

Relatório MVP Sprint Engenharia de Dados

Pós Graduação Ciência de Dados & Analytics - PUC-Rio

Vitor Jannes - outubro/2023

Objetivo

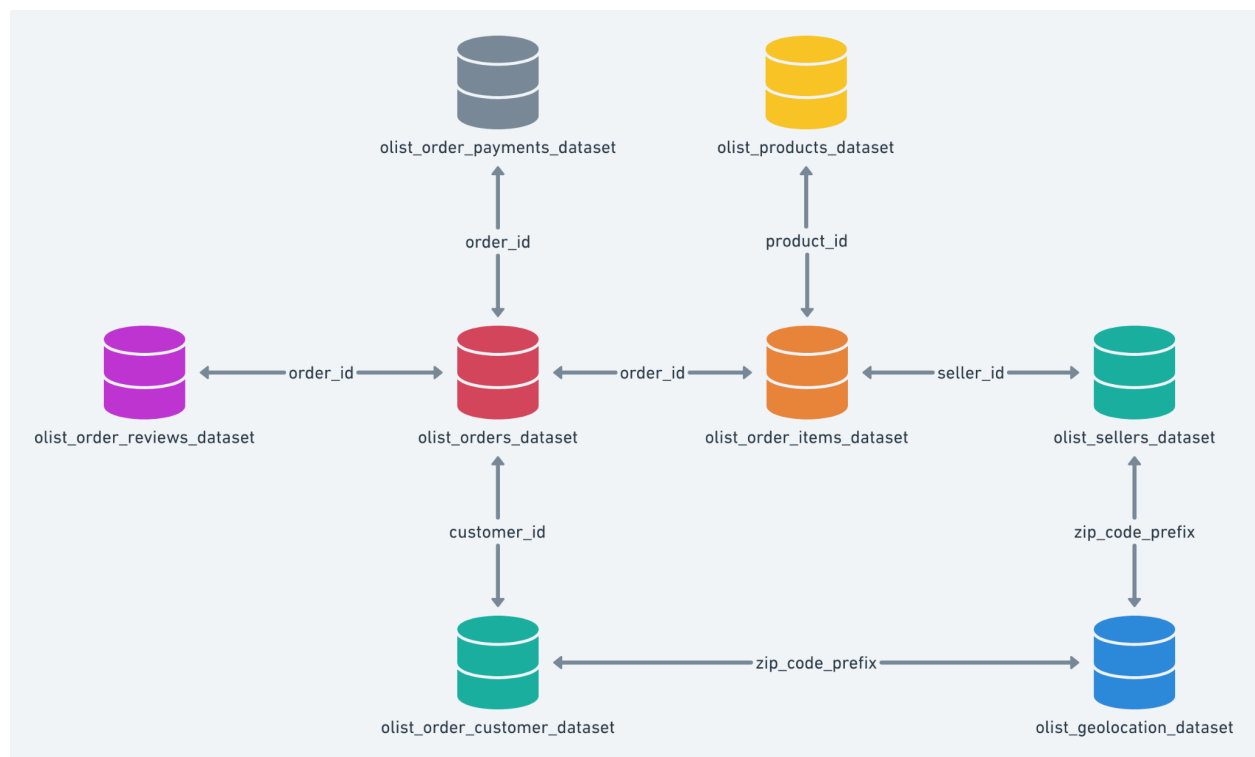
Dado o conjunto de dados de um e-commerce disponibilizado na página <https://www.kaggle.com/datasets/olistbr/brazilian-ecommerce>, as seguintes questões serão investigadas:

1. Quais estados geram maior receita?
2. Qual é a relação valor do frete / valor do produto por Estado?
3. Qual é a média de tempo para entrega por Estado?

Modelo de Dados

Este relatório é baseado em um conjunto de dados disponibilizado por um e-commerce brasileiro sob a licença CC BY-NC-SA 4.0.

Abaixo segue o diagrama de relação das entidades do conjunto que foi disponibilizado em oito arquivos .csv e uma descrição de cada atributo por tabela:



ORDERS

order_id

identificador único do pedido - chave para as tabelas **order_items**, **order_payments** e **order_reviews**.

customer_id

chave para o dataset customer. cada pedido tem um customer_id único - chave para a tabela customers.

order_status

status do pedido (entregue, enviado, cancelado etc).

order_purchase_timestamp

timestamp do ato da compra.

order_approved_at

timestamp da aprovação do pagamento.

order_delivered_carrier_date

timestamp da postagem do pedido, quando ele foi entregue ao parceiro logístico.

order_delivered_customer_date

data que o pedido foi entregue ao cliente.

order_estimated_delivery_date

data estimada de entrega informada ao cliente no ato da compra.

ORDER_ITEMS

Esta tabela lista todos os itens pedidos, sendo um item por entrada.

order_id

identificador único do pedido - chave para a tabela orders

order_item_id

número sequencial identificando o número dos itens inclusos no mesmo pedido

product_id

identificador único do produto - chave para a tabela products

seller_id

identificador único do vendedor - chave para a tabela sellers

shipping_limit_date

data limite para o vendedor entregar o pedido para o parceiro logístico

price

preço do item

freight_value

valor do frete do item - se o pedido tem mais de um item, o valor do frete é dividido entre os itens de acordo com suas medidas e peso.

PRODUCTS

product_id

identificador único do pedido - chave para a tabela order_items

product_category_name

categoria do produto, em português

product_name_lenght

número de caracteres do nome do produto

product_description_lenght

número de caracteres da descrição do produto

product_photos_qty

número de fotos publicadas do produto

product_weight_g

peso do produto em gramas

product_length_cm

comprimento do produto em centímetros

product_height_cm

altura do produto em centímetros

product_width_cm

largura do produto em centímetros

CUSTOMERS

customer_id

cada pedido possui um único identificador - chave para a tabela orders

customer_unique_id

identificador único de um cliente

customer_zip_code_prefix

primeiros 5 dígitos do cep do cliente

customer_city

nome da cidade do cliente

customer_state

nome do Estado do cliente

SELLERS

seller_id

identificador único do vendedor - chave para a tabela order_items

seller_zip_code_prefix
primeiros 5 dígitos do cep do vendedor

seller_city
nome da cidade do vendedor

seller_state
nome do Estado do vendedor

ORDER_PAYMENTS

order_id
identificador único do pedido - chave para a tabela orders

payment_sequential
um cliente pode pagar um pedido com mais de uma forma de pagamento. Se isso acontecer, uma sequência é criada para acomodar todos os pagamentos.

