

ENUNCIADO COMPLETO DE COMANDOS E FUNÇÕES SQL

Este documento apresenta comandos SQL incluindo criação, alteração, consultas, joins e funções agregadoras.

1. CREATE TABLE

```sql

CREATE TABLE Cliente (id INT PRIMARY KEY, nome VARCHAR(100), telefone VARCHAR(20));

CREATE TABLE Produto (codigo INT PRIMARY KEY, descricao VARCHAR(255), preco DECIMAL(10,2));

CREATE TABLE Categoria (id\_categoria INT PRIMARY KEY, nome VARCHAR(100));

...

### ## 2. ALTER TABLE

```sql

ALTER TABLE Cliente ADD email VARCHAR(120);

ALTER TABLE Produto MODIFY preco DECIMAL(12,2);

ALTER TABLE Produto ADD CONSTRAINT fk_categoria FOREIGN KEY (id_categoria) REFERENCES Categoria(id_categoria);

...

3. DROP TABLE

```sql

DROP TABLE Cliente;

DROP TABLE IF EXISTS Produto;

DROP TABLE Categoria;

...

### ## 4. SELECT (Consultas)

```sql

SELECT * FROM Cliente;

SELECT nome, telefone FROM Cliente;

SELECT nome, preco FROM Produto WHERE preco > 50;

...

5. INNER JOIN

```sql

SELECT c.nome, p.valor\_total FROM Pedido p INNER JOIN Cliente c ON c.id = p.id\_cliente;

SELECT f.nome, e.nome FROM Funcionario f INNER JOIN Empresa e ON e.id\_empresa = f.id\_empresa;

```
SELECT p.descricao, c.nome FROM Produto p INNER JOIN Categoria c ON c.id_categoria =
p.id_categoria;
...
```

## ## 6. LEFT JOIN

```
```sql  
SELECT c.nome, p.valor_total FROM Cliente c LEFT JOIN Pedido p ON p.id_cliente = c.id;  
...
```

7. GROUP BY e HAVING

```
```sql  
SELECT estado, COUNT(*) FROM Cliente GROUP BY estado;
SELECT id_cliente, COUNT(*) FROM Pedido GROUP BY id_cliente HAVING COUNT(*) > 5;
...
```

## ## 8. Funções Agregadoras

```
```sql  
SELECT COUNT(*) FROM Cliente;  
SELECT MAX(preco) FROM Produto;  
SELECT AVG(preco) FROM Produto;  
...
```

9. Subqueries

```
```sql  
SELECT * FROM Cliente WHERE id IN (SELECT id_cliente FROM Pedido);
SELECT * FROM Produto WHERE preco > (SELECT AVG(preco) FROM Produto);
...
```

## ## 10. Modelo completo com FK e PK

```
```sql  
CREATE TABLE Cidade (  
id_cidade INT PRIMARY KEY AUTO_INCREMENT,  
nome VARCHAR(100),  
estado VARCHAR(2)  
);  
  
CREATE TABLE Empresa (  
id_empresa INT PRIMARY KEY AUTO_INCREMENT,  
nome VARCHAR(150),  
cnpj VARCHAR(18),
```

```
id_cidade INT,  
FOREIGN KEY (id_cidade) REFERENCES Cidade(id_cidade)  
);  
  
CREATE TABLE Funcionario (  
id_func INT PRIMARY KEY AUTO_INCREMENT,  
nome VARCHAR(150),  
salario DECIMAL(10,2),  
id_empresa INT,  
FOREIGN KEY (id_empresa) REFERENCES Empresa(id_empresa)  
);  
...
```