



Sprint III – Engenharia de dados

Aluno : Arthur Ferreira de F. Lopes

Pós-graduação em ciência de dados e Analytics

Professores : Vitor Almeida e Silvo Alonso

Introdução

O número de restaurantes em Nova York está aumentando dia a dia. Muitos estudantes e profissionais ocupados confiam nesses restaurantes devido ao seu estilo de vida agitado. O serviço de entrega de comida online é uma ótima opção para eles. Fornece-lhes boa comida nos seus restaurantes preferidos. Uma empresa agregadora de alimentos, FoodHub, oferece acesso a vários restaurantes por meio de um único aplicativo de smartphone.

O aplicativo permite que os restaurantes recebam um pedido online direto de um cliente. O aplicativo designa um entregador da empresa para retirar o pedido após a confirmação do restaurante. O entregador então usa o mapa para chegar ao restaurante e aguarda o pacote de comida. Assim que a embalagem de comida é entregue ao entregador, ele confirma a retirada no aplicativo e se desloca até o local do cliente para entregar a comida. O entregador confirma a entrega no aplicativo após entregar a embalagem de comida ao cliente. O cliente pode avaliar o pedido no aplicativo. O agregador de alimentos ganha dinheiro arrecadando uma margem fixa do pedido de entrega dos restaurantes.

Coleta

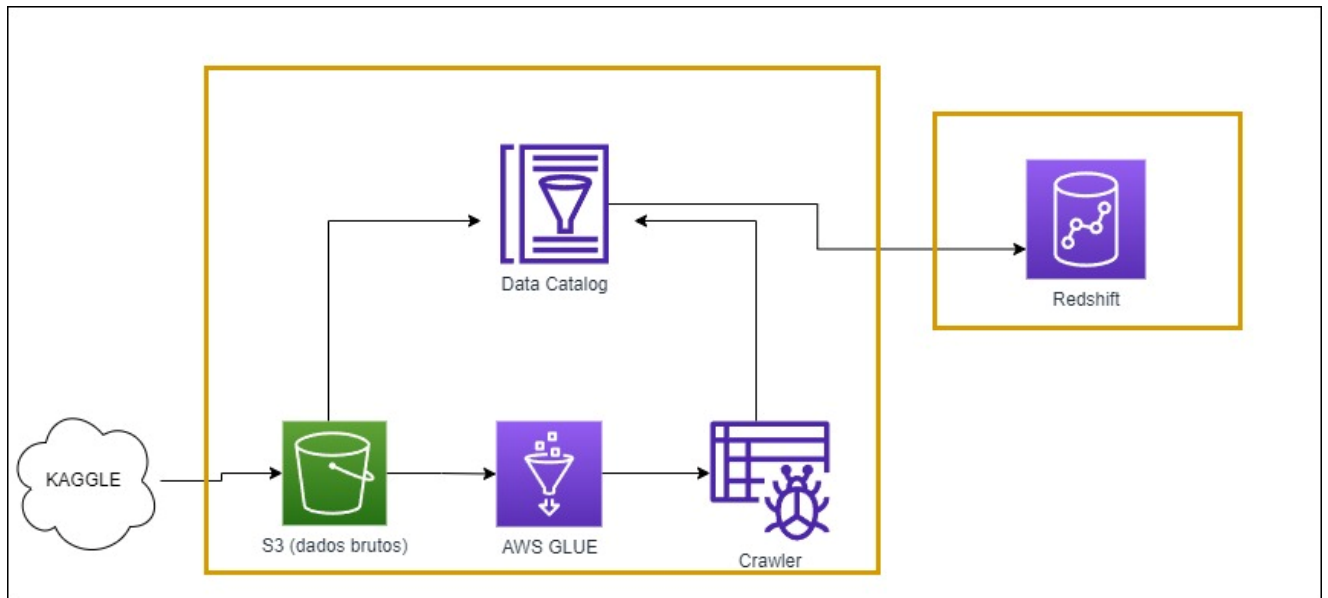
A coleta de dados foi realizada através do repositório de dados Kaggle. Por onde pude ter informações detalhadas do dataset que realizei o downloads.

<https://www.kaggle.com/datasets/ahsan81/food-ordering-and-delivery-app-dataset>

O dataset foi baixado em apenas um arquivo no formato CSV, onde é um formato que facilita o upload do mesmo na cloud AWS, a plataforma de escolhi para fazer o MVP

Modelagem

Um "pipeline de dados" refere-se a um conjunto de etapas sequenciais ou processos que são usados para coletar, processar, transformar e mover dados de uma fonte para um destino específico. Os pipelines de dados são comumente usados em engenharia de dados, análise de dados e em sistemas de processamento de dados em geral. Eles são projetados para automatizar e facilitar o fluxo de dados através de diferentes estágios. Abaixo, está representado o processo desde a coleta de dados até a utilização dos mesmos para responder alguma pergunta.



Catálogo de dados.

Um catálogo de dados é como uma lista organizada de informações sobre todos os dados que temos. É como um catálogo de uma biblioteca, mas em vez de livros, ele lista todos os nossos dados. Com um catálogo de dados, podemos encontrar rapidamente informações sobre onde os dados estão, como estão estruturados e quem os usa. Isso nos ajuda a gerenciar nossos dados de forma eficaz, economizando tempo e tornando mais fácil usar essas informações importantes para tomar decisões informadas. Em resumo, um catálogo de dados é como um guia útil para nossos dados, tornando mais simples encontrá-los e usá-los. Abaixo irei descrever os dados do dataset escolhido.

O dataset contém os diferentes dados relacionados a um pedido de comida em um aplicativo.