payment_type
método de pagamento escolhido pelo cliente

payment_installments
número de parcelas escolhido pelo cliente

payment_value
valor da transação

ORDER_REVIEWS

review_id
identificador único do review

order_id
identificador único do pedido - chave para a tabela orders

review_score
nota de 1 a 5 dada pelo cliente na pesquisa de satisfação

review_comment_title
título do comentário do review feito pelo cliente

review_comment_message
comentário do review feito pelo cliente

review_creation_date
data em que a pesquisa de satisfação foi enviada ao cliente

review_answer_timestamp
data em que a pesquisa de satisfação foi respondida pelo cliente

Atributos selecionados para resolução do problema

Dadas as questões a serem solucionadas, as tabelas e seus atributos necessários são os seguintes:

customers

customer_id

customer_state

orders

order_id

customer_id

order_purchase_timestamp

order_delivered_customer_date

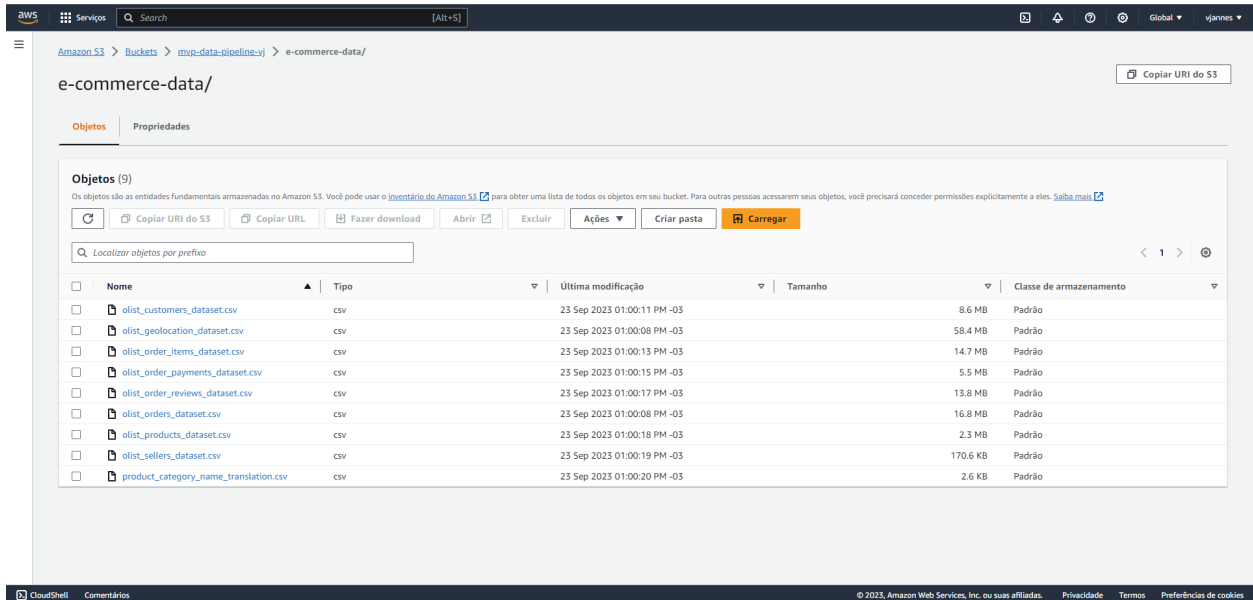
order_items

order_id

price

freight_value

Carregamento de dados



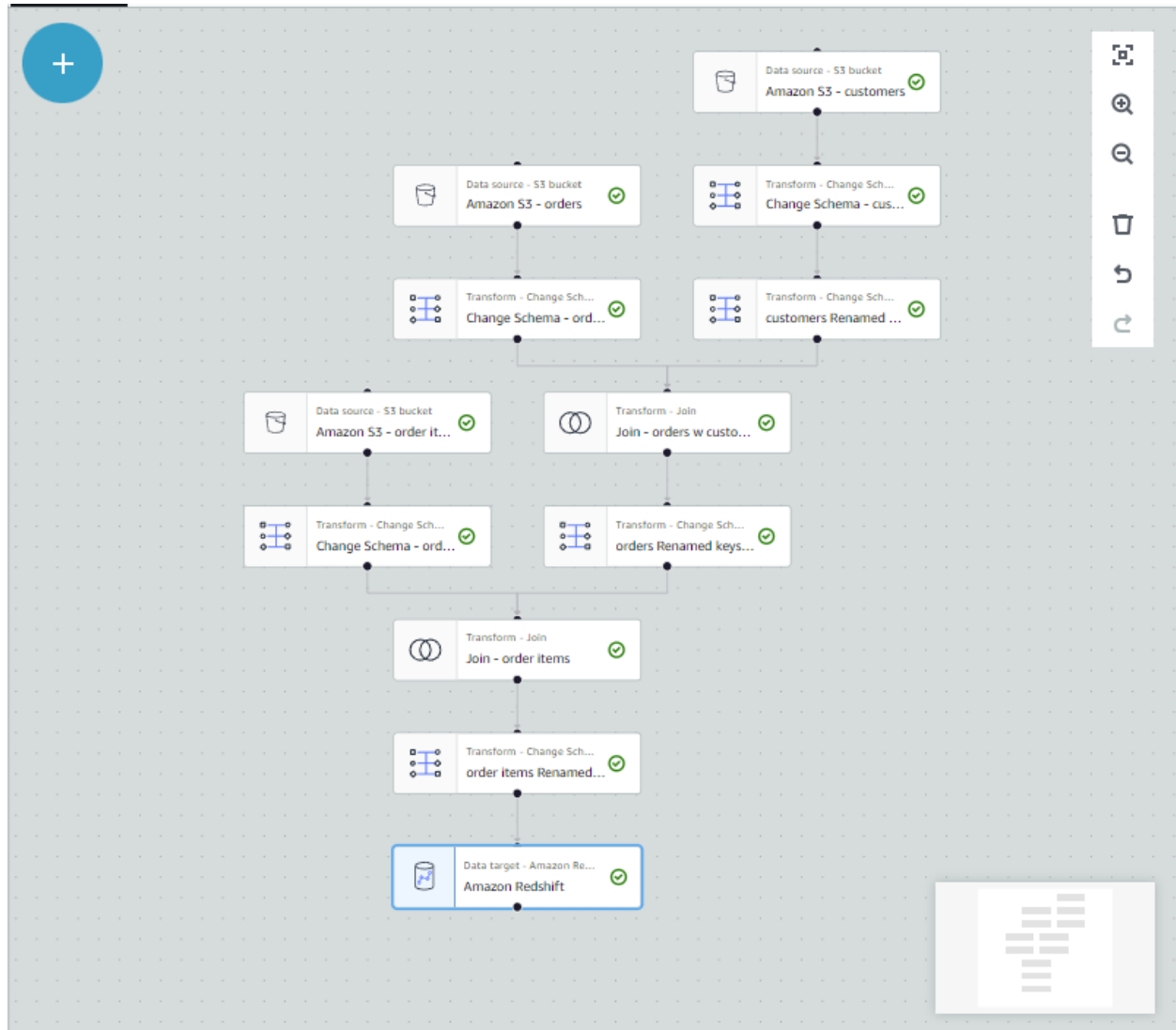
The screenshot shows the Amazon S3 console interface. The breadcrumb navigation indicates the path: Amazon S3 > Buckets > my-data-pipeline-aj > e-commerce-data/. The bucket name 'e-commerce-data/' is displayed at the top. Below the bucket name, there are tabs for 'Objetos' (selected) and 'Propriedades'. A 'Copiar URI do S3' button is visible in the top right corner. The 'Objetos (9)' section shows a list of 9 objects, all of which are CSV files. The table columns are: Nome, Tipo, Última modificação, Tamanho, and Classe de armazenamento. The files listed are: olist_customers_dataset.csv, olist_geolocation_dataset.csv, olist_order_items_dataset.csv, olist_order_payments_dataset.csv, olist_order_reviews_dataset.csv, olist_orders_dataset.csv, olist_products_dataset.csv, olist_sellers_dataset.csv, and product_category_name_translation.csv. All files were last modified on 23 Sep 2023 and are stored in the 'Padrão' storage class. At the bottom of the console, there is a footer with 'CloudShell', 'Comentários', and copyright information for Amazon Web Services, Inc. or its affiliates.

