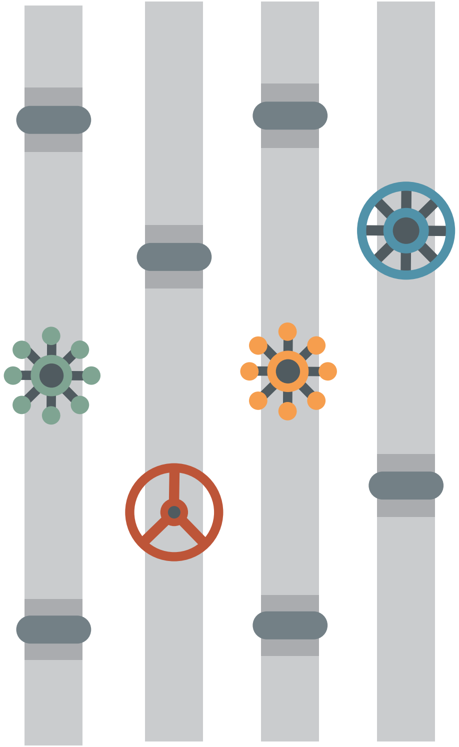
Data Pipeline

Module 2 Project - Group 4



OUR TEAM

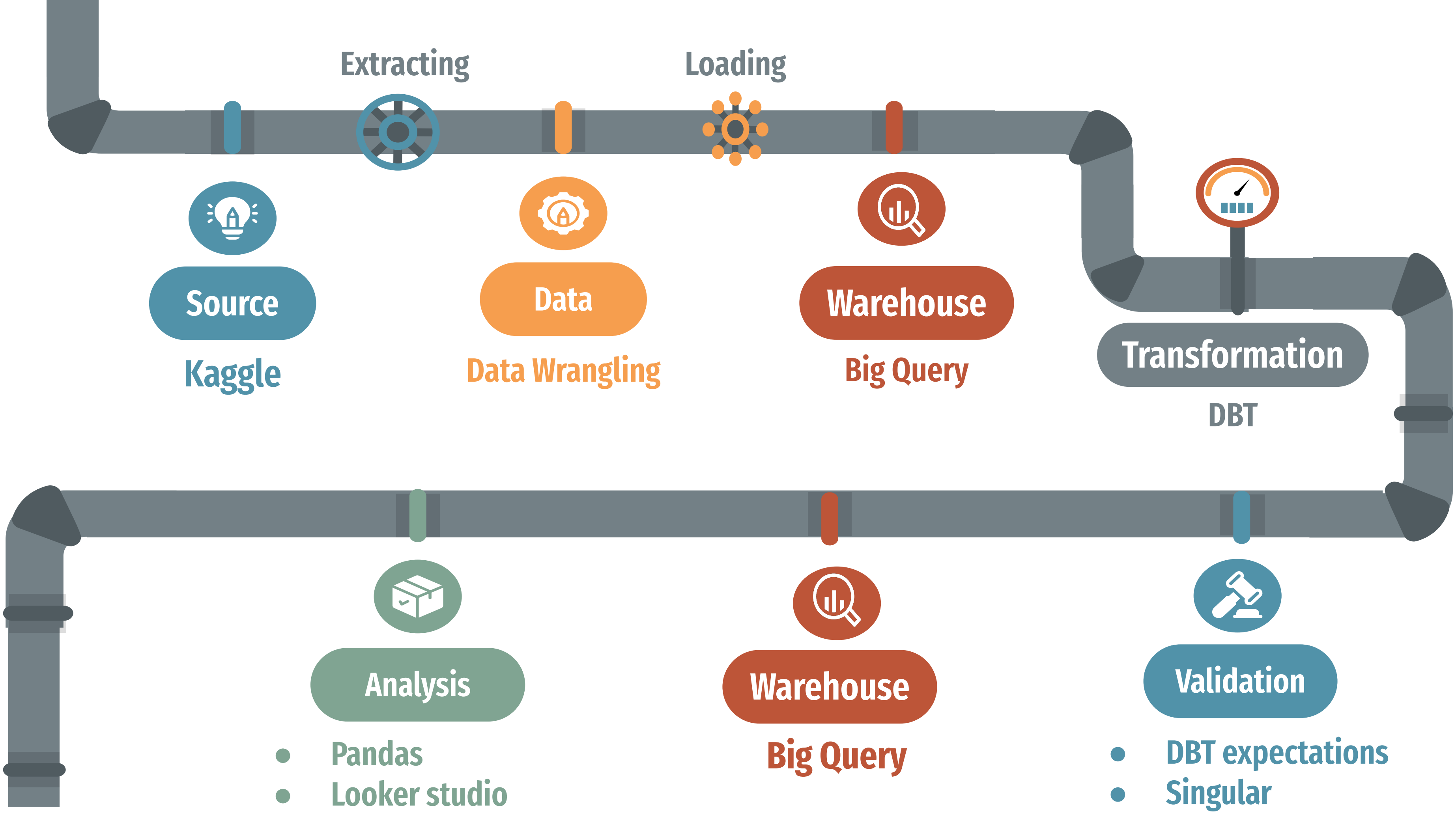
BAI HUIJING

TANG LAI LIN

MATHAVAN ARUGALAIMUTHU

LIM KAH HUI ESTHER

THOMAS TAY



Extracting

Loading

Source

Kaggle

Data

Data Wrangling

Warehouse

Big Query

Transformation

DBT

Validation

- DBT expectations
- Singular

Warehouse

Big Query

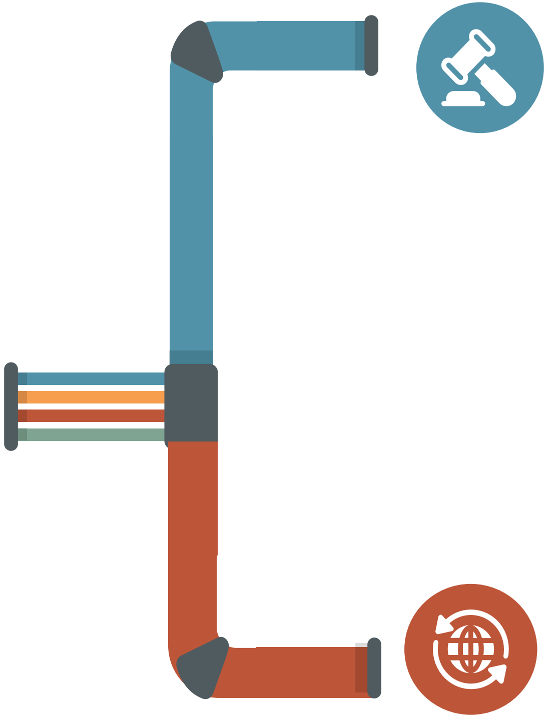
Analysis

- Pandas
- Looker studio

Data wrangling

olist_customers

olist_order_items



#1: Dropping of duplicates

- Duplicates are dropped (keep =first) from *'customer_id'* in olist_cutomers
- Duplicates are dropped (keep = last) from based on *'order_id'* as *order_item_id* is the list of quantity purchased

