**Para saber mais: filtros**

Na linguagem SQL, filtros são usados para recuperar dados específicos de uma tabela. Eles funcionam como critérios de pesquisa que limitam os resultados de uma consulta. Filtros permitem que você especifique condições que os dados devem atender para serem incluídos no resultado da consulta. Então, vamos conhecer alguns conceitos-chave relacionados aos filtros em SQL:

**Cláusula WHERE**: A maioria dos filtros é especificada na cláusula WHERE de uma consulta SELECT. Por exemplo, a consulta a seguir retorna todos os clientes com idade superior a 30 anos:

**SELECT** \* **FROM** Clientes **WHERE** Idade > 30;

**Operadores de Comparação**: Para criar filtros, você usa operadores de comparação, como >, <, =, != (diferente de), >=, <=, etc. Esses operadores comparam valores em colunas de tabelas com valores específicos.

Além dos operadores de comparação, temos diversos outros que também podem ser especificados na cláusula WHERE e aplicar filtros.

Operadores Lógicos: Você pode combinar várias condições usando operadores lógicos, como AND, OR e NOT. Isso permite criar filtros complexos. Por exemplo, recuperar clientes com idade superior a 30 e que não sejam do sexo masculino:

**SELECT** \* **FROM** Clientes **WHERE** Idade > 30 **AND** Sexo <> 'Masculino';

Filtros de Texto: Para strings, você pode usar operadores de comparação ou funções como LIKE para filtrar registros com base em padrões de texto. Por exemplo, recuperar produtos que contenham a palavra "computador" no nome:

**SELECT** \* **FROM** Produtos **WHERE** Nome **LIKE** 'computador';

Filtros de Data: Para datas, você pode usar operadores como =, >, <, etc., para filtrar registros com base em datas. Por exemplo, recuperar pedidos feitos após uma data específica:

**SELECT** \* **FROM** Pedidos **WHERE** DataPedido > '2023-01-01';

Em resumo, filtros são fundamentais em SQL, permitindo que você controle quais registros são retornados em suas consultas. Eles são usados para encontrar dados específicos em grandes conjuntos de informações, tornando a SQL uma linguagem poderosa para recuperar informações de bancos de dados.

**Para saber mais: renomeando campos e tabelas**

Na linguagem SQL, um "ALIAS" é um nome temporário atribuído a uma tabela ou coluna em uma consulta. Isso é útil para tornar os resultados das consultas mais legíveis ou para renomear tabelas e colunas temporariamente. Os ALIAS são usados com a palavra-chave "AS".

Você pode usar ALIAS para renomear colunas em resultados de consulta.

Por exemplo: suponha que tenhamos uma tabela "Clientes" com as colunas "Nome" e "Sobrenome," e queremos apresentar o nome completo como uma coluna renomeada nos resultados:

**SELECT** Nome **AS** NomeCompleto **FROM** Clientes;

Neste exemplo, a coluna "Nome" é renomeada como "NomeCompleto" nos resultados da consulta.

ALIAS também pode ser usado para renomear tabelas em consultas, especialmente quando se realizam junções de tabelas.

Por exemplo: suponha que estamos unindo as tabelas "Pedidos" e "Clientes," e queremos evitar ambiguidades nos nomes das colunas "ID" de ambas as tabelas:

**SELECT** P.ID **AS** IDPedido, C.ID **AS** IDCliente

**FROM** Pedidos **AS** P

**JOIN** Clientes **AS** C **ON** P.ClienteID = C.ID;

Neste caso, usamos ALIAS para renomear as tabelas "Pedidos" e "Clientes" como "P" e "C," respectivamente.

Os ALIAS são uma ferramenta útil para tornar as consultas mais legíveis e evitar conflitos de nome em consultas complexas. Eles são amplamente usados em SQL para melhorar a clareza dos resultados e facilitar a compreensão do significado dos dados.