	Nome	Tipo	Última modificação	Tamanho	Classe de armazenamento
<input type="checkbox"/>	olist_customers_dataset.csv	csv	23 Sep 2023 01:00:11 PM -03	8.6 MB	Padrão
<input type="checkbox"/>	olist_geolocation_dataset.csv	csv	23 Sep 2023 01:00:08 PM -03	58.4 MB	Padrão
<input type="checkbox"/>	olist_order_items_dataset.csv	csv	23 Sep 2023 01:00:13 PM -03	14.7 MB	Padrão
<input type="checkbox"/>	olist_order_payments_dataset.csv	csv	23 Sep 2023 01:00:15 PM -03	5.5 MB	Padrão
<input type="checkbox"/>	olist_order_reviews_dataset.csv	csv	23 Sep 2023 01:00:17 PM -03	13.8 MB	Padrão
<input type="checkbox"/>	olist_orders_dataset.csv	csv	23 Sep 2023 01:00:08 PM -03	16.8 MB	Padrão
<input type="checkbox"/>	olist_products_dataset.csv	csv	23 Sep 2023 01:00:18 PM -03	2.3 MB	Padrão
<input type="checkbox"/>	olist_sellers_dataset.csv	csv	23 Sep 2023 01:00:19 PM -03	170.6 KB	Padrão
<input type="checkbox"/>	product_category_name_translation.csv	csv	23 Sep 2023 01:00:20 PM -03	2.6 KB	Padrão

A ferramenta escolhida para criar o pipeline de dados foi a Amazon AWS e, como primeiro passo, os dados foram carregados na nuvem em um Bucket S3.

ETL

Esquema geral do ETL Job



Com o conjunto de dados armazenados na nuvem, foi criado um ETL job na ferramenta AWS Glue para extrair os dados necessários do S3, transformá-los e carregá-los na ferramenta de data warehouse Amazon Redshift para realizar as análises propostas.

A figura acima demonstra o esquema geral do ETL Job criado. O objetivo desse job é que os atributos seleccionados contendo os dados de todos os itens pedidos sejam disponibilizados em uma única tabela para consulta no Redshift. Portanto, primeiro é realizado um “left join” da tabela “orders” com a “customers” a fim de somente adicionar a coluna “customer_state” aos dados dos pedidos. Em seguida, é realizado outro “left join”, agora da tabela “order_items” com a “orders” que contém a informação de “customer_state”. Por fim, obtemos uma tabela com

todos os itens pedidos (base da tabela “order_items”) mas com as colunas de “order_purchase_timestamp”, “order_delivered_customer_date” e “customer_state” adicionadas.

Extração dos dados

Primeiro, o Job extrai os dados do bucket S3 de acordo com os nós demonstrados nas três figuras abaixo.

Como são necessários dados de três tabelas diferentes, foram criados três nós.

The screenshot displays the Databricks interface for a job named "e-commerce-data-job". The top navigation bar includes tabs for "Visual", "Script", "Job details", "Runs", "Data quality", "Schedules", and "Version Control". The main workspace shows a visual pipeline with four nodes: a "Data source - S3 bucket" node labeled "Amazon S3 - customers", followed by three "Transform - Change Schema" nodes. The right-hand sidebar is open to the "Data source properties - S3" configuration panel. This panel includes fields for "Name" (Amazon S3 - customers), "S3 source type" (S3 location), "S3 URL" (s3://mvp-data-pipeline-vj/e-commerce-data/olist_customers_datz), and options for "Recursive" (checked), "Data format" (CSV), "Delimiter" (Comma (,)), "Escape character" (optional), "Quote character" (Double quote (")), and "First line of source file contains column headers" (checked). The "Visual" tab on the left shows a grid of nodes, with the "Data source - S3 bucket" node highlighted.

e-commerce-data-job

Last modified on 26/09/2023, 20:47:55 Try new UI Actions Save Run

Visual Script Job details Runs Data quality New Schedules Version Control

Data source properties - S3 Output schema Data preview

Name
Amazon S3 - orders

S3 source type [Info](#)
☒ S3 location
Choose a file or folder in an S3 bucket.
☐ Data Catalog table

S3 URL
 View Browse S3

☒ Recursive
Read files in all subdirectories.

