Modelagem, Extração e Manipulação de Dados

BLOCO: B.I. E ANÁLISE DE DADOS

PROF. RODRIGO EIRAS, M.SC.

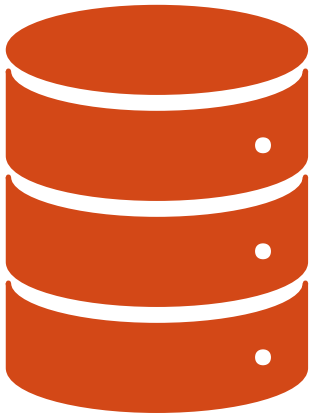
[ETAPA 8] AULAS 1 E 2 – EXTRAIR DADOS DE SISTEMAS DE INFORMAÇÃO UTILIZANDO SQL - PARTE 2

Na última aula...

- Linguagem SQL
  - Seleção
  - Projeção
  - Produto Cartesiano
  - União
  - Diferença entre conjuntos

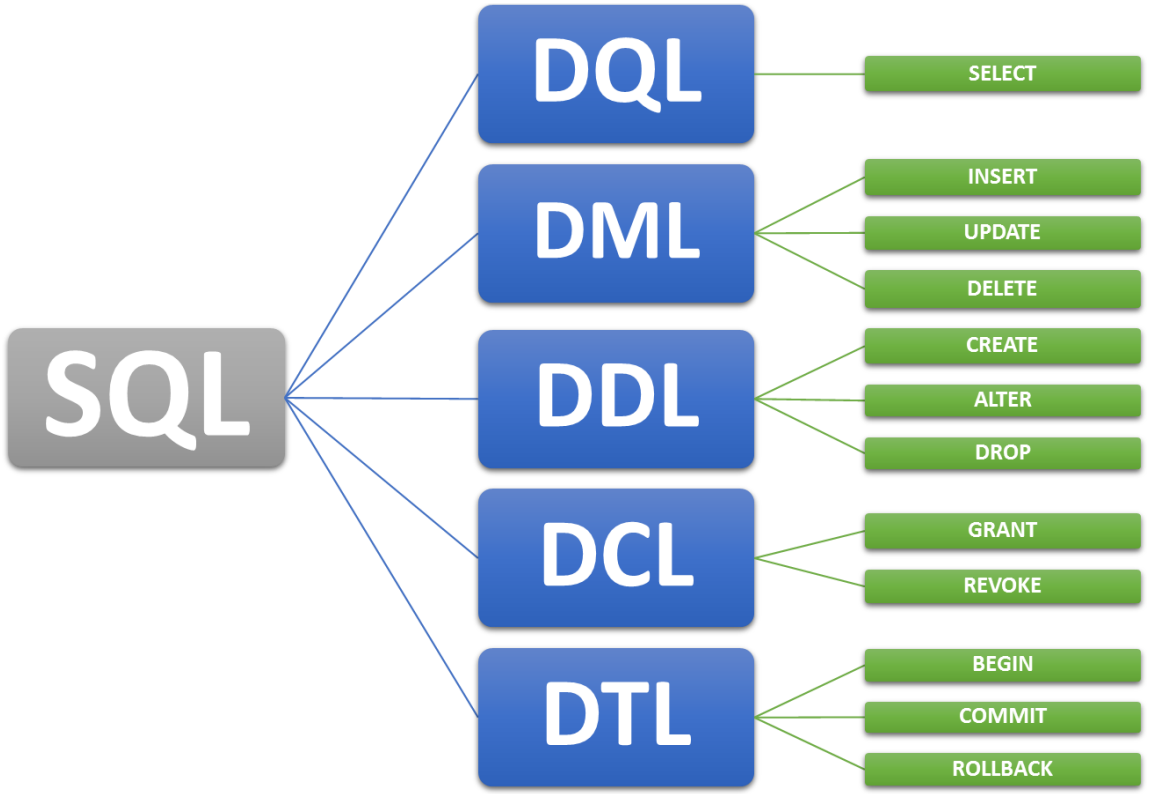
Agenda

- Manipular dados com SQL
- Direct Query vs Importação
- Funções de agregação em SQL



Manipular dados com SQL

Linguagem SQL - Relembrando



Linguagem para:

- Definição de dados: criação das estruturas
  - Data Definition Language (DDL)
- Manipulação de dados: atualização e consultas
  - Data Manipulation Language (DML)
  - Data Query Language (DQL)

Manipular dados com SQL

As instruções para manipulação de dados em SQL são denominadas DML - Data Manipulation Language (Linguagem de Manipulação de Dados) e estão disponíveis para Consulta, Inclusão, Alteração e Exclusão de dados.

