**GUSTAVO HENRIQUE PEREIRA SILVA**

Turma: Back-end com JS (seg e qua - Noite)

1. Pesquisar sobre tipos de dados em SQL

**TIPOS DE DADOS EM SQL:**

1. **Tipos Numéricos**

**1.1 TINYINT : Armazena numeros inteiros de tamanho minúsculo.**

**1.2 SMALLINT : Amarzena números inteiros de tamanho pequeno.**

**1.3 MEDIUMINT : Armazena números inteiros de tamanho médio.**

**1.4 INT : Armazena números inteiros.**

**1.5 BIGINT : Armazena núemros inteiros grandes.**

1. **Decimais e Ponto Flutuante**

**2.1 FLOAT : Armazena números com ponto flutuante de precisão simples**

**2.2 DOUBLE : Armazena números com ponto flutante de precisão dupla.**

**2.3 DECIMAL : Armazena números com ponto flutuante de precisão fixa, (definição de precisão específica)**

1. **Tipos de Dados strings**

**3.1 CHAR : Armazena uma cadeia de caracteres com quantidade fixa.**

**3.2 VARCHAR : Armazena uma cadeia de caracteres com quantidade variável.**

**3.3 TEXT : Armazena textos longos.**

**3.4 TINYTEXT : Armazena textos pequenos.**

**3.5 MEDIUMTEXT : Armazena textos médios.**

**3.6 LONGTEXT : Armazena textos extremamente longos.**

1. **Tipos de Dados de Data e Hora**

**4.1 DATE : Armazena a hora no formato inglês(**YYYY-MM-DD**).**

**4.2 TIME : Armazena a hora (**HH:MM**)**

**4.3 DATETIME : Armazena data e hora juntos.**

**4.4 TIMESTAMP : Armazena data e hora, registrando a hora da última atualização .**

**4.5 YEAR : Armazena um ano (**YYYY**)**

1. **Tipos de Dados Binários**

**5.1 BINARY : Armazena dados binarios fixos.**

**5.2 VARBINARY : Armazena dados binários de tamanho variável.**

**5.3 BLOB : Armazena grande quantidade de dados binários.**

**5.4 TINYBLOB: Armazena pequenos blobs de dados binários.**

**5.5 MEDIUMBLOB: Armazena blobs de tamanho médio.**

**5.6 LONGBLOB: Armazena grandes blobs de dados binários.**

1. **Tipos de Dados Booleanos**

**6.1 BOOLEAN : Armazena TRUE ou FALSE.**

1. **Tipos de Dados de Enumerados e Conjuntos**

**7.1 ENUM : Armazena um valor em uma lista de valores ( ENUM(‘picoleAgua, aoLeite’))**

**7.2 SET : Armazena zero ou mais valores de uma lsita de valores permitidos.**

2. Quais são as diferenças entre INNER JOIN, LEFT JOIN,

RIGHT JOIN e FULL OUTER JOIN?

1. *Inner Join :* Retorna apenas as linhas que têm correspondências em ambas as tabelas.
2. *Left Join* ***:*** Retorna todas as linhas da tabela à esquerda e as linhas correspondentes da tabela à direita. Se não houver correspondência na tabela à direita, retorna nulo para essas colunas.
3. Right Join : Retorna todas as linhas da tabela à direita e as linhas correspondentes da tabela à esquerda. Se não houver correspondência na tabela à esquerda, retorna nulo para essas colunas.
4. Full outer join : Retorna todas as linhas de ambas as tabelas. Se não houver correspondência em uma das tabelas, retorna nulo para as colunas da tabela que não tem correspondência.

3. Como inserir, atualizar e deletar registros em uma tabela

SQL?

**Resposta:**

1. INSERIR : **INSERT INTO** nome da tabela (atributos da tabela) **VALUES** (digitar a informação referente a cada atributo da tabela );
2. ATUALIZAR : **UPDATE** nome da tabela **SET** coluna= valor que será atribuido a coluna **WHERE** coluna = valor
3. DELETAR: **DELETE FROM** nome da tabela **WHERE** coluna= valor

4. Quais são as boas práticas para manipulação de grandes

volumes de dados?

* Escolha da ferramenta correta para o tipo de dado.
* Otimização de consultas.
* Armazenamento eficiente.
* Processamento distribuído.
* Monitoramento e logging.
* Segurança e privacidade dos dados.
* Utilização de armazenamento em nuvem.
* Backup regular e recuperação de dados.
* Utilização de ferramentas de ETL.
* Gerenciamento de metadados.
* Utilização de cache para consultas frequentes.
* Utilização de ferramentas de visualização e análise.

