

## Lista de Consultas SQL por Nível de Dificuldade

Aqui está a lista organizada das perguntas SQL, classificadas da mais fácil para a mais difícil, formatada como uma checklist:

### ● Fáceis (comandos básicos com JOIN ou WHERE simples)

- [ X ] **SELECT com INNER JOIN:** Selecione o nome dos alunos e o nome da escola em que estão matriculados.

```
SELECT alunos.nome as Nome_aluno, escolas.nome as
Nome_escolas
FROM alunos
INNER JOIN escolas on alunos.id_escola = escolas.id_escola;
```

- [ X ] **SELECT com LEFT JOIN:** Selecione todos os alunos, incluindo aqueles que não estão matriculados em nenhuma escola.

```
SELECT alunos.nome AS nomeAluno, escolas.nome AS NomeEscola
FROM alunos
LEFT JOIN escolas ON alunos.id_escola = escolas.id_escola;
```

- [ X ] **DELETE com WHERE:** Exclua todos os alunos que não estão matriculados em nenhuma escola (sem id\_escola).

```
DELETE FROM alunos
WHERE id_escola IS NULL;

-- VERIFICA SE FOI APAGADO
SELECT * FROM alunos;
```

- [ X ] **SELECT com JOIN e ORDER BY:** Selecione o nome e a cidade dos alunos, juntamente com o nome da escola, ordenando os resultados pela cidade.

```
SELECT alunos.nome as Aluno_Nome, escolas.nome as
Escola_Nome, escolas.cidade as Cidade
```

```
FROM alunos  
INNER JOIN escolas on alunos.id_escola = escolas.id_escola  
ORDER BY Cidade;
```

- [ X ] **SELECT com JOIN e LIKE:** Selecione o nome dos alunos e o nome da escola onde o nome da escola contém a palavra "Pública".

```
SELECT alunos.nome AS Nome_Aluno, escolas.nome AS Nome_Escola  
FROM alunos  
INNER JOIN escolas ON alunos.id_escola = escolas.id_escola  
WHERE escolas.tipo LIKE 'Pública';
```

### ● Intermediárias (JOINS com filtros, agrupamentos ou atualizações)

- [ X ] **SELECT com WHERE e JOIN:** Selecione o nome dos alunos e o nome das escolas, mas só mostre os alunos que estão matriculados em escolas públicas.

```
SELECT alunos.nome AS Nome_Aluno, escolas.nome AS Nome_Escola  
FROM alunos  
JOIN escolas ON alunos.id_escola = escolas.id_escola  
WHERE escolas.tipo = 'Pública';
```

- [ X ] **SELECT com COUNT e GROUP BY:** Selecione o número de alunos matriculados em cada escola, juntamente com o nome da escola.

```
SELECT escolas.nome AS Nome_Aluno, COUNT(alunos.id_aluno) AS  
Quantidade  
FROM alunos  
INNER JOIN escolas ON alunos.id_escola = escolas.id_escola  
GROUP BY escolas.nome,alunos.id_aluno;
```

- [ X ] **UPDATE com INNER JOIN:** Atualize o nome da escola do aluno de ID 5 para "Escola Nova".

```
UPDATE escolas as e
JOIN alunos as a ON a.id_escola = e.id_escola
SET e.nome = 'Escola Nova'
WHERE a.id_aluno = 5;

SELECT a.id_aluno, a.nome AS nome_aluno, e.nome AS
nome_escola
FROM alunos a
JOIN escolas e ON a.id_escola = e.id_escola
WHERE a.id_aluno = 5;
```

- [ X ] **UPDATE com JOIN e WHERE:** Atualize o nome da escola para "Escola Melhor" do aluno com id 8.

```
UPDATE escolas e
JOIN alunos a ON e.id_escola = a.id_escola
SET e.nome = 'Escola Melhor'
WHERE a.id_aluno = 8;

SELECT * FROM escolas;
```

- [ X ] **DELETE com JOIN:** Exclua todas as escolas que não têm nenhum aluno matriculado.

```
DELETE escolas
FROM escolas
LEFT JOIN alunos A ON escolas.id_escola = A.id_escola
WHERE A.id_escola IS NULL;

SELECT * FROM escolas
```

- [ X ] **SELECT com LEFT JOIN e WHERE:** Selecione todos os alunos e o nome da escola, mas somente mostre alunos que têm um status ativo.

```
SELECT alunos.nome AS Alunos, escolas.nome AS Escolas
FROM alunos
LEFT JOIN escolas ON escolas.id_escola = alunos.id_escola
WHERE alunos.status = 'Ativo';
```