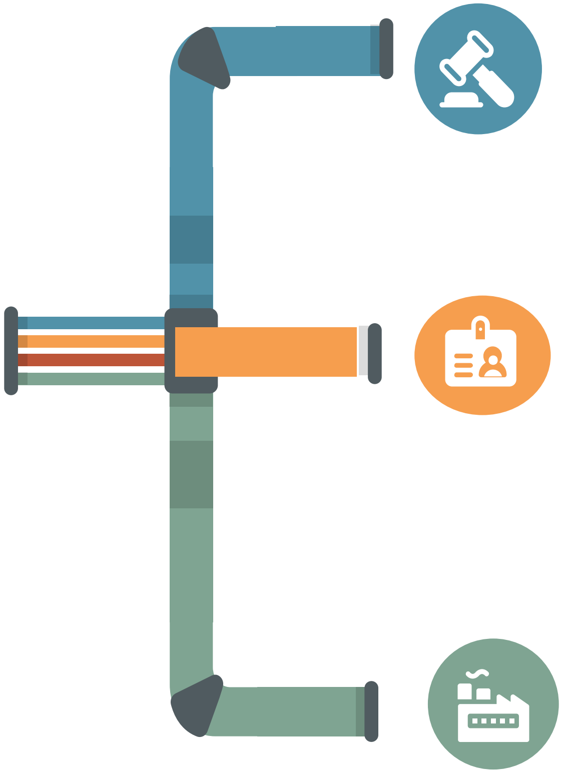
#2: Exporting dataframe to csv

- Clean data frame is exported to csv

Data wrangling

olist_payments

olist_orders



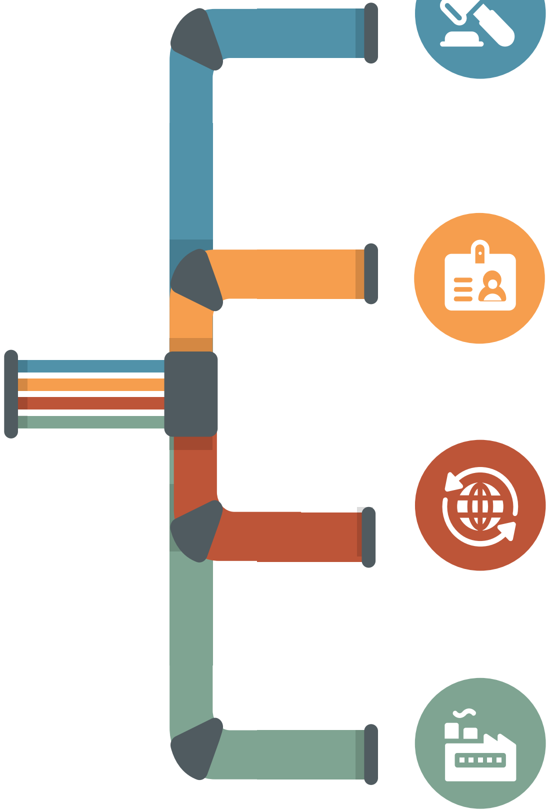
#1: Dropping of duplicates

#2: Dropping rows with null values

#3: Exporting dataframe to csv

Data wrangling

olist_products



#1: Dropping of Rows/Columns

- Duplicated rows & rows with null values dropped
- Columns dropped:
  - *‘product_name_lenght’,*
  - *‘product_description_lenght’*
  - *‘product_photos_qty’*

#2: Merging of dataframes

- Dataframe is merged with *product_category_name_translation.csv* using *‘product_category_name’*.

#3: Replace null values in English column

- Some Portuguese words might go to English column, but can be fixed later
- Replace *‘product_category_name’* with English column

#4: Exporting dataframe to csv

- Duplicates *‘product_id’* are removed
- Set *‘product_id’* as index
- Clean data frame is exported to csv