5. O que é uma estrutura condicional if em JavaScript?

**Resposta :**

É uma estrutura condicional que é acionada quando a expressão da condição é verdadeira.(TRUE)

**Exemplo:**

let idade = 18;

If(idade >=18){

console.log(“É maior de idade”)

}

6. Como funciona a cláusula else em uma estrutura condicional?

**Resposta:**

O bloco do **else**(falso) é executado quando a bloco if() não é verdadeiro

7. Qual é a sintaxe básica de um if e else em JavaScript?

**Exemplo:**

If(**true**){  
 console.log(“Bloco verdadeiro ativo”);

}else{

Console.log(“ Bloco Falso ativo.”);

8. O que faz a seguinte estrutura if em JavaScript?

A black background with green and white text

Description automatically generated

**Resposta:** Se a variável x for maior que 10 o bloco if é acionado exibindo no console

“x é maior que 10”

9. Como você escreveria uma estrutura if-else que

verifica se uma variável idade é maior ou igual a 18 e

imprime "Adulto" ou "Menor de idade" dependendo do

caso?

**Resposta:**

Let idade = 18;

If(idade >=18){  
 console.log(“ Adulto”);

}else{

console.log(“Menor de idade”);

10. O que é uma função em JavaScript e por que elas são

úteis?

**Resposta :**

Funções em javascript são blocos de códigos que podem ser chamados quando necessário com o propósito de reutilização diminuindo a verbosidade e redundancia de algoritmos no código , as funções podem retornar ou não um valor além de receberem parâmetros que são referencias para as expressões utilizadas dentro do escopo do bloco de código da função.

11. Como você define uma função em JavaScript?

**Resposta:**

Utilizando a palavra reservada function seguido do nome da função e parênteses(com parâmetros ou não) e por ultimo as chaves.

**Exemplo:**

function apresentar(nome){

console.log(`Olá me chamo ${nome} prazer em conhecer-lo.`);

}

**apresentar(“Gustavo”);**

Saida: **Olá me chamo Gustavo prazer em conhercer-lo.**

12. Dê um exemplo de uma função vazia que imprime uma

mensagem no console. Explique o que você fez.

**Resposta:**

function dizerOi(){

console.log(`Oi sou uma função`);

}

dizerOi();

No código acima criei uma função que não recebe nenhum parâmetro e em seguida chamei a função para ela imprimir a mensagem .

13. Como você define uma função que aceita parâmetros

em JavaScript?

**Resposta:**

Uma função que aceita parâmetros é utilizada para fazermos diferentes expressões no nosso algoritmo , onde definimos uma sequência lógica se baseando na captura dos parâmetros para fazermos algo com eles, utilizamos isto com funções para reaproveitarmos o código.

Exemplo :

function calcular(numero1,numero2,operacao){

If(operacao === “somar”){  
 return numero1+ numero2;

}else if (operacao === “subtrair”){

return numero1 - numero2;

} else if (operacao === “multiplicar”){

return numero1 \* numero2;

} else if (operacao === “dividir”){

return numero1/numero2

}

}

let resultSoma = calcular(2,2,”somar”);

let resultSub = calcular(10,2,”subtrair”);

let resultMulti = calcular(2,2,”multiplicar”);

let resultDiv = calcular(10,2,”dividir);

**OBS**: Só exibir tudo no console.log.

14. Dê um exemplo de uma função que aceita dois

parâmetros e retorna a soma deles.

**Resposta:**

function soma(num1,num2){

return `a soma de $(num1) + ${num2} é ${(num1+num2)}`

}

console.log(soma(2,2,)) => saída : A soma de 2 + 2 é 4

15. Como você pode fazer uma função retornar um valor

em JavaScript?

**Resposta:** Utilizando a palavar reservada return seguido do valor.

Exemplo:

function retornar (numero){

return numero;

}

let numero = retornar(5);

console.log(numero);

16. Para dobrar o número 5 usando a função

dobrarNumero e armazenar o resultado em uma variável

chamada resultado, você chamaria a função de que

maneira?

**Resposta:**

function dobrarNumero(numero){

return numero\*2;

}

let resultado = dobrarNumero(5);

console.log(resultado);