

1. Busca pelos dados

Plataforma para a coleta do dataset:

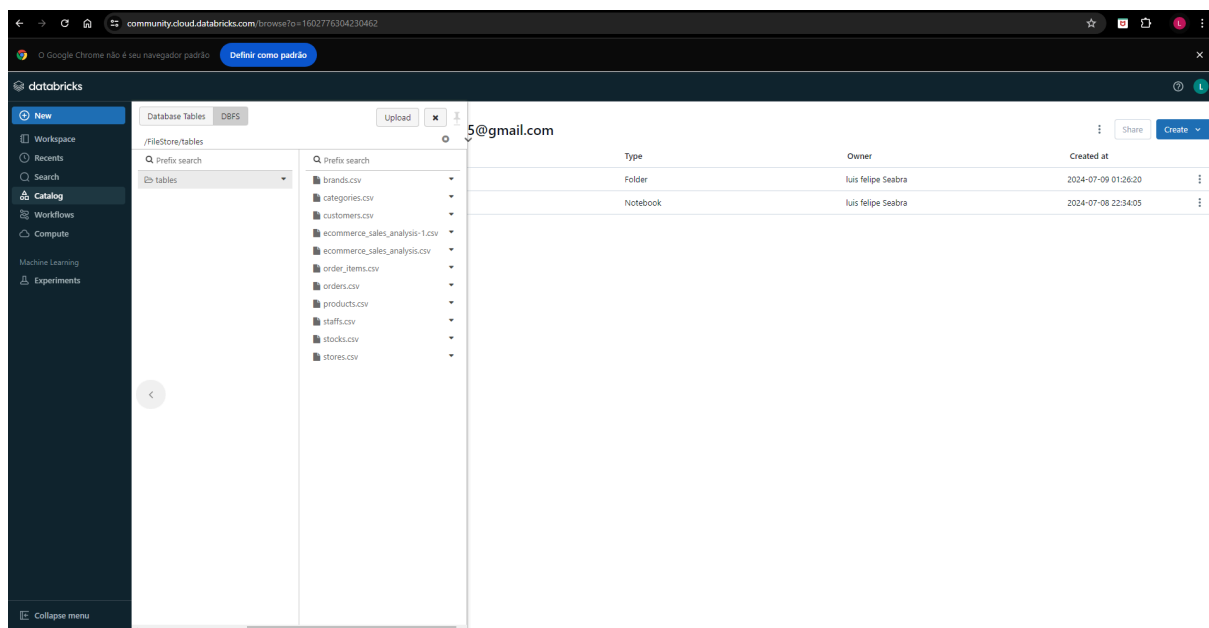
- Kaggle (<https://www.kaggle.com/datasets>)