Data format
CSV

Delimiter
Comma (,)

Escape character - optional
Enter a character to use for escaping

Quote character
Double quote (")

☒ First line of source file contains column headers

Visual Script:

```
graph TD
    DS1[Data source - S3 bucket  
Amazon S3 - orders] --> T1[Transform - Change Schema  
Change Schema - ord...]
    T1 --> DS2[Data source - S3 bucket  
Amazon S3 - order it...]
    DS2 --> T2[Transform - Join - orders w...
```

e-commerce-data-job

Last modified on 26/09/2023, 20:47:55 Try new UI Actions Save Run

Visual Script Job details Runs Data quality New Schedules Version Control

Data source properties - S3 Output schema Data preview

Name
Amazon S3 - order items

S3 source type [Info](#)
☒ S3 location
Choose a file or folder in an S3 bucket.
☐ Data Catalog table

S3 URL
 View Browse S3

☒ Recursive
Read files in all subdirectories.

Data format
CSV

Delimiter
Comma (,)

Escape character - optional
Enter a character to use for escaping

Quote character
Double quote (")

☒ First line of source file contains column headers

Visual Script:

```
graph TD
    DS1[Data source - S3 bucket  
Amazon S3 - orders] --> T1[Transform - Change Schema  
Change Schema - ord...]
    T1 --> DS2[Data source - S3 bucket  
Amazon S3 - order it...]
    DS2 --> T2[Transform - Change Schema  
Change Schema - ord...]
    T2 --> T3[Transform - Join - order items]
```

Transformação

Em seguida, se inicia a etapa de transformação dos dados.

Primeiro, são selecionados somente os atributos necessários para a análise proposta e descartado o restante dos atributos preenchendo o checklist de “Drop”.

Aqui também foi selecionado o tipo de dado para cada atributo - “string”, “timestamp” e “float”.

The screenshot displays the AWS Glue console interface for a job named "e-commerce-data-job". The job is currently in the "Visual" tab, showing a workflow diagram with the following steps:

- Data source - S3 bucket**: Amazon S3 - customers
- Transform - Change Schema**: Change Schema - customers
- Transform - Change Schema**: customers Renamed ...
- Transform - Join**: Join - orders w custo...

The right-hand panel shows the configuration for the "Change Schema - customers" transform. The "Name" field is "Change Schema - customers". The "Node parents" dropdown is set to "Amazon S3 - customers". The "Change Schema (Apply mapping)" table is as follows:

Source key	Target key	Data type	Drop
customer_id	customer_id	string	<input type="checkbox"/>
customer_unique_id			<input checked="" type="checkbox"/>
customer_zip_code_prefix			<input checked="" type="checkbox"/>
customer_city			<input checked="" type="checkbox"/>
customer_state	customer_state	string	<input type="checkbox"/>

e-commerce-data-job

Last modified on 26/09/2023, 20:47:55

Try new UI

Actions

Save

Run

Visual

Script

Job details

Runs

Data quality New

Schedules

Version Control

Data source - S3 bucket

Amazon S3 - orders

Transform - Change Sch...

Change Schema - ord...

Data source - S3 bucket

Amazon S3 - order it...

Transform - Change Sch...

Change Schema - ord...

Transform - Join

Join - orders w custo...

Transform -

orders Renamed keys...

Transform

Output schema

Data preview

Name

Change Schema - orders

Node parents

Choose which nodes will provide inputs for this one.

Choose one or more parent node

Amazon S3 - orders X

S3 - DataSource

Change Schema (Apply mapping)

Source key	Target key	Data type	Drop
order_id	order_id	string	<input type="checkbox"/>
customer_id	customer_id	string	<input type="checkbox"/>
order_status			<input checked="" type="checkbox"/>
order_purchase_timestamp	order_purchase_timestr	timestamp	<input type="checkbox"/>
order_approved_at			<input checked="" type="checkbox"/>
order_delivered_carrier_date			<input checked="" type="checkbox"/>
order_delivered_customer_date	order_delivered_customr	timestamp	<input type="checkbox"/>
order_estimated_delivery_date			<input checked="" type="checkbox"/>

e-commerce-data-job

Last modified on 26/09/2023, 20:47:55

Try new UI

Actions

Save

Run

Visual

Script

Job details

Runs

Data quality New

Schedules

Version Control

Data source - S3 bucket

Amazon S3 - order it...

Transform - Change Sch...

Change Schema - ord...

Transform - Join

Join - order items

Transform -

order items Renamed...

Transform

Output schema

Data preview

Name

Change Schema - order items

Node parents

Choose which nodes will provide inputs for this one.

Choose one or more parent node

Amazon S3 - order items X

S3 - DataSource

Change Schema (Apply mapping)

Source key	Target key	Data type	Drop
order_id	order_id	string	<input type="checkbox"/>
order_item_id			<input checked="" type="checkbox"/>
product_id			<input checked="" type="checkbox"/>
seller_id			<input checked="" type="checkbox"/>
shipping_limit_date			<input checked="" type="checkbox"/>
price	price	float	<input type="checkbox"/>
freight_value	freight_value	float	<input type="checkbox"/>

12

Para realizar os “joins”, o Glue requisita que não haja atributos com nomes idênticos nas tabelas, portanto foi adicionada uma etapa renomeando os atributos com o prefixo do nome da tabela “customers” antes do join:

The screenshot shows the AWS Glue console interface for a job named "e-commerce-data-job". The workflow is visualized on the left, showing a sequence of nodes: "Data source - S3 bucket Amazon S3 - customers", "Transform - Change Schema - customers", "Transform - Change Schema - orders", "Transform - Join", and "Transform - Change Schema - orders Renamed keys...". The "Transform - Change Schema - customers" node is selected, and the right sidebar displays its configuration.

Transform Configuration:

- Name:** customers Renamed keys for Join
- Node parents:** Choose one or more parent node (Change Schema - customers)
- Change Schema (Apply mapping):**

Source key	Target key	Data type	Drop
customer_id	customers_customer_id	string	<input type="checkbox"/>
customer_state	customers_customer_st	string	<input type="checkbox"/>

É realizado o “left join” com a tabela “orders” à esquerda e “customers” à direita (a chave é o atributo “customer_id”), mantendo os dados de “orders” e adicionando a coluna de “customer_state” a esses dados:

The screenshot shows the same AWS Glue console interface, but with the "Transform - Join" node selected. The right sidebar displays the join configuration.

Join Configuration:

- Name:** Join - orders w customers
- Node parents:** Choose one or more parent node (Change Schema - orders, customers Renamed keys for Join)
- Join type:** Left join (Select all rows from the left dataset and the rows that meet the join condition from the right dataset.)
- Join conditions:** Select a field from each parent node for the join condition.

