing, so that each line stays intact and doesn't get split into multiple lines.

Manually import the csv files

supabase

Hosted

PostgresSQL Database

EXTRACT

meltano

Add Postgres Extractor

Configure an extractor to fetch data from Postgres

Add BigQuery Loader

Configure loader to load Postgres data into BigQuery

LOAD

dbt

Data Modelling

Galaxy Schema

Fact and Dimension Tables

Workflow

dbt debug

dbt clean

dbt deps

dbt seed

dbt run

dbt test

dbt docs generate

dbt docs serve

Extract

TRANSFORMED & MATERIALIZED MODELS

Google Cloud

BigQuery

DATA WAREHOUSE

DATA QUALITY TESTING

great expectations

Local File

Feature Engineering

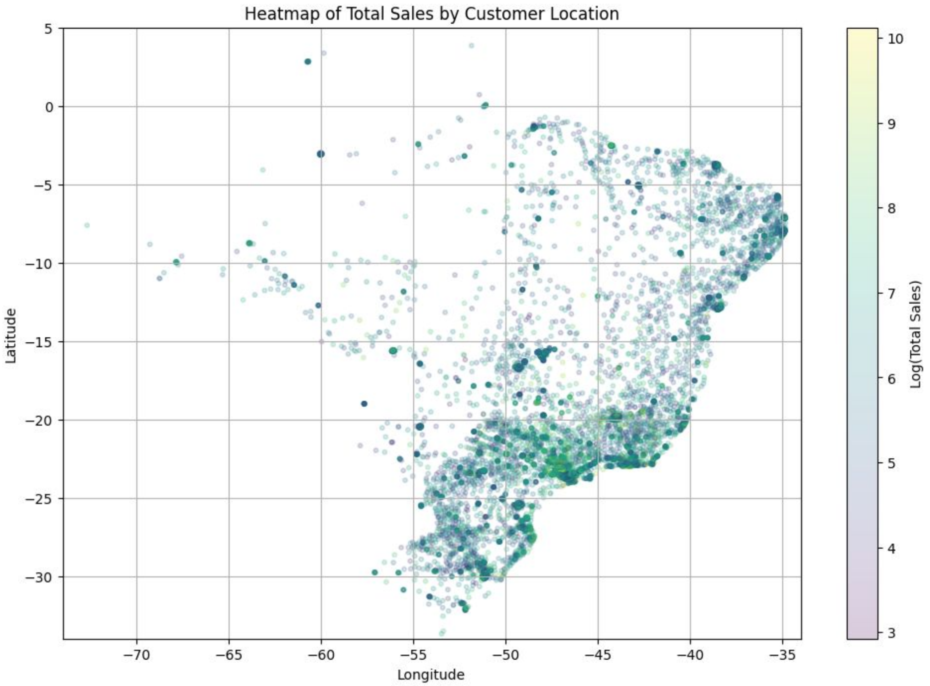
EXPLORATORY DATA ANALYSIS & VISUALIZATIONS

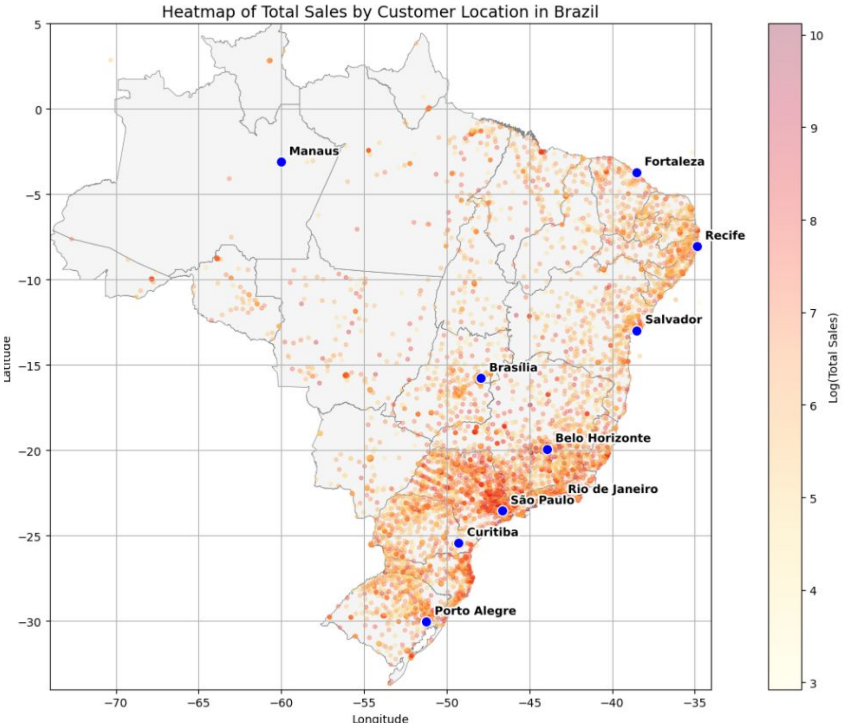
dagster

ORCHESTRATION

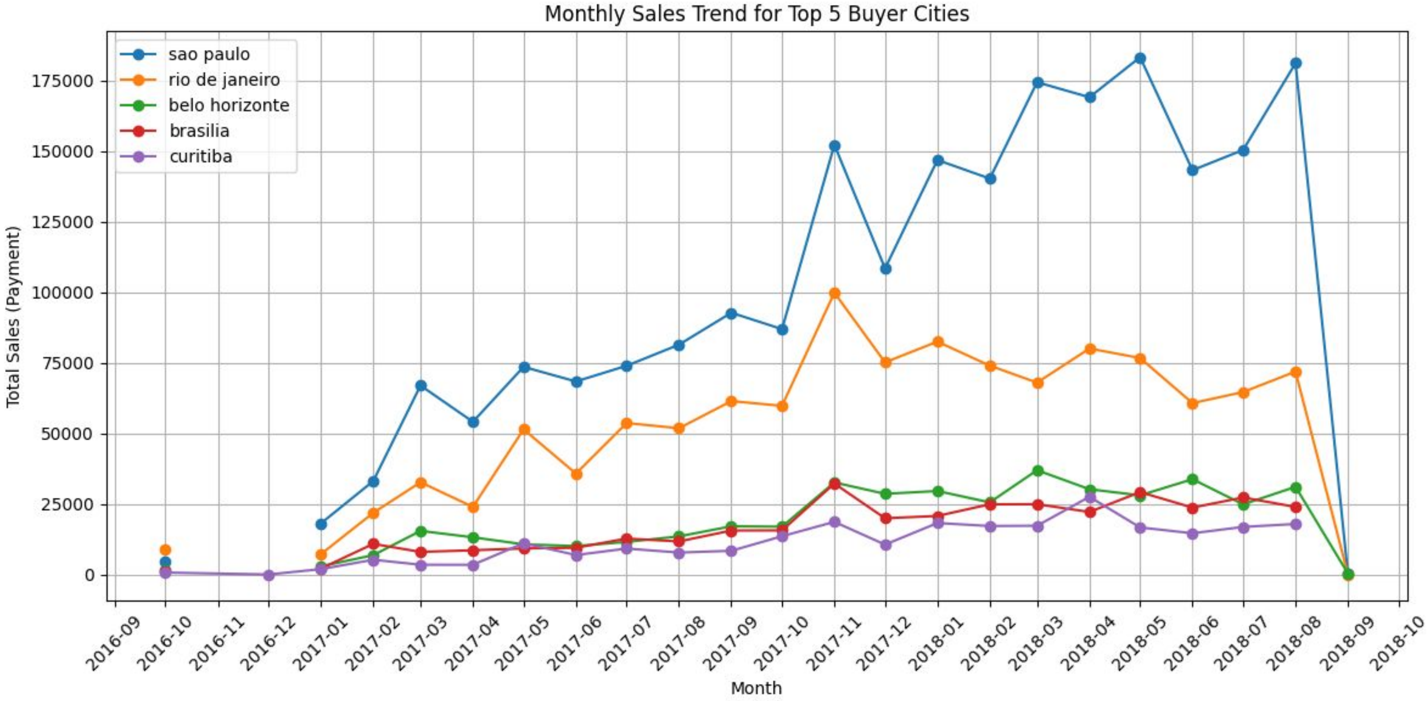
Geo heatmap sales

red= higher sales

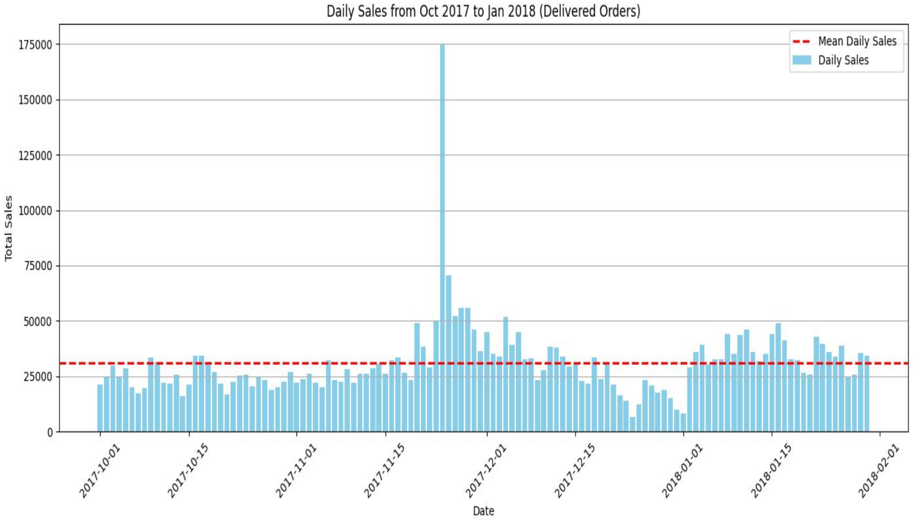


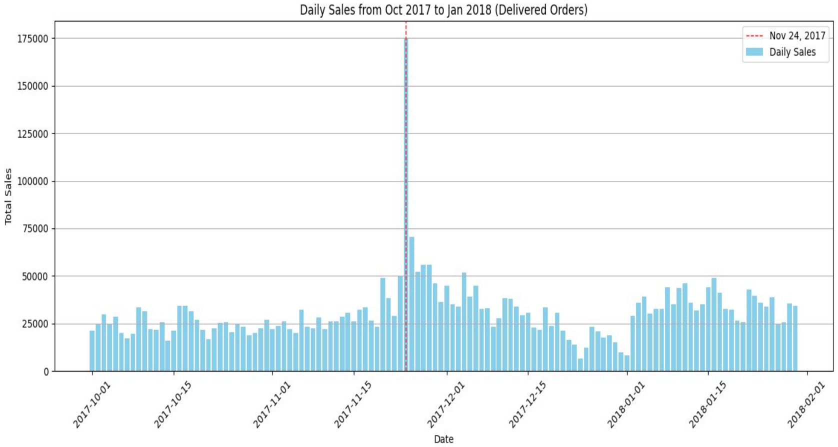


Monthly sales trend for top 5 cities



Daily sales highlighting black friday sales

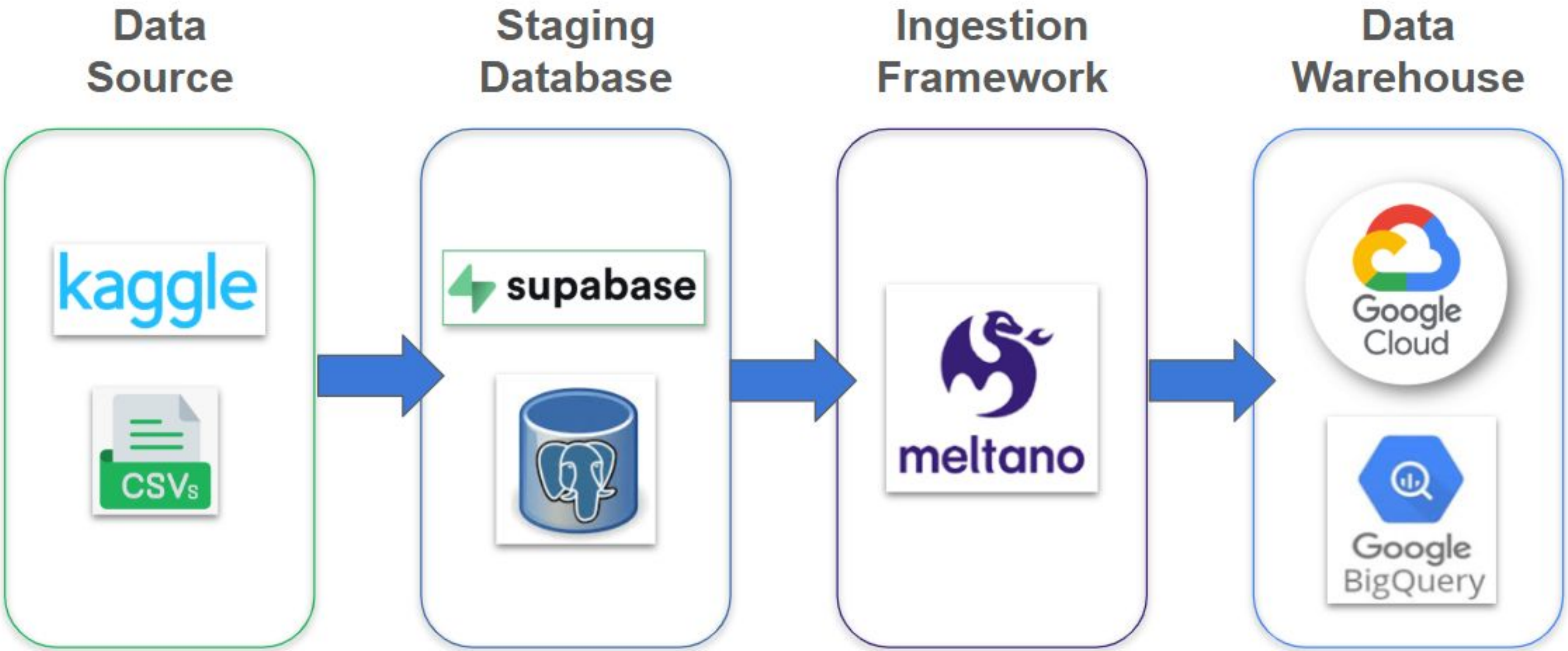




Nov 24

black friday

Data Ingestion



Data Source

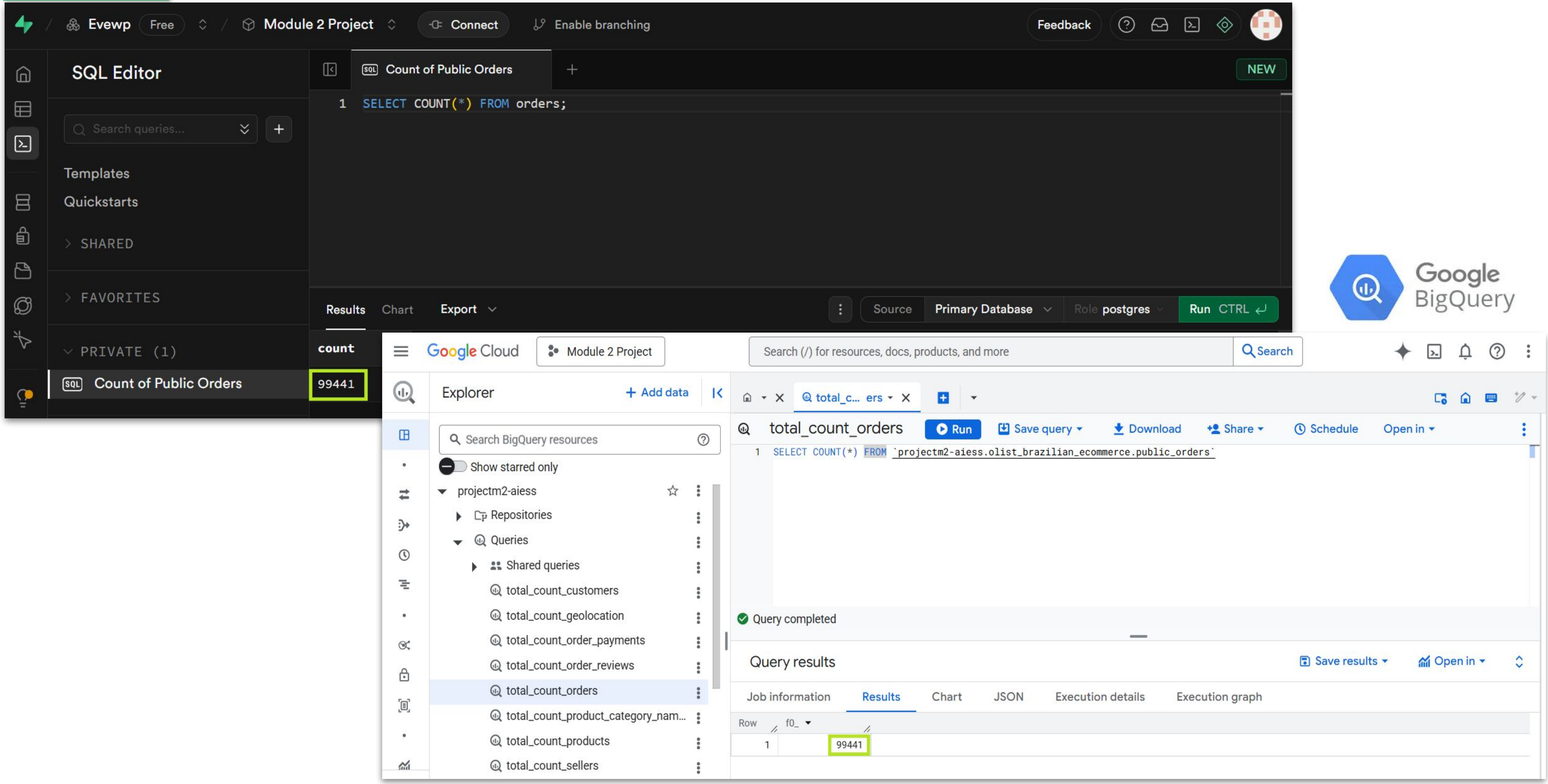
Staging Database

Ingestion Framework