Consulta SQL

As instruções de seleção tem uma estrutura básica que é demonstrada na tabela abaixo.

Comandos	Descrição
SELECT	Especifica as colunas que serão exibidas no resultado da consulta
FROM	Especifica as tabelas que contém os dados que serão manipulados na consulta
WHERE	Especifica as condições utilizadas para filtrar os registros da consulta
ORDER BY	Classifica o resultado da consulta baseado em uma ou mais colunas
GROUP BY	Agrupa os registros das consultas com base nos valores de uma ou mais colunas
HAVING	Especifica as condições utilizadas para filtrar consultas agrupadas (Group by)

Consulta SQL

A sintaxe mínima da instrução SELECT é:

SELECT campos FROM tabela.

- Exemplo de comando para consulta em SQL:

```
SELECT * FROM DM_Localidade
```

DM_Localidade
Pais
Estado
Cidade

Consulta SQL

Selecionando Colunas nas Consultas SQL

- Exemplo de comando para consulta em SQL com seleção de colunas:

```
SELECT DM_Localidade.Pais,
DM_Localidade.Estado
FROM DM_Localidade
```

DM_Localidade
Pais
Estado
Cidade

Consulta SQL

Eliminando Duplicação nas Consultas SQL

- Exemplo de comando para consulta em SQL com distinção de registros
- O objetivo da instrução DISTINCT é somente trazer uma repetição da informação.
- No exemplo, serão listados cada um dos Estados de um País, apenas uma única vez, apesar da tabela conter uma repetição dos mesmos visto que a granularidade da tabela é cidade.

```
SELECT DISTINCT DM_Localidade.Estado

FROM DM_Localidade
```

DM_Localidade
Pais
Estado
Cidade

Consulta SQL

Colunas calculadas nas Consultas SQL

- Exemplo de comando para consulta em SQL com colunas calculadas

- O objetivo das colunas calculadas é não duplicar dados nas tabelas. Essa funcionalidade permite termos uma completude de requisitos de dados.

- Todos os dados solicitados pelos usuários estarão disponíveis, mas fisicamente na gravação dos dados não há duplicação.

```
SELECT Estado + " / " + Cidade  "Cidades"

FROM DM_Localidade
```

DM_Localidade
Pais
Estado
Cidade

Consulta SQL

Cláusula FROM

- A cláusula FROM especifica as tabelas que contém os dados que serão manipulados na consulta.

- É possível selecionar dados de uma única tabela, segue um exemplo:

```
SELECT DM_Localidade.Pais,
DM_Localidade.Estado
FROM DM_Localidade
```

DM_Localidade
Pais
Estado
Cidade

```
SELECT Fato_Vendas.Valor_Vendas,
Fato_Vendas.Quantidade,
DM_Localidade.Estado,
DM_Tempo.Ano
FROM Fato_Vendas, DM_Localidade, DM_Tempo
WHERE Fato_Vendas.Pk_Localidade = DM_Localidade.PK_Localidade
AND Fato_Vendas.Pk_Tempo = DM_Tempo.Pk_Tempo;
```

Consulta SQL

Cláusula FROM

- E também é possível selecionar dados de várias tabelas.
- Para cruzar dados de mais de uma tabela é preciso listar todas as tabelas envolvidas na cláusula FROM.
- As formas de junção dessas tabelas foram abordadas no item “Junções” da Etapa 07.
- Segue o exemplo na imagem abaixo:

```
SELECT Fato_Vendas.Valor_Vendas,
Fato_Vendas.Quantidade,
DM_Localidade.Estado,
DM_Tempo.Ano
FROM Fato_Vendas, DM_Localidade, DM_Tempo
WHERE Fato_Vendas.Pk_Localidade = DM_Localidade.PK_Localidade
AND Fato_Vendas.Pk_Tempo = DM_Tempo.Pk_Tempo;
```

Consulta SQL

- Nesse exemplo houve a junção de 3 tabelas: a Fato Vendas e as dimensões Localidade e Tempo.
- Essa junção forma o Fato do Processo de Negócio Vendas conforme vimos na modelagem dimensional nas Etapas 01 e 02.

