SQL Mastery: Aprenda e Pratique com Exercícios Interativos para Domínio da Linguagem SQL

MAYSA ARRUDA

Structured Query Language

1. O que é SQL?

SQL (Structured Query Language) é uma linguagem padronizada para consultas e manipulação de dados em bancos de dados relacionais. Sua principal função é permitir que os usuários interajam com o banco de dados, realizando operações como inserção, atualização, exclusão e recuperação de dados.

2. Comandos SQL Básicos:

- SELECT: Utilizado para recuperar dados de uma ou mais tabelas. Exemplo:

```
SELECT * FROM tabela;
```

- **INSERT:** Insere novos registros em uma tabela. Exemplo:

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

- **UPDATE:** Atualiza dados existentes em uma tabela. Exemplo:

```
UPDATE tabela SET colunaI = novo_valor WHERE condição;
```

- **DELETE:** Remove registros de uma tabela. Exemplo:

```
DELETE FROM tabela WHERE condição;
```

3. Cláusulas SQL:

- **WHERE:** Utilizada para filtrar os resultados de uma consulta com base em uma condição.

```
SELECT * FROM tabela WHERE condicao;
```

- **ORDER BY:** Ordena os resultados da consulta de acordo com uma ou mais colunas.

```
SELECT * FROM tabela ORDER BY coluna ASC/DESC;
```

- **GROUP BY:** Agrupa resultados com base em valores semelhantes em uma ou mais colunas.

```
SELECT coluna, COUNT(*) FROM tabela GROUP BY coluna;
```

4. Chaves Primárias e Estrangeiras:

- **Chave Primária (PRIMARY KEY):** Identifica de forma exclusiva cada registro em uma tabela.

```
CREATE TABLE tabela ( id INT PRIMARY KEY, nome VARCHAR(50) );
```

- **Chave Estrangeira (FOREIGN KEY):** Estabelece uma relação entre duas tabelas.

```
CREATE TABLE tabela_filha ( id INT PRIMARY KEY, id_tabela_pai INT, FOREIGN KEY (id_tabela_pai) REFERENCES tabela_pai(id) );
```

5. JOINs:

- **INNER JOIN:** Retorna apenas os registros que têm correspondência em ambas as tabelas.

```
SELECT * FROM tabela1 INNER JOIN tabela2 ON tabela1.coluna = tabela2.coluna;
```

- **LEFT JOIN (ou LEFT OUTER JOIN):** Retorna todos os registros da tabela à esquerda e os registros correspondentes da tabela à direita.

```
SELECT * FROM tabela1 LEFT JOIN tabela2 ON tabela1.coluna = tabela2.colun
```

50 exercícios de SQL para ajudar você a praticar

1. **Banco de Dados Simples:**
2. Crie uma tabela chamada **alunos** com as colunas **id**, **nome**, e **nota**.
3. Insira alguns registros na tabela **alunos**.
4. Atualize a nota de um aluno específico.
5. Exclua um aluno da tabela.
6. Selecione todos os alunos com notas acima de 7.
7. **Consultas Simples:**
8. Selecione todos os registros da tabela **alunos**.
9. Selecione apenas o nome e a nota dos alunos.
10. Conte quantos alunos existem na tabela.
11. Encontre a média das notas dos alunos.
12. Liste os alunos em ordem alfabética.
13. **Consultas com Condições:**
14. Selecione apenas os alunos que têm notas maiores que 5.
15. Selecione alunos que se chamam "João".
16. Liste os alunos que não têm nota (NULL).
17. Selecione alunos com notas entre 6 e 8.
18. Encontre o aluno com a nota mais alta.
19. **Ordenação e Agrupamento:**
20. Ordene os alunos por nota de forma descendente.
21. Agrupe os alunos por nota.
22. Conte quantos alunos têm cada nota.
23. Selecione o aluno com a menor nota de cada grupo.
24. Encontre a média de notas por aluno.
25. **Joins:**
26. Crie uma segunda tabela chamada **cursos** com **id_curso** e **nome_curso**.
27. Insira alguns registros na tabela **cursos**.
28. Adicione uma coluna **id_curso** à tabela **alunos**.
29. Associe alunos a cursos usando JOIN.
30. Selecione todos os alunos e os cursos que estão associados a eles.
31. **Subconsultas:**
32. Selecione alunos com notas maiores que a média.
33. Encontre o curso com o maior número de alunos.
34. Selecione alunos que não estão associados a nenhum curso.
35. Encontre alunos que têm notas superiores à média do curso deles.
36. Selecione alunos que têm a mesma nota que outro aluno específico.
37. **Alterações em Tabelas:**
38. Adicione uma nova coluna chamada **situacao** à tabela **alunos**.
39. Atualize a situação de todos os alunos com notas maiores que 7 para "Aprovado".
40. Remova a coluna **situacao** da tabela **alunos**.
41. Modifique o nome de uma coluna na tabela **cursos**.
42. Adicione uma restrição UNIQUE à coluna **nome** na tabela **alunos**.
43. **Relacionamentos e Chaves Estrangeiras:**
44. Crie uma tabela **professores** com **id_professor** e **nome_professor**.

