Nome	Turma:
Anna Gabriela Monteiro da Silva	2° DT - A
Curso	**Unidade Curricular**
Técnico em Desenvolvimento de Sistemas	Banco de Dados

Questionário sobre banco de dados

As questões devem ser entregues em papel sulfite, escritas de forma legível contendo as questões e as respostas. Sinta-se à vontade para caprichar nos desenhos para exemplificar as respostas.

Comandos DDL

1. **O que são comandos DDL e qual é o seu propósito em SQL?**
   Os comandos DDL tem o propósito de definir a estrutura de um banco de dados. Eles são um subconjunto dos comandos da linguagem SQL, cujos comandos alteram, excluem e criam objetos do banco.

2. **Quais são os comandos DDL mais comuns em SQL e como eles são usados na criação e modificação de estruturas de banco de dados?**
   Os comados DDL mais comuns são CREATE, ALTER e DROP. O comando CREATE serve para a criação de objetos no banco de dados, o ALTER é utilizado para fazer alterações nos objetos criados e o DROP que exclui os objetos do banco.

3. **Explique a diferença entre os comandos CREATE e ALTER em SQL. Como eles são aplicados de maneira prática?**
   A diferença entre os comandos CREATE e ALTER é que o CREATE cria objetos no banco, enquanto o ALTER altera os objetos. Exemplos:

```
CREATE TABLE endereco ( -- Está criando uma nova tabela
        id_endereco         INT,
        CEP                 CHAR(9),
);
ALTER TABLE endereco ADD num_residencia VARCHAR(20); -- Altera a tabela endereco para adicionar um novo atributo/coluna
```

4. **Descreva o comando CREATE TABLE em detalhes. Quais são os**

principais parâmetros usados com esse comando e como eles influenciam a estrutura da tabela?

O comando CREATE TABLE serve para criar uma nova tabela no banco de dados. Seus principais paramentros são:

- Nome da tabela, que deve ser exclusivo.
- Colunas que são incluidas na tabela com um nome unico
- Tipos de dados que cada coluna pode armazenar
- Restrições que podem ser aplicadas caso seja necessário

Exemplo:

```
CREATE TABLE nome_tabela (
        nome_coluna tipo_dado restricao,
        nome_coluna tipo_dado,
        nome_coluna tipo_dado
);
```

5. **Quais são os diferentes tipos de restrições que podem ser aplicadas usando o comando CREATE TABLE? Forneça exemplos de cada tipo.**

Os diferentes tipos de restrições que podem ser aplicadas são: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, e DEFAULT. A restrição NOT NULL faz com que a coluna não receba valores nulos, a UNIQUE garante que um valor na coluna seja unico, a PRIMARY KEY identifica uma coluna de forma unica e a FOREIGN KEY indica relacionamentos entre tabelas. Já a restrição CHECK funciona como uma estrutura condicional fazendo com que uma coluna satisfaça uma restrição e o DEFAULT indica que uma coluna tem valor padrão

Exemplo:

```
CREATE TABLE pessoa(
        pk_idpessoa     INT             NOT NULL    UNIQUE,
        nome            VARCHAR(100)    NOT NULL,
        telefone        CHAR(12)        NOT NULL,
        idade           INT             NOT NULL,
        pais            CHAR(2)         DEFAULT 'BR',
        CHECK (idade >= 18)

        PRIMARY KEY(pk_idpessoa)
```

```
);
CREATE TABLE vendedor(
        pk_idvendedor       INT NOT NULL UNIQUE,
        fk _idpessoa        INT NOT NULL,

        PRIMARY KEY(pk_idvendedor),
        FOREING KEY (fk _idpessoa) REFERENCES
    pessoa(pk_idpessoa)
    );
```

6. **Explique o uso do comando DROP TABLE em SQL. Quais são os cuidados a serem tomados ao utilizar esse comando para evitar perda de dados críticos?**

   O comando DROP TABLE é usado para deletar uma tabela existente no banco de dados, apagando sua estrutura e todos os dados associados a ela. Alguns cuidados que devem ser tomados são: verificar se a tabela a ser excluida é a correta, se ela está sendo referenciada por chaves estrangeiras em outras tabelas e se foi feito o backup do banco de dados inteiro, pois em caso de uma exclusão acidental ainda haverá uma cópia.
   Exemplo de uso do comando:
   DROP TABLE nome_tabela.