As descrições das colunas são :

order_id: ID exclusivo do pedido - Tipo : bigint

customer_id: ID do cliente que pediu a comida – Tipo: bigint

nome_restaurante: Nome do restaurante – Tipo : Varchar

cuisine_type: Culinária pedida pelo cliente – Tipo : Varchar

day_of_the_week: Indica se o pedido é feito em dia de semana ou final de semana (o dia da semana é de segunda a sexta e o final de semana é sábado e domingo) obs: Existem pedidos que não foram classificados pelo cliente – Varchar

rating: Classificação dada pelo cliente em 5 (Existem pedidos que o cliente não informou a classificação) – Tipo: Varchar – Valor mínimo 3 – Valor Máximo 5

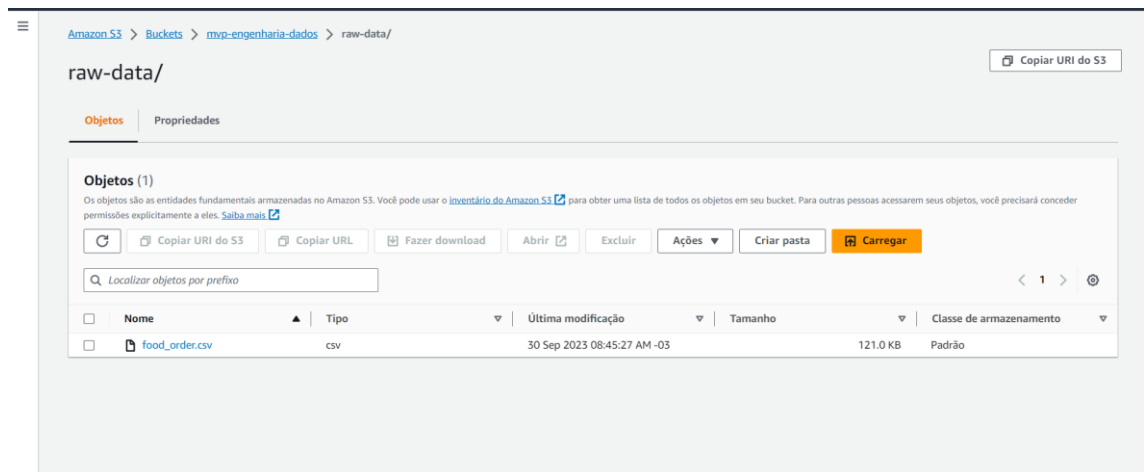
food_preparation_time: Tempo (em minutos) que o restaurante leva para preparar a comida. Isto é calculado tomando a diferença entre os carimbos de data/hora da confirmação do pedido do restaurante e a confirmação de retirada do entregador. – Tipo: bigint - Valor mínimo 20 – Valor máximo – 35

delivery_time: Tempo (em minutos) que o entregador leva para entregar a embalagem de comida. Isso é calculado tomando a diferença entre os carimbos de data/hora da confirmação de coleta do entregador e as informações de entrega – Tipo : bigint – Valor mínimo 15 – Valor máximo 33

Carga dos dados

Como visto acima, os dados foram obtidos através de um repositório na internet e que somente depois do downloads pude iniciar a carga dos dados , abaixo, irei explicar cada passo do pipeline :

- 1- A Ingestão de dados no S3 (bucket) foi feita através do upload no arquivo na plataforma. Realizei a criação da pasta (mvp-engndados) e subir o arquivo CSV na casa raw-data, dessa forma consigo organizar os arquivos brutos, aqueles que ainda não foram processados.



- 2- Para a próxima etapa, foi necessário criar um crawler , que nada mais é do que uma ferramenta do AWS GLUE onde é possível processar e catalogar os dados e os metadados.

The screenshot shows the 'Set crawler properties' page in the AWS Glue console. On the left, a sidebar lists five steps: Step 1 (Set crawler properties), Step 2 (Choose data sources and classifiers), Step 3 (Configure security settings), Step 4 (Set output and scheduling), and Step 5 (Review and create). The main area is titled 'Set crawler properties' and contains a 'Crawler details' section. This section has a 'Name' field with the value 'crawler-food', a 'Description - optional' field with the placeholder 'Enter a description', and a 'Tags - optional' section with the instruction 'Use tags to organize and identify your resources.' At the bottom right, there are 'Cancel' and 'Next' buttons.

- 2.1- Nessa etapa será necessário localizar o arquivo ou pasta que contém os dados e informar a localização

The screenshot shows the 'Add data source' dialog box in the AWS Glue console. The 'Data source' dropdown is set to 'S3'. The 'Network connection - optional' section has a dropdown menu and a 'Clear selection' button. The 'Location of S3 data' section has two radio buttons: 'In this account' (selected) and 'In a different account'. The 'S3 path' section has a text input field containing 'haria-dados/raw-data/food_order.csv', a 'View' button, and a 'Browse S3' button. The 'Subsequent crawler runs' section has three radio buttons: 'Crawl all sub-folders' (selected), 'Crawl new sub-folders only', and 'Crawl based on events'. At the bottom, there are 'Cancel' and 'Add an S3 data source' buttons.

2.2 - – Será necessário configuração algumas permissões de segurança e liberar no IAM para que prosseguíssemos no processo.

The screenshot shows the 'Configure security settings' page in the AWS Glue console. It has three main sections: 1. 'IAM role' with a dropdown menu showing 'AWSGlueServiceRole-glue', buttons for 'Create new IAM role', 'Update chosen IAM role', and a 'View' link. 2. 'Lake Formation configuration - optional' with a checkbox 'Use Lake Formation credentials for crawling S3 data source' and explanatory text. 3. 'Security configuration - optional' with a checkbox 'Enable at-rest encryption with a security configuration'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

2.2- Teremos também que criar ou apenas informar o local de saída dos dados processados. No meu caso criei uma pasta com o nome “MVP-database”

The screenshot shows the 'Set output and scheduling' page in the AWS Glue console. It includes a sidebar with steps: Step 1 (Set crawler properties), Step 2 (Choose data sources and classifiers), Step 3 (Configure security settings), Step 4 (Set output and scheduling), and Step 5 (Review and create). The main content area has two sections: 1. 'Output configuration' with a 'Target database' dropdown set to 'mvp-database', a 'Table name prefix' field, and a 'Maximum table threshold' field. 2. 'Crawler schedule' with a 'Frequency' dropdown set to 'On demand'. At the bottom right are 'Cancel', 'Previous', and 'Next' buttons.

2.3- Após o crawler criado será necessário executá-lo para que ele possa rodar e fazer o catálogo dos dados.

AWS Glue > Crawlers

Crawlers
A crawler connects to a data store, progresses through a prioritized list of classifiers to determine the schema for your data, and then creates metadata tables in your data catalog.