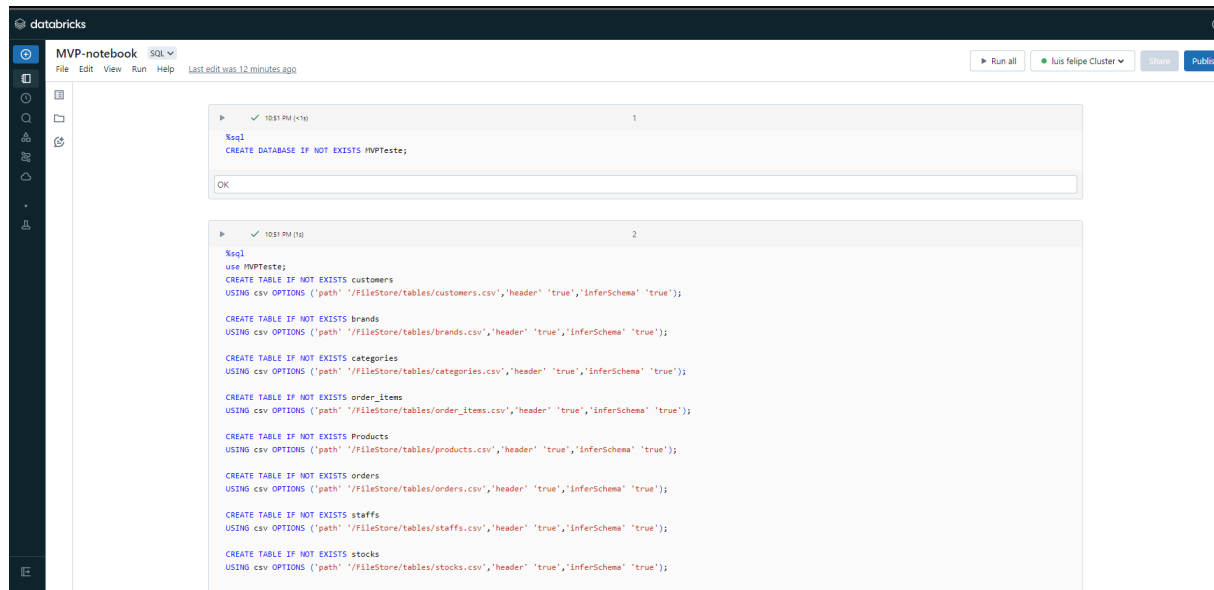
2. Coleta

Base de dados para utilizar no MVP:

Uma vez definido o conjunto de dados, devemos coletar e armazená-los na nuvem.

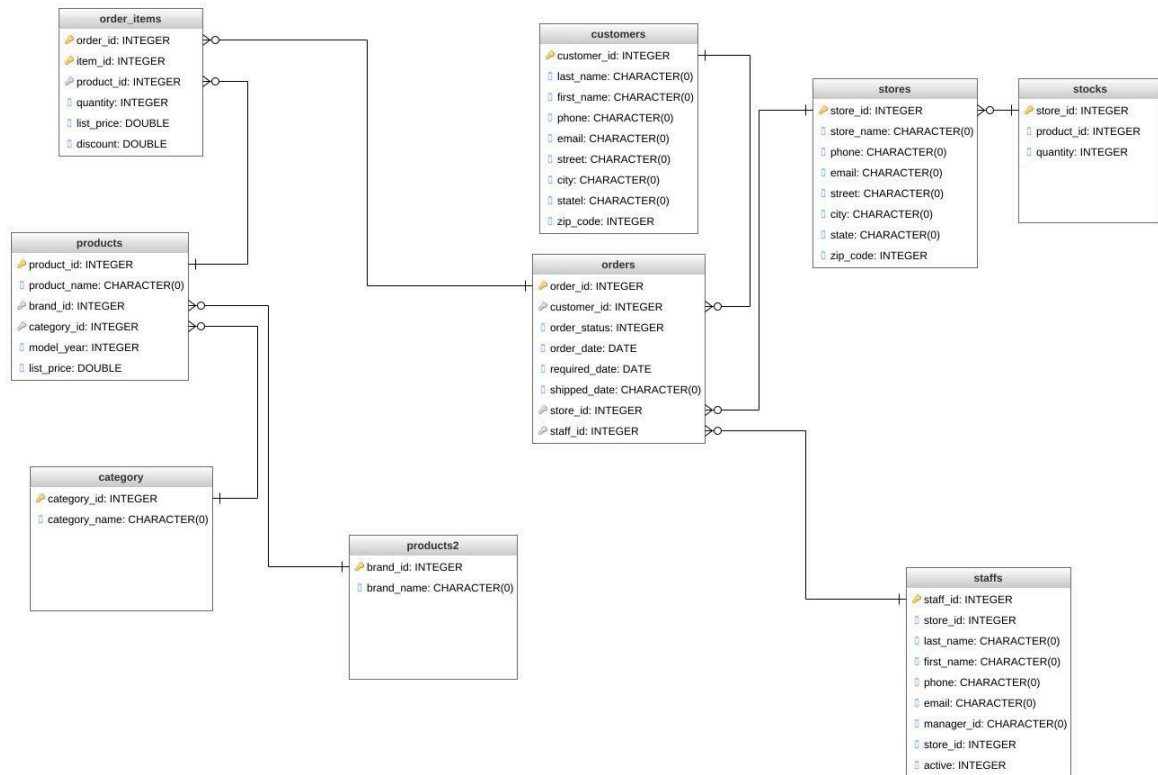
<https://www.kaggle.com/datasets/dillonmyrick/bike-store-sample-database>





3. Modelagem

A modelagem Snowflake é uma técnica de design de banco de dados que normaliza dados de dimensão em várias tabelas menores, criando uma estrutura mais organizada e sem redundâncias. Embora essa abordagem possa trazer benefícios em termos de integridade e economia de espaço, ela também pode introduzir complexidade e impactar o desempenho das consultas. Portanto, a escolha entre um esquema Snowflake e um esquema estrela deve ser baseada nos requisitos específicos do projeto e nas prioridades de desempenho e manutenção.