7. **Como o comando ALTER TABLE é usado para modificar a estrutura de uma tabela existente? Dê exemplos práticos de modificações que podemser feitas usando esse comando.**

   O comando ALTER TABLE é usado para adicionar, excluir ou modificar colunas ou restrições de uma tabela. Exemplos:

   - Adiciona uma coluna

   ALTER TABLE nome_tabela ADD nome_coluna tipo_dado;

   - Apaga uma coluna

   ALTER TABLE nome_tabela DROP COLUMN nome_coluna;

   - Renomeia uma coluna

   ALTER TABLE nome_tabela RENAME COLUMN nome_coluna TO novo_nome_coluna;

- Altera o tipo de dado de uma coluna

ALTER TABLE nome_tabela ALTER COLUMN nome_coluna tipo_dado;

8. **Qual a diferença entre o comando ALTER TABLE e o comando UPDATE em SQL? Em que contextos específicos cada um desses comandos é mais apropriado?**

   O comando ALTER TABLE é usado para fazer alterações na estrutura de uma tabela, em suas restrições e para acomodar novos requisitos antes da inserção dos dados. Já o comando UPDATE TABLE faz alterações de dados ou de registros e modifica valores dentro das colunas. Ao utilizar o UPDATE TABLE deve-se tomar cuidado, pois se não forem especificados os registros a serem atualizados, todos serão atualizados.

   O comando ALTER TABLE é DDL enquanto o UPDATE TABLE é um comando DML.

   Exemplo de uso do UPDATE TABLE com especificação de registro:
   UPDATE vendedor

   SET nome = 'Vitor Santos'

   WHERE id_vendedor = 8;

9. **Explique o propósito do comando RENAME em SQL. Em que situações ele é útil e como ele pode ser usado para modificar a estrutura de uma tabela?**

   O comando RENAME serve para renomear objetos no banco de dados. Ele é util em situações como: manutenção do banco de dados, para deixar o objeto organizado e claro e para adequar os objetos aos requisitos de negócios sem recriar sua estrutura. Exemplo de uso:
   RENAME TABLE nome_tabela TO novo nome_tabela;

10. **Quais são os usos comuns do comando TRUNCATE TABLE em SQL? Como ele difere do comando DELETE e quais são os possíveis impactos no desempenho do banco de dados?**

    O comando TRUNCATE TABLE deleta todos os dados de uma tabela de uma vez, sem apagar sua estrutura. Ele difere do comando DELETE, pois o comando DELETE exclui apenas dados especificos (linha por linha) indicados pela instrução WHERE. A clausula TRUNCATE TABLE é mais

rápida e utiliza menos recursos e transações durante a sua execução. O TRUNCATE TABLE é um comando DDL e não pode ser desfeito, enquanto o DELETE é um comando DML e pode ser desfeito em determinados contextos.

11. **Descreva o uso do comando CREATE INDEX em SQL. Quais são os benefícios de criar índices e quais são os possíveis cenários em que eles devem ser evitados?**
    O comando CREATE INDEX cria indices em tabelas, que são utilizados para recuperar dados mais rapido no banco de dados. Eles são beneficos, pois permitem o acesso mais rapido aos dados, melhorando o desempenho de consultas. Os indices devem ser evitados em tabelas ou colunas que não serão pesquisadas constantemente e em tabelas que já possuem um numero consideravel de indices para não causar a redução do desempenho.

12. **Como o comando DROP INDEX é usado para remover índices em SQL?**

    O comando DROP INDEX é usado para apagar o índice de um tabela.
    Para utiliza-lo deve-se indicar a tabela que deseja apagar o indice e em seguida qual o indice que se deseja apagar/ remover.