ELT infrastructure

Data Warehouse : BigQuery vs DuckDb

#1: Concurrency

- Challenge in managing DuckDb file lock

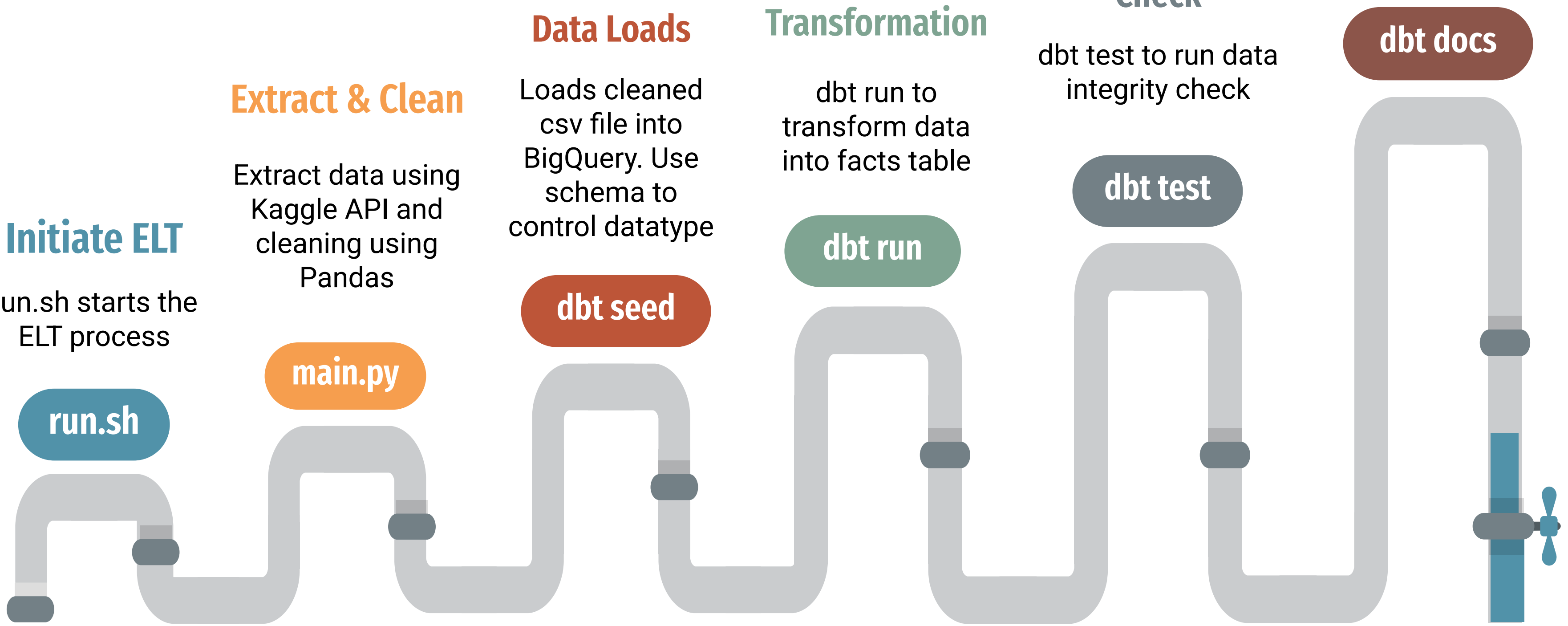
#2: Scalability

- No scalability concern for BigQuery

#3: Availability

- Available 24/7, include friendly tools: Looker Studio

Technical Pipeline



Initiate ELT

run.sh starts the ELT process

run.sh

Extract & Clean

Extract data using Kaggle API and cleaning using Pandas

main.py

Data Loads

Loads cleaned csv file into BigQuery. Use schema to control datatype

dbt seed

Transformation

dbt run to transform data into facts table

dbt run

Data Integrity Check

dbt test to run data integrity check

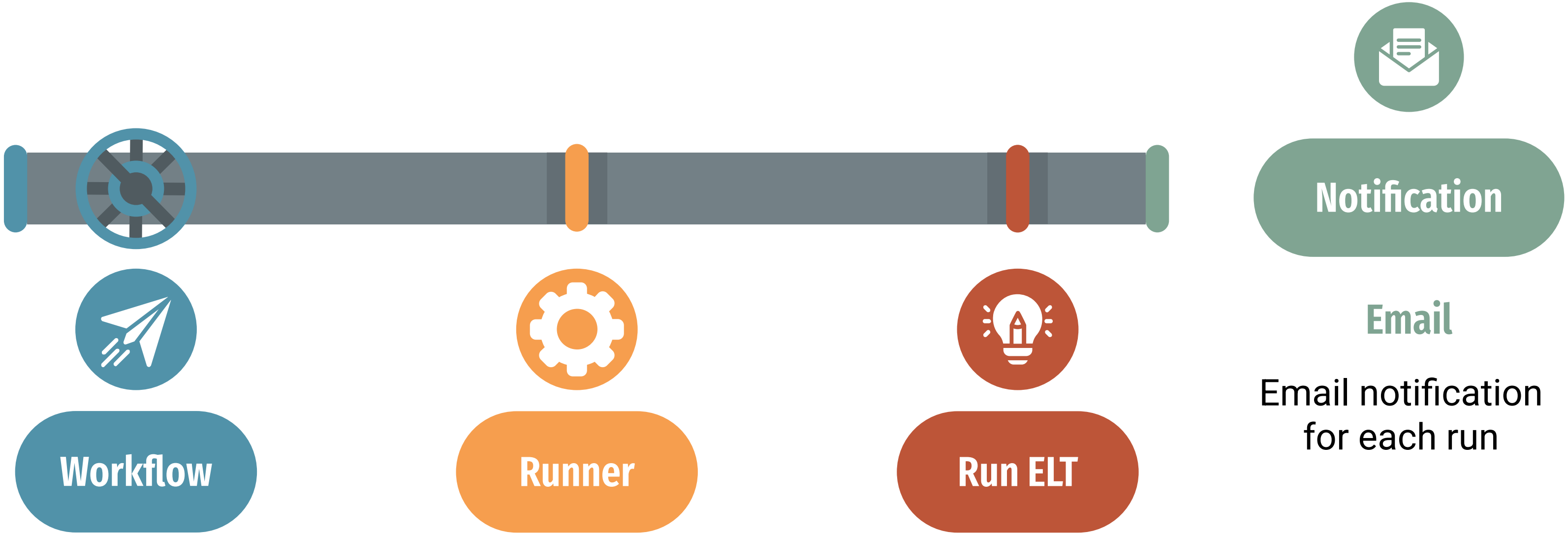
dbt test

Documentation

Generate documentation for project

dbt docs

Github Actions - CI/CD Platform



Workflow

Define Schedule & Workflow

Nightly on git push and manual

Runner

Launch VM

Install required packages

Run ELT

run.sh

Python scripts and dbt commands

Notification

Email

Email notification for each run

ELT infrastructure

GitHub Actions Implementations and Challenges

#1: Ease of Implementation

- Only one yaml file required

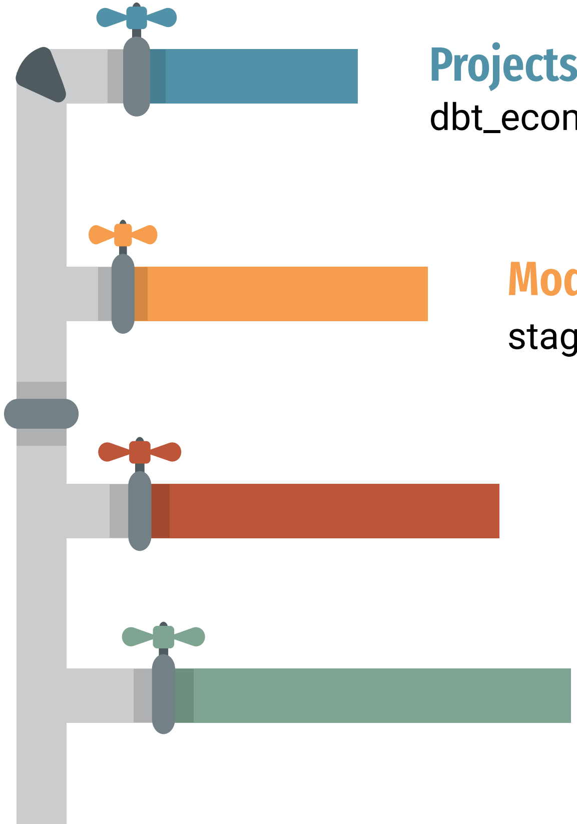
#2: Schedule Not Guaranteed

- Not suitable for mission critical job, need external scheduler

#3: Challenge to Maintain Github Secrets

- Cannot edit secrets, app password can only be copied directly.

DBT transformation | validation | documentation



Projects

dbt_ecomm | dbt_expectations

Models (8)

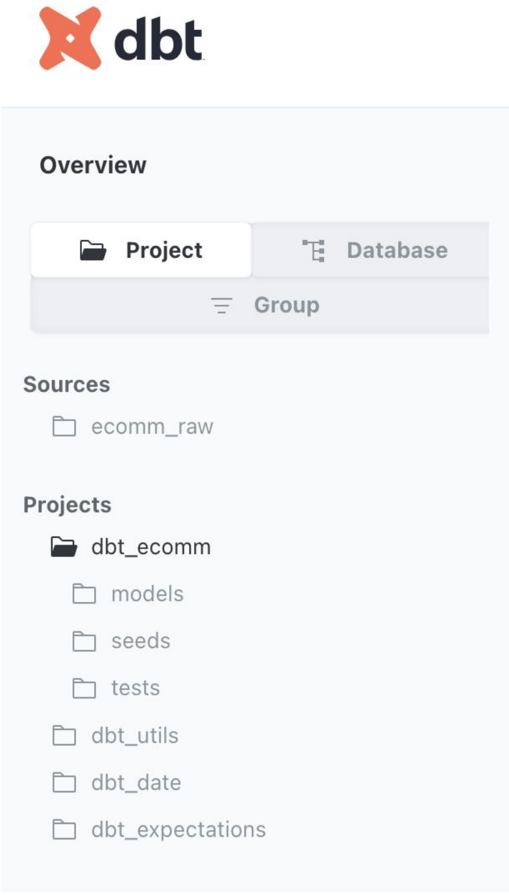
staging (5) | dimensions (2) | facts (1)

Tests / data tests

Generic tests (48) | singular tests (3)