- = (igual)
- != ou <> (diferente)
- > (maior que)
- < (menor que)
- >= (maior ou igual a)
- <= (menor ou igual a)
- LIKE (similar a)
- BETWEEN (entre X e Y)
- IN (vários valores dentro da lista)
- AND (e)
- OR (ou)
- XOR (mistura do OR com
- NOT (negação)
- IS (valores iguais)

Consulta SQL

Cláusula WHERE

- O comando WHERE é utilizado para filtrarmos os atributos escolhidos.
- O WHERE é opcional, sendo assim você omiti-lo, então você terá todos os registros com os atributos escolhidos.
- As condições mais comuns são:

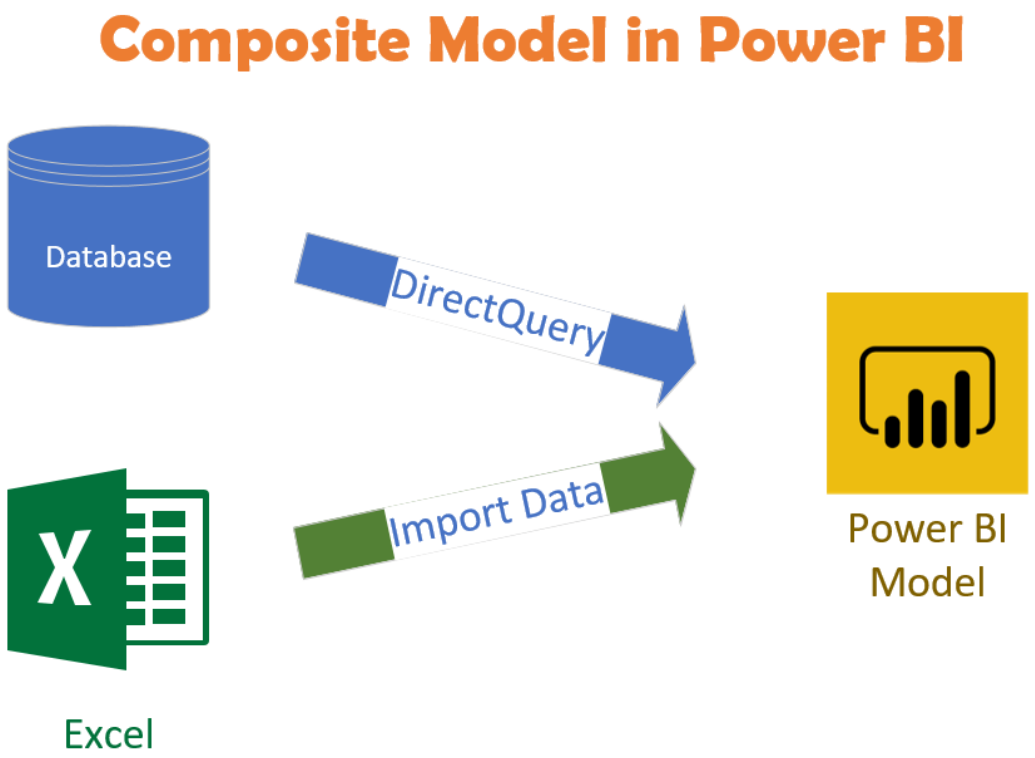
```
SELECT Fato_Vendas.Valor_Vendas,
Fato_Vendas.Quantidade,
DM_Localidade.Estado,
DM_Tempo.Ano
FROM Fato_Vendas, DM_Localidade, DM_Tempo
WHERE Fato_Vendas.Pk_Localidade = DM_Localidade.PK_Localidade
AND Fato_Vendas.Pk_Tempo = DM_Tempo.Pk_Tempo
AND DM_Tempo.Ano = 2017
AND DM_Localidade. Estado = "RJ";
```

Consulta SQL

Nesse exemplo a cláusula WHERE propiciou a forma de junção da Fato Vendas com as dimensões Tempo e Localidade, assim restringiu a consulta para trazer somente os dados de 2017 do Estado do Rio de Janeiro.

Direct Query

FALANDO DE ARQUITETURA DE CONSUMO DE DADOS



Direct Query

- Você pode se conectar a todos os tipos de fontes de dados diferentes ao usar o Power BI Desktop ou o serviço do Power BI e fazer essas conexões de dados de maneiras diferentes.

