

O que é SQL?

SQL é a sigla para *Structured Query Language*, ou Linguagem de Consulta Estruturada, em português. É uma linguagem de programação utilizada para trabalhar com banco de dados relacionais.

Um banco de dados relacional armazena os dados em formatos similares a tabelas, com as quais é possível construir relações entre si, facilitando a inserção e a recuperação das informações.

Com o SQL, é possível realizar consultas à base de dados informando diversos tipos de parâmetros (seguindo algumas regras e normas da linguagem), podendo obter grandes quantidades de informação de diferentes fontes (no caso, tabelas) com poucos ou até mesmo apenas uma instrução.

A linguagem também possui outros subconjuntos de operações utilizadas para gerenciar e manipular as informações e o próprio banco de dados, como inserção, atualização e exclusão de informações, criação de novas tabelas, etc

Quais comandos nós veremos?

Comando/Cláusula	Descrição
SELECT *	Seleciona todas as colunas de uma tabela.
SELECT coluna1, coluna2	Selecionar colunas específicas de uma tabela.
LIMIT	Limita o número de resultados retornados.
DISTINCT	Remove linhas duplicadas, mostrando apenas valores únicos.
WHERE	Filtra os resultados com base em uma condição.
OR	Aplica uma condição alternativa no filtro.
AND	Aplica múltiplas condições no filtro.
ORDER BY	Ordena os resultados em ordem crescente (padrão) ou decrescente.
BETWEEN	Filtra resultados entre um intervalo de valores.
NOT BETWEEN	Filtra resultados fora de um intervalo de valores.
IN	Filtra resultados com base em uma lista de valores.
COUNT(*)	Conta todas as linhas de uma tabela ou resultado de uma coluna.
MAX	Retorna o valor máximo de uma coluna.
GROUP BY	Agrupar os resultados por uma ou mais colunas, normalmente usado com funções agregadas.
HAVING	Filtra grupos de resultados após uma agregação.
AS	Renomeia uma coluna ou cria uma coluna temporária com um nome novo.

Base de dados:

Relação Anual de Informações Sociais (RAIS)

O que é? A Relação Anual de Informações Sociais (RAIS) é um relatório de informações socioeconômicas solicitado pela Secretaria de Trabalho do Ministério da Economia brasileiro às pessoas jurídicas e outros empregadores anualmente. Foi instituída pelo Decreto nº 76.900, de 23 de dezembro de 1975.

Variável	Descrição	Tipo	Período	Observação
ano	Ano	INT64	1985 - 2023	-
sigla_uf	Sigla da Unidade da Federação	STRING	1985 - 2023	-
id_municipio	ID do Município - IBGE (7 dígitos)	STRING	1985 - 2023	-
quantidade_vinculos_ativos	Estoque de vínculos ativos em 31/12	INT64	1985 - 2023	Soma dos vínculos ativos quando acumulada
quantidade_vinculos_clt	Estoque de vínculos CLT e outros ativos em 31/12	INT64	1985 - 2023	Soma dos vínculos ativos quando acumulada
quantidade_vinculos_estatutarios	Estoque de vínculos estatutários ativos em 31/12	INT64	1985 - 2023	Soma dos vínculos ativos quando acumulada
natureza_estabelecimento	Natureza do Estabelecimento	STRING	1985 - 2023	-
natureza_juridica	Natureza jurídica (CONCLA/2002)	STRING	1985 - 2023	-
tamanho_estabelecimento	Tamanho - empregados ativos em 31/12	STRING	1985 - 2023	-
tipo_estabelecimento	Tipo do Estabelecimento	STRING	1985 - 2023	-
indicador_cei_vinculado	Indicador de CEI vinculado	INT64	1985 - 2023	-
indicador_pat	Indicador de estabelecimento no PAT	INT64	1985 - 2023	-
indicador_simples	Indicador de optante pelo SIMPLES	INT64	1985 - 2023	-
indicador_rais_negativa	Indicador de Rais Negativa	INT64	1985 - 2023	-

indicador_atividade_ano	Indicador de atividade no ano de referência	INT64	1985 - 2023	-
cnae_1	Código Nacional de Atividades Econômicas 1.0	STRING	1985 - 2023	-
cnae_2	Código Nacional de Atividades Econômicas 2.0	STRING	1985 - 2023	-
cnae_2_subclasse	Subclasse do Código Nacional de Atividades Econômicas 2.0	STRING	1985 - 2023	-
subsetor_ibge	Subsetor IBGE	STRING	1985 - 2023	-
subatividade_ibge	Subatividade IBGE	STRING	1985 - 2023	-
cep	Código de Endereçamento Postal	STRING	1985 - 2023	-
bairros_sp	Bairros do município de São Paulo	STRING	1985 - 2023	-
distritos_sp	Distritos do município de São Paulo	STRING	1985 - 2023	-
bairros_fortaleza	Bairros do município de Fortaleza	STRING	1985 - 2023	-
bairros_rj	Bairros do município do Rio de Janeiro	STRING	1985 - 2023	-
regioes_administrativas_df	Regiões Administrativas do Distrito Federal	STRING	1985 - 2023	-

Vamos abrir por onde?

