Curso: ADS – Análise e Desenvolvimento de Sistema

Ano atualização: 2023/1

Nome da Trilha: OT4 – Linguagem SQL: Funções de Agregação, operadores

lógicos e Cálculos.

FICHAMENTO:

Nesta OT abordaremos sobre Funções de Agregação e operadores lógicos e funções de cálculos, disponibilizados pela linguagem SQL . Para isso você deverá fazer uma pesquisa sobre as principais funções, seus objetivos, trazendo exemplos para cada função. Lembrando que na criação do seu fichamento será necessário obedecer às normas ABNT.

1.Funções

De acordo com W3 SCHOOLS (2024), funções agregadoras são responsáveis por realizar operações em um conjunto de valores em vez de valores individuais, sendo geralmente usadas em conjunto com a cláusula GROUP BY.

1. SUM

Calcula a soma de todos os valores em uma coluna específica. É útil para obter totais, como somar valores em uma coluna de preços.

Exemplo:

SELECT SUM(preco) FROM vendas;

2. MIN

Retorna o menor valor em uma coluna. É útil para encontrar o valor mínimo, como o menor preço ou data mais antiga.

Exemplo:

SELECT MIN(preco) FROM vendas;

3. MAX

Retorna o maior valor em uma coluna. É útil para localizar o valor máximo, como o preço mais alto de um produto.

Exemplo:

SELECT MAX(preco) FROM vendas;

4. AVG

Calcula a média dos valores em uma coluna. Ideal para descobrir médias de valores, como o preço médio de produtos.

Exemplo:

SELECT AVG(preco) FROM vendas;

5. COUNT

Conta o número de linhas que correspondem a uma condição. Pode contar todas as linhas ou apenas aquelas com valores específicos.

Exemplo:

SELECT COUNT(\*) FROM vendas;

2.Utilização das cláusulas/operadores:

1. BETWEEN

Filtra dados entre dois valores, incluindo os valores limites. Funciona com números, datas e até strings

Exemplo:

SELECT \* FROM vendas WHERE preco BETWEEN 10 AND 50;

2. GROUP BY

Agrupa linhas com valores iguais em colunas especificadas. Frequentemente usado com funções agregadoras para realizar cálculos em grupos.

Exemplo:

SELECT categoria, SUM(preco) FROM produtos GROUP BY categoria;

3. DISTINCT

Retorna apenas valores únicos em uma coluna, eliminando duplicatas. Útil para ver todos os valores únicos em uma coluna.

Exemplo:

SELECT DISTINCT categoria FROM produtos;

4. ORDER BY

Ordena os resultados em ordem crescente (ASC) ou decrescente (DESC) com base em uma ou mais colunas.

Exemplo:

SELECT \* FROM produtos ORDER BY preco DESC;

5. LIKE

Filtra valores com base em um padrão específico. Usado com curingas (% para vários caracteres e \_ para um único caractere).

Exemplo:

SELECT \* FROM produtos WHERE descricao LIKE 'Escova%';

6. ALIAS

Cria nomes temporários para colunas ou tabelas, facilitando a leitura e organização dos resultados.

Exemplo:

SELECT preco AS 'Preço do Produto' FROM produtos;

7. OPERADORES DE COMPARAÇÃO

Compara valores para filtrar dados. Inclui =, <> (ou !=), >, <, >=, e <=.

Exemplo:

SELECT \* FROM produtos WHERE preco >= 20;

8. OPERADORES LÓGICOS

O autor define que operadores lógicos combinam condições para filtrar dados mais complexos.

1. AND

Ambas as condições precisam ser verdadeiras.

Exemplo:

SELECT \* FROM produtos WHERE preco > 10 AND categoria = 'Higiene';

2. OR

Pelo menos uma das condições deve ser verdadeira.

Exemplo:

SELECT \* FROM produtos WHERE preco < 10 OR categoria = 'Higiene';

3. NOT

Inverte o resultado da condição. Seleciona registros que não correspondem a uma condição.

Exemplo:

SELECT \* FROM produtos WHERE NOT categoria = 'Higiene';

**REFERÊNCIAS**

W3 SCHOOLS. **SQL SUM() Function**. 2024. Disponível em: https://www.w3schools.com/sql/sql\_sum.asp. Acesso em: 13 nov. 2024.