

Téc em Desenvolvimento
de Sistemas Bilíngue

**Planejar e Administrar
Banco de Dados**

Téc em Desenvolvimento
de Sistemas Bilíngue

Comandos Básicos SQL

Comandos Básicos SQL

DDL - Data Definition Language (Linguagem de Definição de Dados)

Ela consiste em um conjunto de comandos usados para criar, alterar e excluir objetos dentro do banco de dados.

Permite criar novos objetos no banco de dados (como tabelas), alterar a estrutura de objetos existentes (como adicionar colunas) e excluir objetos quando não forem mais necessários.

Comandos Básicos SQL

Como criar uma tabela

```
CREATE TABLE clientes (  
    -- coluna / tipo de dado / restrição  
    idClientes INT PRIMARY KEY,  
    nome VARCHAR(100),  
    telefone VARCHAR(100)  
);
```

Comandos Básicos SQL

Como evitar tentar criar uma tabela que já existe

```
CREATE TABLE IF NOT EXISTS clientes (  
    -- coluna / tipo de dado / restrição  
    idClientes INT PRIMARY KEY,  
    nome VARCHAR(100),  
    telefone VARCHAR(100)  
);
```

Comandos Básicos SQL

Como alterar uma tabela já criada

```
ALTER TABLE clientes
```

E então, adicionar o resto do comando para alterar o que você quer, como por exemplo adicionar uma nova coluna. Vamos ver no exemplo a seguir.

Comandos Básicos SQL

Como alterar uma tabela já criada e adicionar uma coluna

```
ALTER TABLE clientes  
ADD email VARCHAR(100);
```

Comandos Básicos SQL

Como alterar uma tabela já criada e modificar o tipo de uma coluna

```
ALTER TABLE clientes  
MODIFY email TEXT;
```


Comandos Básicos SQL

Como alterar uma tabela já criada e renomear uma coluna

```
ALTER TABLE clientes  
CHANGE COLUMN email emailCliente VARCHAR(255);
```

Comandos Básicos SQL

Como alterar uma tabela já criada e remover uma coluna

```
ALTER TABLE clientes  
DROP COLUMN telefone
```

Comandos Básicos SQL

Como alterar uma tabela já criada e
alterar o nome dela

```
ALTER TABLE clientes  
RENAME TO Clientes;
```

Comandos Básicos SQL

Como deletar uma tabela

```
DROP TABLE clientes;
```

Comandos Básicos SQL

DML - Data Manipulation Language (Linguagem de Manipulação de Dados)

É usada para manipular dados dentro de um banco de dados. Isso inclui a inserção, atualização, exclusão e consulta de dados armazenados nas tabelas.

Comandos Básicos SQL

Como inserir dados em uma tabela

```
INSERT INTO clientes (idCliente, nome, email)
VALUES (1, 'João Silva', 'joao.silva@email.com');
```

Comandos Básicos SQL

Como atualizar dados em uma tabela

```
UPDATE clientes  
SET email = 'novo.email@email.com'  
WHERE idCliente = 1; -- condição
```

Comandos Básicos SQL

Como deletar dados em uma tabela

```
DELETE FROM Clientes  
WHERE idCliente = 1;
```


Comandos Básicos SQL

Como fazer consultas

Consultas em bancos de dados são pedidos ou perguntas que fazemos ao sistema para obter informações específicas dos dados armazenados em tabelas.

Permitem selecionar, filtrar, ordenar e manipular dados conforme necessário. Basicamente, uma consulta permite buscar e recuperar informações específicas de um banco de dados de acordo com critérios definidos, como por exemplo, selecionar todos os clientes com um determinado email ou ordenar produtos por preço.

