

SQL, sintaxe e principais conceitos.

Os comandos SQL por convenção são escritos em maiúsculo e finalizados com “ ; “.

`SELECT * FROM table;`

No entanto, caso o comando seja escrito em minúsculo, ele também funcionará.

Os principais comandos incluem:

- `SELECT` - extrai dados de um database/tabela
 - `SELECT DISTINCT` – extrai valores únicos da tabela
- `UPDATE` - atualiza dados numa tabela
- `DELETE` - deleta dados
- `INSERT INTO` - insere novos dados
- `CREATE DATABASE` - cria uma nova database
- `ALTER DATABASE` - modifica uma database existente
- `CREATE TABLE` - cria uma nova tabela
- `ALTER TABLE` - modifica uma tabela existente
- `DROP TABLE` - deleta uma tabela
- `CREATE INDEX` - cria um índice (chave de pesquisa)
- `DROP INDEX` - deleta um índice

Além dos comando gerais, temos comandos que apoiam estes gerais, os principais são:

- `WHERE` – Filtra dados do select.
 - `WHERE` aceita operadores lógicos `AND` `OR` `NOT`
- `ORDER BY` – Organiza os dados segundo uma coluna escolhida.
- `GROUP BY` – Agrupa dados a partir de uma coluna ou função.

Constantemente também utilizamos funções para trabalhar os dados:

- `MIN()` - retorna o menor valor da coluna escolhida
- `MAX()` - retorna o maior valor da coluna escolhida
- `COUNT()` - retorna o número de linhas de um determinado recorte

- SUM() - retorna a soma de valores de uma coluna numérica
- AVG() - retorna a média dos valores de uma coluna numérica

Alguns exemplos de queries comuns no dia-a-dia são expostos abaixo:

Existem 3 tabelas com algumas colunas para os exemplos:

- Clientes
 - ClienteID, Nome, Email, Cidade, Estado
- Produtos
 - ProdutoID, NomeProduto, Preço, Categoria
- Pedidos
 - PedidoID, ClienteID, DataPedido, ValorTotal
- Retorna todas as colunas de uma tabela:
 - `SELECT * FROM Clientes;`
- Selecionar colunas específicas:
 - `SELECT Nome, Email FROM Clientes;`
- Filtrar dados:
 - `SELECT * FROM Clientes WHERE Cidade = "Niterói";`
- Filtrar dados com and, or, not:
 - `SELECT * FROM Clientes WHERE Cidade = "Niterói" AND Estado = "RJ";`
- Selecionando valores distintos (dentro de tabelas com muitas repetições)
 - `SELECT DISTINCT Cidade FROM Clientes;`
- Ordenando resultados:
 - `SELECT NomeProduto, Preço FROM Produtos ORDER BY Preço DESC;`
- Lista clientes em ordem alfabética
 - `SELECT Nome, Cidade FROM Clientes ORDER BY Nome ASC;`
- Inserindo dados:
 - `INSERT INTO Clientes (ClienteID, Nome, Email, Cidade, Estado) VALUES (101, "Caio Marins", "caio@email.com", "Niterói", "RJ");`
- Atualizando dados:
 - `UPDATE Clientes SET Email = "novo@email.com" WHERE ClienteID = 101;`
- Deletando Dados

- `DELETE FROM Clientes WHERE ClienteID = 101;`
- Contando Linhas:
 - `SELECT COUNT(*) FROM Clientes;`
- Conta quantos clientes são do estado do Rio de Janeiro.
 - `SELECT COUNT(*) FROM Clientes WHERE Estado = 'RJ';`
- Somando Valores:
 - `SELECT SUM(ValorTotal) FROM Pedidos;`
- Calculando a Média:
 - `SELECT AVG(Preco) FROM Produtos;`
- Encontrando o Maior e o Menor Valor:
 - `SELECT MAX(Preco) AS PrecoMaisCaro, MIN(Preco) AS PrecoMaisBarato FROM Produtos;`
 - Repare que o AS cria um “Alias” para uma nova coluna que conterá os valores, mas não existe na tabela original!
- Agrupando Dados:
 - `SELECT Cidade, COUNT(ClienteID) AS NumeroDeClientes`
 - `FROM Clientes`
 - `GROUP BY Cidade;`
 - Repare que todas as linhas acima são uma mesma querysql, podemos dividir em várias linhas sem nenhum problema.
- Calcula o valor total de pedidos por cliente.
 - `SELECT ClienteID, SUM(ValorTotal) AS TotalGasto`
 - `FROM Pedidos`
 - `GROUP BY ClienteID;`

Alguns comandos são reservados a definição de dados (DDL), servem para criar e manipular a modelagem do banco. São alguns deles:

- Criando um Banco de Dados:
 - `CREATE DATABASE Vendas;`
- Criando uma Tabela:
 - `CREATE TABLE Fornecedores (`
 - `FornecedorID INT PRIMARY KEY,`
 - `Nome VARCHAR(255) NOT NULL,`
 - `Contato VARCHAR(255),`
 - `CNPJ VARCHAR(18) UNIQUE`
 - `);`
- Modificando uma Tabela:
 - `ALTER TABLE Clientes ADD Telefone VARCHAR(20);`
- Remove a coluna Telefone da tabela Clientes.
 - `ALTER TABLE Clientes DROP COLUMN Telefone;`
- Deletando uma Tabela:
 - `DROP TABLE Fornecedores;`
- Criando um Índice, criar um índice na coluna Nome da tabela Clientes irá acelerar as buscas por nome.
 - `CREATE INDEX idx_nome_cliente ON Clientes (Nome);`
- Deletando um Índice. A sintaxe pode variar entre sistemas de banco de dados (MySQL, PostgreSQL, etc.).
 - `DROP INDEX idx_nome_cliente ON Clientes;`