Data Warehouse

supabase

Google BigQuery



Count of Public Orders

```
SELECT COUNT(*) FROM orders;
```

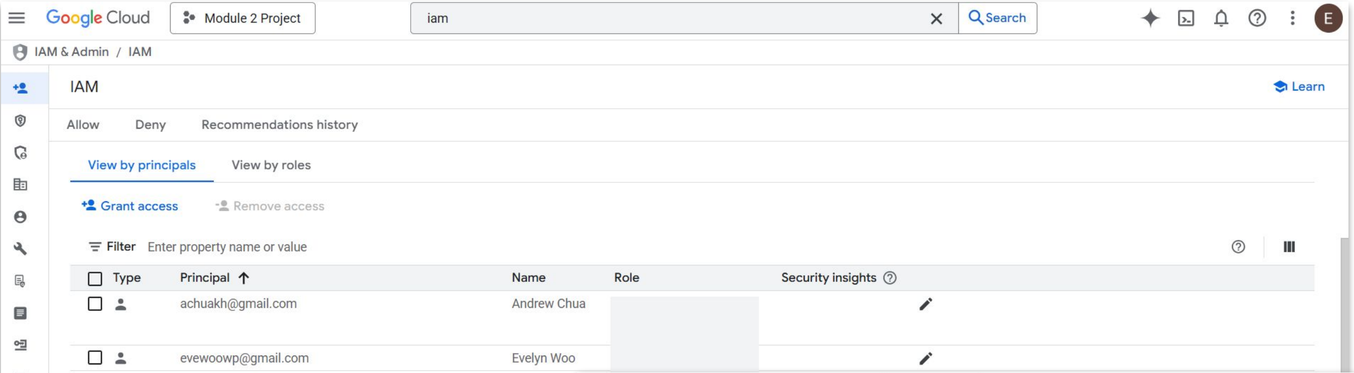
```
count
99441
```

total_count_orders

```
SELECT COUNT(*) FROM `projectm2-aiess.olist_brazilian_ecommerce.public_orders`
```

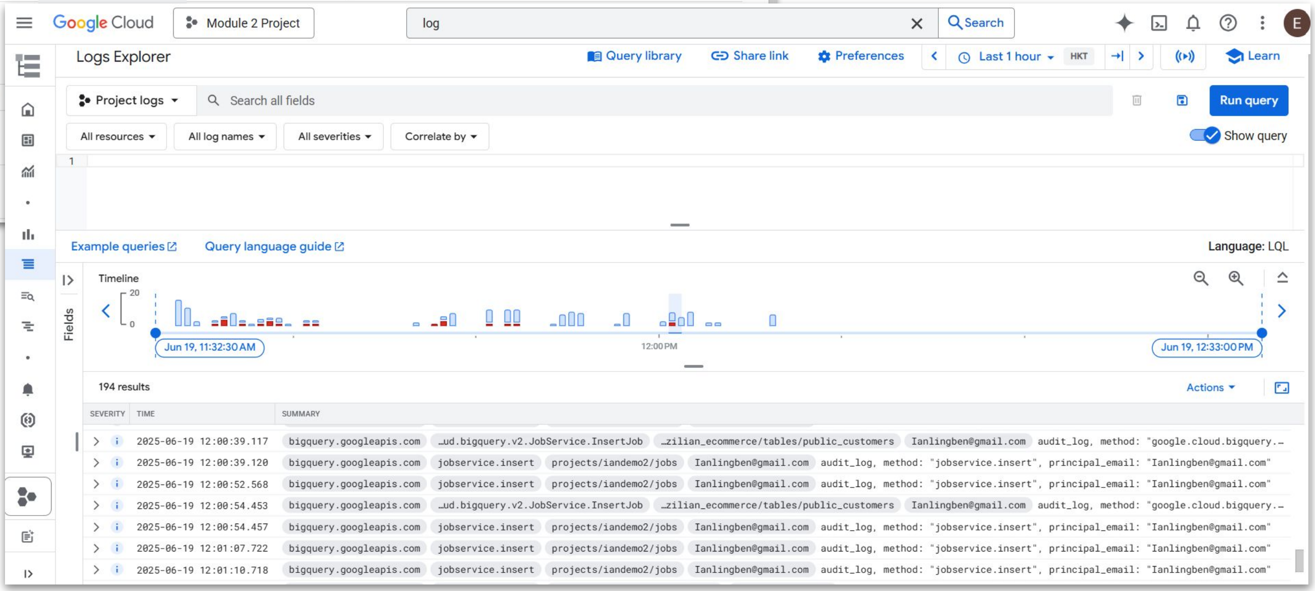
```
Row   f0_
1     99441
```

Grant User Access



Type	Principal ↑	Name	Role	Security insights
□	achuakh@gmail.com	Andrew Chua		
□	evewoowp@gmail.com	Evelyn Woo		
□	Ianlingben@gmail.com			
□	projectm2@projectm2-aiess.iam.gserviceaccount.com	projectm2		
□	selinasun68@gmail.com			
□	swlai1112@gmail.com			