    ALTER TABLE nome_tabela
    DROP INDEX nome_index

13. **Explique o conceito de restrições de chave estrangeira (foreign key constraints) e como elas são aplicadas usando o comando ALTER TABLE em SQL.**
    A FOREIGN KEY é uma restrição que faz o relacionamento entre tabelas, permitindo que uma tabela referencie a chave primaria de outra tabela. Os valores contidos na coluna referenciadora (foreign key - fk) devem estar na coluna referenciada (primary key). Deve-se utilizar o comando ALTER TABLE quando for necessario deletar ou acrescentar uma FOREIGN KEY a uma tabela já criada.

```
ALTER TABLE nome_tabela_ referenciadora
ADD CONSTRAINT nome_fk
FOREINGN KEY (coluna_referenciadora) REFERENCES nome_tabela_referenciada(coluna_referenciada);

ALTER TABLE nome_tabela_ referenciadora
DROP FOREINGN KEY nome_fk
```

14. **Discuta a importância do comando CHECK em SQL. Como ele é usado para impor restrições específicas aos valores em uma tabela?**

O comando CHECK permite restringir o número de dados aceitos que podem ser inseridos em uma coluna. Ele é importante, pois com essa limitação pode diminuir a chance de inserir dados errados e assim manter a integridade de dados da tabela.

```
CREATE TABLE produto(
    pk_idproduto    INT          NOT NULL UNIQUE PRIMARY KEY,
    preco           DEC(10,2)    NOT NULL,
    lote            INT          NOT NULL,
    descricao       VARCHAR(100) NOT NULL,

    CHECK(preco>=0)
);
```

15. **Quais são os diferentes tipos de restrições de integridade que podem ser aplicadas usando comandos DDL em SQL? Como cada tipo de restrição contribui para manter a integridade dos dados em um banco de dados?**

Os diferentes tipos de restrições que podem ser aplicadas são: NOT NULL, UNIQUE, PRIMARY KEY, FOREIGN KEY, CHECK, DEFAULT e CREATE INDEX. A restrição NOT NULL faz com que a coluna não receba valores nulos, assegurando que campos obrigatorios que podem impactar em consultas sejam preenchidos, a UNIQUE garante que um valor na coluna seja unico, impedindo a duplicação de valores, mantendo a consistencia dos dados, a PRIMARY KEY identifica uma coluna de forma unica, impedindo a duplicação de registros e a FOREIGN KEY indica

relacionamentos entre tabelas, garantindo a integridade referencial. Já a restrição CHECK funciona como uma estrutura condicional fazendo com que os valores da coluna satisfaçam uma condição, o DEFAULT indica que uma coluna tem valor padrão, evitando a inserção de valores vazios, mantendo a consistencia dos dados, e o CREATE INDEX que cria um index para criar e recuperar dados, o que auxilia na manutenção de dados.

16. **Explique como o comando CREATE VIEW é usado para criar uma visualização em SQL. Quais são os benefícios de usar visualizações e como elas podem melhorar a eficiência das consultas?**

    O comado CREATE VIEW cria uma tabela virtual em forma de consulta, onde os campos são preenchidos a partir de dados de uma ou mais tabelas reais ja criadas. A VIEW é benefica, pois sempre mostra dados atualizados, já que o mecanismo do banco de dados recria a VIEW sempre que um usuário a consulta, ela simplifica a manutenção, manipulação e analise de dados, melhora o desempenho otimizando as consultas e reduz a complexiadade das consultas que podem ser encapsulas em uma VIEW.

    ```
    CREATE VIEW nome_view AS
    SELECT nome_coluna,
    FROM nome_tabela,
    WHERE condição;
    ```

17. **Quais são as principais diferenças entre o comando CREATE e o comando CREATE OR REPLACE em SQL? Quando é apropriado usar cada um desses comandos?**

    O comando CREATE cria objetos no banco de dados como tabelas, funções e views, enquanto o comando CREATE OR REPLACE cria ou substitui objetos existentes no banco de dados como procedimentos, funções ou views. O comando CREATE só deve ser usado com a certeza de que o objeto a ser criado não existe no banco dados para não gerar erros. Caso haja incerteza quanto a existencia de um objeto no banco é indicado usar o comando CREATE OR REPLACE.

