

Guia SQL: primeiros passos

Assim como os humanos usam linguagens diferentes para se comunicar com outras pessoas, o mesmo ocorre com os computadores. A **linguagem de consulta estruturada** (ou **SQL**, geralmente pronunciada como “sequel”) permite que os analistas de dados se comuniquem com os bancos de dados. O SQL é uma das ferramentas mais úteis para os analistas de dados, especialmente ao trabalhar com grandes conjuntos de dados em tabelas. Ela pode ajudar a investigar bancos de dados enormes, rastrear textos (conhecidos como strings) e números e filtrar o tipo exato de dados de que você precisa de maneira muito mais rápida do que uma planilha. Se você nunca usou SQL antes, neste texto ajudaremos a aprender o básico para apreciar como ele é útil e como as consultas SQL são particularmente úteis. Você escreverá consultas SQL em um piscar de olhos.

O QUE É UMA CONSULTA?

Uma **consulta** é uma solicitação de dados ou informações de um banco de dados. Ao consultar bancos de dados, você usa o SQL para comunicar sua pergunta ou solicitação. Você e o banco de dados sempre podem trocar informações, desde que falem a mesma língua.

Cada linguagem de programação, incluindo SQL, segue um conjunto exclusivo de diretrizes, conhecido como **sintaxe**. A **sintaxe** é a estrutura predeterminada de um idioma que inclui todas as palavras, pontuações e símbolos necessários, bem como o posicionamento adequado deles. Assim que você inserir critérios de pesquisa usando a sintaxe correta, a consulta começará a trabalhar para extrair os dados solicitados do banco de dados de destino.

A sintaxe de cada consulta SQL é a mesma:

- Use **SELECT** para escolher as colunas que deseja retornar.
- Use **FROM** para escolher as tabelas em que estão localizadas as colunas que você quer.
- Use **WHERE** para filtrar por certas informações.

Uma consulta **SQL** é como preencher um modelo. Você descobrirá que, ao escrever uma consulta **SQL** do zero, é útil iniciá-la escrevendo as palavras-chave **SELECT**, **FROM** e **WHERE** no seguinte formato:

SELECT

FROM

WHERE

Em seguida, insira o nome da tabela após **FROM**; as colunas da tabela que você quer após o **SELECT**; e, por fim, as condições que você quer colocar em sua consulta após **WHERE**. Não deixe de adicionar uma linha nova e um recuo ao adicionar esses comandos, conforme mostrado abaixo:

SELECT

Columns you want to look at

FROM

Table the data lives in

WHERE

Certain condition is met

Seguir esse método sempre facilita a escrita de consultas SQL. Isso também pode ajudar você a cometer menos erros de sintaxe.

EXEMPLO DE UMA CONSULTA

Veja como uma consulta simples apareceria no BigQuery, um data warehouse no Google Cloud Platform.

```
SELECT
  first_name
FROM
  customer_data.customer_name
WHERE
  first_name = 'Tony'
```

A consulta acima usa três comandos para localizar clientes com o nome Tony:

1. **SELECT** (selecione) a coluna chamada **first_name**
2. **FROM** (de) uma tabela chamada **customer_name** (em um conjunto de dados chamado **customer_data**). (O nome do conjunto de dados é sempre seguido por um ponto e depois pelo nome da tabela.)
3. Mas só retorne os dados **WHERE** (onde) o **first_name** (nome) for **Tony**

Os resultados da consulta podem ser semelhantes aos mostrados a seguir:

first_name
Tony
Tony
Tony

Como você pode concluir, essa consulta tinha a sintaxe correta, mas não foi muito útil depois que os dados foram retornados.

VÁRIAS COLUNAS EM UMA CONSULTA

Na vida real, você precisará trabalhar com mais dados, além de clientes chamados Tony. Várias colunas escolhidas pelo mesmo comando **SELECT** podem ser recuadas e agrupadas.

Se você solicitar vários campos de dados de uma tabela, precisará incluir essas colunas no comando **SELECT**. Cada coluna é separada por uma vírgula, conforme mostrado abaixo:

```
SELECT
    ColumnA,
    ColumnB,
    ColumnC
FROM
    Table where the data lives
WHERE
    Certain condition is met
```

Veja aqui um exemplo de como isso ficaria no **BigQuery**:

```
SELECT
    customer_id,
    first_name,
    last_name
FROM
    customer_data.customer_name
WHERE
    first_name = 'Tony'
```

A consulta acima usa três comandos para localizar clientes com o nome Tony.

1. **SELECT** (selecione) as colunas denominadas **customer_id**, **first_name** e **last_name**
2. **FROM** (de) uma tabela chamada **customer_name** (em um conjunto de dados chamado **customer_data**). (O nome do conjunto de dados é sempre seguido por um ponto e depois pelo nome da tabela.)
3. Mas só retorne os dados **WHERE** (onde) o **first_name** (nome) for **Tony**

A única diferença entre essa consulta e a anterior é que mais colunas de dados foram selecionadas. A consulta anterior selecionou apenas `first_name`, enquanto essa consulta selecionou também `customer_id` e `last_name` além de `first_name`. Em geral, selecionar apenas as colunas de que você precisa faz um uso mais eficiente dos recursos. Por exemplo, faz sentido selecionar mais colunas se você realmente usar os campos adicionais em sua cláusula **WHERE**. Se você tiver várias condições em sua cláusula **WHERE**, elas poderão ser escritas assim:

```
SELECT
    ColumnA,
    ColumnB,
    ColumnC
FROM
    Table where the data lives
WHERE
    Condition 1
    AND Condition 2
    AND Condition 3
```

Observe que, ao contrário do comando **SELECT** (que usa uma vírgula para separar campos/variáveis/parâmetros), o comando **WHERE** usa a instrução **AND** para conectar as condições. À medida que você se tornar um criador de consultas mais experiente, fará uso de outros conectores/operadores, como **OR** e **NOT**.

Veja aqui um exemplo do **BigQuery** com vários campos usados em uma cláusula **WHERE**:

```
SELECT
    customer_id,
    first_name,
    last_name
FROM
    customer_data.customer_name
WHERE
    customer_id > 0
    AND first_name = 'Tony'
    AND last_name = 'Magnolia'
```

A consulta acima usa três comandos para localizar clientes com um ID de cliente válido (maior que 0) com nome Tony e sobrenome Magnolia.

1. **SELECT** (selecione) as colunas denominadas **customer_id**, **first_name** e **last_name**
2. **FROM** (de) uma tabela chamada **customer_name** (em um conjunto de dados chamado **customer_data**). (O nome do conjunto de dados é sempre seguido por um ponto e depois pelo nome da tabela.)
3. Mas só retorne os dados **WHERE** (onde) **customer_id** for maior que **0**, o **first_name** (nome) for **Tony** e o **last_name** (sobrenome) for **Magnolia**.

Observe que uma das condições é uma condição lógica que verifica se **customer_id** é maior que zero.

Se apenas um cliente se chamar Tony Magnolia, os resultados da consulta poderão ser os mostrados a seguir:

customer_id	first_name	last_name
1967	Tony	Magnolia

Se mais de um cliente tiver o mesmo nome, os resultados da consulta poderão ser os mostrados a seguir:

customer_id	first_name	last_name
1967	Tony	Magnolia
7689	Tony	Magnolia

Principal conclusão

O mais importante a ser lembrado é como usar **SELECT**, **FROM** e **WHERE** em uma consulta. As consultas com vários campos ficarão mais simples depois que você praticar a escrita de suas próprias consultas SQL posteriormente no programa.