- [ X ] **UPDATE com JOIN:** Atualize o status de todos os alunos que estão matriculados em escolas inativas para "Transferido".

```
UPDATE alunos
JOIN escolas ON alunos.id_escola = escolas.id_escola
SET alunos.Status = 'Transferido'
WHERE escolas.Status = 'Inativa';

SELECT * FROM alunos
```

### ● Difíceis (com múltiplas condições, subconsultas, CASE, INSERTs dinâmicos)

- [ X ] **UPDATE com CASE:** Atualize o status dos alunos para "Graduado" se a data de matrícula for anterior a 2015. Caso contrário, mantenha o status como "Ativo".

```
UPDATE alunos
SET Status = CASE
    WHEN YEAR(data_matricula) < 2015 THEN 'Graduado'
    ELSE 'Ativo'
END;

SELECT * FROM alunos
```

- [ X ] **SELECT com múltiplos JOINS:** Selecione o nome do aluno, nome da escola e o telefone da escola. Exiba apenas alunos que têm data de nascimento

posterior a 2000.

```
SELECT alunos.nome alunos_nome, escolas.nome escola_nome,
       telefones.id_escola telefone_escola
FROM alunos
JOIN escolas ON alunos.id_escola = escolas.id_escola
JOIN telefones ON escolas.id_escola = telefones.id_escola
WHERE alunos.data_matricula > '2000-01-01';
```

- **[ X ] UPDATE com JOIN e SET múltiplos campos:** Atualize o nome de um aluno e da escola associada a ele. O nome do aluno é "Carlos" e o nome da escola é "Escola Estadual".

```
UPDATE alunos
JOIN escolas ON alunos.id_escola = escolas.id_escola
SET alunos.nome = 'Pedro teste',
    escolas.nome = 'Escolas teste'
WHERE alunos.nome = 'Carlos' AND escolas.nome = 'Escola
Pública Central';

SELECT * FROM alunos
```

- **[ X ] INSERT com SELECT e JOIN:** Crie uma nova escola e insira todos os alunos com id\_escola = NULL nessa escola recém-criada.

```
INSERT INTO escolas (nome, cidade, tipo, status)
VALUES ('Escola Nova Centralizada', 'Belo Horizonte',
       'Pública', 'Ativa');

UPDATE alunos
SET id_escola = (
    SELECT id_escola
    FROM escolas
```

```

        WHERE nome = 'Escola Nova Centralizada'
        LIMIT 1
    )
WHERE id_escola IS NULL;

SELECT alunos.nome AS nome_aluno, escolas.nome AS nome_escola
FROM alunos
JOIN escolas ON alunos.id_escola = escolas.id_escola;

```

- [ X ] **SELECT com INNER JOIN e subconsulta:** Selecione o nome dos alunos que estão matriculados em escolas que têm mais de 50 alunos.

```

SELECT alunos.nome
FROM alunos
INNER JOIN (
    SELECT id_escola
    FROM alunos
    GROUP BY id_escola
    HAVING COUNT(*) > 50
) AS escolas_com_mais_de_50
ON alunos.id_escola = escolas_com_mais_de_50.id_escola;

```

- [ X ] **DELETE com WHERE e NOT EXISTS:** Exclua todas as escolas que não têm alunos matriculados.

```

DELETE FROM escolas
WHERE NOT EXISTS (
    SELECT 1
    FROM alunos
    WHERE alunos.id_escola = escolas.id_escola
);

```

```
SELECT * FROM escolas;
```

- [ X ] **DELETE com JOIN e LIMIT:** Exclua 5 alunos que estão com status "Inativo" e não têm nenhuma escola associada a eles.

```
DELETE FROM alunos
WHERE status = 'Inativo' AND ID_Escola IS NULL
LIMIT 5;

SELECT * FROM alunos
```

- [ X ] **DELETE com JOIN, WHERE e LIMIT:** Exclua os alunos com status "Transferido" de escolas que têm menos de 5 alunos matriculados.

```
DELETE alunos
FROM alunos
JOIN (
    SELECT id_escola
    FROM alunos
    WHERE id_escola IS NOT NULL
    GROUP BY id_escola
    HAVING COUNT(*) < 5
) AS escolas_com_poucos_alunos
ON alunos.id_escola = escolas_com_poucos_alunos.id_escola
WHERE alunos.status = 'Transferido';

SELECT * FROM alunos
```