- Você pode importar dados para o Power BI, que é a maneira mais comum de obter dados, ou conectar-se diretamente aos dados no repositório de origem original, que é conhecido como DirectQuery.

Direct Query - Considerações

- Você deve importar dados para o Power BI sempre que possível.
- A importação aproveita o mecanismo de consulta de alto desempenho do Power BI e proporciona uma experiência altamente interativa e completa.
- Caso as suas metas não possam ser atendidas com a importação de dados, considere o uso do DirectQuery.
- Por exemplo, se os dados estão frequentemente em alteração e os relatórios devem refletir os dados mais recentes, o DirectQuery pode ser a melhor opção.

Direct Query - Considerações

- No entanto, o uso do DirectQuery só é possível quando a fonte de dados subjacente pode fornecer consultas interativas, menos de 5 segundos para a consulta de agregação típica, e pode processar a carga de consulta que será gerada.
  - Por isso somente alguns bancos suportam.
- Além disso, a lista de limitações do uso do DirectQuery deve ser considerada cuidadosamente.
- Está em constante evolução.

Direct Query – Comportamento

Para o DirectQuery, ao usar Obter Dados no Power BI Desktop para se conectar a uma fonte de dados, o comportamento dessa conexão será o seguinte:

- Durante a experiência inicial de Obter Dados, a fonte é selecionada.
- Para fontes relacionais, um conjunto de tabelas é selecionado e cada uma ainda define uma consulta que retorna logicamente um conjunto de dados.
- Para fontes multidimensionais como SAP BW, somente a fonte é selecionada.

Direct Query – Comportamento

- No entanto, após o carregamento, nenhum dado é importado para o repositório do Power BI.
- Em vez disso, após a criação de um visual no Power BI Desktop, as consultas são enviadas à fonte de dados subjacente para recuperar os dados necessários.\
- O tempo necessário para atualizar o visual depende do desempenho da fonte de dados subjacente.

Direct Query – Comportamento

- As alterações nos dados subjacentes não são imediatamente refletidas em nenhum visual existente.
- Ainda é necessário fazer a atualização.
- As consultas necessárias são reenviadas para cada visual e o visual é atualizado conforme necessário.
- Após a publicação do relatório no serviço do Power BI, isso resultará novamente em um conjunto de dados no serviço do Power BI, o mesmo usado para a importação.
- No entanto, nenhum dado será incluído nesse conjunto de dados.

Direct Query – Comportamento

- Ao abrir um relatório existente no serviço do Power BI ou criar um relatório, a fonte de dados subjacente é novamente consultada para recuperar os dados necessários.

- Dependendo da localização da fonte de dados original, poderá ser necessário configurar um gateway de dados local, assim como será necessário para o modo de importação se os dados forem atualizados.

Direct Query – Comportamento

- Páginas inteiras de relatório ou visuais podem ser fixadas como blocos de Dashboard.
- Para garantir que a abertura de um dashboard seja rápida, os blocos são atualizados automaticamente de acordo com um agendamento, por exemplo, a cada hora.
- A frequência dessa atualização pode ser controlada, para refletir a frequência com que os dados são alterados e o nível de importância de ver os dados mais recentes.
- Ao abrir um dashboard, os blocos refletem os dados na hora da última atualização, não necessariamente as últimas alterações feitas na fonte subjacente.
- Você pode atualizar um dashboard aberto para garantir que ele seja atual.

Importação – Comportamento

Para a importação, ao usar Obter Dados no Power BI Desktop para se conectar a uma fonte de dados como o SQL Server, o comportamento dessa conexão será o seguinte:

- Durante a experiência de Obter Dados, o conjunto de tabelas selecionado define cada uma das consultas que retornará um conjunto de dados.
- Essas consultas podem ser editadas antes do carregamento dos dados, por exemplo, para aplicar filtros, agregar os dados ou unir tabelas diferentes.
- Após o carregamento, todos os dados definidos por essas consultas serão importados para o cache do Power BI.

Importação – Comportamento

- Com a criação de um visual no Power BI Desktop, os dados importados serão consultados.
- O repositório do Power BI garante que a consulta será rápida.
- Todas as alterações no visual são refletidas imediatamente.
- As alterações nos dados subjacentes não são refletidas em nenhum visual.
- É necessário atualizar para reimportar os dados.