4. Carga

```

%sql
use MVPTeste;
CREATE TABLE IF NOT EXISTS customers
USING csv OPTIONS ('path' '/FileStore/tables/customers.csv','header' 'true','inferSchema' 'true');

CREATE TABLE IF NOT EXISTS brands
USING csv OPTIONS ('path' '/FileStore/tables/brands.csv','header' 'true','inferSchema' 'true');

CREATE TABLE IF NOT EXISTS categories
USING csv OPTIONS ('path' '/FileStore/tables/categories.csv','header' 'true','inferSchema' 'true');

CREATE TABLE IF NOT EXISTS order_items
USING csv OPTIONS ('path' '/FileStore/tables/order_items.csv','header' 'true','inferSchema' 'true');

CREATE TABLE IF NOT EXISTS Products
USING csv OPTIONS ('path' '/FileStore/tables/products.csv','header' 'true','inferSchema' 'true');

CREATE TABLE IF NOT EXISTS orders
USING csv OPTIONS ('path' '/FileStore/tables/orders.csv','header' 'true','inferSchema' 'true');
  
```

▶

✓

10:51 PM (3s)

23

SQL

⌵

⋮

```

%sql
-- verificar quais produtos, quem foi o atendente e qual foi a loja houve a compra para o estado do texas
SELECT customers.first_name as nome_cliente, orders.order_id, orders.customer_id, orders.order_status, staffs.first_name as nome_atendente,
stores.store_name as nome_loja, products.product_name
FROM customers
INNER JOIN orders ON customers.customer_id = orders.customer_id
INNER JOIN staffs ON staffs.staff_id = orders.staff_id
INNER JOIN stores ON stores.store_id = orders.store_id
INNER JOIN order_items ON order_items.order_id = orders.order_id
INNER JOIN products ON products.product_id = order_items.product_id
WHERE stores.store_id IN (SELECT store_id FROM stores WHERE stores.state LIKE 'TX');

```

▶ (7) Spark Jobs

Table

⌵

+

🔍

🔍

📄

	A _C nome_cliente	I ₃ order_id	I ₃ customer_id	I ₃ order_status	A _C nome_atendente	A _C nome_loja	A _C product_name
1	Edgar	31	1238	4	Kali	Rowlett Bikes	Trek Conduit+ - 2016
2	Edgar	31	1238	4	Kali	Rowlett Bikes	Surly Straggler 650b - 2016
3	Silas	50	872	4	Kali	Rowlett Bikes	Electra Cruiser 1 (24-Inch) - 2016
4	Silas	50	872	4	Kali	Rowlett Bikes	Electra Townie Original 7D EQ
5	Silas	50	872	4	Kali	Rowlett Bikes	Surly Wednesday Frameset - 2016
6	Lazaro	67	526	4	Kali	Rowlett Bikes	Pure Cycles William 3-Speed - 2016

5. Análise

A análise da qualidade dos dados é um componente crítico de qualquer estratégia de gestão de dados. Ela ajuda a assegurar que as decisões sejam baseadas em informações precisas e confiáveis, melhorando a eficiência operacional, a satisfação do cliente, e a conformidade regulatória, além de reduzir custos e promover a segurança e a privacidade dos dados.

No item a. verificaremos todas as análises que foram efetuadas com o intuito de entender melhor o conjunto de dados.

Passos seguidos:

Entendimento do Conjunto de Dados; Análise de Valores Nulos; Verificação de Valores Duplicados; Consistência de Dados; Análise de Outliers; Verificação de Formatos de Dados

a. Qualidade de dados

Entendimento do Conjunto de Dados

1 hour ago (1s) 14

```
%sql
-- verificação da tabela stores, limitado a 2 registros
select * FROM stores
limit 2;
```

(1) Spark Jobs

Table	store_id	store_name	phone	email	street	city	state	zip_code
1	1	Santa Cruz Bikes	(831) 476-4321	santacruz@bikes.sh...	3700 Portola Drive	Santa Cruz	CA	95060
2	2	Baldwin Bikes	(516) 379-8888	baldwin@bikes.shop	4200 Chestnut La...	Baldwin	NY	11432