Change Schema - orders	customers Renamed keys for Join
customer_id	customers_customer_id

Da mesma forma que foi feito com a tabela “customers” antes do join, foi incluído o prefixo no nome dos atributos da tabela resultante do join de “orders” com “customers”:

The screenshot shows the AWS Glue console interface for a workflow named "e-commerce-data-job". The workflow consists of several nodes: "Data source - S3 bucket", "Transform - Change Schema", "Transform - Join", and "Transform - Change Schema". The "Transform - Join" node is selected, and the "Change Schema (Apply mapping)" panel is open. The panel shows the mapping of source keys to target keys for the join operation.

Source key	Target key	Data type	Drop
order_id	orders_order_id	string	<input type="checkbox"/>
customer_id	orders_customer_id	string	<input type="checkbox"/>
order_purchase_timestamp	orders_order_purchase	timestamp	<input type="checkbox"/>
order_delivered_customer_date	orders_order_delivered	timestamp	<input type="checkbox"/>
customers_customer_id	orders_customers_custc	string	<input type="checkbox"/>
customers_customer_state	orders_customers_custc	string	<input type="checkbox"/>

É realizado o “left join” de “order_items” à esquerda com a tabela resultante do primeiro join à direita, a chave é o atributo “order_id”:

The screenshot shows the AWS Glue console interface for a workflow named "e-commerce-data-job". The workflow consists of several nodes: "Data source - S3 bucket", "Transform - Change Schema", "Transform - Join", "Transform - Change Schema", and "Data target - Amazon Redshift". The "Transform - Join" node is selected, and the "Join type" is set to "Left join". The "Join conditions" panel is open, showing the condition "order_id = orders_order_id".

Join type: Left join
Select all rows from the left dataset and the rows that meet the join condition from the right dataset.

Join conditions: Select a field from each parent node for the join condition.

Change Schema - order items = orders Renamed keys for Join
order_id = orders_order_id

Assim, é obtida a tabela com os atributos necessários. É feita mais uma etapa para renomear os atributos finais e realizar o “drop” de atributos duplicados:

The screenshot shows the AWS Glue console interface for a job named 'e-commerce-data-job'. The job is configured in the 'Visual' tab, displaying a workflow with the following nodes:

- Change Schema - order items** (Transform - Change Schema)
- Join - order items** (Transform - Join)
- order items Renamed keys** (Transform - Change Schema)
- Amazon Redshift** (Data target - Amazon Redshift)

The 'Transform' node 'order items Renamed keys' is selected, and the 'Change Schema (Apply mapping)' panel is open on the right. This panel shows a table with columns: Source key, Target key, Data type, and Drop. The 'Drop' column has checkboxes for 'orders_order_id' and 'orders_customers_customer_id', which are checked.

Source key	Target key	Data type	Drop
order_id	order_id	string	<input type="checkbox"/>
price	price	float	<input type="checkbox"/>
freight_value	freight_value	float	<input type="checkbox"/>
orders_order_id			<input checked="" type="checkbox"/>
orders_customer_id	customer_id	string	<input type="checkbox"/>
orders_order_purchase_timestamp	order_purchase_timestamp	timestamp	<input type="checkbox"/>
orders_order_delivered_customer_id	order_delivered_customer_id	timestamp	<input type="checkbox"/>
orders_customers_customer_id			<input checked="" type="checkbox"/>
orders_customers_customer_state	customer_state	string	<input type="checkbox"/>

Foi criada uma tabela vazia “orders_joined” no Redshift para ser populada pelo job:

The screenshot shows the AWS Redshift console interface for the 'Redshift query editor v2'. The editor displays a SQL query:

```
1 create table public.orders_joined (order_id varchar);
```

The 'Result 1' panel shows the query execution summary, including the number of rows returned (0) and the elapsed time (966ms). The 'Summary' panel shows the query text and the result set query.

Field	Type	NL	CMP
A	order_id	character varying(256)	NULL

Carga no Redshift

Por fim, foi configurada a etapa de carga no Redshift selecionando a tabela recém criada “orders_joined” como destino dos dados:

The screenshot shows the AWS Glue console interface for a workflow named "e-commerce-data-job". The workflow is visualized on the left, showing a sequence of nodes: "Join - order items", "Transform - Change Sch... order items Renamed...", and "Data target - Amazon Re... Amazon Redshift". The "Data target - Amazon Re... Amazon Redshift" node is selected, and its configuration is shown on the right. The configuration includes:

- Name:** Amazon Redshift
- Node parents:** Choose one or more parent node (dropdown menu)
- Redshift access type:** ☒ Direct data connection - recommended, ☐ Glue Data Catalog tables
- Redshift connection:** Choose the AWS Glue connection for Amazon Redshift, or create a new connection (dropdown menu showing "mvp-glue-redshift")
- Connection:** View properties (link)
- Database:** dev
- Schema:** Choose your Amazon Redshift schema. (dropdown menu showing "public")
- Table:** Search and enter the name of the source Amazon Redshift table. (input field showing "orders_joined")
- Handling of data and target table:** ☒ APPEND (insert) to target table, ☐ MERGE data into target table, ☐ TRUNCATE target table, ☐ DROP and recreate target table
- ☐ Also update existing records in target table
- Performance and security**
- Custom Redshift parameters - optional**

The screenshot shows the AWS Glue console interface for a workflow named "e-commerce-data-job". The workflow is visualized on the left, showing a sequence of nodes: "Join - order items", "Transform - Change Sch... order items Renamed...", and "Data target - Amazon Re... Amazon Redshift". The "Data target - Amazon Re... Amazon Redshift" node is selected, and its configuration is shown on the right. The configuration includes:

- Name:** Amazon Redshift
- Node parents:** Choose one or more parent node (dropdown menu)
- Redshift access type:** ☒ Direct data connection - recommended, ☐ Glue Data Catalog tables
- Redshift connection:** Choose the AWS Glue connection for Amazon Redshift, or create a new connection (dropdown menu showing "mvp-glue-redshift")
- Connection:** View properties (link)
- Database:** dev
- Schema:** Choose your Amazon Redshift schema. (dropdown menu showing "public")
- Table:** Search and enter the name of the source Amazon Redshift table. (input field showing "orders_joined")
- Handling of data and target table:** ☒ APPEND (insert) to target table, ☐ MERGE data into target table, ☐ TRUNCATE target table, ☐ DROP and recreate target table
- ☐ Also update existing records in target table
- Performance and security**
- Custom Redshift parameters - optional**