Crawlers (2) [Info](#)
View and manage all available crawlers.

Filter crawlers

<input type="checkbox"/>	Name	State	Schedule	Last run	Last run timestamp	Log	Table changes from last run
<input type="checkbox"/>	carwler-food	Ready		Succeeded	September 30, 2023 at 13:...	View log	1 created
<input type="checkbox"/>	crawler-food	Ready		-	-	-	-

[Create crawler](#)

3 – Após os dados serem catalogados, ele ficam armazenados no “data catalog”, onde é possível obter informações da tabela processada, como mostrado a imagem abaixo. É possível ter informações do metadados.

AWS Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)

▼ **Data Catalog**
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings

▼ **Data Integration and ETL**
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Interactive Sessions
Data classification tools

Location: [REDACTED]
Connection: -
Deprecated: -
Last updated: September 30, 2023 at 13:17:39

Input format: org.apache.hadoop.mapred.TextInputFormat
Output format: org.apache.hadoop.hive.q1.io.HiveIgnoreKeyTextOutputFormat
Serde serialization lib: org.apache.hadoop.hive.serde2.LazySimpleSerDe

Schema | Partitions | Indexes

Schema (9)
View and manage the table schema.

Filter schemas

#	Column name	Data type	Partition key	Comment
1	order_id	bigint	-	-
2	customer_id	bigint	-	-
3	restaurant_name	string	-	-
4	cuisine_type	string	-	-
5	cost_of_the_order	double	-	-
6	day_of_the_week	string	-	-
7	rating	string	-	-
8	food_preparation_time	bigint	-	-
9	delivery_time	bigint	-	-

[Edit schema as JSON](#)
[Edit schema](#)

4 – Agora com os dados já processados será necessário criar uma conexão para podermos conectar no redshift através do Job e assim injetar nossos dados dentro do banco. Obs: É sempre uma boa prática testar o conector criado antes de executar o job que irá ser utilizado.

AWS Glue

Getting started
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Data Catalog tables
Data connections
Workflows (orchestration)

▼ **Data Catalog**
Databases
Tables
Stream schema registries
Schemas
Connections
Crawlers
Classifiers
Catalog settings

▼ **Data Integration and ETL**
ETL jobs
Visual ETL
Notebooks
Job run monitoring
Interactive Sessions
Data classification tools

AWS Glue > Connections

Connectors

Marketplace connectors
Subscribe to connectors from AWS partners to expand your data sources. [Go to AWS Marketplace](#)

Custom connectors
Provide your own connector to expand your data sources. [Create custom connector](#)

Test Connection

Successfully connected to the data store with connection connection-redshift. [View log](#)

Cancel

Connections (1) [Info](#)
View and manage your connections from a connector to a job.

Filter connections by provider

Name	Type	Last modified
connection-redshift	JDBC	Sep 25, 2023

[Create connection](#)
[Create job](#)

5 – Agora que já subimos o arquivo no S3, já criamos o crawler para catalogar os dados, conseguimos visualizar os dados catalogados e a conexão com o conector foi validada, agora é o momento de criar o Job e subir os dados do S3 para o redshift.

5.1- Para iniciar a criação do Job, adicionei o S3(bucket) onde estão salvo os dados processados e optei por escolher a opção de catalogo de dados, onde está salva a table processada pelo crawler

The screenshot displays the AWS Glue console interface for a job named 's3-redshift-jobs'. The left sidebar shows the navigation menu with categories like 'Getting started', 'ETL jobs', 'Data Catalog', and 'Data Integration and ETL'. The main panel shows the job configuration with a visual workflow diagram and a properties panel on the right.

Visual Workflow:

- Data source - S3 bucket Amazon S3** (with a green checkmark)
- Transform - Change Schema... Change Schema** (with a green checkmark)
- Data target - Amazon Redshift** (with a green checkmark)

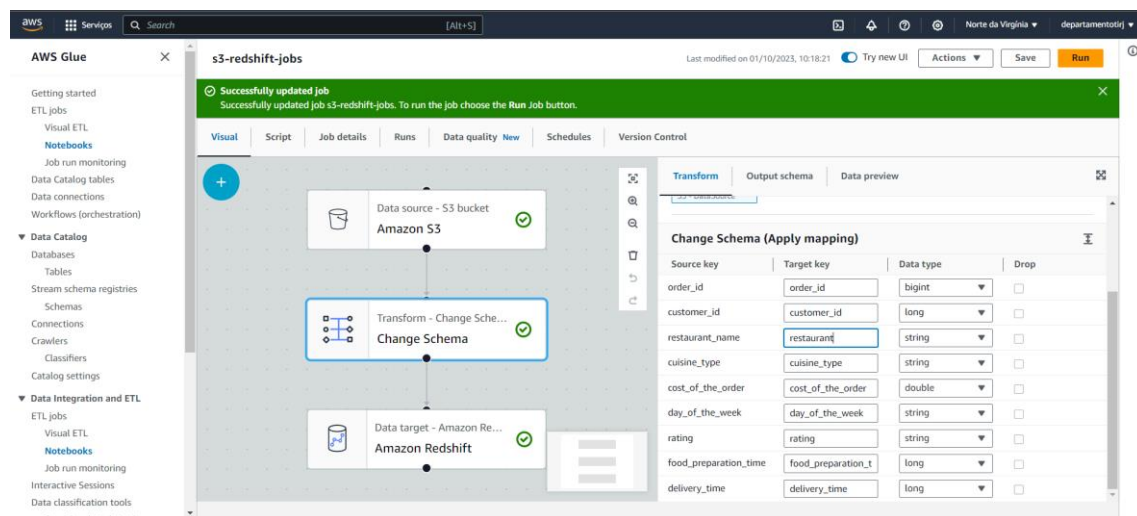
Data source properties - S3:

- Name:** Amazon S3
- S3 source type:** ☒ **Data Catalog table** (selected), ☐ S3 location
- Database:** mvp-database
- Table:** raw_data