Importação – Comportamento

- Após a publicação do relatório como um arquivo .pbix no serviço do Power BI, um conjunto de dados é criado e carregado no serviço do Power BI.
- Os dados importados são incluídos no conjunto de dados.
- Em seguida, é possível agendar a atualização desses dados, por exemplo, para reimportar os dados todos os dias.
- Dependendo da localização da fonte de dados original, poderá ser necessário configurar um gateway de dados local.

Importação – Comportamento

- Ao abrir um relatório existente no serviço do Power BI ou criar um relatório, os dados importados são consultados novamente, garantindo a interatividade.
- Páginas inteiras de relatório ou visuais podem ser fixadas como blocos de dashboard.
- Os blocos são atualizados automaticamente sempre que o conjunto de dados subjacente é atualizado.

Direct Query – Quando é útil?

Em resumo, considerando as funcionalidades atuais do DirectQuery no Power BI, ele oferece os benefícios nos seguintes cenários:

- Dados com alterações frequentes e a necessidade de relatórios quase em tempo real.
- Manipulação de dados muito grandes, sem a necessidade de pré-agregação.
- Aplicação de restrições de soberania de dados.
  - Maiores infos: https://docs.microsoft.com/pt-br/power-bi/connect-data/desktop-directquery-about
  - Fontes que suportam DirectQuery: https://docs.microsoft.com/en-us/power-bi/connect-data/power-bi-data-sources

Demo DirectQuery

Exercício 1

Servidor: vmi578219.contaboserver.net (Login e Senha serão passados no chat do zoom.)

Database: imdb

Usando o conector MySQL (preferencialmente) ou MariaDB, responda as questões abaixo:

1) Qual o ano com mais lançamentos de filmes?

2) No ano em que ocorreu mais lançamentos de cinema, qual o gênero de filme que predominou? Quanto esse gênero representa percentualmente em relação aos demais?

3) Mostre os TOP 10 atores e atrizes em popularidade (elimine atores com valores vazios)

4) Qual o filme com maior duração? Quando adicionado popularidade, qual o filme que desponta agora? Qual a duração desse filme?

Funções de Agregação em SQL

O SQL possui algumas funções de agregação de dados para facilitar a manipulação de dados mais complexos e volumosos. São elas:

Agregação	Descrição
AVG	Calcula o valor médio dos dados selecionados
MIN	Calcula o valor mínimo dos dados selecionados
MAX	Calcula o valor máximo dos dados selecionados
SUM	Calcula o somatório dos dados selecionados
COUNT	Calcula a quantidade de dados selecionados

Funções de Agregação em SQL

Agregador AVG em SQL

- O objetivo é retornar a média dos dados da consulta de forma simples.

```
SELECT AVG(Fato_Vendas.Valor_Vendas),
DM_Tempo.Ano
FROM Fato_Vendas, DM_Localidade, DM_Tempo
WHERE Fato_Vendas.Pk_Localidade = DM_Localidade.PK_Localidade
AND Fato_Vendas.Pk_Tempo = DM_Tempo.Pk_Tempo ;
```

Funções de Agregação em SQL

Agregadores MIN e MAX em SQL

- O objetivo é retornar o valor mínimo e o valor máximo dos dados da consulta sem ter que fazer alguma programação para isso.

```
SELECT MIN(Fato_Vendas.Valor_Vendas),
       MAX(Fato_Vendas.Valor_Vendas),

DM_Tempo.Ano

FROM Fato_Vendas, DM_Localidade, DM_Tempo

WHERE Fato_Vendas.Pk_Localidade = DM_Localidade.PK_Localidade

AND Fato_Vendas.Pk_Tempo = DM_Tempo.Pk_Tempo ;
```

Funções de Agregação em SQL

```
SELECT SUM(Fato_Vendas.Valor_Vendas),
DM_Tempo.Ano
FROM Fato_Vendas, DM_Localidade, DM_Tempo
WHERE Fato_Vendas.Pk_Localidade = DM_Localidade.PK_Localidade
AND Fato_Vendas.Pk_Tempo = DM_Tempo.Pk_Tempo ;
```

Agregador SUM em SQL

- O objetivo desse agregador é retornar o somatório dos valores.
- Esse é um dos agrupadores mais utilizados, e é o padrão para as ferramentas de manipulação de dados, exemplo Power BI, para as medidas das consulta de dados.