É possível observar o esquema de dados final resultante do job:

e-commerce-data-job Last modified on 26/09/2023, 20:47:55 Try new UI Actions Save Run

Visual Script Job details Runs Data quality New Schedules Version Control

Transform - Join **Join - order items** ✓

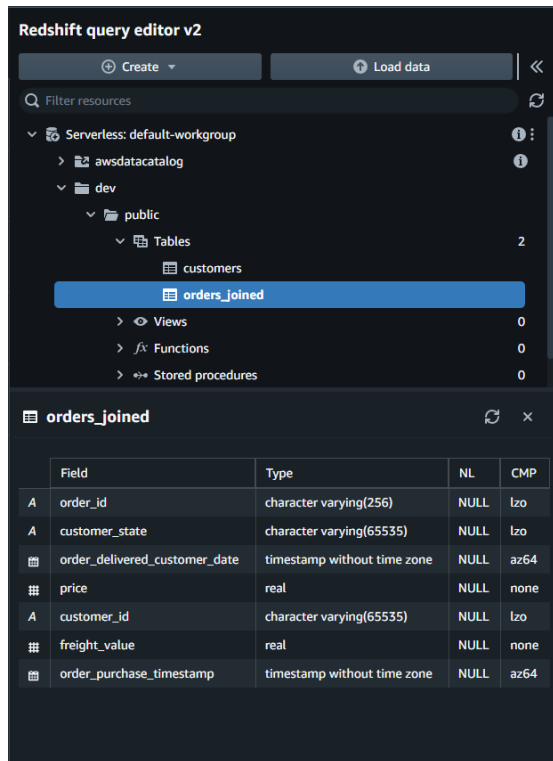
Transform - Change Sch... **order items Renamed...** ✓

Data target - Amazon Re... **Amazon Redshift** ✓

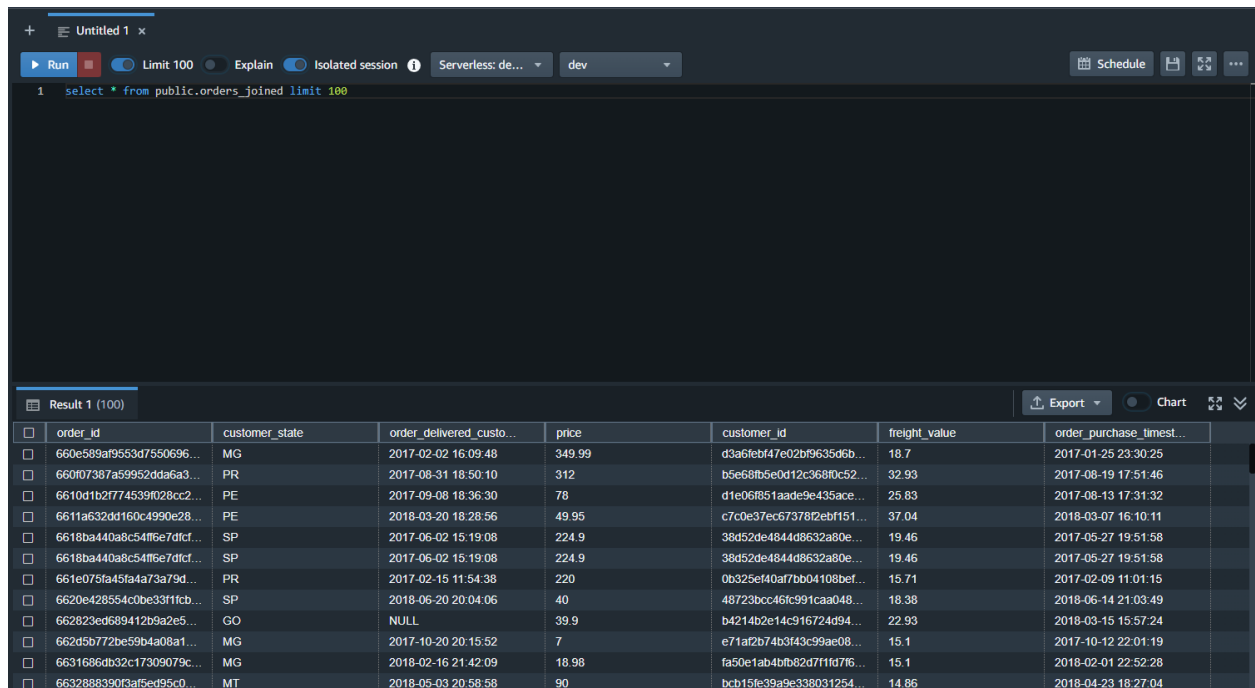
Data target properties - Amazon Redshift **Output schema** **Data preview**

Schema

Key	Data type
order_id	string
price	float
freight_value	float
customer_id	string
order_purchase_timestamp	timestamp
order_delivered_customer_date	timestamp
customer_state	string



Após rodar o job, a tabela foi populada com sucesso:



Análise

Qualidade dos dados

Foi utilizado um notebook Python para analisar os arquivos .csv originais e a tabela “orders_joined” resultante do job.

A função “describe” traz um resumo estatístico dos dados, onde é possível observar que a tabela “orders” contém dados de 99.441 pedidos únicos realizados de set/2016 a out/2018, todos com “order_id”, “customer_id” e “order_purchase_timestamp” não nulos. Já “order_delivered_customer_date” possui valores nulos pois sua contagem é 96.476, que é menor do que 99.441. Ou seja, possui 2.965 valores nulos.