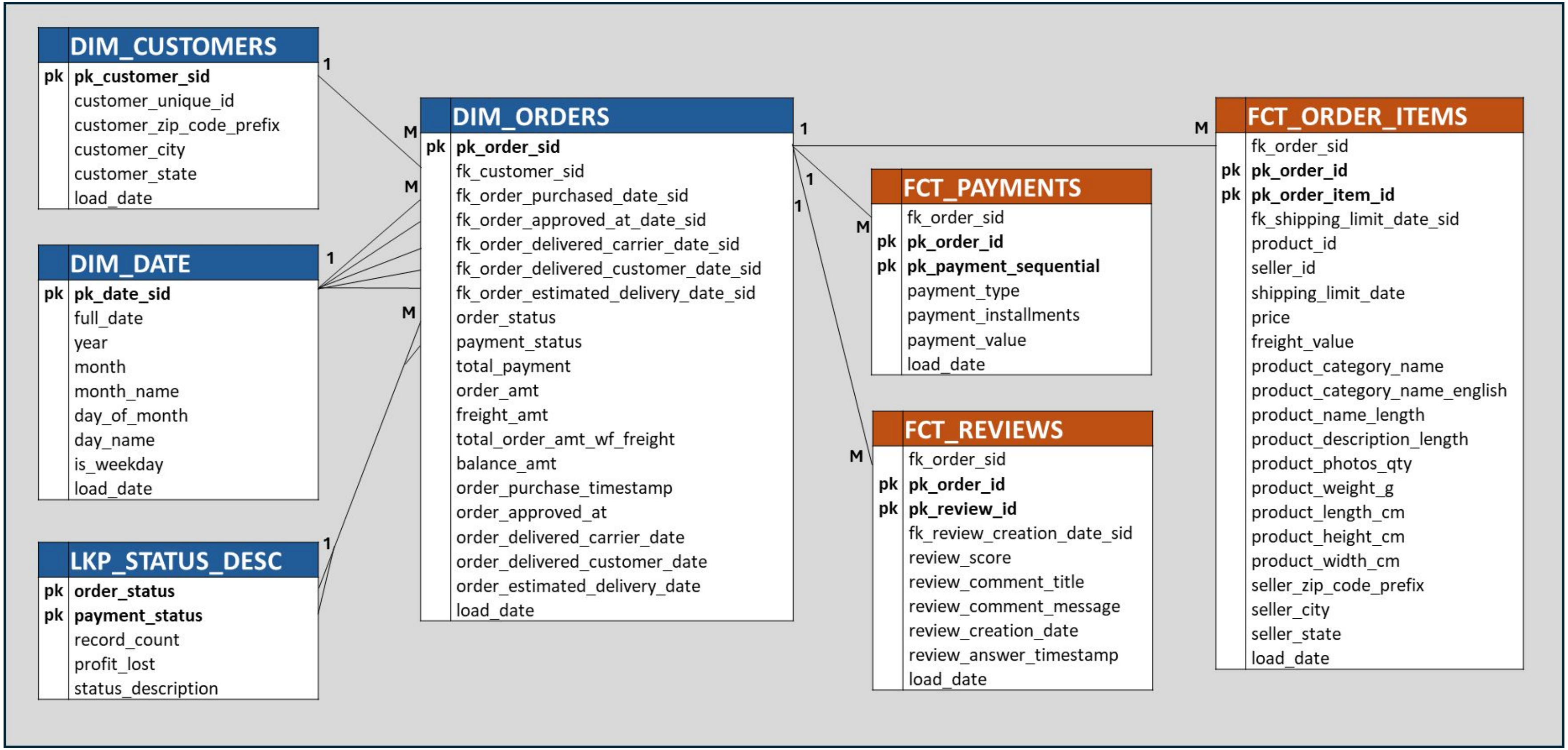
Logs Explorer



194 results

```
SEVERITY  TIME                     SUMMARY
>  i  2025-06-19 12:00:39.117  bigquery.googleapis.com  …ud.bigquery.v2.JobService.InsertJob  …zilian_ecommerce/tables/public_customers  Ianlingben@gmail.com  audit_log, method: "google.cloud.bigquery.…
>  i  2025-06-19 12:00:39.120  bigquery.googleapis.com  jobservice.insert  projects/iandemo2/jobs  Ianlingben@gmail.com  audit_log, method: "jobservice.insert", principal_email: "Ianlingben@gmail.com"
>  i  2025-06-19 12:00:52.568  bigquery.googleapis.com  jobservice.insert  projects/iandemo2/jobs  Ianlingben@gmail.com  audit_log, method: "jobservice.insert", principal_email: "Ianlingben@gmail.com"
>  i  2025-06-19 12:00:54.453  bigquery.googleapis.com  …ud.bigquery.v2.JobService.InsertJob  …zilian_ecommerce/tables/public_customers  Ianlingben@gmail.com  audit_log, method: "google.cloud.bigquery.…
>  i  2025-06-19 12:00:54.457  bigquery.googleapis.com  jobservice.insert  projects/iandemo2/jobs  Ianlingben@gmail.com  audit_log, method: "jobservice.insert", principal_email: "Ianlingben@gmail.com"
>  i  2025-06-19 12:01:07.722  bigquery.googleapis.com  jobservice.insert  projects/iandemo2/jobs  Ianlingben@gmail.com  audit_log, method: "jobservice.insert", principal_email: "Ianlingben@gmail.com"
>  i  2025-06-19 12:01:10.718  bigquery.googleapis.com  jobservice.insert  projects/iandemo2/jobs  Ianlingben@gmail.com  audit_log, method: "jobservice.insert", principal_email: "Ianlingben@gmail.com"
```

Olist Brazilian Ecommerce - Galaxy Schema



	DIM_CUSTOMERS
pk	**pk_customer_sid**
	customer_unique_id
	customer_zip_code_prefix
	customer_city
	customer_state
	load_date

	DIM_DATE
pk	**pk_date_sid**
	full_date
	year
	month
	month_name
	day_of_month
	day_name
	is_weekday
	load_date

	LKP_STATUS_DESC
pk	**order_status**
pk	**payment_status**
	record_count
	profit_lost
	status_description