DBT Docs

dbt docs generate | dbt docs serve

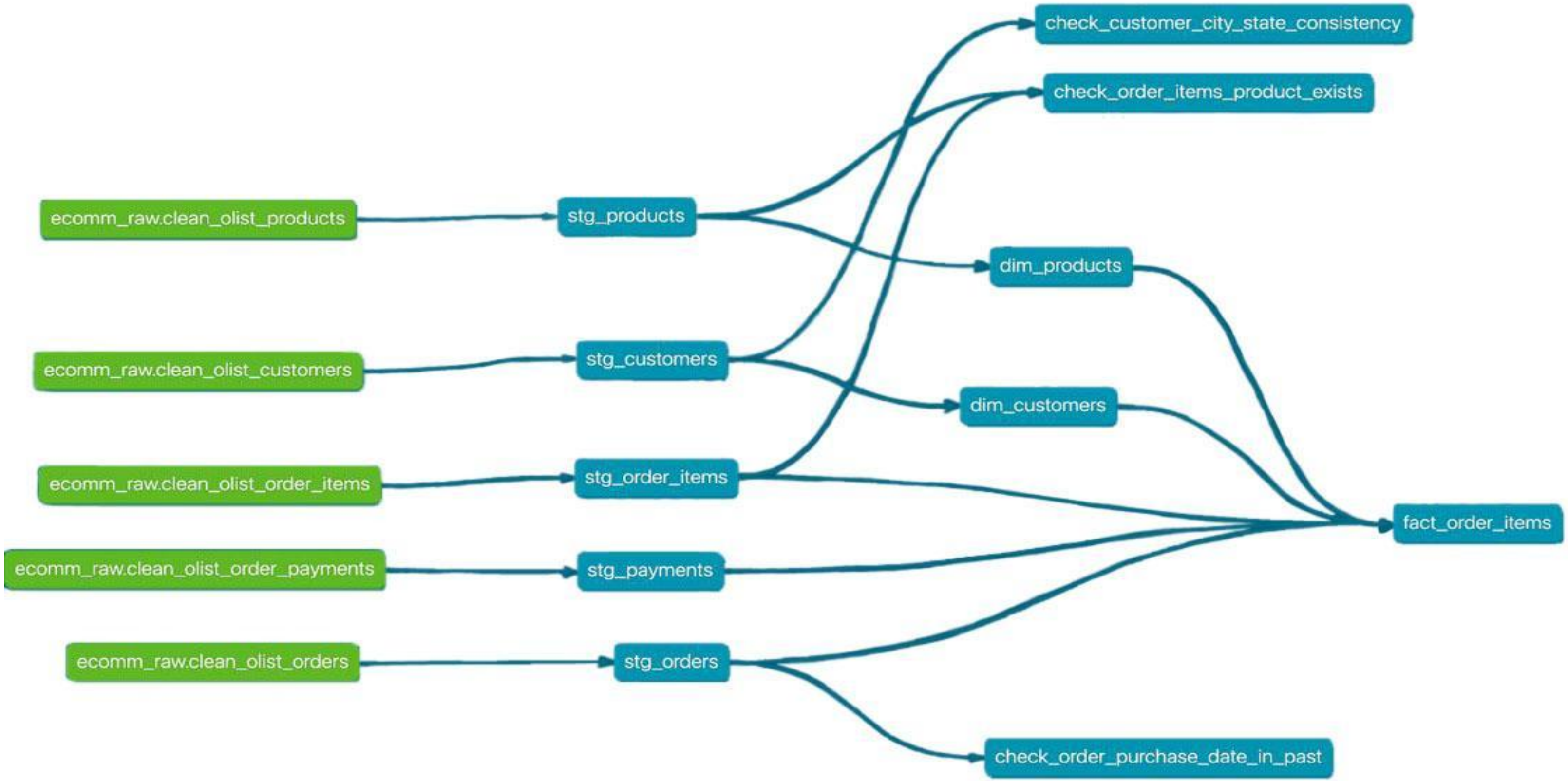


DBT lineage graph

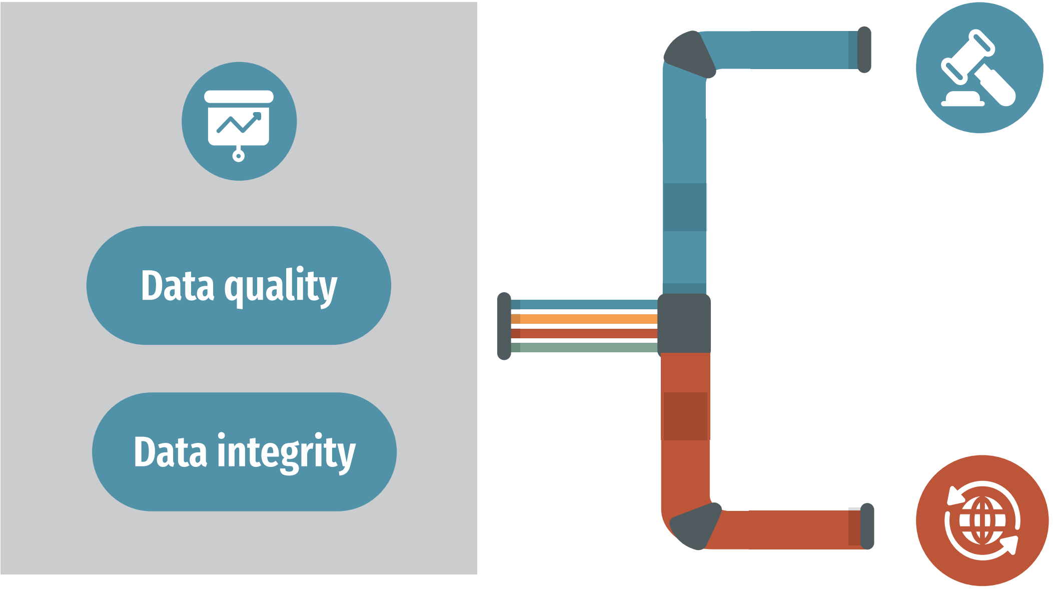
Source

Models

Tests


check_customer_city_state_consistency
check_order_items_product_exists
ecomm_raw.clean_olist_products
stg_products
dim_products
ecomm_raw.clean_olist_customers
stg_customers
dim_customers
ecomm_raw.clean_olist_order_items
stg_order_items
fact_order_items
ecomm_raw.clean_olist_order_payments
stg_payments
ecomm_raw.clean_olist_orders
stg_orders
check_order_purchase_date_in_past

DBT validation



Data quality

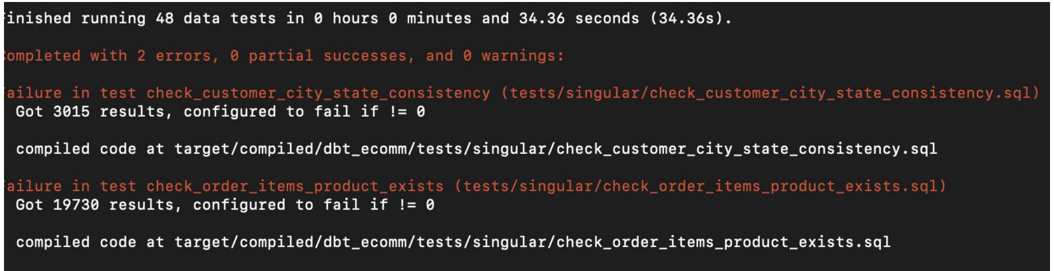
Data integrity

Test cases (51)