2 rows | 1.22 seconds runtime Refreshed 1 hour ago

1 hour ago (<1s) 8 SQL

```
%sql
DESCRIBE Products
```

Table

	col_name	data_type	comment
1	product_id	int	null
2	product_name	string	null
3	brand_id	int	null
4	category_id	int	null
5	model_year	int	null
6	list_price	double	null

6 rows | 0.12 seconds runtime Refreshed 1 hour ago

Análise de Valores Nulos

Verifique a presença de valores nulos em cada coluna. Valores nulos podem indicar dados ausentes ou incompletos.

1 minute ago (1s) 13 SQL

```
%sql
-- Análise de dados: Análise de Valores Nulos
SELECT first_name, COUNT(*) AS total,
SUM(CASE WHEN first_name IS NULL THEN 1 ELSE 0 END) AS null_values
FROM customers
GROUP BY first_name;
```

(2) Spark Jobs

	first_name	total	null_values
1	Merlene	1	0
2	Kiana	1	0
3	Leola	1	0
4	Eliz	2	0
5	Laurence	1	0
6	Julianne	1	0
7	Angelina	1	0
8	Myesha	1	0
9	Rod	1	0
10	Aubrey	1	0

Verificação de Valores Duplicados

Valores duplicados podem ser problemáticos em certos contextos. Verifique se há registros duplicados que não deveriam existir.

Just now (1s)

14

SQL

```
%sql
-- Análise de dados: Verificação de Valores Duplicados
SELECT first_name, COUNT(*) AS total,
       COUNT(DISTINCT first_name) AS unique_values
FROM customers
GROUP BY first_name;
```

(2) Spark Jobs

Table

+

Q

F

	first_name	total	unique_values
130	Shirely	2	1
131	Gabriel	2	1
132	Ira	2	1
133	Reatha	1	1
134	Joe	1	1
135	Florrie	1	1
136	Petronila	2	1
137	Cindi	2	1
138	Basilila	1	1
139	Arielle	1	1
140	Carson	1	1

Consistência de Dados

Verifique a consistência dos dados, especialmente em colunas que deveriam ter valores consistentes.

Just now (1s)

15

SQL

```
%sql
-- Análise de dados: Consistência de Dados
SELECT DISTINCT(order_status)
FROM orders
WHERE order_status IS NOT NULL;
```

(2) Spark Jobs

Table

+

Q

F

	order_status
1	1
2	3
3	4
4	2

4 rows

0.92 seconds runtime

Refreshed now

b. Solução do problema

Após análise dos dados, foram respondidas várias perguntas, como:

1. Selecionar produtos comprados por determinado cliente:

2 hours ago (1s) 17

```
%sql
-- Selecionar pedidos de um determinado cliente
select * from orders where customer_id = 2
```

(1) Spark Jobs

	order_id	customer_id	order_status	order_date	required_date	shipped_date	store_id	staff_id
1	692	2	3	2017-02-05	2017-02-05	NULL	1	
2	1084	2	4	2017-08-21	2017-08-24	2017-08-23	1	
3	1509	2	1	2018-04-09	2018-04-09	NULL	1	

3 rows | 1.42 seconds runtime Refreshed 1 hour ago

2. Verificar quais produtos foram vendidos, quem foi o atendente, qual foi a loja onde o produto foi vendido para determinado cliente.

2 hours ago (6s) 18

```
%sql
-- verificar quais produtos, quem foi o atendente e qual foi a loja
SELECT customers.first_name as nome_cliente, orders.order_id, orders.customer_id, orders.order_status, staffs.first_name as nome_atendente,
stores.store_name as nome_loja, products.product_name
FROM customers
INNER JOIN orders ON customers.customer_id = orders.customer_id
INNER JOIN staffs ON staffs.staff_id = orders.staff_id
INNER JOIN stores ON stores.store_id = orders.store_id
INNER JOIN order_items ON order_items.order_id = orders.order_id
INNER JOIN products ON products.product_id = order_items.product_id
WHERE customers.customer_id = 259;
```