A princípio, não é necessária nenhuma ação em relação aos valores nulos.

```
# resumo estatístico do dataset 'orders'
# include=all inclui todos os atributos do dataframe no resumo
# datetime_is_numeric=True trata os atributos datetime como numéricos neste resumo
orders.describe(include='all', datetime_is_numeric=True)
```

	order_id	customer_id	order_status	order_purchase_timestamp	order_approved_at	order_delivered_carrier_date	order_delivered_customer_date	order_estimated_delivery_date
count	99441	99441	99441	99441	99281	97658	96476	99441
unique	99441	99441	8	NaN	NaN	NaN	NaN	NaN
top	e481f51cbd54678b7cc49136f2d6aff7	9ef432eb6251297304e76186b10a928d	delivered	NaN	NaN	NaN	NaN	NaN
freq	1	1	96478	NaN	NaN	NaN	NaN	NaN
mean	NaN	NaN	NaN	2017-12-31 08:43:12.776581120	2017-12-31 18:35:24.098800128	2018-01-04 21:49:48.138278656	2018-01-14 12:09:19.035542272	2018-01-24 03:08:37.730111232
min	NaN	NaN	NaN	2016-09-04 21:15:19	2016-09-15 12:16:38	2016-10-08 10:34:01	2016-10-11 13:46:32	2016-09-30 00:00:00
25%	NaN	NaN	NaN	2017-09-12 14:46:19	2017-09-12 23:24:16	2017-09-15 22:28:50.249999872	2017-09-25 22:07:22.249999872	2017-10-03 00:00:00
50%	NaN	NaN	NaN	2018-01-18 23:04:36	2018-01-19 11:36:13	2018-01-24 16:10:58	2018-02-02 19:28:10.5000000	2018-02-15 00:00:00
75%	NaN	NaN	NaN	2018-05-04 15:42:16	2018-05-04 20:35:10	2018-05-08 13:37:45	2018-05-15 22:48:52.249999872	2018-05-25 00:00:00
max	NaN	NaN	NaN	2018-10-17 17:30:18	2018-09-03 17:40:06	2018-09-11 19:48:28	2018-10-17 13:22:46	2018-11-12 00:00:00

A tabela “customers” também possui dados de 99.441 pedidos e todos possuem dados de “customer_state” preenchidos. A cidade com mais vendas é São Paulo, assim como o Estado com mais vendas também é São Paulo.

```
customers.describe(include='all')
```

	customer_id	customer_unique_id	customer_zip_code_prefix	customer_city	customer_state
count	99441	99441	99441.000000	99441	99441
unique	99441	96096	NaN	4119	27
top	06b89999e2fba1a1fbc88172c00ba8bc7	8d50f5eadf50201ccdcdfb9e2ac8455	NaN	sao paulo	SP
freq	1	17	NaN	15540	41746
mean	NaN	NaN	35137.474583	NaN	NaN
std	NaN	NaN	29797.938996	NaN	NaN
min	NaN	NaN	1003.000000	NaN	NaN
25%	NaN	NaN	11347.000000	NaN	NaN
50%	NaN	NaN	24416.000000	NaN	NaN
75%	NaN	NaN	58900.000000	NaN	NaN
max	NaN	NaN	99990.000000	NaN	NaN

A tabela “order_items” possui dados de 112.650 itens pedidos com dados de “price” e “freight_value” sem nulos. A média do preço do produto é de R\$ 120 e do frete é de R\$ 20.

```
order_items.describe(include='all', datetime_is_numeric=True)
```

	order_id	order_item_id	product_id	seller_id	shipping_limit_date	price	freight_value
count	112650	112650.000000	112650	112650	112650	112650.000000	112650.000000
unique	98666	NaN	32951	3095	NaN	NaN	NaN
top	8272b63d03f5f79c56e9e4120aec44ef	NaN	aca2eb7d00ea1a7b8ebd4e68314663af	6560211a19b47992c3666cc44a7e94c0	NaN	NaN	NaN
freq	21	NaN	527	2033	NaN	NaN	NaN
mean	NaN	1.197834	NaN	NaN	2018-01-07 15:36:52.192685312	120.653739	19.990320
min	NaN	1.000000	NaN	NaN	2016-09-19 00:15:34	0.850000	0.000000
25%	NaN	1.000000	NaN	NaN	2017-09-20 20:57:27.500000	39.900000	13.080000
50%	NaN	1.000000	NaN	NaN	2018-01-26 13:59:35	74.990000	16.260000
75%	NaN	1.000000	NaN	NaN	2018-05-10 14:34:00.750000128	134.900000	21.150000
max	NaN	21.000000	NaN	NaN	2020-04-09 22:35:08	6735.000000	409.680000
std	NaN	0.705124	NaN	NaN	NaN	183.633928	15.806405

A tabela resultante do job contém 112.650 itens pedidos e valores nulos somente para a variável de “order_delivered_customer_date” como o esperado.

```
orders_joined.describe(include='all', datetime_is_numeric=True)
```

	order_id	price	freight_value	order_purchase_timestamp	order_delivered_customer_date	customer_state
count	112650	112650.000000	112650.000000	112650	110196	112650
unique	98666	NaN	NaN	NaN	NaN	27
top	8272b63d03f5f79c56e9e4120aec44ef	NaN	NaN	NaN	NaN	SP
freq	21	NaN	NaN	NaN	NaN	47449
mean	NaN	120.653739	19.990320	2018-01-01 00:09:48.464376576	2018-01-14 13:25:24.023939328	NaN
min	NaN	0.850000	0.000000	2016-09-04 21:15:19	2016-10-11 13:46:32	NaN
25%	NaN	39.900000	13.080000	2017-09-13 19:17:04	2017-09-26 20:09:44.500000	NaN
50%	NaN	74.990000	16.260000	2018-01-19 23:02:16	2018-02-02 20:57:23	NaN
75%	NaN	134.900000	21.150000	2018-05-04 17:30:36.750000128	2018-05-15 20:09:21.500000	NaN
max	NaN	6735.000000	409.680000	2018-09-03 09:06:57	2018-10-17 13:22:46	NaN
std	NaN	183.633928	15.806405	NaN	NaN	NaN

Solução do Problema

1. Quais estados geram maior receita?

```
1 select customer_state, round(sum(price),0) as receita
2 from public.orders_joined
3 group by customer_state
4 order by receita desc
```

Result 1 (27)		
<input type="checkbox"/>	customer_state	receita
<input type="checkbox"/>	SP	5202955
<input type="checkbox"/>	RJ	1824093
<input type="checkbox"/>	MG	1585308
<input type="checkbox"/>	RS	750304
<input type="checkbox"/>	PR	683084
<input type="checkbox"/>	SC	520553
<input type="checkbox"/>	BA	511350
<input type="checkbox"/>	DF	302604
<input type="checkbox"/>	GO	294592
<input type="checkbox"/>	ES	275037
<input type="checkbox"/>	PE	262788
<input type="checkbox"/>	CE	227255
<input type="checkbox"/>	PA	178948
<input type="checkbox"/>	MT	156454
<input type="checkbox"/>	MA	119648
<input type="checkbox"/>	MS	116813
<input type="checkbox"/>	PB	115268
<input type="checkbox"/>	PI	86914
<input type="checkbox"/>	RN	83035
<input type="checkbox"/>	AL	80315
<input type="checkbox"/>	SE	58921
<input type="checkbox"/>	TO	49622
<input type="checkbox"/>	RO	46141
<input type="checkbox"/>	AM	22357
<input type="checkbox"/>	AC	15983
<input type="checkbox"/>	AP	13474
<input type="checkbox"/>	RR	7829

Observa-se que São Paulo é o Estado que gerou maior receita no período (R\$ 5,2 mm), seguido por Rio de Janeiro (R\$ 1,8 mm) e Minas Gerais (R\$ 1,6 mm), três componentes do Sudeste. Na sequência vêm os três Estados do Sul do país, seguidos por Bahia, DF, Goiás e Espírito Santo. Os Estados do Nordeste e Norte figuram na parte de baixo da lista, com as menores receitas geradas.