45. Associe professores a cursos usando uma tabela de relacionamento.
46. Selecione os cursos e os professores associados a eles.
47. Atualize o nome de um professor específico.
48. Exclua um professor da tabela.
49. **Funções Agregadas:**
50. Encontre a nota mais baixa de cada curso.
51. Calcule a soma total das notas dos alunos.
52. Encontre a média de notas de cada curso.
53. Conte quantos alunos existem em cada curso.
54. Selecione os cursos com uma média de notas superior a 8.
55. **Operações Avançadas:**
56. Utilize a cláusula CASE para classificar os alunos em "Aprovado" ou "Reprovado".
57. Concatene o nome e a nota dos alunos em uma coluna.
58. Encontre os alunos que têm notas idênticas a outros alunos.
59. Utilize a função EXISTS para encontrar cursos sem alunos.
60. Utilize a cláusula HAVING para filtrar cursos com uma média de notas superior a 7.

Múltiplas escolhas

1. **Criação de Tabelas:**

a. Qual comando SQL é usado para criar uma tabela chamada produtos com colunas id_produto, nome, e preco?

1. CREATE TABLE produtos (id_produto INT, nome VARCHAR(255), preco DECIMAL(10, 2));
2. NEW TABLE produtos (id_produto INT, nome VARCHAR(255), preco DECIMAL(10, 2));
3. ADD TABLE produtos (id_produto INT, nome VARCHAR(255), preco DECIMAL(10, 2));

2. **Consultas Básicas:**

a. Qual comando SQL é usado para selecionar todos os registros da tabela clientes?

1. SELECT * FROM clientes;
2. SHOW registros FROM clientes;
3. LIST clientes;

3. **Cláusula WHERE:**

a. Qual comando SQL é usado para selecionar clientes cujo nome é "João"?

1. SELECT * FROM clientes WHERE nome = 'João';
2. SELECT clientes WHERE nome LIKE 'João';
3. GET clientes WHERE nome = 'João';

4. **Atualização de Registros:**

a. Qual comando SQL é usado para atualizar o telefone do cliente com id_cliente igual a 1?

1. UPDATE clientes SET telefone = '123-4567' WHERE id_cliente = 1;
2. MODIFY clientes SET telefone = '123-4567' WHERE id_cliente = 1;
3. ALTER clientes SET telefone = '123-4567' WHERE id_cliente = 1;

5. **Exclusão de Registros**:

a. Qual comando SQL é usado para excluir todos os clientes com e-mail igual a "cliente@gmail.com"?

1. DELETE FROM clientes WHERE email = 'cliente@gmail.com';
2. REMOVE clientes WHERE email = 'cliente@gmail.com';
3. DROP clientes WHERE email = 'cliente@gmail.com';

6. **Ordenação e Limitação de Resultados:**

a. Qual comando SQL é usado para selecionar todos os clientes ordenados pelo nome em ordem decrescente?

1. SELECT * FROM clientes ORDER BY nome DESC;

2. SELECT * FROM clientes SORT BY nome DESC;
3. SORT clientes BY nome DESC;

b. Como você limitaria o resultado da consulta para mostrar apenas os 10 primeiros clientes?
1. SELECT * FROM clientes LIMIT 10;
2. SELECT * FROM clientes TOP 10;
3. LIMIT clientes TO 10;

7. **Cláusula GROUP BY:**

a. Como você conta o número de clientes em cada cidade usando a cláusula GROUP BY?
1. SELECT cidade, COUNT(*) FROM clientes GROUP BY cidade;
2. COUNT clientes BY cidade FROM clientes;
3. GROUP clientes BY cidade COUNT;

8. **Funções Agregadas:**

a. Qual função SQL é usada para encontrar o valor mínimo em uma coluna?
1. MIN(coluna);
2. FIND_MIN(coluna);
3. GET_MIN(coluna);

b. Como você calcularia a média das idades dos clientes?
1. SELECT AVG(idade) FROM clientes;
2. CALCULATE AVG(idade) FROM clientes;
3. AVG clientes(idade);