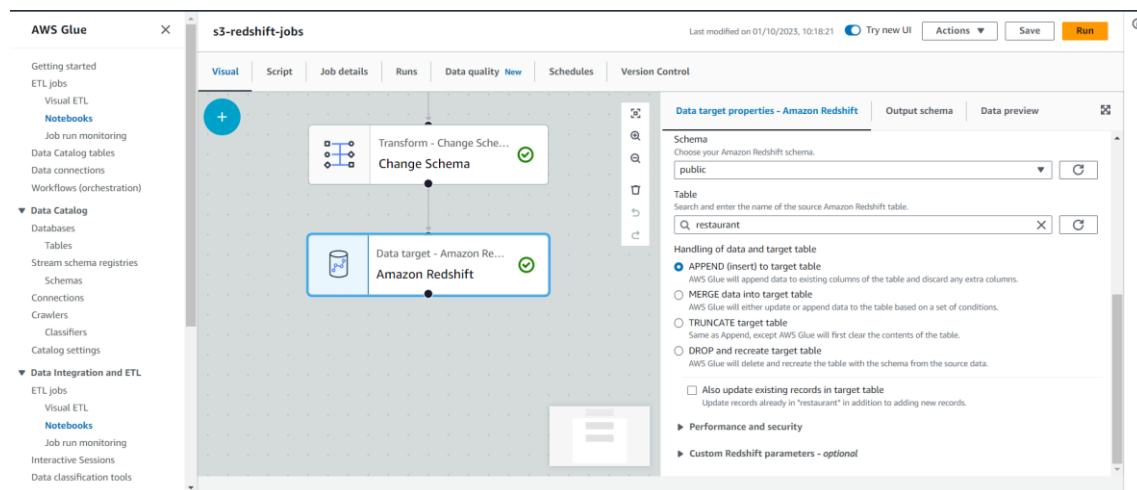
Job details: Successfully updated job s3-redshift-jobs. To run the job choose the Run Job button.

Actions: Try new UI, Actions, Save, Run

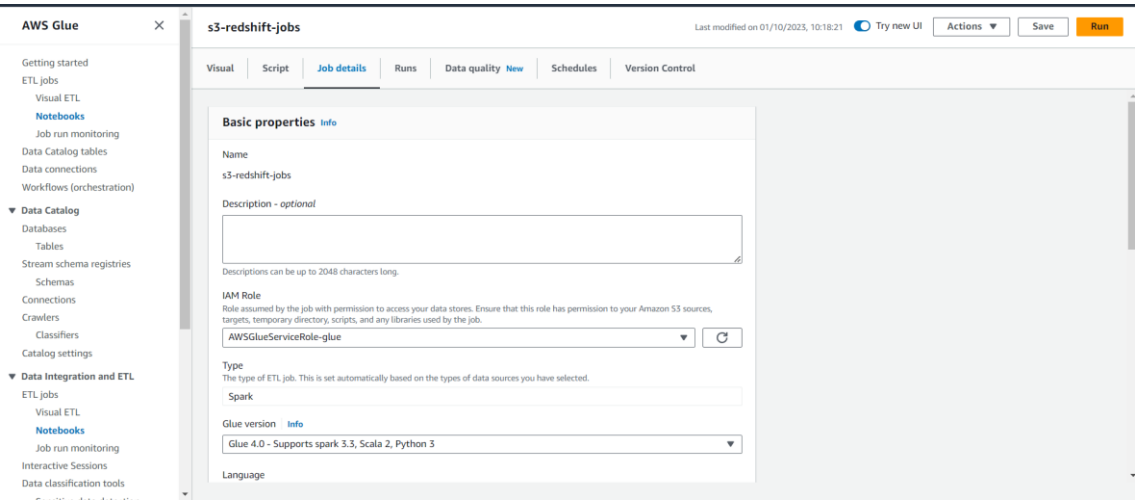
5.2 – Nessa etapa irei aplicar algumas transformações no nome de cada e coluna e também no tipo de dados de cada uma delas.



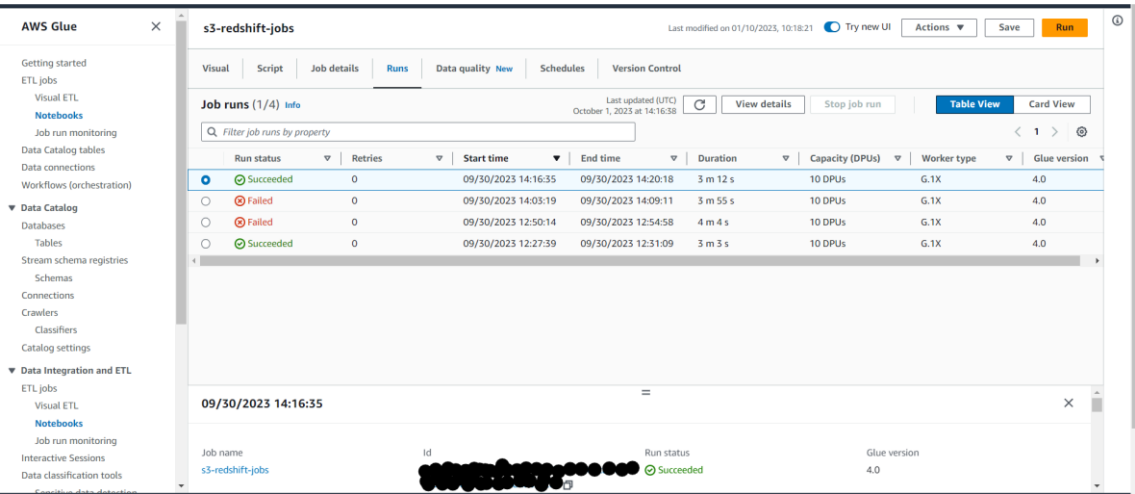
5.3 – Na etapa para salvar a saída dos dados, foi necessário selecionar o conector entre o s3 e o Redshift, que foi criado anteriormente, e também selecionar a tabela dentro do redshift que esse novos dados irão pertencer. No meu caso , criei uma tabela chamada “restaurant” dentro do schema “Public” e optei pela opção do Job dar um insert dos dados na tabela especificada.



5.4 - Para finalizar o Job e coloca-lo em produção, Precisei acessar a aba “Job Details” e seleccionar a IAM ROLE que permite obter esses acessos.