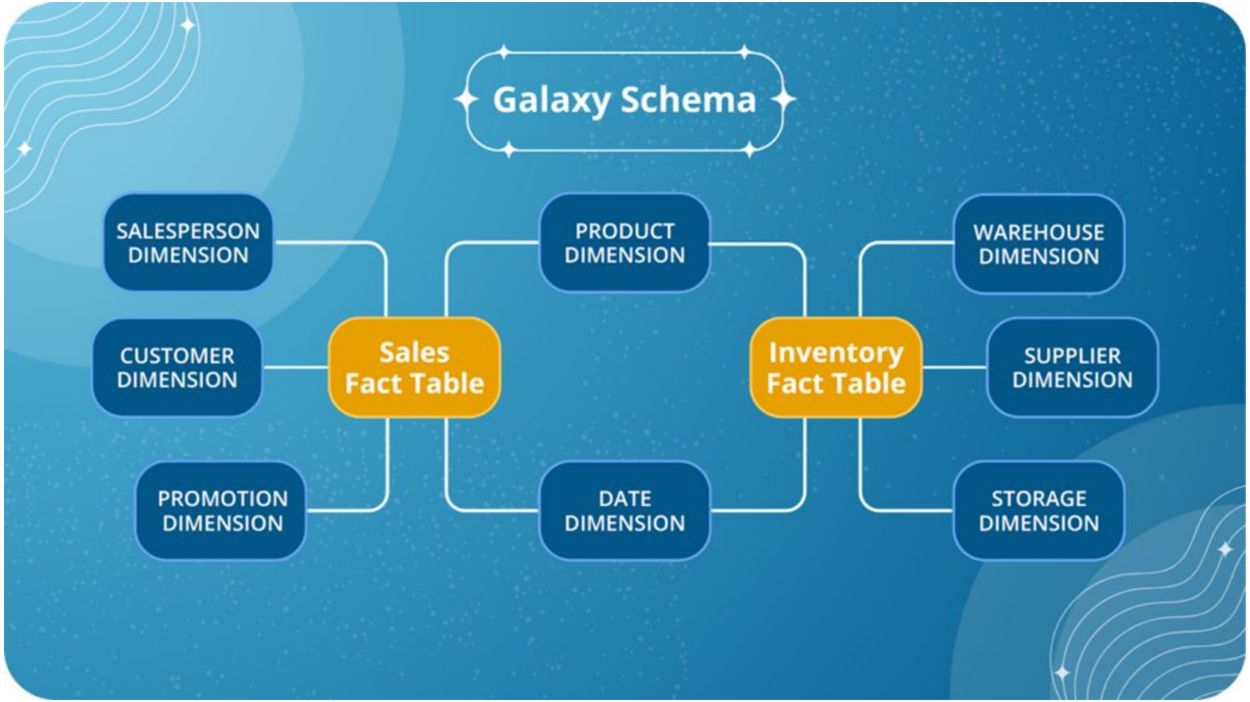
	DIM_ORDERS
pk	**pk_order_sid**
	fk_customer_sid
	fk_order_purchased_date_sid
	fk_order_approved_at_date_sid
	fk_order_delivered_carrier_date_sid
	fk_order_delivered_customer_date_sid
	fk_order_estimated_delivery_date_sid
	order_status
	payment_status
	total_payment
	order_amt
	freight_amt
	total_order_amt_wf_freight
	balance_amt
	order_purchase_timestamp
	order_approved_at
	order_delivered_carrier_date
	order_delivered_customer_date
	order_estimated_delivery_date
	load_date

	FCT_PAYMENTS
	fk_order_sid
pk	**pk_order_id**
pk	**pk_payment_sequential**
	payment_type
	payment_installments
	payment_value
	load_date

	FCT_REVIEWS
	fk_order_sid
pk	**pk_order_id**
pk	**pk_review_id**
	fk_review_creation_date_sid
	review_score
	review_comment_title
	review_comment_message
	review_creation_date
	review_answer_timestamp
	load_date

	FCT_ORDER_ITEMS
	fk_order_sid
pk	**pk_order_id**
pk	**pk_order_item_id**
	fk_shipping_limit_date_sid
	product_id
	seller_id
	shipping_limit_date
	price
	freight_value
	product_category_name
	product_category_name_english
	product_name_length
	product_description_length
	product_photos_qty
	product_weight_g
	product_length_cm
	product_height_cm
	product_width_cm
	seller_zip_code_prefix
	seller_city
	seller_state
	load_date

Galaxy Schema - When One Star Isn’t Enough



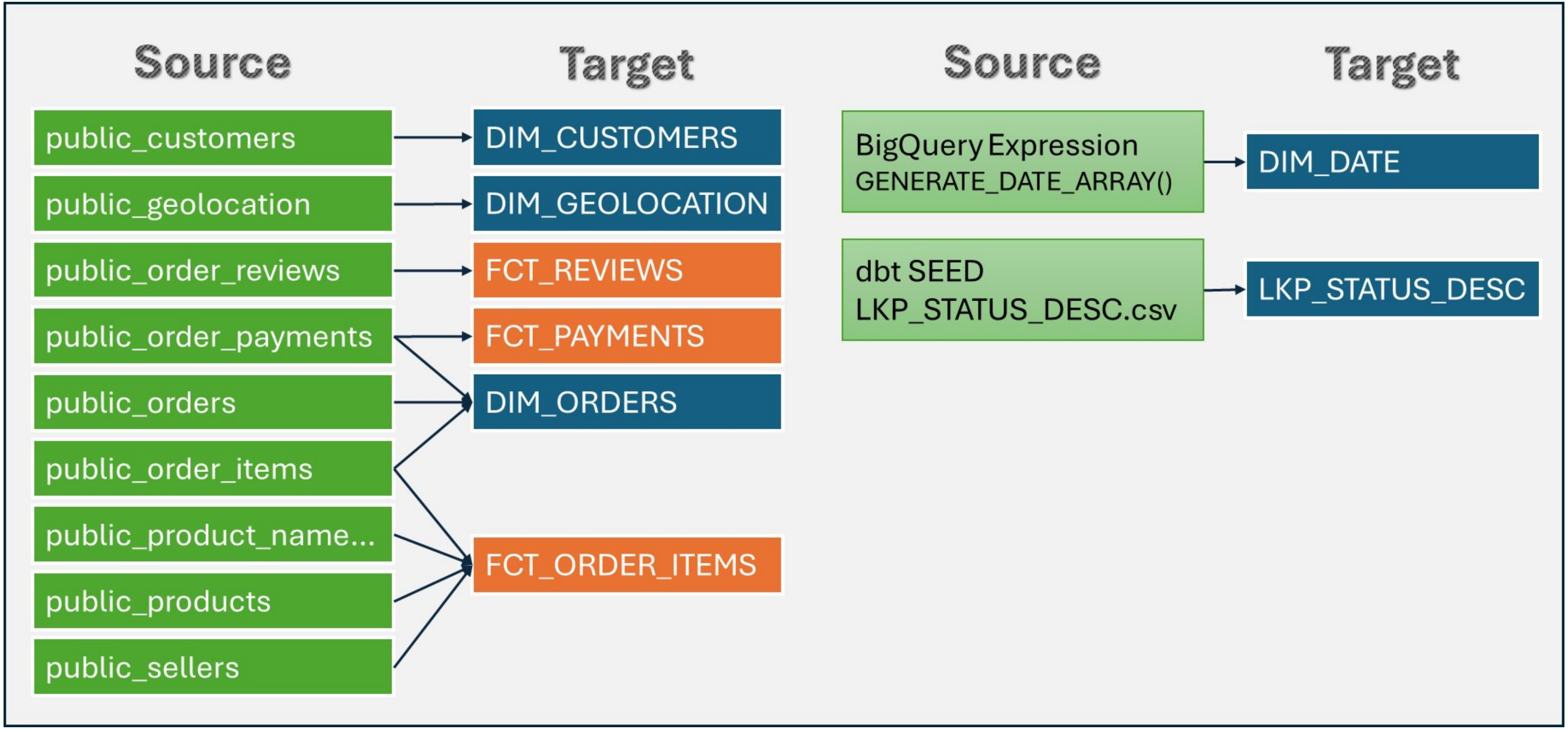
Why It Matters:

Share Dimension

Avoid Cartesian Joins

Granular Control

Data Lineage in dbt



Source

Target

public_customers

public_geolocation

public_order_reviews

public_order_payments

public_orders

public_order_items

public_product_name...

public_products

public_sellers

DIM_CUSTOMERS

DIM_GEOLOCATION

FCT_REVIEWS

FCT_PAYMENTS

DIM_ORDERS

FCT_ORDER_ITEMS

Source

Target

BigQuery Expression
GENERATE_DATE_ARRAY()