- Generic test cases: 48 (Unique, not_null, accepted_values, dbt expectation) focus on validation
- Singular test case: 3 focus on integrity

Failed test case (4)*

- Accepted_values mismatch
- Order lists product_id mismatch with product's product_id



```
inished running 48 data tests in 0 hours 0 minutes and 34.36 seconds (34.36s).

ompleted with 2 errors, 0 partial successes, and 0 warnings:

ailure in test check_customer_city_state_consistency (tests/singular/check_customer_city_state_consistency.sql)
 Got 3015 results, configured to fail if != 0

 compiled code at target/compiled/dbt_ecomm/tests/singular/check_customer_city_state_consistency.sql

ailure in test check_order_items_product_exists (tests/singular/check_order_items_product_exists.sql)
 Got 19730 results, configured to fail if != 0

 compiled code at target/compiled/dbt_ecomm/tests/singular/check_order_items_product_exists.sql
```

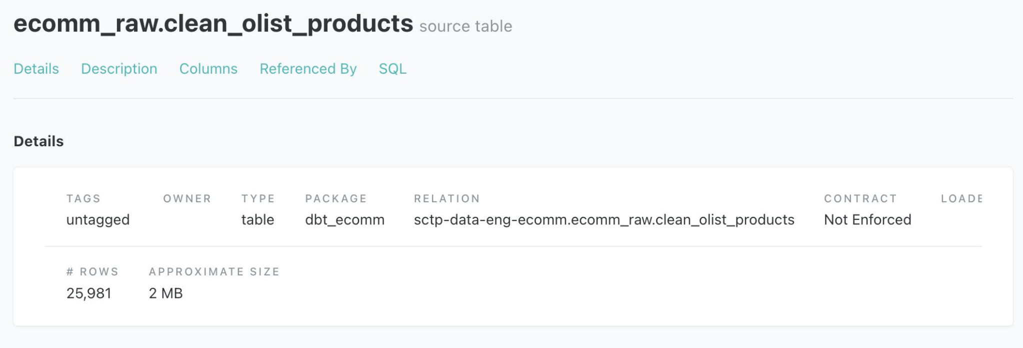
*bugs will be raised for the 4 failed test cases

DBT documentations

Docs generation

Docs serve

#1: Docs generation (catalog.json)



ecomm_raw.clean_olist_products source table

Details Description Columns Referenced By SQL

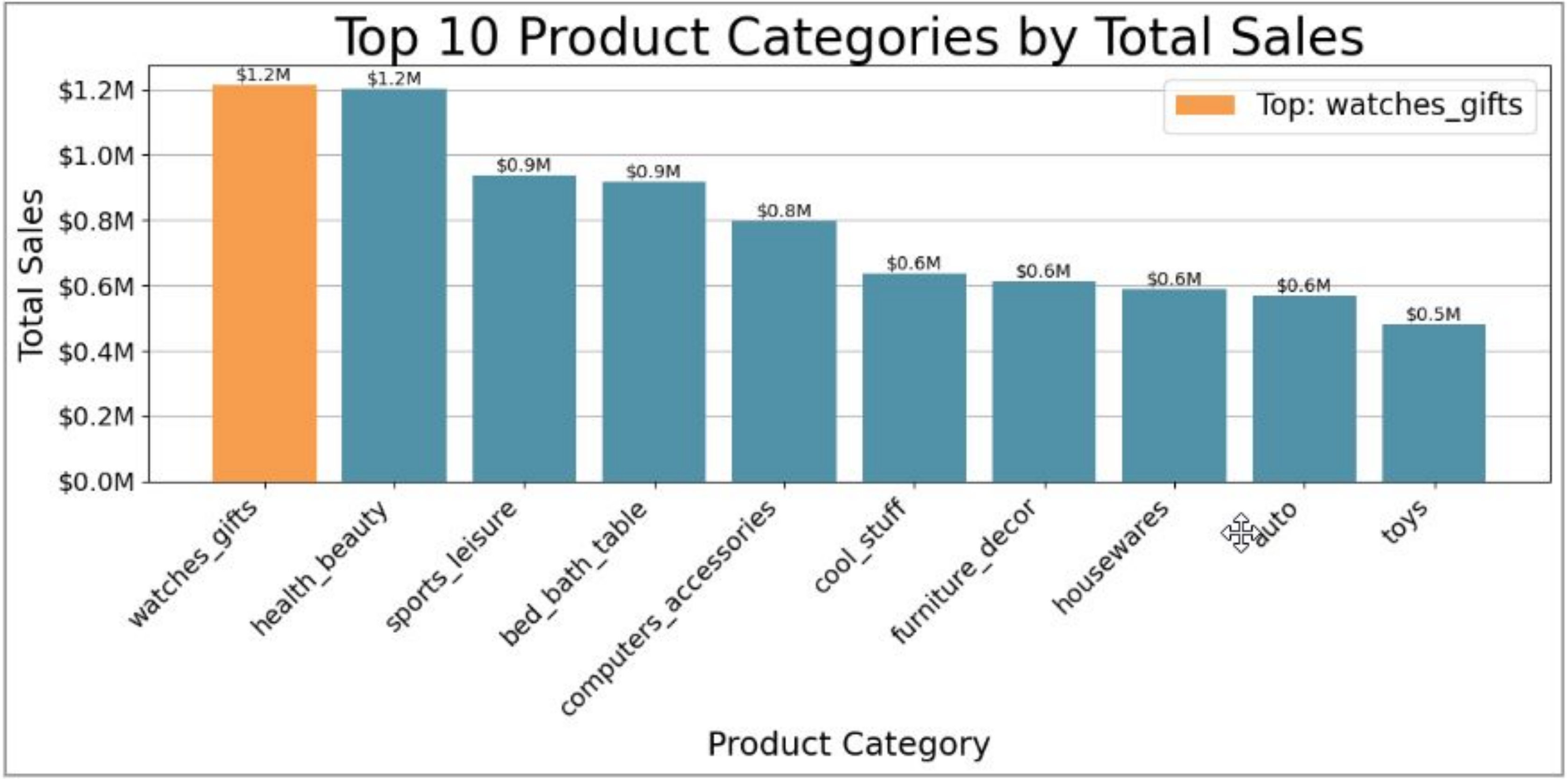
Details

TAGS	OWNER	TYPE	PACKAGE	RELATION	CONTRACT	LOADE
untagged		table	dbt_ecomm	sctp-data-eng-ecomm.ecomm_raw.clean_olist_products	Not Enforced	