5.5 – Agora basta rodar o Job e aguardar que ele faça a ingestão de dados no redshift.



6 – Após a execução do Jobs, é possível visualizar e realizar consulta dentro do redshift.

The screenshot displays the AWS Redshift Query Editor v2 interface. The left sidebar shows the 'Serverless: default-workg...' workspace with a tree view of resources including 'awsdatacatalog', 'dev', 'public', 'Tables' (1), 'Views' (0), 'Functions' (0), 'Stored proc...' (0), and 'sample_data_dev'. The main editor area contains a SQL query:

```
1 SELECT *
2 FROM "dev"."public"."restaurant";
```

The query has been executed, and the results are displayed in a table with 100 rows. The table has the following columns: order_id, customer_id, rating, cost_of_the_order, delivery_time, restaurant, and cuisine_type. The results show various restaurants and their associated data.

order_id	customer_id	rating	cost_of_the_order	delivery_time	restaurant	cuisine_type
1477607	396995	Not given	9.75	25	Han Dynasty	Chinese
1477607	206039	5	12.66	15	Shake Shack	American
1478263	201535	Not given	9.51	29	Blue Ribbon Fried Chicken	American
1477410	39011	Not given	9.07	24	Mamoun's Falafel	Mediterranean
1478359	361616	3	24.2	32	Blue Ribbon Brooklyn	American
1477249	351561	Not given	20.13	31	The Kati Roll Company	Indian
1476650	88201	5	8.35	23	J. G. Melon	American
1477904	48131	4	16.1	29	RedFarm Hudson	Chinese
1477740	129798	4	25.27	24	TAO	Japanese

The bottom of the interface shows the 'Elapsed time: 38 ms' and 'Total rows: 100'.

Análise de dados

A análise de dados é o processo de examinar informações para encontrar respostas, identificar tendências e tomar decisões mais inteligentes. Ela envolve olhar para números, fatos e figuras para descobrir o que eles nos dizem. A análise de dados é uma ferramenta valiosa em muitos campos, ajudando a resolver problemas, melhorar processos e impulsionar o sucesso. É como usar um superpoder para entender e agir com base nos dados disponíveis. Com isso, desenvolvi algumas query que é possível extrair informações para novas tomadas de decisões como mostrarei abaixo.

1 – Listando todas informações do dataset e entender como ele está sendo distribuído.

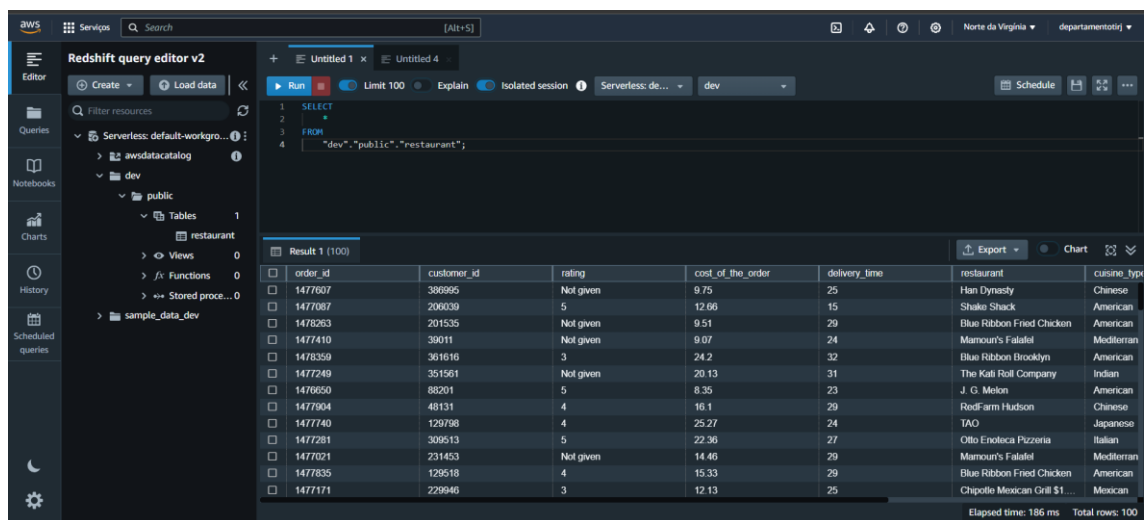
O dataset possui 1898 pedidos.

SELECT

*

FROM

"dev"."public"."restaurant";



The screenshot shows the AWS Redshift Query Editor v2 interface. The SQL query editor contains the following query:

```
1 SELECT
2 *
3 FROM
4 "dev"."public"."restaurant";
```