DIM_DATE

dbt SEED
LKP_STATUS_DESC.csv

LKP_STATUS_DESC

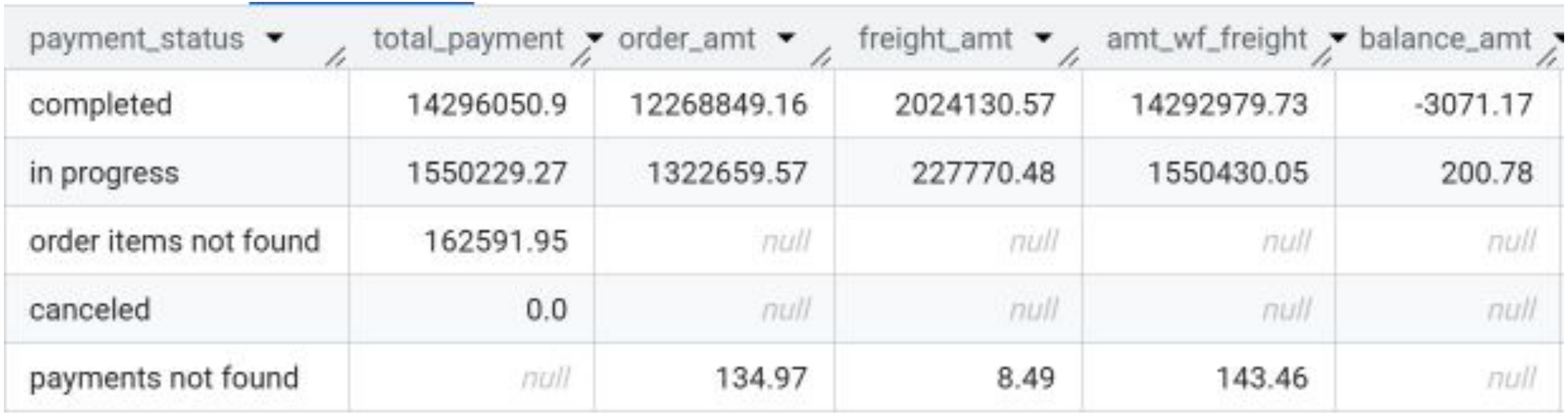
Pre-Aggregated Metrics

Why It Matters:

Improve Report Performance

Consistency across reports

Simplified Storytelling

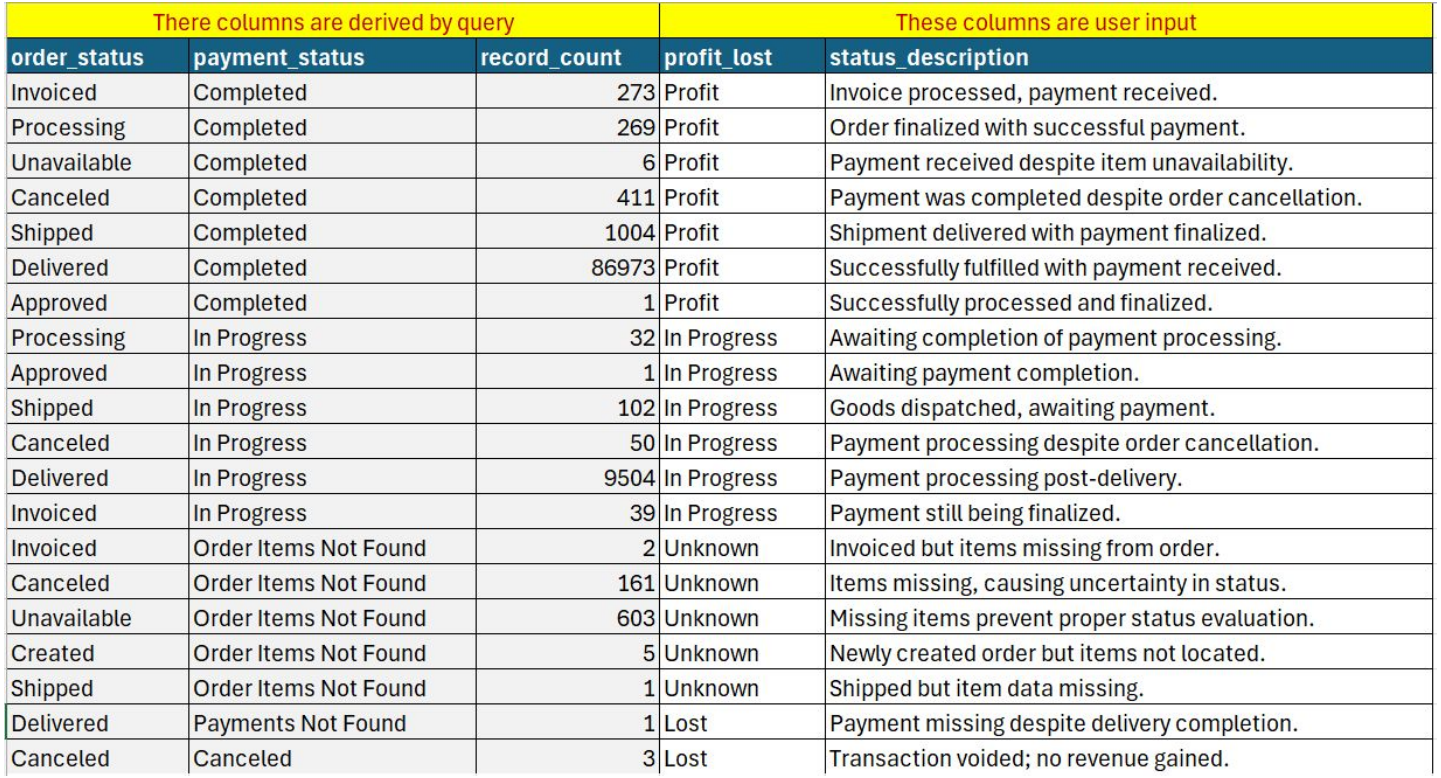


payment_status	total_payment	order_amt	freight_amt	amt_wf_freight	balance_amt
completed	14296050.9	12268849.16	2024130.57	14292979.73	-3071.17
in progress	1550229.27	1322659.57	227770.48	1550430.05	200.78
order items not found	162591.95	null	null	null	null
canceled	0.0	null	null	null	null
payments not found	null	134.97	8.49	143.46	null

Screenshot: Pre-aggregated columns from DIM_ORDER in BigQuery.

Lookup Tables for Decoupling Logic

The LKP_STATUS_DESC table adds business logic without cluttering DIM_ORDERS.



There columns are derived by query			These columns are user input	
order_status	payment_status	record_count	profit_lost	status_description
Invoiced	Completed	273	Profit	Invoice processed, payment received.
Processing	Completed	269	Profit	Order finalized with successful payment.
Unavailable	Completed	6	Profit	Payment received despite item unavailability.
Canceled	Completed	411	Profit	Payment was completed despite order cancellation.
Shipped	Completed	1004	Profit	Shipment delivered with payment finalized.
Delivered	Completed	86973	Profit	Successfully fulfilled with payment received.
Approved	Completed	1	Profit	Successfully processed and finalized.
Processing	In Progress	32	In Progress	Awaiting completion of payment processing.
Approved	In Progress	1	In Progress	Awaiting payment completion.
Shipped	In Progress	102	In Progress	Goods dispatched, awaiting payment.
Canceled	In Progress	50	In Progress	Payment processing despite order cancellation.
Delivered	In Progress	9504	In Progress	Payment processing post-delivery.
Invoiced	In Progress	39	In Progress	Payment still being finalized.
Invoiced	Order Items Not Found	2	Unknown	Invoiced but items missing from order.
Canceled	Order Items Not Found	161	Unknown	Items missing, causing uncertainty in status.
Unavailable	Order Items Not Found	603	Unknown	Missing items prevent proper status evaluation.
Created	Order Items Not Found	5	Unknown	Newly created order but items not located.
Shipped	Order Items Not Found	1	Unknown	Shipped but item data missing.
Delivered	Payments Not Found	1	Lost	Payment missing despite delivery completion.
Canceled	Canceled	3	Lost	Transaction voided; no revenue gained.