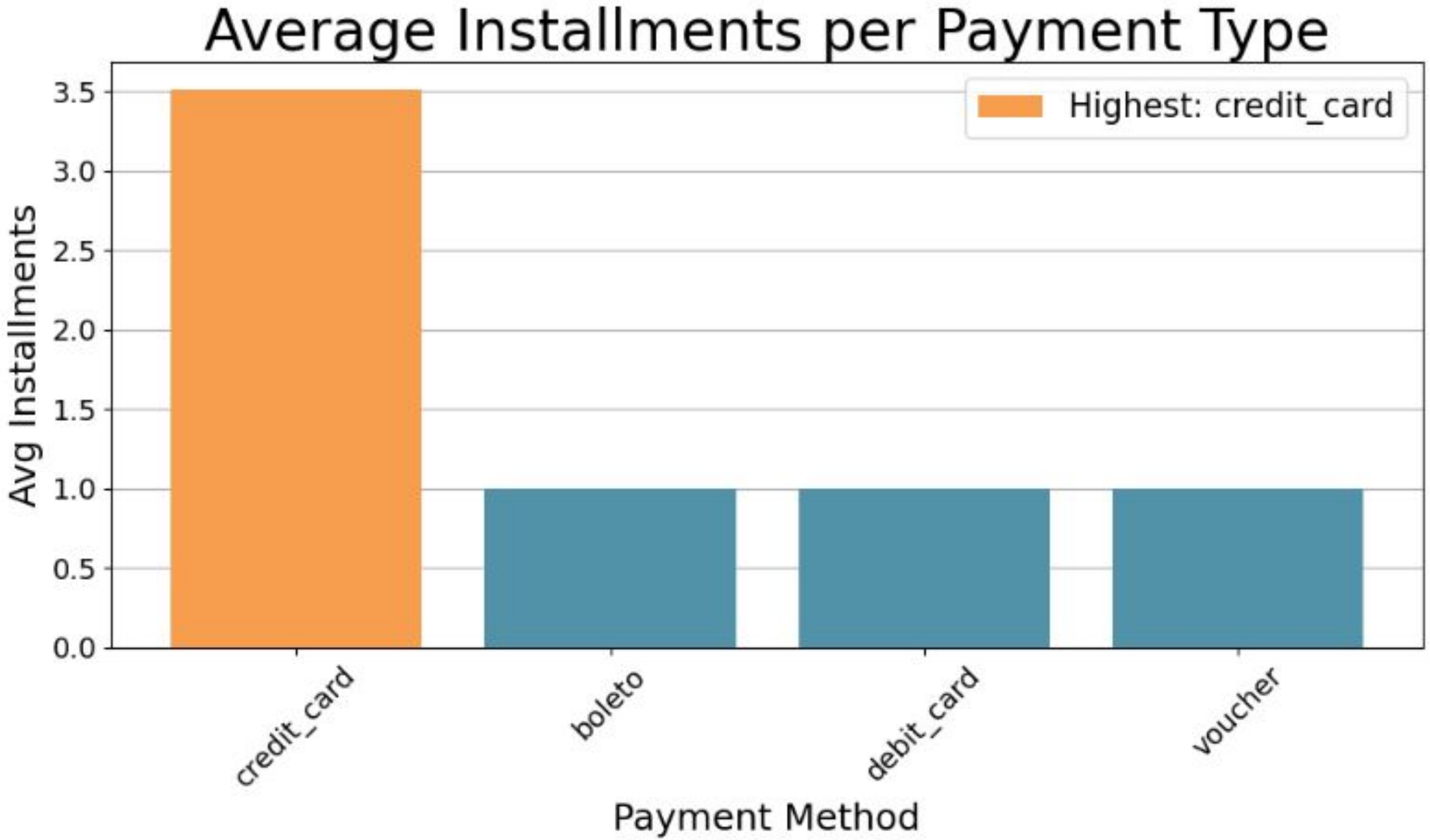
# ROWS	APPROXIMATE SIZE
25,981	2 MB

#2: Docs serve (localhost)

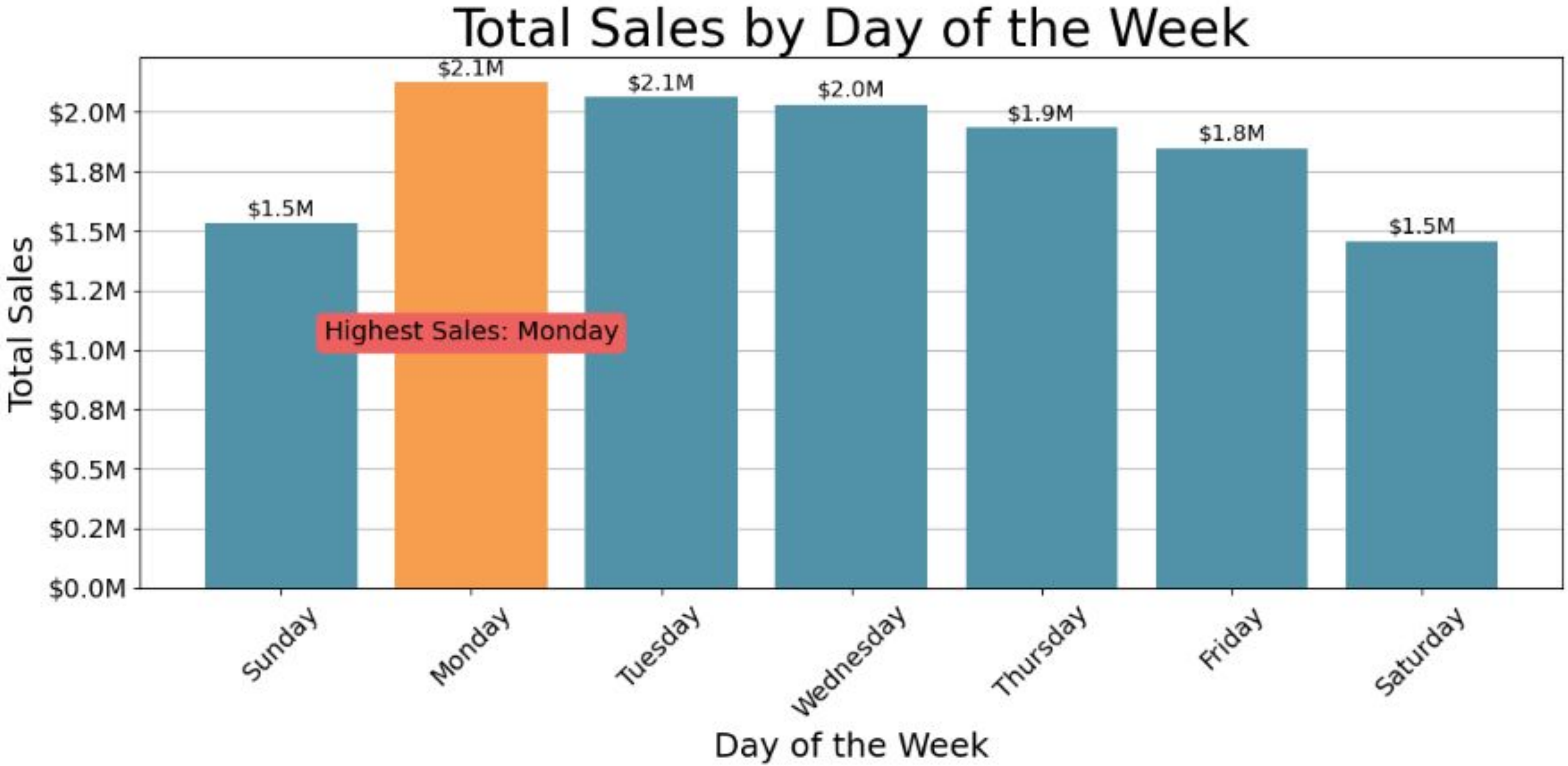
Analysis


Top 10 Product Categories by Total Sales
Top: watches_gifts
Total Sales
$1.2M
$1.0M
$0.8M
$0.6M
$0.4M
$0.2M
$0.0M
$1.2M
$1.2M
$0.9M
$0.9M
$0.8M
$0.6M
$0.6M
$0.6M
$0.6M
$0.5M
watches_gifts
health_beauty
sports_leisure
bed_bath_table
computers_accessories
cool_stuff
furniture_decor
housewares
auto
toys
Product Category