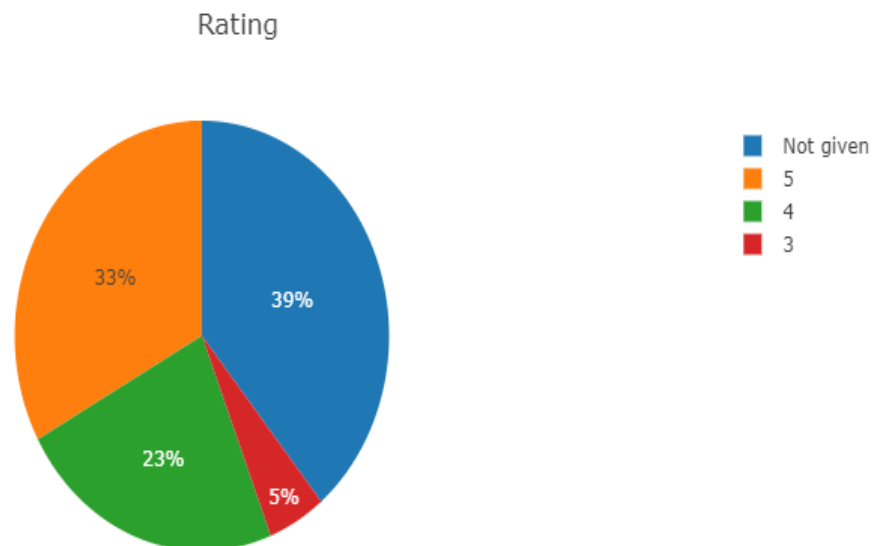
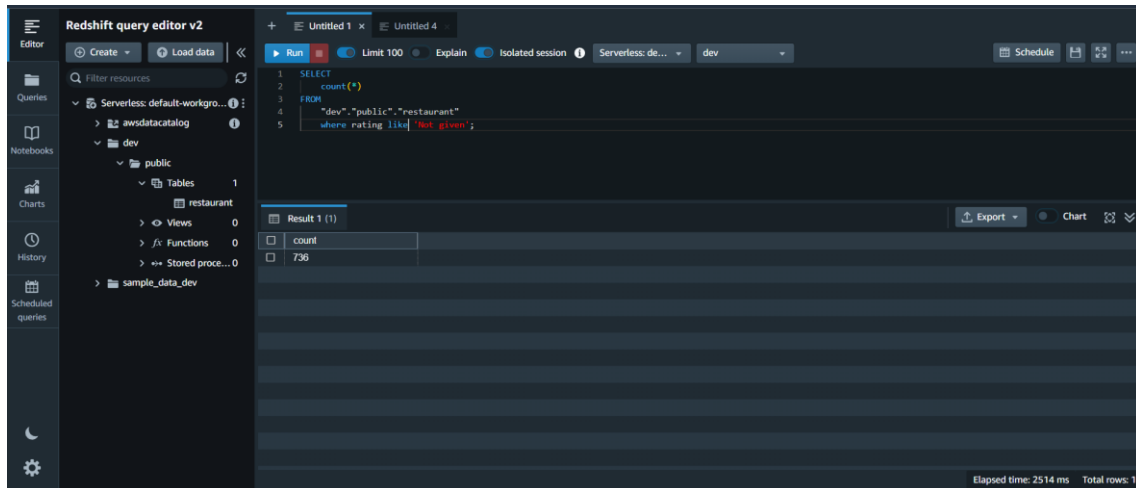
The results are displayed in a table with the following columns: order_id, customer_id, rating, cost_of_the_order, delivery_time, restaurant, and cuisine_type. The table shows 100 rows of data.

order_id	customer_id	rating	cost_of_the_order	delivery_time	restaurant	cuisine_type
1477607	386995	Not given	9.75	25	Han Dynasty	Chinese
1477607	206039	5	12.68	15	Shake Shack	American
1478263	201535	Not given	9.51	29	Blue Ribbon Fried Chicken	American
1477410	39011	Not given	9.07	24	Mamoun's Falafel	Mediterran
1478359	361616	3	24.2	32	Blue Ribbon Brooklyn	American
1477249	351561	Not given	20.13	31	The Kati Roll Company	Indian
1476650	88201	5	8.35	23	J. G. Melon	American
1477904	48131	4	16.1	29	Redf-arm Hudson	Chinese
1477740	129798	4	25.27	24	TAO	Japanese
1477281	309513	5	22.36	27	Otto Enoteca Pizzeria	Italian
1477021	231453	Not given	14.46	29	Mamoun's Falafel	Mediterran
1477835	129518	4	15.33	29	Blue Ribbon Fried Chicken	American
1477171	229946	3	12.13	25	Chipotle Mexican Grill \$1...	Mexican

Elapsed time: 186 ms Total rows: 100

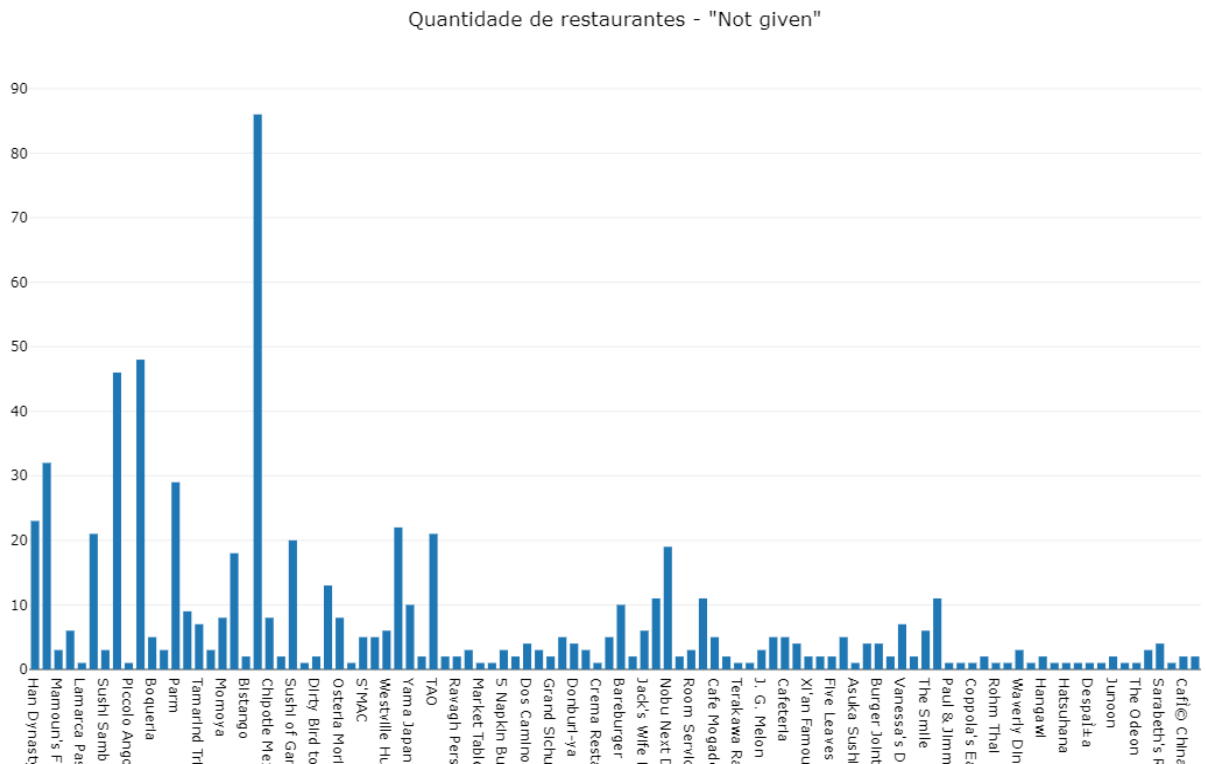
2 – Todos os pedidos possui alguma tipo de classificação entre 3 e 5 e se o usuário não quiser dar alguma classificação, a classificação fica registrado no banco de dados como “Not given”.

