

Regras Básicas do SQL

1. Selecionar dados

- Selecionar todos os campos de uma tabela:

```
SELECT * FROM nome_tabela;
```

- Selecionar colunas específicas:

```
SELECT nome_coluna1, nome_coluna2 FROM nome_tabela;
```

- Selecionar uma coluna com condição:

```
SELECT nome_coluna FROM nome_tabela WHERE condicao;
```

- Selecionar campos cujo título comece com 'o':

```
SELECT * FROM nome_tabela WHERE nome_coluna LIKE 'o%';
```

- Selecionar campos cujo título termine com 'o':

```
SELECT * FROM nome_tabela WHERE nome_coluna LIKE '%o';
```

- Ordenar uma coluna em ordem decrescente:

```
SELECT * FROM nome_tabela ORDER BY nome_coluna DESC;
```

2. Modificação de tabelas

- Renomear tabela:

```
SP_RENAME 'nome_atual', 'novo_nome';
```

- Adicionar coluna:

```
ALTER TABLE nome_tabela ADD nome_nova_coluna tipo_dado;
```

- Alterar tipo de dado de uma coluna:

```
ALTER TABLE nome_tabela ALTER COLUMN nome_coluna novo_tipo_dado;
```

- Alterar nome de uma coluna:

```
SP_RENAME 'nome_tabela.nome_coluna', 'novo_nome_coluna', 'COLUMN';
```

- Apagar coluna:

```
ALTER TABLE nome_tabela DROP COLUMN nome_coluna;
```

3. Manipulação de Dados

- Inserir dados em uma tabela:

```
INSERT INTO nome_tabela (coluna1, coluna2) VALUES (valor1, valor2);
```

- Atualizar dados:

```
UPDATE nome_tabela SET nome_coluna = novo_valor WHERE condicao;
```

- Apagar linha com condição:

```
DELETE FROM nome_tabela WHERE condicao;
```

- Apagar todas as linhas sem apagar a estrutura da tabela:

```
TRUNCATE TABLE nome_tabela;
```

4. Estruturas Avançadas

- Criar uma chave primária:

```
ALTER TABLE nome_tabela ADD CONSTRAINT pk_nome PRIMARY KEY (nome_coluna);
```

- Criar uma chave estrangeira:

```
ALTER TABLE nome_tabela ADD CONSTRAINT fk_nome FOREIGN KEY (nome_coluna)  
REFERENCES outra_tabela(coluna_referenciada);
```

- Pesquisar valores dentro de um intervalo:

```
SELECT * FROM nome_tabela WHERE nome_coluna BETWEEN valor1 AND valor2;
```

- Pesquisar valores dentro de uma lista:

```
SELECT * FROM nome_tabela WHERE nome_coluna IN (valor1, valor2, valor3);
```

- Filtrar valores nulos:

```
SELECT * FROM nome_tabela WHERE nome_coluna IS NULL;
```

- Contar quantidade de registros:

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

- Agrupar dados:

```
SELECT nome_coluna, COUNT(*) FROM nome_tabela GROUP BY nome_coluna;
```

- Filtrar grupos:

```
SELECT nome_coluna, COUNT(*) FROM nome_tabela GROUP BY nome_coluna HAVING  
COUNT(*) > valor;
```

5. Criar uma View

- Criar uma view para reutilizar consultas complexas:

```
CREATE VIEW nome_view AS SELECT coluna1, coluna2 FROM nome_tabela WHERE condicao;
```

- Consultar a view:

```
SELECT * FROM nome_view;
```

Essas são algumas das regras básicas do SQL para ajudar no entendimento e aplicação prática.