

SELECT — Buscar dados

O que faz: Busca informações no banco.

Quando usar: Sempre que quiser consultar dados.

Exemplo:

```
SELECT name, salary FROM employee;
```

FROM — Origem dos dados

O que faz: Define a tabela base da consulta.

Exemplo:

```
SELECT * FROM employee;
```

WHERE — Filtro de linhas

O que faz: Filtra registros antes do agrupamento.

Exemplo:

```
SELECT name FROM employee WHERE salary > 5000;
```

INSERT — Inserir dados

O que faz: Adiciona novos registros.

Exemplo:

```
INSERT INTO employee (name, salary) VALUES ('Ana', 4500);
```

UPDATE — Atualizar dados

O que faz: Modifica registros existentes.

Exemplo:

```
UPDATE employee SET salary = 5000 WHERE id = 1;
```

DELETE — Remover dados

O que faz: Exclui registros.

Exemplo:

```
DELETE FROM employee WHERE id = 10;
```

JOIN — Relacionar tabelas

O que faz: Une dados de tabelas relacionadas.

Exemplo:

```
SELECT e.name, d.name FROM employee e JOIN department d ON e.department_id = d.id;
```

GROUP BY — Agrupar dados

O que faz: Agrupa linhas para cálculo.

Exemplo:

```
SELECT department_id, AVG(salary) FROM employee GROUP BY department_id;
```

HAVING — Filtro de grupos

O que faz: Filtra resultados após GROUP BY.

Exemplo:

```
HAVING AVG(salary) > 6000;
```

ORDER BY — Ordenar resultados

O que faz: Ordena o resultado final.

Exemplo:

```
ORDER BY salary DESC;
```

LIMIT — Limitar linhas

O que faz: Limita quantidade de registros.

Exemplo:

```
LIMIT 5;
```

SUBQUERY — Consulta aninhada

O que faz: Usa o resultado de uma consulta dentro de outra.

Exemplo:

```
SELECT name FROM employee WHERE salary > (SELECT AVG(salary) FROM employee);
```

EXISTS — Verificar existência

O que faz: Verifica se existe relacionamento.

Exemplo:

```
SELECT name FROM employee e WHERE EXISTS (SELECT 1 FROM works_on w WHERE w.employee_id = e.id);
```

ALTER TABLE — Manutenção

O que faz: Altera estrutura da tabela.

Exemplo:

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