Nessa análise, percebi que a proporção de pedidos não classificados “ **Not given**” chegou a ter uma porcentagem expressiva de 39% que corresponde num total de 736 pedidos.



3 - Para entender melhor o motivo dessa alta taxa, listei todos os restaurantes e verifiquei a quantidade de pedidos com a classificação "Not given" de cada um dele e com isso.

O primeiro restaurante da lista é o **Shake Shack** e possui 86 pedidos sem classificação e o restaurante que está na **10º Sushi of Gari 46** possui 20 pedidos nessa classificação. Existe exatamente 134 restaurantes nesta lista.



```
SELECT
    restaurant, count(order_id) as order_id
FROM
    "dev"."public"."restaurant"
where rating like 'Not given'
group by restaurant
order by order_id desc;
```


Redshift query editor v2

Filter resources

Serverless default-workgr...
awsdatacatalog
dev
public
restaurant

restaurant

cost_of_the_order
delivery_time
restaurant
cuisine_type
food_preparation_time
day_of_the_week

SQL Query:

```
1 SELECT
2   restaurant, count(order_id) as order_id
3 FROM
4   "dev"."public"."restaurant"
5 where rating like 'Not given'
6 group by restaurant
7 order by order_id desc;
```

Result 1 (100)

restaurant	order_id
Shake Shack	86
The Meatball Shop	48
Blue Ribbon Sushi	46
Blue Ribbon Fried Chicken	32
Parm	29
Han Dynasty	23
Blue Ribbon Sushi Bar & ...	22
RedFarm Hudson	21
TAO	21
Sushi of Gari 46	20
Nobu Next Door	19
RedFarm Broadway	18
Rubrosa	13

Elapsed time: 2732 ms Total rows: 100

Listando a quantidade de restaurante com possui a classificação “ Not given”

SELECT

count(distinct restaurant)

FROM

"dev"."public"."restaurant"

where rating like 'Not given';

Redshift query editor v2

Filter resources

Serverless default-workgr...
awsdatacatalog
dev
public
restaurant

restaurant

cost_of_the_order
delivery_time
restaurant
cuisine_type
food_preparation_time
day_of_the_week

SQL Query:

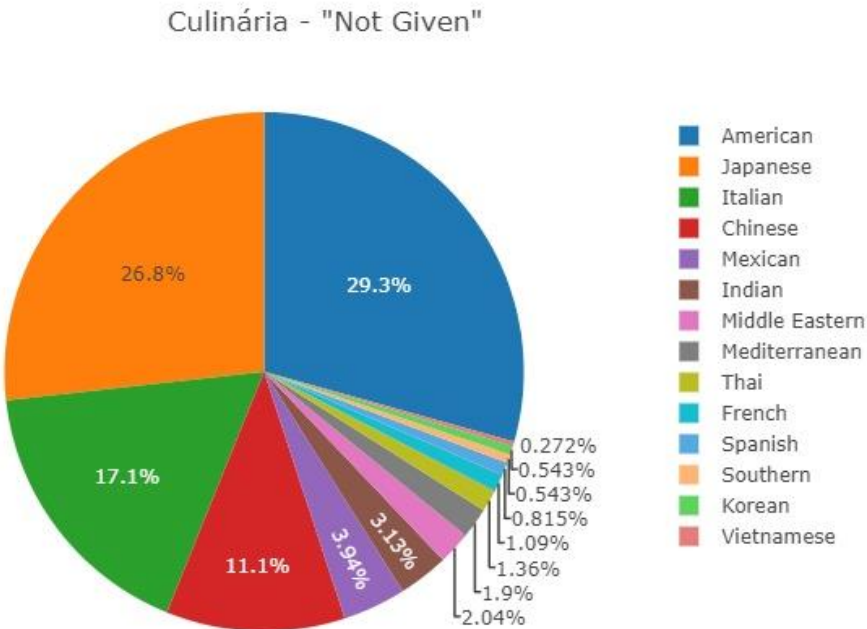
```
1 SELECT
2   count(distinct restaurant)
3 FROM
4   "dev"."public"."restaurant"
5 where rating like 'Not given';
6
```

Result 1 (1)

count
134

Elapsed time: 3281 ms Total rows: 1

4 – Após analisar os restaurantes, agora diminuir o filtro e pesquisei quais são os tipos de culinária que estão com índice ele levado de “Not given”



SELECT

cuisine_type, count(order_id) as order_id

FROM

"dev"."public"."restaurant"

where rating like 'Not given'

group by cuisine_type;

The screenshot displays the AWS Redshift query editor v2 interface. On the left, a sidebar shows the 'restaurant' table schema with fields: order_id (bigint), customer_id (bigint), rating (character varying(65535)), cost_of_the_order (double precision), delivery_time (bigint), restaurant (character varying(65535)), cuisine_type (character varying(65535)), food_preparation_time (bigint), and day_of_the_week (character varying(65535)). The main editor area contains the following SQL query:

```
1 SELECT
2   cuisine_type, count(order_id) as order_id
3 FROM
4   "dev"."public"."restaurant"
5 where rating like 'Not given'
6 group by cuisine_type;
```