18. **Como o comando COMMENT é usado em SQL para adicionar comentários a objetos de banco de dados? Qual é a sua utilidade prática no desenvolvimento e manutenção de bancos de dados?**

    O comando COMMENT é utilizado para adicionar comentarios no banco de dados. Na prática o comando COMMENToferece informações sobre os objetos no banco de dados, ajudando na compreensão do código, na manutenção, uma vez que estes dados podem dar informações para correção ou identificação de erros.

    Cometarios de linha unica contém “--“, enquanto os de multiplas linhas começam com “/*” e terminam com “*/”.

    -- Texto

    /*Texto*/

19. **Explique o uso do comando CREATE SCHEMA em SQL. Quais são as vantagens de agrupar objetos de banco de dados em um único esquema?**

    O comado CREATE SCHEMA cria um esquema no banco de dados. Um esquema é uma coleção de objetos de banco de dados que estão relacionados entre si e que podem ser agrupados para facilitar a organização e a manutenção do banco de dados. No MySQL, CREATE SCHEMA é sinônimo de CREATE DATABASE.

20. **Qual é a importância do comando GRANT em SQL e como ele é usado para conceder permissões a usuários específicos em um banco de dados? Como o comando GRANT contribui para a segurança e o controle de acesso aos dados?**

    O comando GRANT concede privilégios específicos para usuarios a objetos no banco de dados. Ele é importante, pois auxilia na segurança e controle de acesso aos dados, como o controle do que os usuarios podem fazer banco de dados, garantindo que somente pessoas autorizadas acessem ou manipulem os dados.

    GRANT tipo_de_permissao ON nome_do_banco_de_dados.nome_da_tabela TO 'nome_do_usuario'@'host';

Comandos DML

21. **Explique o que são os comandos DML no SQL e forneça exemplos de pelo menos dois comandos DML comumente usados.**

    Os comandos DML são comando de manipulação de dados, ou seja eles interagem ou modificam os dados das tabelas. Dois exemplos de comandos DML são o INSERT e o UPDATE. O comando INSERT insere dados em uma ou mais tabelas de um banco de dados, enquanto o comando UPDATE atualiza os dados de uma ou mais tabelas.

    INSERT INTO nome_tabela (linha1, linha2) VALUES (dado1, dado2)

    UPDATE nome_tabela SET linha = 'novo nome' WHERE condição

22. **Como você insere dados em uma tabela no SQL? Forneça um exemplo prático de um comando de inserção (INSERT).**

    O comando INSERT insere dados em uma ou mais tabelas de um banco de dados. Para utiliza-lo deve-se fornecer o nome da tabela, nome das colunas e os valores que serão inseridos.

    INSERT INTO usuario (idUsuario, nome, idade) VALUES (1, 'Maria Vitoria Silva', 40)

23. **Explique como você atualiza os dados existentes em uma tabela usando o SQL. Forneça um exemplo prático de um comando de atualização (UPDATE).**

    O comando UPDATE atualiza os dados de uma ou mais tabelas. Para utiliza-lo deve-se fornecer o nome da tabela, nome do objeto que se deseja atualizar e uma condição contenha um identificador deste objeto.

    UPDATE usuario SET idade = 55 WHERE idUsuario = 1;

24. **Como você exclui registros específicos de uma tabela no SQL? Forneça um exemplo prático de um comando de exclusão (DELETE).**

    O comando DELETE exclui dados de uma ou mais tabelas. Para utiliza-lo deve-se fornecer o nome da tabela e uma condição contenha um identificador deste objeto.

    DELETE FROM usuario WHERE idUsuario = 1;

25. **Como você recupera dados específicos de uma tabela no SQL? Forneça um exemplo prático de um comando de recuperação (SELECT).**

    O SELECT é um comando utilizado para selecionar dados. Para utiliza-lo deve-se fornecer o nome da tabela e as linhas que deseja vizualizar.

