Aqui está uma lista dos principais comandos SQL, organizados do básico ao avançado, com suas funções explicadas:

**Comandos Básicos**

1. **SELECT**
   * Recupera dados de uma tabela.
   * Exemplo: SELECT \* FROM tabela;
2. **INSERT INTO**
   * Insere novos registros em uma tabela.
   * Exemplo: INSERT INTO tabela (coluna1, coluna2) VALUES (valor1, valor2);
3. **UPDATE**
   * Atualiza dados existentes em uma tabela.
   * Exemplo: UPDATE tabela SET coluna1 = valor1 WHERE condicao;
4. **DELETE**
   * Remove registros de uma tabela.
   * Exemplo: DELETE FROM tabela WHERE condicao;
5. **CREATE TABLE**
   * Cria uma nova tabela no banco de dados.
   * Exemplo:

sql

Copiar código

CREATE TABLE tabela (

id INT PRIMARY KEY,

nome VARCHAR(50),

idade INT

);

1. **DROP TABLE**
   * Exclui uma tabela do banco de dados.
   * Exemplo: DROP TABLE tabela;
2. **ALTER TABLE**
   * Modifica a estrutura de uma tabela (adicionar, excluir ou modificar colunas).
   * Exemplo: ALTER TABLE tabela ADD coluna\_nova VARCHAR(100);
3. **WHERE**
   * Filtra registros com base em uma condição.
   * Exemplo: SELECT \* FROM tabela WHERE idade > 30;

**Comandos Intermediários**

1. **GROUP BY**
   * Agrupa registros com base em uma ou mais colunas.
   * Exemplo: SELECT categoria, COUNT(\*) FROM produtos GROUP BY categoria;
2. **HAVING**
   * Filtra grupos após a agregação.
   * Exemplo: SELECT categoria, COUNT(\*) FROM produtos GROUP BY categoria HAVING COUNT(\*) > 10;
3. **ORDER BY**
   * Ordena os resultados em ordem crescente (ASC) ou decrescente (DESC).
   * Exemplo: SELECT \* FROM tabela ORDER BY idade DESC;
4. **JOIN**
   * Combina registros de duas ou mais tabelas com base em uma condição.
     + **INNER JOIN**: Retorna registros correspondentes nas tabelas.
     + **LEFT JOIN**: Retorna todos os registros da tabela esquerda e correspondências na tabela direita.
     + **RIGHT JOIN**: Retorna todos os registros da tabela direita e correspondências na tabela esquerda.
     + **FULL OUTER JOIN**: Retorna todos os registros quando há uma correspondência em qualquer uma das tabelas.
   * Exemplo:

sql

Copiar código

SELECT a.nome, b.departamento

FROM funcionarios a

INNER JOIN departamentos b ON a.departamento\_id = b.id;

1. **LIKE**
   * Pesquisa por padrões específicos.
   * Exemplo: SELECT \* FROM tabela WHERE nome LIKE 'A%'; (nomes que começam com "A").
2. **DISTINCT**
   * Retorna valores únicos.
   * Exemplo: SELECT DISTINCT cidade FROM clientes;
3. **LIMIT** (ou TOP em alguns sistemas)
   * Restringe o número de registros retornados.
   * Exemplo: SELECT \* FROM tabela LIMIT 10;

**Comandos Avançados**

1. **SUBQUERIES**
   * Consulta dentro de outra consulta.
   * Exemplo:

sql

Copiar código

SELECT nome FROM clientes

WHERE id IN (SELECT cliente\_id FROM pedidos WHERE total > 1000);

1. **CASE**
   * Estrutura condicional dentro de uma consulta.
   * Exemplo:

sql

Copiar código

SELECT nome,

CASE

WHEN idade < 18 THEN 'Menor'

WHEN idade BETWEEN 18 AND 60 THEN 'Adulto'

ELSE 'Idoso'

END AS faixa\_etaria

FROM clientes;

1. **TRIGGERS**
   * Executa uma ação automaticamente antes ou depois de um evento em uma tabela.
   * Exemplo:

sql

Copiar código

CREATE TRIGGER antes\_insert

BEFORE INSERT ON tabela

FOR EACH ROW

BEGIN

SET NEW.criado\_em = NOW();

END;

1. **VIEWS**
   * Cria uma "tabela virtual" com base em uma consulta.
   * Exemplo:

sql

Copiar código

CREATE VIEW clientes\_vips AS

SELECT nome, total\_gasto

FROM clientes

WHERE total\_gasto > 10000;

1. **INDEX**
   * Cria um índice para melhorar o desempenho das consultas.
   * Exemplo: CREATE INDEX idx\_nome ON tabela(nome);
2. **TRANSACTIONS**
   * Garante a consistência dos dados (com COMMIT e ROLLBACK).
   * Exemplo:

sql

Copiar código

BEGIN;

UPDATE conta SET saldo = saldo - 100 WHERE id = 1;

UPDATE conta SET saldo = saldo + 100 WHERE id = 2;

COMMIT;

1. **PARTITION**
   * Divide tabelas grandes em partes menores para melhorar o desempenho.
   * Exemplo:

sql

Copiar código

CREATE TABLE historico (

id INT,

data DATE

)

PARTITION BY RANGE (YEAR(data)) (

PARTITION p1 VALUES LESS THAN (2000),

PARTITION p2 VALUES LESS THAN (2010),

PARTITION p3 VALUES LESS THAN MAXVALUE

);

1. **WINDOW FUNCTIONS**
   * Realiza cálculos em conjuntos de linhas relacionados.
   * Exemplo:

sql

Copiar código

SELECT nome, salario,

RANK() OVER (PARTITION BY departamento\_id ORDER BY salario DESC) AS rank

FROM funcionarios;

1. **CTE (Common Table Expressions)**
   * Define subconsultas reutilizáveis.
   * Exemplo:

sql

Copiar código

WITH total\_vendas AS (

SELECT cliente\_id, SUM(valor) AS total

FROM vendas

GROUP BY cliente\_id

)

SELECT \* FROM total\_vendas WHERE total > 5000;

1. **FULL-TEXT SEARCH**
   * Realiza buscas textuais avançadas.
   * Exemplo: SELECT \* FROM artigos WHERE MATCH(titulo, conteudo) AGAINST ('SQL tutorial');