Base dos Dados: <https://basedosdados.org/>

Base dos Dados - Rais:

<https://basedosdados.org/dataset/3e7c4d58-96ba-448e-b053-d385a829ef00?table=86b69f96-0bfe-45da-833b-6edc9a0af213>

Como acessar a tela para inserir os comandos?

SQL > Acessar Big Query

SQL

Python

R

No editor de consultas do BigQuery, digite a seguinte instrução:

Primeira vez usando o BigQuery? Siga o passo a passo.

Acessar o BigQuery

Clicar em 'consulta' > 'Em uma nova guia'

The screenshot shows the Google Cloud BigQuery interface. The top navigation bar includes the Google Cloud logo, a dropdown menu for 'Aula Unesp', a search bar, and user profile information. The left sidebar contains navigation links for 'Análise', 'Migração', and 'Administração'. The main area is divided into three panels: 'Explorer', 'Schema', and 'Table Data'. The 'Explorer' panel shows a list of datasets under the 'basedosdados.br_me_rais' project, including 'microdados_estabelecimentos'. The 'Schema' panel displays the table schema for 'microdados_estabelecimentos', showing columns like 'ano', 'sigla_uf', 'id_municipio', and 'quantidade_vinculos_ativos'.

Nome do campo	Tipo	Modo	Chave	Compilação	Valor padrão	Tags de políticas
ano	INTEGER	NULLABLE	-	-	-	-
sigla_uf	STRING	NULLABLE	-	-	-	-
id_municipio	STRING	NULLABLE	-	-	-	-
quantidade_vinculos_ativos	INTEGER	NULLABLE	-	-	-	-
quantidade_vinculos_clt	INTEGER	NULLABLE	-	-	-	-
quantidade_vinculos_estatutarios	INTEGER	NULLABLE	-	-	-	-
natureza_estabelecimento	STRING	NULLABLE	-	-	-	-
natureza_juridica	STRING	NULLABLE	-	-	-	-
tamanho_estabelecimento	STRING	NULLABLE	-	-	-	-
tipo_estabelecimento	STRING	NULLABLE	-	-	-	-
indicador_rai_vinculado	INTEGER	NULLABLE	-	-	-	-

SELECT

- Selecionar Tabelas
- Todas as tabelas - (*)
- Tabelas específicas - Indica o nome da variável

Selecionando Todos os Dados, limitando as 10 primeiras linhas.

SELECT *

FROM `basedosdados.br_me_rais.microdados_estabelecimentos`

LIMIT 10

Selecionando dados específicos, limitando as 10 primeiras linhas

SELECT ano, id_municipio, cnae_2, cep

FROM `basedosdados.br_me_rais.microdados_estabelecimentos`

WHERE id_municipio = '3506003' AND ano = 2023

LIMIT 10

DISTINCT

- Remover linhas duplicadas e mostrar apenas linhas distintas muito usado na etapa de exploração das bases

Verificando os anos que minha base de dados tem

```
SELECT DISTINCT ano  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`
```

Verificando Combinações Distintas

```
SELECT DISTINCT ano, cnae_1  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`
```

WHERE

- Serve Para Filtrar Linhas Com Determinado Código
- SQL tem diferença minúsculo e maiores
- Em texto, usamos aspas simples

Selecionando todos os dados de Bauru - Mostrando as 10 primeiras linhas

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE id_municipio = '3506003'  
LIMIT 10
```

Encontrar um dado a partir de um filtro com mais de uma condição.

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE id_municipio = '3506003' OR id_municipio = '3526803'  
LIMIT 50
```

Encontrar um dado a partir de um filtro com mais de uma condição.

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE id_municipio = '3506003' OR id_municipio = '3526803' AND cnae_1 = '01139'  
LIMIT 50
```

Encontrar um dado a partir de um filtro com mais de uma condição, mostrando apenas colunas selecionadas.