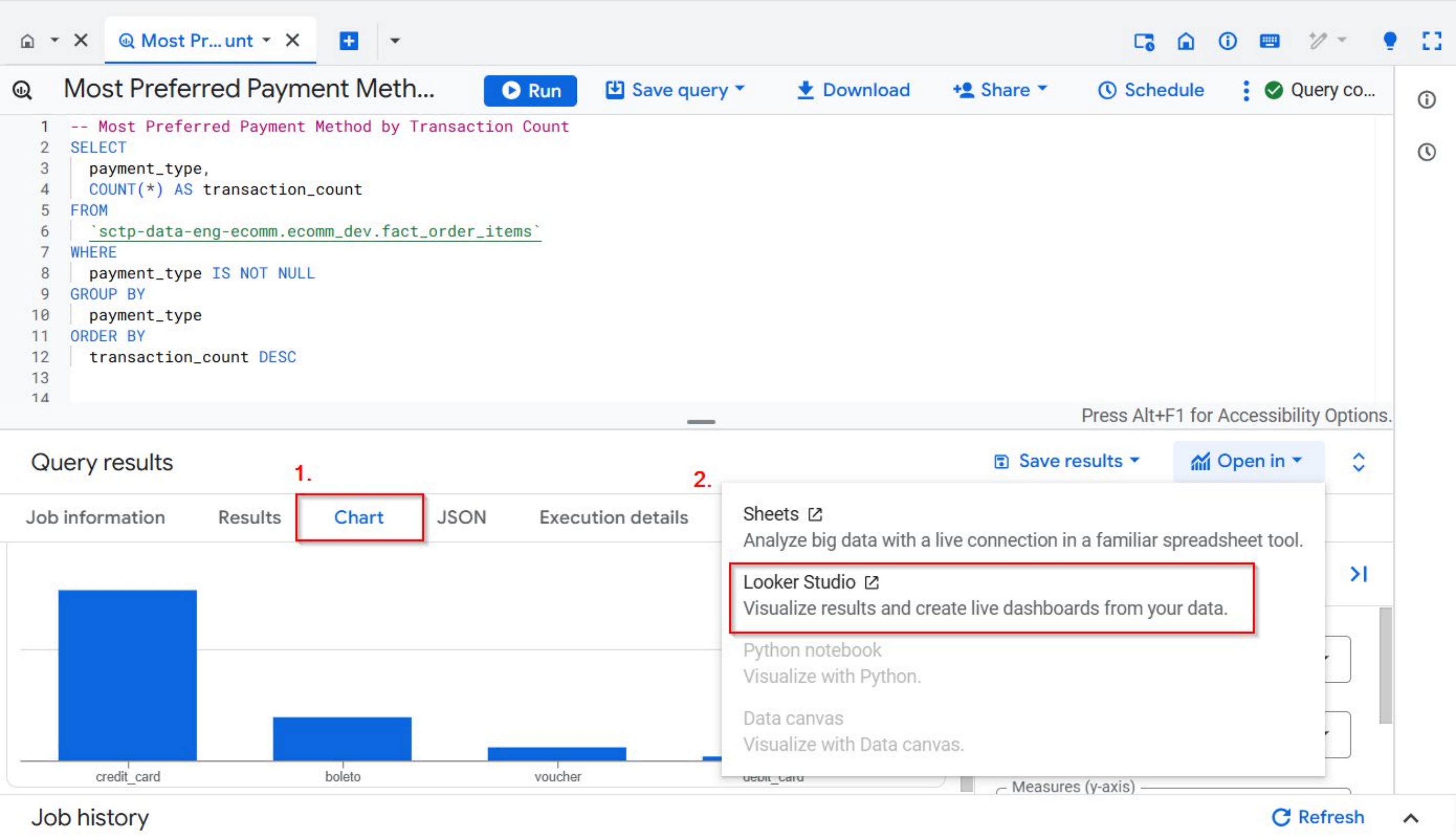
Analysis


Average Installments per Payment Type
Highest: credit_card
Avg Installments
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5
credit_card
boleto
debit_card
voucher
Payment Method

Analysis


Total Sales by Day of the Week
Total Sales
$0.0M
$0.2M
$0.5M
$0.8M
$1.0M
$1.2M
$1.5M
$1.8M
$2.0M
$1.5M
$2.1M
$2.1M
$2.0M
$1.9M
$1.8M
$1.5M
Highest Sales: Monday
Sunday
Monday
Tuesday
Wednesday
Thursday
Friday
Saturday
Day of the Week

Looker Studio



```
-- Most Preferred Payment Method by Transaction Count
SELECT
  payment_type,
  COUNT(*) AS transaction_count
FROM
  `sctp-data-eng-ecomm.ecomm_dev.fact_order_items`
WHERE
  payment_type IS NOT NULL
GROUP BY
  payment_type
ORDER BY
  transaction_count DESC
```

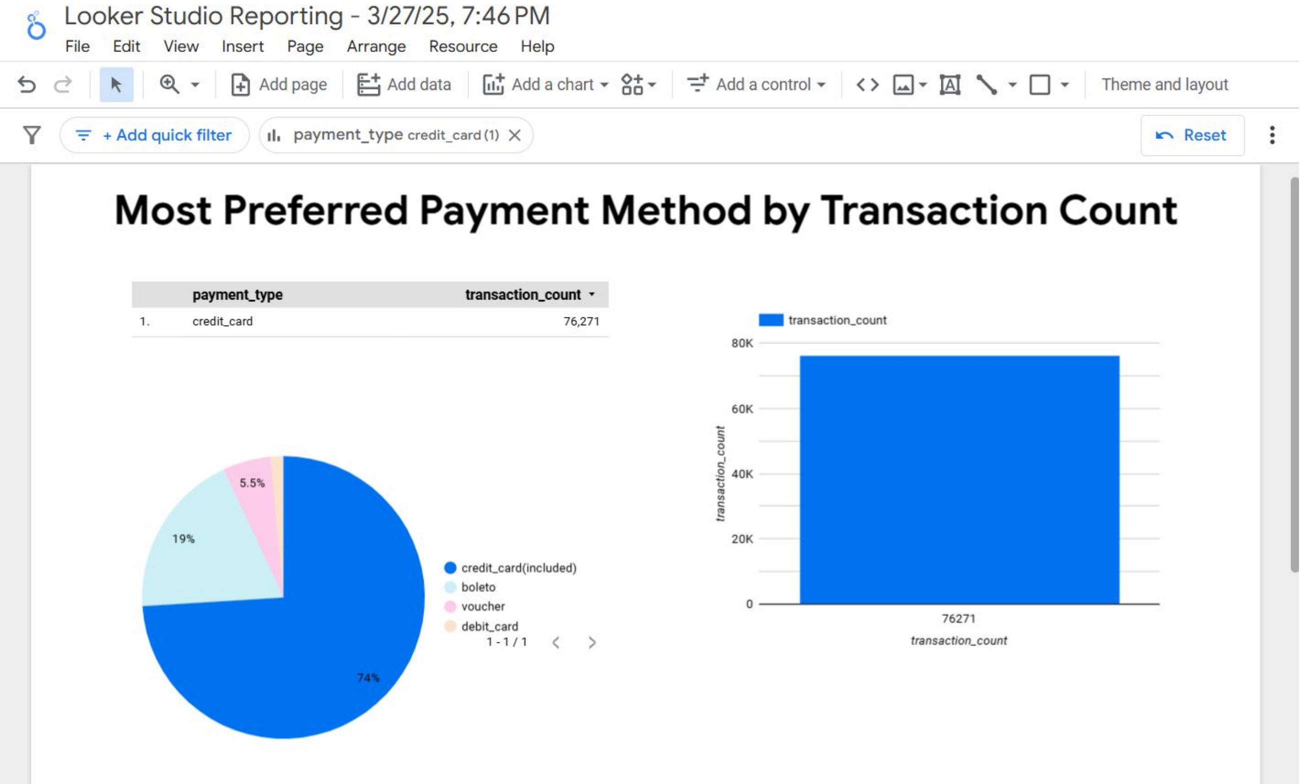
Sheets
Analyze big data with a live connection in a familiar spreadsheet tool.