Ensuring Primary Key Integrity in dbt

DBT Test & DBT Expectations

Implemented 13 dbt tests to validate that all primary keys are

🔑 **unique**

🚫 **not null**

❌ **Failure Handling:**

Failing rows are automatically logged in

📁 *target/run_results.json*

Why It Matters:

🛢️ Prevents duplicate records

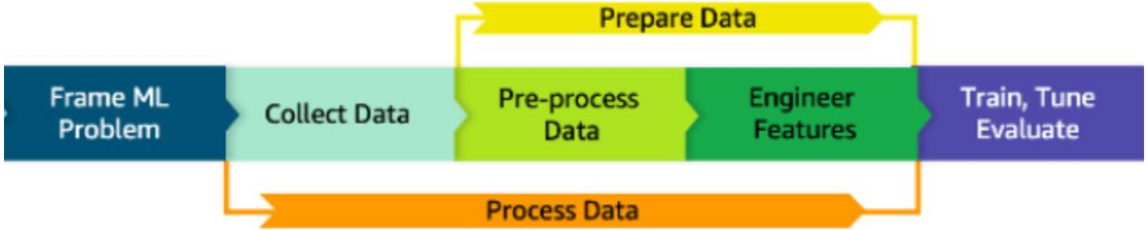
🔗 Clean join across model

📊 Trustworthy downstream reporting

Role of feature engineering

Data reusability

Feature creation, transform



Prepare Data

Frame ML Problem

Collect Data

Pre-process Data

Engineer Features

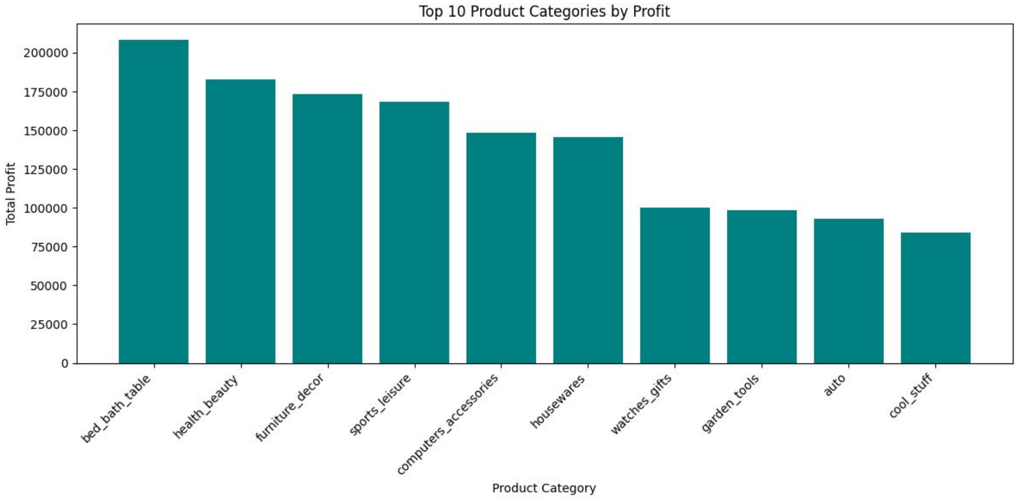
Train, Tune Evaluate

Process Data

One to many mapping

Solution?

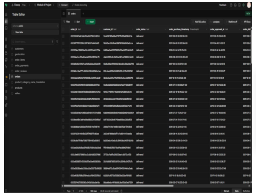
Aggregation



Data Quality

great expectations

supabase



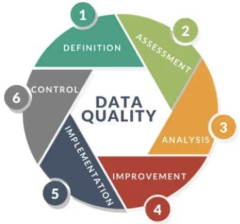
Your data assets: database tables, flat files, dataframes...

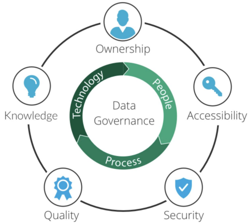
Data validation with Great Expectations

High quality data in your data products

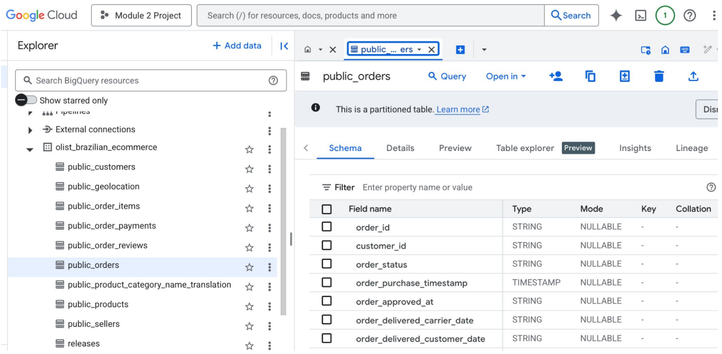
Data documentation & data quality reports

Logging & alerting

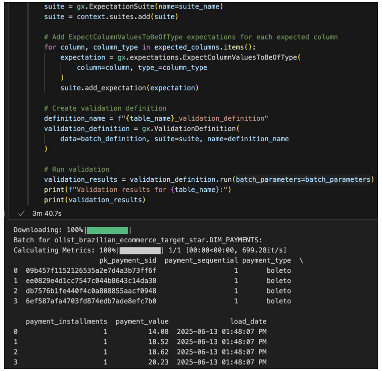




Google BigQuery



☐	Field name	Type	Mode	Key	Collation
☐	order_id	STRING	NULLABLE	-	-
☐	customer_id	STRING	NULLABLE	-	-
☐	order_status	STRING	NULLABLE	-	-
☐	order_purchase_timestamp	TIMESTAMP	NULLABLE	-	-
☐	order_approved_at	STRING	NULLABLE	-	-
☐	order_delivered_carrier_date	STRING	NULLABLE	-	-
☐	order_delivered_customer_date	STRING	NULLABLE	-	-

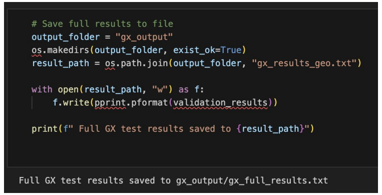


```
    suite = gx.ExpectationSuite(name=suite_name)
    suite = context.suites.add(suite)

    # Add ExpectColumnValuesToBeOfType expectations for each expected column
    for column, column_type in expected_columns.items():
        expectation = gx.expectations.ExpectColumnValuesToBeOfType(
            column=column, type_=column_type
        )
        suite.add_expectation(expectation)

    # Create validation definition
    definition_name = f"{table_name}_validation_definition"
    validation_definition = gx.ValidationDefinition(
        data=batch_definition, suite=suite, name=definition_name
    )

    # Run validation
    validation_results = validation_definition.run(batch_parameters=batch_parameters)
    print(f"Validation results for {table_name}:")
    print(validation_results)
```

✓ 3m 40.7s

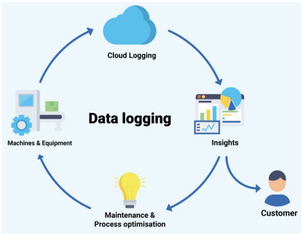
```
Downloading: 100%|██████████|
Batch for olist_brazilian_ecommerce_target_star.DIM_PAYMENTS:
Calculating Metrics: 100%|██████████| 1/1 [00:00<00:00, 699.28it/s]
                  pk_payment_sid  payment_sequential payment_type  \
0  09b457f1152126535a2e7d4a3b73ff6f                   1       boleto
1  ee0829e4d1cc7547c044b8643c14da38                   1       boleto
2  db7576b1fe440f4c0a808855aacf0948                   1       boleto
3  6ef587afa4703fd874edb7ade8efc7b0                   1       boleto

   payment_installments  payment_value              load_date
0                     1          14.08  2025-06-13 01:48:07 PM
1                     1          18.52  2025-06-13 01:48:07 PM
2                     1          18.62  2025-06-13 01:48:07 PM
3                     1          20.23  2025-06-13 01:48:07 PM
```

great-expectations 1.5.2

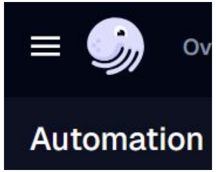


```
# Save full results to file
output_folder = "gx_output"
os.makedirs(output_folder, exist_ok=True)
result_path = os.path.join(output_folder, "gx_results_geo.txt")

with open(result_path, "w") as f:
    f.write(pprint.pformat(validation_results))

print(f" Full GX test results saved to {result_path}")
```

```
Full GX test results saved to gx_output/gx_full_results.txt
```