    SELECT nome FROM usuario;

26. **Explique como você pode usar a cláusula WHERE no SQL para filtrar resultados específicos em uma consulta SELECT. Forneça um exemplo prático.**

    A clausula WHERE extrai apenas registros que atendam a uma determinada condição.

    SELECT nome FROM usuario WHERE idUsuario<10

27. **Qual é a diferença entre as cláusulas WHERE e HAVING no SQL? Forneça exemplos para ilustrar essa diferença.**

    A clausula WHERE extrai apenas registros que atendam a uma determinada condição e é utilizada antes do agrupamento (GROUP BY) filtrando todas as linhas antes dele. Já a clausula HAVING filtra as consultas que depois que o agrupamento é realizado.

    SELECT nome FROM usuario WHERE idUsuario<10

    SELECT

        nome,

        COUNT(idUsuario) AS Total

    FROM usuario

    GROUP BY nome

    HAVING COUNT(idUsuario) <20

28. **Como você ordena os resultados de uma consulta SQL em ordem ascendente e descendente? Forneça exemplos práticos.**

    Para ordenar os resultados de uma consulta SQL usa-se o ORDER BY, que classica os resultados em ordem crescente por padrão. Para ordenar em ordem decrescente acrescenta-se o DESC e para ordenar em ordem crescente também pode-se usar o ASC.

    SELECT coluna FROM tabela ORDER BY coluna;

    SELECT coluna FROM tabela ORDER BY coluna ASC;

SELECT coluna FROM tabela ORDER BY coluna DESC;

SELECT * FROM fabricacao ORDER BY produto DESC, maquina ASC;

29. **Explique como você pode usar a cláusula LIKE no SQL para realizar consultas que correspondam a padrões específicos em uma coluna. Forneça exemplos práticos.**

    LIKE é a clausula usada para procurar um padrão especificado em uma coluna e que utiliza caracteres curingas que serão associados ao termo pesquisado. O caracter “%” representa zero, um ou muitos caracteres antes ou depois do termo pesquisado, enquanto o “_” representa um único caracter.

    SELECT colunas FROM nome_tabela WHERE nome_coluna LIKE modelo;

    SELECT * FROM usuario WHERE nome LIKE ‘%a%’;

    SELECT * FROM usuario WHERE nome LIKE ‘Mar_’;

30. **Qual é a finalidade da cláusula LIMIT no SQL e como ela é usada em consultas SELECT? Forneça exemplos práticos.**

    A clausula LIMIT é usada para especificar o número de registros de retorno, uma vez que sem ela pode-se afetar o desempenho das consultas em tabelas com grande número de resgistros

    SELECT colunas FROM nome_tabela WHERE condição LIMIT numero_regitros;

    SELECT * FROM usuario WHERE pais = ’BR’ LIMIT 3;

31. **Como você pode usar a cláusula ORDER BY e a cláusula GROUP BY em conjunto para classificar e agrupar os resultados de uma consulta SQL? Forneça exemplos práticos.**

    A cláusula GROUP BY faz com que seja encontrada as caracteristicas de grupos de linhas, contudo agrupar estas linhas não significa ordenar. Para ordenar as linhas deve-se usar a cláusula ORDER BY que classica os resultados em ordem crescente por padrão.

    SELECT coluna
    FROM tabela
    GROUP BY nome_coluna
    ORDER BY coluna;

```
SELECT COUNT(idUsuario), pais
FROM usuario
GROUP BY pais
ORDER BY COUNT(idUsuario) DESC;
```

32. Explique o conceito de transações no SQL. Por que as transações são importantes e como você pode usar o COMMIT e o ROLLBACK para controlar transações?