Looker Studio
Visualize results and create live dashboards from your data.

Python notebook
Visualize with Python.

Data canvas
Visualize with Data canvas.

Looker Studio



Looker Studio Reporting - 3/27/25, 7:46 PM

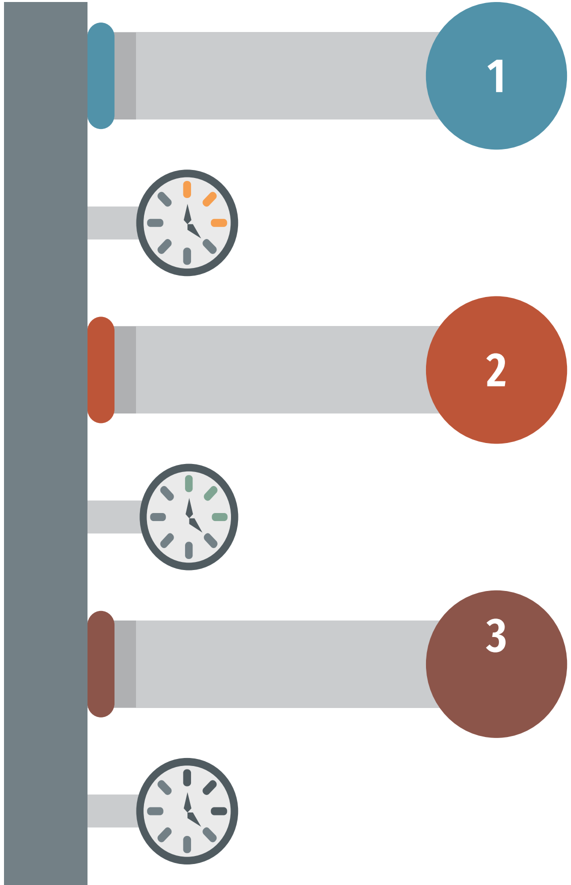
payment_type credit_card (1)

Most Preferred Payment Method by Transaction Count

	payment_type	transaction_count
1.	credit_card	76,271

- Web-based data visualization tool
- create interactive dashboards and reports from various data sources
- To explore, visualize, and share insights easily.

Improvements



Github Secrets

- For better security convert service key to Github Secrets

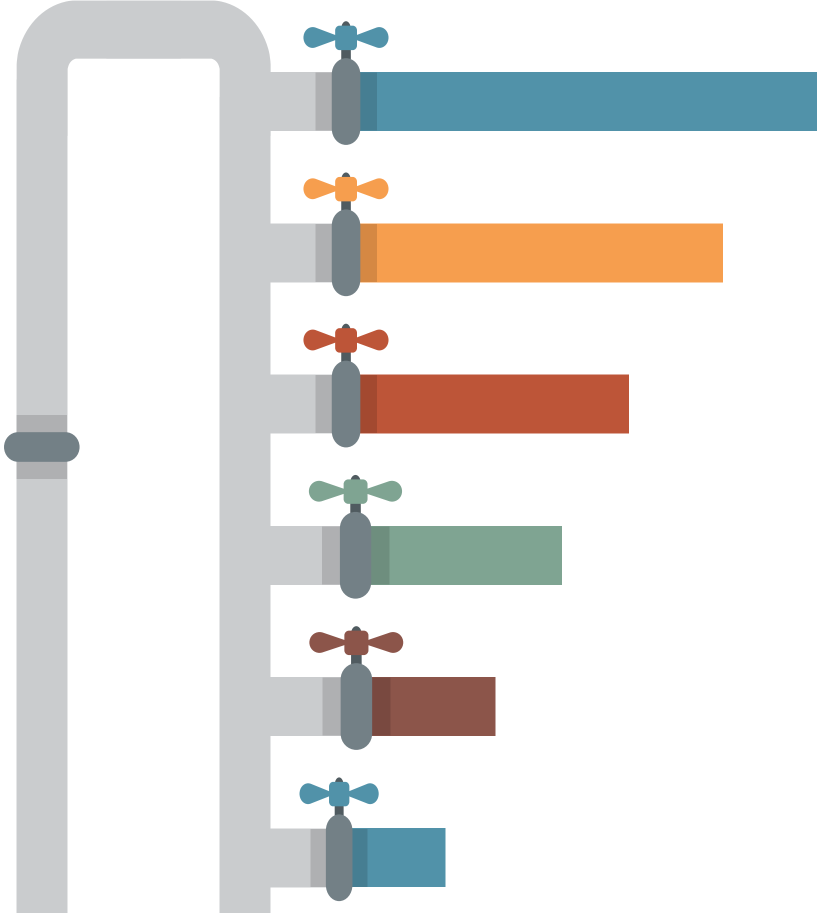
Google Job Scheduler

- Use Google job scheduler to guarantee workflow runs on time

dbt Macro and dbt Unit Test

- Improve data quality and integrity

References



rudderstack

https://www.rudderstack.com/guides/how-to-access-and-query-your-bigquery-data-using-python-and-r/

kaggle

https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce?select=olist_customers_dataset.csv

mbvyn.medium

https://mbvyn.medium.com/understanding-dbt-seeds-and-sources-c5611be17d32

drlee.io

https://drlee.io/unlock-the-power-of-kaggle-how-to-call-datasets-directly-with-python-and-rule-the-data-science-2303cbae4e8a

Chatgpt

https://chatgpt.com/share/67e4e54f-b340-8000-8d01-fce47c0f4870

Slides template

Business Pipeline Infographics by Slidesgo

THANKS!

CREDITS: This template is created by **Slidesgo**