Comandos Básicos SQL

Consulta Simples: retorna todos os dados da tabela

```
SELECT idCliente, nome, email  
FROM clientes;
```

```
SELECT *  
FROM clientes;
```

Comandos Básicos SQL

Consulta com filtro: retorna dados com base em uma condição

```
SELECT email  
FROM Clientes  
WHERE idCliente = 1;
```

Comandos Básicos SQL

Consulta com ordenação: retorna dados ordenados

```
SELECT idCliente, nome, email  
FROM Clientes  
ORDER BY nome ASC; -- OU DESC
```

Comandos Básicos SQL

Consulta com filtro e com ordenação:
retorna dados ordenados e com base
em uma condição

```
SELECT nome  
FROM Clientes  
WHERE email = 'joao.silva@email.com'  
ORDER BY idCliente DESC;
```

Comandos Básicos SQL

Consulta com limite de resultados

```
SELECT idCliente, nome, email  
FROM Clientes  
LIMIT 3;
```

Comandos Básicos SQL

Consulta de valores únicos

```
SELECT DISTINCT email  
FROM Clientes;
```

Comandos Básicos SQL

Agora, crie esta tabela:

```
CREATE TABLE Pedidos (  
    idPedido INT AUTO_INCREMENT PRIMARY KEY,  
    idCliente INT,  
    numeroPedido VARCHAR(50),  
    valorTotal DECIMAL(10, 2),  
    dataPedido DATE,  
    FOREIGN KEY (idCliente) REFERENCES Clientes(idCliente)  
);
```


Comandos Básicos SQL

Insira estes dados:

```
INSERT INTO Pedidos (idCliente, numeroPedido, valorTotal, dataPedido)
VALUES (1, 'PED-001', 150.00, '2024-07-12'),
       (2, 'PED-002', 250.00, '2024-07-12'),
       (1, 'PED-003', 180.00, '2024-07-13');
```

Comandos Básicos SQL

Consulta com função SUM:

```
SELECT SUM(valorTotal) AS valor_total_pedidos  
FROM Pedidos;
```

Soma o valor total de pedidos.

Comandos Básicos SQL

Consulta com função AVG:

```
SELECT AVG(valorTotal) AS valor_medio_pedidos  
FROM Pedidos;
```

Calcula o valor médio dos pedidos.

Comandos Básicos SQL

Consulta com função MAX:

```
SELECT MAX(numeroPedido) AS pedido_com_numero_mais_alto  
FROM Pedidos;
```

Encontra o pedido com o número mais alto em ordem alfabética.

Comandos Básicos SQL

Consulta com função MIN:

```
SELECT MIN(numeroPedido) AS pedido_com_numero_mais_baixo  
FROM Pedidos;
```

Encontra o pedido com o número mais baixo em ordem alfabética.

Comandos Básicos SQL

Consulta com Junção de Tabelas (JOIN):

```
SELECT Clientes.idCliente, Clientes.nome, Clientes.email, Pedidos.numeroPedido  
FROM Clientes  
INNER JOIN Pedidos ON Clientes.idCliente = Pedidos.idCliente;
```

Seleciona clientes e seus pedidos correspondentes usando uma junção entre a tabela Clientes e Pedidos. Neste exemplo, INNER JOIN combina registros de ambas as tabelas onde existem correspondências entre os campos idCliente.

Comandos Básicos SQL

Consulta com Junção Externa (LEFT JOIN):

```
SELECT Clientes.idCliente, Clientes.nome, Pedidos.numeroPedido  
FROM Clientes  
LEFT JOIN Pedidos ON Clientes.idCliente = Pedidos.idCliente;
```

Seleciona todos os clientes, incluindo aqueles que não têm pedidos correspondentes. LEFT JOIN retorna todos os registros da tabela à esquerda (Clientes), e os registros correspondentes da tabela à direita (Pedidos). Se não houver correspondência, são retornados NULLs para os campos da tabela à direita.

Comandos Básicos SQL

Consulta com Junção Condicional (JOIN com WHERE):

```
SELECT Clientes.idCliente, Clientes.nome, Pedidos.numeroPedido, Pedidos.valorTotal  
FROM Clientes  
INNER JOIN Pedidos ON Clientes.idCliente = Pedidos.idCliente  
WHERE Pedidos.valorTotal > 200;
```

Seleciona clientes e seus pedidos apenas se o valor total do pedido for superior a 200. Esta consulta usa INNER JOIN para combinar registros onde há correspondência nos idCliente, filtrando apenas pedidos com valor total superior a 200.

Comandos Básicos SQL

Consulta com função COUNT:

```
SELECT COUNT(*)  
FROM Pedidos;
```

Conta o número total de pedidos.