Uma transação é um conjunto de declarações combinadas em uma única unidade de trabalho, o que faz com que sejam executadas como se fossem uma unica operação. As transações são importantes, pois elas mantem o banco de dados consitente em caso de falhas, permite uma recuperação correta e aplica as propriedades de ACID. Quando uma transação é efetuada com sucesso e o banco de dados é alterado com os dados da transação salvos, acontece um COMMIT. O ROLLBACK ocorre caso haja falha em qualquer operação das transações e o banco retorna ao estado em que se encontrava antes da execução da transação.

```
START TRANSACTION;
UPDATE usuario SET idade = 55 WHERE idUsuario = 1;
COMMIT | ROLLBACK;
```

33. Como você pode usar a cláusula INNER JOIN para combinar dados de duas tabelas relacionadas no SQL? Forneça um exemplo prático.

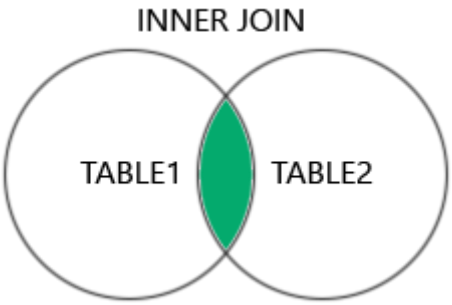
A cláusula INNER JOIN retorna apenas linhas com correspondecia entre tabelas pode ser usada da seguinte forma:

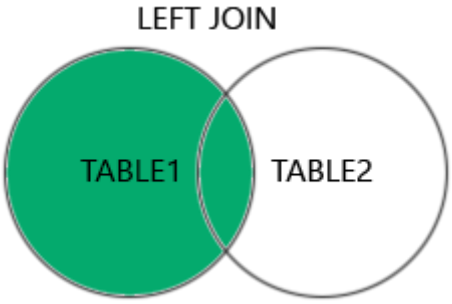
```
SELECT colunas
FROM tabela1
INNER JOIN tabela2 ON  tabela1.nome_coluna = tabela2.nome_coluna;
```

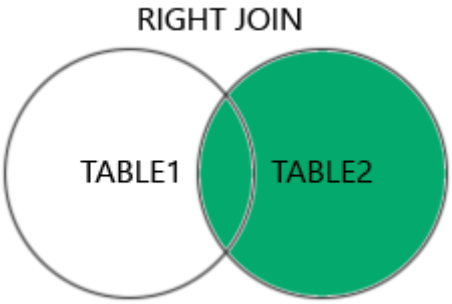
34. Explique a diferença entre as junções INNER JOIN, LEFT JOIN e RIGHT JOIN no SQL. Forneça exemplos práticos para ilustrar cada tipo de junção.

A cláusula INNER JOIN retorna apenas linhas com correspondecia entre ambas as tabelas, o LEFT JOIN retorna todos os registros da tabela

esquerda e os correspondentes da tabela direita, enquanto o RIGHT JOIN retorna todos os registros da tabela direita e os correspondentes da tabela esquerda.







- Exemplo de INNER JOIN

```
SELECT p. pk_idpessoa,
FROM pessoa p
INNER JOIN vendedor v ON p. pk_idpessoa = v.fk_idpessoa;
```

- Exemplo de LEFT JOIN

```
SELECT p. pk_idpessoa,
FROM pessoa p
LEFT JOIN vendedor v ON p. pk_idpessoa = v.fk_idpessoa;
```

- RIGHT JOIN

```
SELECT p. pk_idpessoa,
```

FROM pessoa p

RIGHT JOIN vendedor v ON p. pk_idpessoa = v.fk_idpessoa

35. **Como você pode usar a cláusula UNION no SQL para combinar os resultados de duas ou mais consultas em uma única tabela de resultados? Forneça exemplos práticos.**

A cláusula UNION combina o conjunto de duas ou mais SELECT, que devem ter o mesmo número e ordem de colunas, com tipos de dados semelhantes. Esta clausula permite apenas valores distintos, pode ser usada como UNION ALL para permitir os valores duplicados. A clausula também pode ser usada com WHERE que extrai apenas registros que atendam a uma determinada condição e com o ORDER BY que ordena os registros em ordem crescente.