Funções de Agregação em SQL

Agregador COUNT em SQL

- A contagem, dependendo da coluna selecionada, gera um resultado específico.

- Se nenhuma coluna for selecionada, ou a instrução "*" (todas as colunas) for selecionada, o retorno é o número total de linha da consulta.
  - Sintaxe: SELECT COUNT(*) FROM <tabela>;
- Se uma coluna específica for selecionada o retorno é o número de linhas em que a coluna está preenchida nos dados.
  - Sintaxe: SELECT COUNT(coluna) FROM <tabela>;
- Se foi utilizada a instrução DISTINCT o retorno é o número de valores diferentes que a coluna possui nos dados.

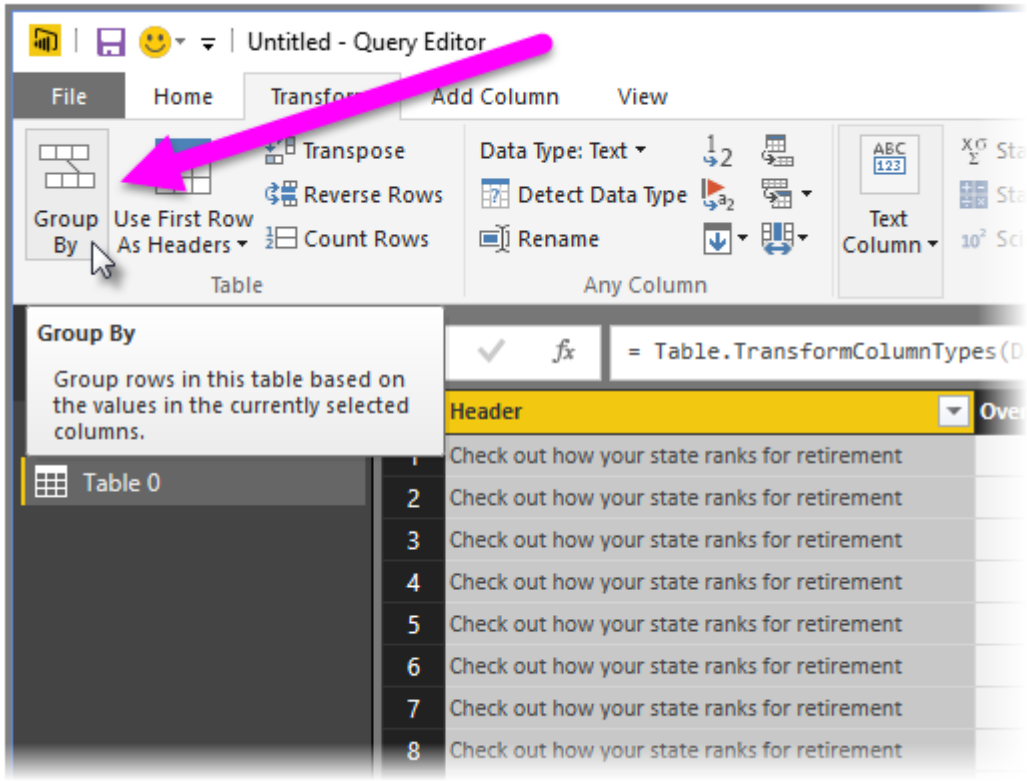
Funções de Agregação em SQL

```
SELECT COUNT(*) Quantidade de Pedidos,
COUNT( DISTINCT DM_Localidade.Estados ) Quantidade Estados,
COUNT(DM_Localidade.Cidades ) Quantidade Cidades Informadas
DM_Tempo.Ano
FROM Fato_Vendas, DM_Localidade, DM_Tempo
WHERE Fato_Vendas.Pk_Localidade = DM_Localidade.PK_Localidade
AND Fato_Vendas.Pk_Tempo = DM_Tempo.Pk_Tempo ;
```

Agregador COUNT em SQL

- COUNT(*) traz a informação da quantidade de linhas da consulta
- COUNT(DISTINCT Estados) traz a quantidade distinta de Estados que realizaram vendas
- COUNT(Estados) traz a quantidade de cidades informadas.

Group By no Power BI



Na próxima aula...

Vamos continuar nosso estudo de manipulação de dados no Power BI.