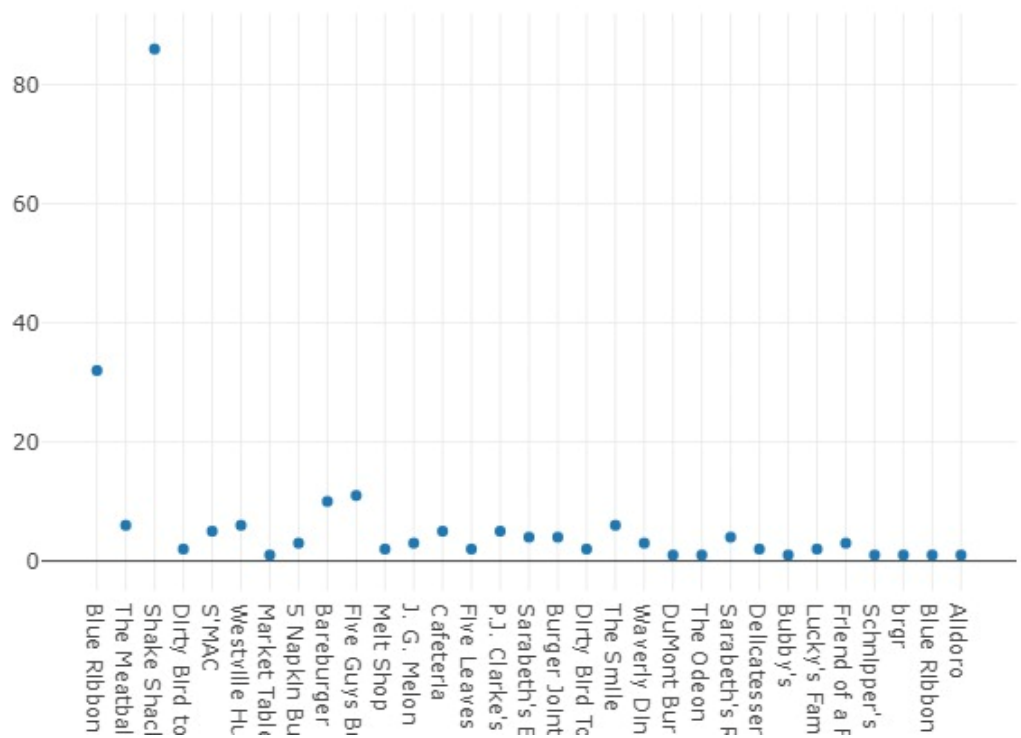
Below the query, the results are displayed as a table with 14 rows. The table has two columns: cuisine_type and order_id. The results are as follows:

cuisine_type	order_id
Chinese	82
American	216
Mediterranean	14
Indian	23
Italian	126
Japanese	197
Spanish	6
Mexican	29
Middle Eastern	15
Southern	4
Thai	10
French	8
Korean	4
Vietnamese	2

The interface also shows a status bar at the bottom right indicating 'Elapsed time: 225 ms' and 'Total rows: 14'.

5 – Continuando com a análise mais a fundo, verifiquei pode existem 31 restaurantes que vendem comida America, em um total de 134 restaurantes e que o restaurante **Shake Shack** possui o maior índice de pedido de culinária américa sem classificação.

Restaurante com culinaria Americana - " Not Given"



Select

distinct restaurant, count(order_id) as order_id

FROM

"dev"."public"."restaurant"

where rating like 'Not given' and cuisine_type like 'American'

group by restaurant ;

The screenshot shows the Redshift query editor interface. On the left, a table named 'restaurant' is displayed with columns: order_id, customer_id, rating, cost_of_the_order, delivery_time, restaurant, cuisine_type, food_preparation_time, and day_of_the_week. The main editor shows a SQL query:

```
1 Select
2 distinct restaurant, count(order_id) as order_id
3 FROM
4 "dev"."public"."restaurant"
5 where rating like 'Not given' and cuisine_type like 'American'
6 group by restaurant ;
```

 The results pane on the right shows 31 rows. The first row is the header: restaurant, order_id. Subsequent rows list restaurants and their corresponding order counts, such as 'Blue Ribbon Fried Chicken' with 32 orders and 'The Meatball Shop' with 6 orders. The bottom status bar indicates 'Elapsed time: 2754 ms' and 'Total rows: 31'.

restaurant	order_id
Blue Ribbon Fried Chicken	32
The Meatball Shop	6
Shake Shack	86
Dirty Bird to Go	2
SMAC	5
Westville Hudson	6
Market Table	1
5 Napkin Burger	3
Bareburger	10
Five Guys Burgers and Fr...	11
Melt Shop	2
J. G. Melon	3
Cafeteria	5
Five Leaves	2
P.J. Clarke's	6

6 – A última análise feita em cima dos dados, me mostrou que o motivo do alto índice de “Not Given “ em cima da Culinária Americana está vindo do Restaurante “ Shake Shack” que em outra analise aparecia de forma significativa. Analise também mostrou que entre 10 pedidos com maior tempo de preparo, 6 foram feito pela Shake Shack

The screenshot shows the Redshift query editor interface. On the left, the same 'restaurant' table is visible. The main editor shows a SQL query:

```
1 Select
2 restaurant, cuisine_type, food_preparation_time, delivery_time, day_of_the_week
3 FROM
4 "dev"."public"."restaurant"
5 where rating like 'Not given' and cuisine_type like 'American'
6 order by food_preparation_time desc
```

 The results pane on the right shows 10 rows. The first row is the header: restaurant, cuisine_type, food_preparation_time, delivery_time, day_of_the_week. Subsequent rows list restaurants and their food preparation times, such as 'Shake Shack' with 35 minutes and 'Blue Ribbon Fried Chicken' with 35 minutes. The bottom status bar indicates 'Elapsed time: 236 ms' and 'Total rows: 10'.

restaurant	cuisine_type	food_preparation_time	delivery_time	day_of_the_week
Shake Shack	American	35	17	Weekend
Blue Ribbon Fried Chicken	American	35	29	Weekend
Shake Shack	American	35	18	Weekend
Shake Shack	American	35	29	Weekend
Shake Shack	American	35	28	Weekend
Shake Shack	American	35	21	Weekend
Shake Shack	American	35	33	Weekday
DuMort Burger	American	35	19	Weekend
5 Napkin Burger	American	35	24	Weekday
Blue Ribbon Fried Chicken	American	35	24	Weekday

Select

restaurant,cuisine_type,food_preparation_time,delivery_time,day_of_the_week

FROM

"dev"."public"."restaurant"

where rating like 'Not given' and cuisine_type like 'American'

order by food_preparation_time desc

limit 10;

Conclusão

Com a análise feita em cima do dataset, objetivo para entender o motivo da alta classificação dos pedidos com “ Not given” se deu pela Culinária americana vir do Restaurante Shake Shack, onde nele o tempo de preparação dos alimentos alcançou o máximo de 35 min, tempo que se reflete na maioria dos pedidos do restaurante.