- Usando UNION, WHERE e ORDER BY

```
SELECT coluna FROM tabela1
WHERE condicao
UNION
SELECT coluna FROM tabela2
WHERE condicao
ORDER BY coluna;
```

- Usando o UNION

```
SELECT coluna FROM tabela1 UNION SELECT coluna FROM tabela2
```

- Usando o UNION ALL

```
SELECT coluna FROM tabela1
UNION ALL
SELECT coluna FROM tabela2
```

36. **Explique a diferença entre as operações de INSERT e UPDATE no SQL. Em que situações você usaria uma em vez da outra?**

O comando INSERT insere dados em uma ou mais tabelas de um banco de dados, enquanto o comando UPDATE atualiza os dados de uma ou mais tabelas. O INSERT é util quando se quer adicionar dados ao banco

de dados e o UPDATE quando já se tem estes dados mas deseja modifica-los.

37. **Como você pode usar a cláusula DISTINCT no SQL para remover valores duplicados de uma consulta SELECT? Forneça um exemplo prático.**

    A clausula DISTINCT é usada para retornar valores diferentes.

    SELECT DISTINCT colunas FROM tabela;
    SELECT DISTICT pais FROM usuario;

38. **Qual é a diferença entre os operadores AND, OR e NOT no SQL? Forneça exemplos práticos de como esses operadores são utilizados em consultas.**

    A clausula condicional WHERE pode conter um ou vários operadores. O operador AND é usado para filtrar registros com base em mais de uma condição, onde todas devem ser verdadeiras, enquanto o OR também filtra registros com base em mais de uma condição, porém apenas uma condição deve ser verdadeira. Já o NOT é usado em combinação com outros operadores para fornecer um resultado oposto ou negativo.

    SELECT colunas
    FROM tabela
    WHERE condição1 AND condição2;

    SELECT colunas
    FROM tabela
    WHERE condição1 OR condição2;

    SELECT colunas
    FROM tabela
    WHERE NOT condição;

39. **Explique como você pode usar as funções de agregação, como COUNT,SUM, AVG, MAX e MIN, para realizar cálculos em conjuntos de dados noSQL. Forneça exemplos práticos de cada função.**

A função COUNT retorna o número de linhas correspondente a critério especificado, a SUM retorna soma de uma coluna numérica e o AVG retorna a média de uma coluna numérica. Já a função MIN retorna o menor valor da coluna selecionada, enquanto o MAX retorna o maior valor.

```
SELECT COUNT(nome_coluna)
FROM nome_tabela
WHERE condição;
```

```
SELECT SUM(nome_coluna)
FROM nome_tabela
WHERE condição;
```

```
SELECT AVG(nome_coluna)
FROM nome_tabela
WHERE condição;
```

```
SELECT MIN(nome_coluna)
FROM nome_tabela
WHERE condição;
```

```
SELECT MAX(nome_coluna)
FROM nome_tabela
WHERE condição
```

40. **Como você pode usar as subconsultas (subqueries) no SQL para realizar consultas mais complexas ou consultas que dependem dos resultados de outra consulta? Forneça exemplos práticos para ilustrar o uso de subconsulta.**

    Uma subconsulta (consulta interna) é uma consulta dentro de outra consulta (consulta externa), que passa os resultados para a consulta externa por meio das clausulas WHERE ou HAVING. Desta forma os dados retornados pela consulta tem filtros bem aprimorados. As subconsultas podem ser utilizadas juntos com as clausulas SELECT, UPDATE, INSERT e DELETE. Em subconsultas não é possivel usar a clausula ORDER BY e nem o operador BETWEEN.

```
SELECT coluna
FROM tabela
WHERE coluna operador (
                        SELECT coluna
                        FROM tabela
                        WHERE condição);
SELECT nome_livro, preco_livro, IdEditora
FROM livro
WHERE IdEditora = (
                        SELECT IdEditora
                        FROM editora
                        WHERE nomeEditora = ‘Intríseca’);
```