Orchestration



5 Stages:

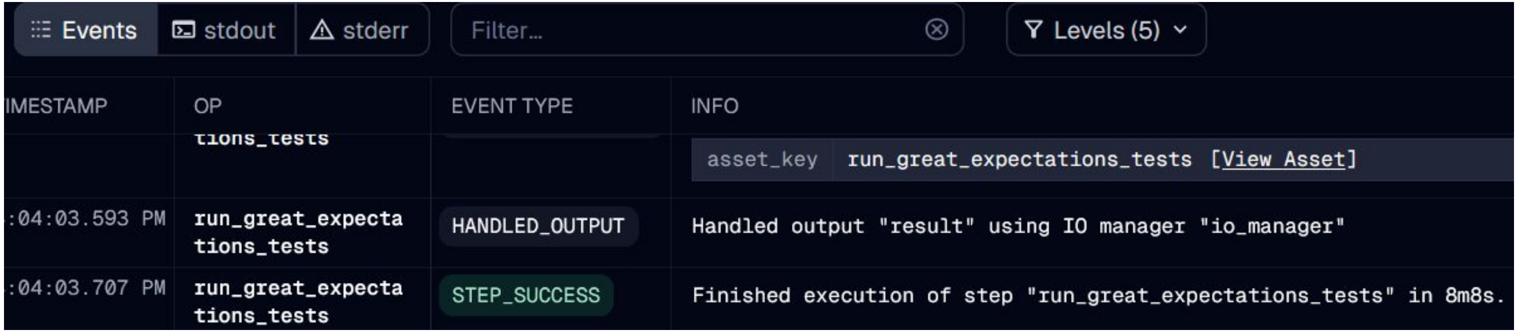
Meltano

dbt run

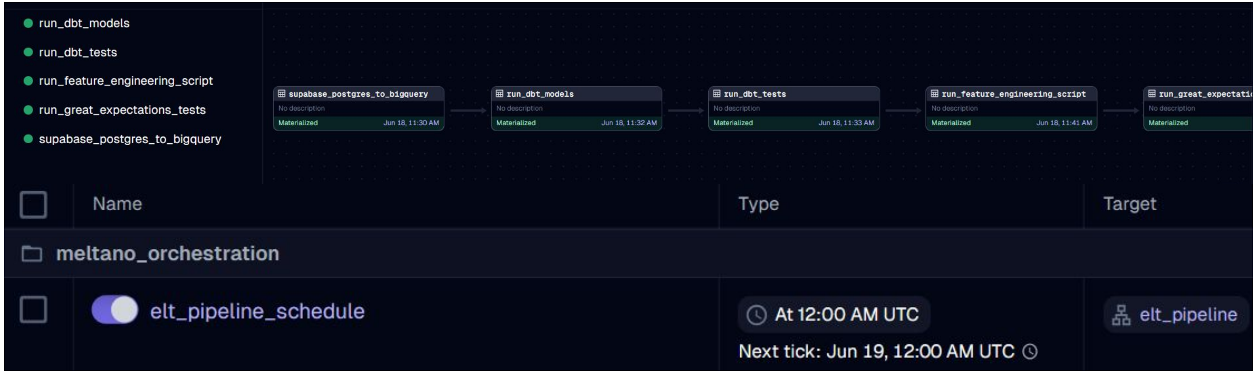
dbt test

Feature engineering

Great expectation



IMESTAMP	OP	EVENT TYPE	INFO
	tions_tests		asset_key run_great_expectations_tests [View Asset]
:04:03.593 PM	run_great_expectations_tests	HANDLED_OUTPUT	Handled output "result" using IO manager "io_manager"
:04:03.707 PM	run_great_expectations_tests	STEP_SUCCESS	Finished execution of step "run_great_expectations_tests" in 8m8s.



- run_dbt_models
- run_dbt_tests
- run_feature_engineering_script
- run_great_expectations_tests
- supabase_postgres_to_bigquery

□	Name	Type	Target
	meltano_orchestration		
□	elt_pipeline_schedule	At 12:00 AM UTC Next tick: Jun 19, 12:00 AM UTC	elt_pipeline

Q&A

How about some questions...

Benefits of Galaxy Schema on Olist Dataset

Shared Dimensions	**Avoids Cartesian Joins**	**Granular Control**
All dimension tables are shared across multiple fact tables, reducing redundancy and promoting consistency.	Clear one-to-many relationships between DIM_ORDERS and each fact table eliminate unintentional row multiplication.	Supports fine-grained analysis and improves query efficiency by separating transactional components.
	Fact Tables Differ in Granular Detail: - DIM_ORDERS: One record per unique order - FCT_ORDER_ITEMS: Captures multiple items per order - FCT_PAYMENTS: Tracks multiple payments or installments per order - FCT_REVIEWS: Stores multiple customer reviews per order	

Benefits of Lookup Tables

Business Logic Separation	Code Translation	Explanation Standardization	Debugging
Keeps DIM_ORDERS clean by offloading business logic to a dedicated table.	Converts technical codes into business-friendly descriptions.	Ensures consistent language across dashboards, reports, and analytical outputs.	Improves traceability.

Pre-Aggregated & Derived Metrics

Listed below are the derived columns in DIM_ORDERS and the SQL logic used to generate them.

Derived Column Name	Transformation

dim_order.total_payment	SELECT order_id, SUM(payment_value) AS total_payment
	FROM {{ source('olist_brazilian_ecommerce', 'public_order_payments') }}
	GROUP BY order_id

dim_order.order_amt	SELECT order_id, SUM(price) order_amt,
	FROM {{ source('olist_brazilian_ecommerce', 'public_order_items') }}
	GROUP BY order_id

dim_order.freight_amt	SELECT order_id, SUM(freight_value) as freight_amt,
	FROM {{ source('olist_brazilian_ecommerce', 'public_order_items') }}
	GROUP BY order_id

dim_order. total_order_amt_wf_freight	SELECT order_id, SUM(price) + sum(freight_value) as total_order_amt_wf_freight
	FROM {{ source('olist_brazilian_ecommerce', 'public_order_items') }}
	GROUP BY order_id

dim_order.balance_amt	i.total_order_amt_wf_freight - p.total_payment
dim_order.payment_status	CASE WHEN p.total_payment>0 and i.order_amt>0 and i.total_order_amt_wf_freight - p.total_payment = 0 THEN 'completed' WHEN p.total_payment>0 and i.order_amt>0 and i.total_order_amt_wf_freight - p.total_payment < 0 THEN 'completed' WHEN p.total_payment>0 and i.order_amt>0 and i.total_order_amt_wf_freight - p.total_payment > 0 THEN 'in progress' WHEN p.total_payment > 0 and i.order_amt IS NULL THEN 'order items not found' WHEN o.order_status in ('delivered','shipped') and (p.total_payment IS NULL OR p.total_payment <= 0) THEN 'payments not found' WHEN o.order_status in ('canceled','unavailable') and (p.total_payment IS NULL OR p.total_payment <= 0) THEN 'canceled' ELSE NULL END AS payment_status,