2. Qual é a relação valor do frete / valor do produto por Estado?

```
1 select customer_state, round(sum(freight_value)/sum(price),2) as razao_frete
2 from public.orders_joined
3 group by customer_state
4 order by razao_frete desc
```

Result 1 (27)		
<input type="checkbox"/>	customer_state	razao_frete
<input type="checkbox"/>	RR	0.29
<input type="checkbox"/>	MA	0.26
<input type="checkbox"/>	RO	0.25
<input type="checkbox"/>	AM	0.25
<input type="checkbox"/>	SE	0.24
<input type="checkbox"/>	PI	0.24
<input type="checkbox"/>	TO	0.24
<input type="checkbox"/>	RN	0.22999999999999998
<input type="checkbox"/>	AC	0.22999999999999998
<input type="checkbox"/>	PE	0.22999999999999998
<input type="checkbox"/>	PB	0.22000000000000003
<input type="checkbox"/>	PA	0.22000000000000003
<input type="checkbox"/>	CE	0.21000000000000002
<input type="checkbox"/>	AP	0.21000000000000002
<input type="checkbox"/>	AL	0.2
<input type="checkbox"/>	BA	0.2
<input type="checkbox"/>	MT	0.19
<input type="checkbox"/>	GO	0.18
<input type="checkbox"/>	RS	0.18
<input type="checkbox"/>	ES	0.18
<input type="checkbox"/>	SC	0.16999999999999998
<input type="checkbox"/>	RJ	0.16999999999999998
<input type="checkbox"/>	DF	0.16999999999999998
<input type="checkbox"/>	PR	0.16999999999999998
<input type="checkbox"/>	MG	0.16999999999999998
<input type="checkbox"/>	MS	0.16
<input type="checkbox"/>	SP	0.13999999999999999

É possível observar que os Estados do Sudeste e Sul figuram na parte de baixo da lista, sendo São Paulo o de menor relação custo de frete por valor do produto. Enquanto os Estados do

Norte e Nordeste aparecem no topo da lista com as maiores relações frete por valor do produto. Em geral, as regiões Norte e Nordeste possuem maiores custos logísticos.

3. Qual é a média de tempo para entrega por Estado?

```
1 select customer_state,  
2 round(AVG(DATEDIFF(day, order_purchase_timestamp, order_delivered_customer_date)),0) as media_dias_entrega  
3 from public.orders_joined  
4 group by customer_state  
5 order by media_dias_entrega desc
```

Nesta query, os valores nulos de “order_delivered_customer_date” não impactam o resultado porque a função “AVG” ignora tais valores em seu cálculo, utilizando somente os valores não nulos para cálculo da média.

Result 1 (27)		
<input type="checkbox"/>	customer_state	media_dias_entrega
<input type="checkbox"/>	AP	28
<input type="checkbox"/>	RR	28
<input type="checkbox"/>	AM	26
<input type="checkbox"/>	AL	24
<input type="checkbox"/>	PA	23
<input type="checkbox"/>	SE	21
<input type="checkbox"/>	MA	21
<input type="checkbox"/>	PB	20
<input type="checkbox"/>	CE	20
<input type="checkbox"/>	AC	20
<input type="checkbox"/>	BA	19
<input type="checkbox"/>	RO	19
<input type="checkbox"/>	PI	19
<input type="checkbox"/>	RN	19
<input type="checkbox"/>	PE	18
<input type="checkbox"/>	MT	17
<input type="checkbox"/>	TO	17
<input type="checkbox"/>	MS	15
<input type="checkbox"/>	GO	15
<input type="checkbox"/>	RS	15
<input type="checkbox"/>	ES	15
<input type="checkbox"/>	RJ	15
<input type="checkbox"/>	SC	14
<input type="checkbox"/>	DF	12
<input type="checkbox"/>	PR	11
<input type="checkbox"/>	MG	11
<input type="checkbox"/>	SP	8

Novamente é possível observar uma diferença entre os Estados do Sul e Sudeste em relação aos Estados do Norte e Nordeste. São Paulo, Minas Gerais, Paraná, DF, Santa Catarina, Rio de Janeiro, Espírito Santo e Rio Grande do Sul possuem os menores tempos para entrega em média. Enquanto os Estados nortistas e nordestinos possuem os maiores tempos para entrega. O que faz sentido de acordo com o custo de frete demonstrado.

Conclusão

É nítido o maior custo logístico para as regiões Norte e Nordeste do país, o que se reflete nos maiores tempos para entrega e certamente nas receitas menores. Pela perspectiva da empresa, seria interessante pensar maneiras de reduzir esses custos e tempo para entrega a fim de aumentar a sua receita e expandir seus negócios para além do eixo Sul-Sudeste.

Autoavaliação

As três questões traçadas como objetivo neste trabalho foram respondidas com sucesso e foi possível avaliar um aspecto do negócio interessante e similar à realidade do país. Certamente o conjunto de dados limpo e sem grandes problemas facilitou a tarefa.

Próximos Passos

Como próximos passos, é indicado aprofundar as análises iniciadas nos seguintes pontos:

- Abertura por categorias de produtos - alguns produtos possuem maiores dificuldades logísticas que outros?
- Atrasos por Estado - como as dificuldades logísticas se traduzem em atrasos?
- Relação de atrasos com review score - como as dificuldades logísticas se traduzem nas notas das avaliações dos clientes?
- Abertura por Cidade e CEP - zonas mais periféricas dentro da mesma cidade tendem a apresentar maiores dificuldades logísticas