```
SELECT id_municipio, cnae_1, cnae_2_subclasse  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE id_municipio = '3506003' OR id_municipio = '3526803' AND cnae_1 = '01139'  
LIMIT 50
```

Selecionando Todos os Dados de Bauru de 2023 - Mostrando as 10 primeiras linhas

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`
```

```
WHERE id_municipio = '3506003' AND ano = 2023  
LIMIT 10
```

ORDER BY

- Serve para ordenar a seleção de acordo com a regra definida pelo usuário

Ordena em valor crescente

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
ORDER BY quantidade_vinculos_ativos  
limit 1000
```

Ordena em valor decrescente

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
ORDER BY quantidade_vinculos_ativos desc  
limit 1000
```

Em valor de texto é a mesma coisa.

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
ORDER BY sigla_uf  
limit 1000
```

Ordena em valor decrescente

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
ORDER BY sigla_uf desc  
limit 1000
```

LIMIT

- Ajuda na fase exploratória para entender o que tem nas tabelas

Olhamos as 10 primeiras linhas

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
LIMIT 10
```

ORDER BY

- Você usa o order by para entender os N valores específicos (5 com mais vinculos por exemplo)

Ordena em valor decrescente

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
ORDER BY quantidade_vinculos_ativos desc
```

limit 5

OPERADORES DE COMPARAÇÃO (AS)

- Servem para comparar dois valores retornando true or false
- Muito utilizado em conjunto com a função WHERE para filtrar uma linha de seleção

Função AS, cria uma nova coluna Chamada 'Bauru' e toda vez que o id_municipio for o de bauru ele insere TRUE

```
SELECT id_municipio, cnae_1, cnae_2_subclasse,  
(id_municipio = '3506003') as Bauru  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE id_municipio = '3506003' OR id_municipio = '3526803' AND cnae_1 = '01139'  
LIMIT 50
```

OPERADORES LÓGICOS

- **AND:** Retorna resultados onde ambas as condições são verdadeiras.
- **OR:** Retorna resultados onde pelo menos uma das condições é verdadeira.
- **BETWEEN:** Filtra resultados dentro de um intervalo de valores, inclusivo nas extremidades.

Valores entre 100 mil e 200 mil. Utiliza função WHERE com operadores lógicos

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE quantidade_vinculos_ativos >= 10000 AND quantidade_vinculos_ativos <= 11000 limit 10
```

Uso do between para a mesma finalidade

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE quantidade_vinculos_ativos between 10000 AND 11000 limit 10
```

valores abaixo de 100 mil e acima de 200 mil

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE (quantidade_vinculos_ativos < 1000 OR quantidade_vinculos_ativos > 2000) and  
quantidade_vinculos_ativos IS NOT NULL  
limit 10
```

Mesma Finalidade Usando Not

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE quantidade_vinculos_ativos not between 1000 and 2000 and quantidade_vinculos_ativos IS NOT NULL  
limit 10
```

Selecionar Estados Que São Ou Ap Ou Sp

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE sigla_uf = 'SP' OR sigla_uf = 'AP'  
limit 10
```

Sintaxe diferente com a mesma finalidade

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE sigla_uf in ('SP','AP')  
limit 1000
```

FUNÇÕES AGREGADAS

- Servem para executar funções aritméticas nos registros de uma coluna
- Não computam células vazias

Contar todas as linhas de uma tabela - Primeira coisa que você faz ao ter acesso a uma tabela

```
SELECT count (*)  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`
```

Contar todas as linhas de uma coluna específica

```
SELECT count (quantidade_vinculos_ativos)  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`
```

Conta a maior renda total de toda tabela

```
SELECT *  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
WHERE quantidade_vinculos_ativos = (SELECT MAX(quantidade_vinculos_ativos) FROM  
`basedosdados.br_me_rais.microdados_estabelecimentos`)
```

GROUP BY

- Serve Para Agrupar Registros Semelhantes De Uma Coluna
- Normalmente Utilizado Em Conjunto Com Funções De Agregação

Calcular o número de respostas por estado

```
SELECT sigla_uf, count(*)  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
GROUP BY sigla_uf
```

Calcular o número de respostas por estado, criando nova coluna e ordenando por ordem crescente

```
SELECT sigla_uf, count(*) as contagem  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
GROUP BY sigla_uf  
ORDER BY contagem
```


Calcular o número de respostas por estado e id do quadro

```
SELECT sigla_uf, cnae_1, count(*) as contagem  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
GROUP BY sigla_uf, cnae_1
```

HAVING

- Serve para filtrar linhas da seleção por uma coluna agrupada
- aqui eu chamei os estados e cnae e contei/agrupei chamando de contagem.
- Selecionei porém, os menores que 3000

```
SELECT sigla_uf, cnae_1, count(*) as contagem  
FROM `basedosdados.br_me_rais.microdados_estabelecimentos`  
GROUP BY sigla_uf, cnae_1  
Having count (*) < 3000
```