(6) Spark Jobs

	nome_cliente	order_id	customer_id	order_status	nome_atendente	nome_loja	product_name
1	Johnathan	1	259	4	Mireya	Santa Cruz Bikes	Trek Fuel EX 8 29 - 2016
2	Johnathan	1	259	4	Mireya	Santa Cruz Bikes	Electra Townie Original 7D EQ
3	Johnathan	1	259	4	Mireya	Santa Cruz Bikes	Surly Straggler - 2016
4	Johnathan	1	259	4	Mireya	Santa Cruz Bikes	Trek Remedy 29 Carbon Fram

3. Verificar qual foi o top e clientes que mais fizeram pedidos

2 hours ago (4s) 19 SQL

```
%sql
-- verificar top 3 clientes que mais fizeram pedidos
SELECT count(orders.order_id) as quantidade_pedidos, customers.first_name, customers.customer_id
FROM customers
INNER JOIN orders ON customers.customer_id = orders.customer_id
GROUP BY customers.customer_id, customers.first_name
ORDER BY COUNT(orders.order_id) DESC
LIMIT 3;
```

(3) Spark Jobs

	quantidade_pedidos	first_name	customer_id
1	3	Aleta	20
2	3	Charolette	5
3	2	Kasha	2

4. Verificar quais funcionários efetuaram mais vendas

2 hours ago (2s) 20

```
%sql
-- verificar quais atendentes mais efetuaram pedidos de vendas
SELECT count(orders.order_id) as quantidade_pedidos, staffs.first_name, staffs.staff_id
FROM staffs
INNER JOIN orders ON staffs.staff_id = orders.staff_id
GROUP BY staffs.staff_id, staffs.first_name
ORDER BY COUNT(orders.order_id) DESC
LIMIT 3;
```

(3) Spark Jobs

	quantidade_pedidos	first_name	staff_id
1	553	Marcelene	6
2	540	Venita	7
3	184	Genna	3

5. Verificar quais marcas possuem mais produtos cadastrados:

2 hours ago (3s) 21

```
%sql
-- Marcas com mais produtos cadastrados
SELECT count(products.brand_id), brands.brand_name
FROM products
INNER JOIN brands ON brands.brand_id = products.brand_id
GROUP BY brands.brand_name
ORDER BY COUNT(products.product_id) DESC
LIMIT 3;
```

(3) Spark Jobs

Table	+	Q	F	□
	count(brand_id)	brand_name		
1	135	Trek		
2	118	Electra		
3	25	Surly		

6. Verificar quais cidades precisamos expandir(loja). Retorno de cidades onde foram feitos menos de 4 pedidos.

2 hours ago (1s) 24 SQL [] :

```
%sql
-- Quantos Clientes Por Cidade com menos de 4 clientes (Quais cidades precisamos expandir)
SELECT COUNT(customer_id) AS Quantidade, city
FROM customers
GROUP BY city
HAVING COUNT(customer_id) < 4
ORDER BY Quantidade DESC;
```

(2) Spark Jobs

Table	+	Q	F	□
	Quantidade	city		
1	3	Holbrook		
2	3	Los Angeles		
3	3	Oakland Gardens		

7. verificar quais produtos, quem foi o atendente e qual foi a loja que houve a compra no estado do texas

```
-- verificar quais produtos, quem foi o atendente e qual foi a loja houve a compra para o estado do texas
SELECT customers.first_name as nome_cliente, orders.order_id, orders.customer_id, orders.order_status, staffs.
first_name as nome_atendente, stores.store_name as nome_loja, products.product_name
FROM customers
INNER JOIN orders ON customers.customer_id = orders.customer_id
INNER JOIN staffs ON staffs.staff_id = orders.staff_id
INNER JOIN stores ON stores.store_id = orders.store_id
INNER JOIN order_items ON order_items.order_id = orders.order_id
INNER JOIN products ON products.product_id = order_items.product_id
WHERE stores.store_id IN (SELECT store_id FROM stores WHERE stores.state LIKE 'TX');
```

(7) Spark Jobs

Table



	A _C nome_cliente	1 ₃ order_id	1 ₃ customer_id	1 ₃ order_status	A _C nome_atendente	A _C nome_loja
1	Edgar	31	1238	4	Kali	Rowlett Bikes
2	Edgar	31	1238	4	Kali	Rowlett Bikes