

Banco de dados

Consultas SQL – Parte 1

Prof. Eldane Vieira

Introdução

- A DQL (Data Query Language) é a parte do SQL que trata das consultas e muitas vezes está embutida na DML.
- O resultado de uma consulta na linguagem SQL é uma tabela, que pode ter uma ou mais linhas, com um ou mais atributos.

Comando SELECT

- A sintaxe básica do comando SELECT é a seguinte:
 - SELECT <nome dos campos>
FROM <nome da tabela>
[WHERE <condições>];

Esquema de tabelas dos exemplos

- Os exemplos nesta aula serão feitos sobre os seguintes esquemas de tabelas:
 - Cliente (ID_Cliente, nome, endereco, cidade, telefone, tipo)
 - Empresa (ID_Empresa, CNPJ, nome, endereco, cidade, telefone)
 - Cliente_Empresa (ID_Empresa, ID_Cliente)
 - Produto (ID_Produto, nome, qnt_vendida, valor, total_produzido)
 - Produto_Empresa(ID_Produto,ID_Empresa)
 - Produto_Comprado_Cliente(ID_Produto,ID_Cliente, quantidade)
- Um script para criação e inserção de dados na tabela está disponível no disco virtual com o nome BaseConsultas.
- Mas lembre que para elaborar a consulta basta conhecer o esquema das tabelas.

Comando SELECT

- Consulta 1: Listar todos os dados da tabela Cliente.
 - SELECT *
 - FROM Cliente;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

Comando SELECT

- Consulta 2: Listar o endereço e o telefone de todas as empresas presentes na tabela Empresa.
 - SELECT endereco, telefone
FROM Empresa;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

Comando SELECT

- Consulta 3: Listar o código das empresas com os respectivos códigos dos clientes.
 - SELECT ID_Empresa, ID_Cliente
FROM Cliente_Empresa;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

AS

- Podemos nomear as colunas das tabelas que apresentam o resultado de uma consulta, de tal modo que faça mais sentido quando for visualizado o resultado.
- Utiliza-se o comando AS para renomear campos (atributos) retornados na busca.
 - `SELECT <nome do atributo> AS <nome fantasia>`
`FROM <nome da tabela>;`
- O comando AS só renomeia o nome do campo na tabela que apresenta o resultado da consulta, e não na estrutura da tabela presente no banco.

AS

- Consulta 4: Lista o código das empresas com os respectivos códigos dos clientes, renomeando o nome das colunas.
 - SELECT ID_Empresa AS Empresa, ID_Cliente AS Cliente
FROM Cliente_Empresa;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

LIMIT

- Usado para limitar a quantidade de registros apresentados em uma consulta.
 - SELECT <nome do atributo>
FROM <nome da tabela>
LIMIT <número de registros a ser mostrado>;

LIMIT

- Consulta 5: Apresenta todos os dados de 3 empresas.
 - SELECT *
FROM Empresa
LIMIT 3;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

DISTINCT

- Dependendo da consulta, seu resultado pode trazer informações repetidas, para evitar isso, utilizamos a cláusula DISTINCT que permite apresentar somente dados diferentes entre si.
 - `SELECT DISTINCT <nome do atributo>`
`FROM <nome da tabela>;`

DISTINCT

- Consulta 6: Selecionar as cidade dos clientes.
 - SELECT DISTINCT cidade
FROM Cliente;

Se existir mais de um cliente em uma mesma cidade, essa cidade vai aparecer uma única vez, e não uma vez para cada cliente.

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

ORDER BY

- O recurso ORDER BY serve para definir qual atributo será usado para estabelecer a ordem em que os dados serão apresentados.
 - SELECT <nome dos atributos>
FROM <nome da tabela>
ORDER BY <nome do atributo>;

ORDER BY

- A ordenação pode ser feita de maneira crescente ou decrescente, para isto utilizamos, respectivamente, os recursos ASC e DESC.
- A ordenação crescente é a forma de apresentação padrão dos dados caso não seja definido o contrário.
 - SELECT <nome dos atributos>
FROM <nome da tabela>
ORDER BY <nome do atributo> [ASC] [DESC];

ORDER BY

- Consulta 7: Consultar o nome, endereço e telefone dos clientes, sendo ordenados pelo nome de maneira inversa a ordem alfabética.
 - SELECT nome, endereco, telefone
FROM Cliente
ORDER BY nome DESC;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

Cálculos sobre consultas

- Para melhor visualizar os dados podemos alterá-los para serem apresentados de uma maneira que atenda melhor as nossas necessidades.
- É possível realizar, por exemplo, operações de aritmética básica, soma (+), subtração (-), multiplicação (*) e divisão (/), com atributos cujos dados são numéricos.

Cálculos sobre consultas

- Consulta 8: Será apresentado na consulta o nome dos produtos com a quantidade que resta dos mesmos no estoque.
 - SELECT nome AS Produto, (total_produzido – qnt_vendida) AS QuantidadeDisponível
FROM Produto;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

Where

- No comando SELECT podemos utilizar a cláusula WHERE, que é opcional e só faz parte do comando quando queremos estabelecer condições para a consulta, que funcionarão como um filtro.

Operadores	Descrição
=	Igual a
>	Maior que
<	Menor que
>=	Maior ou igual a
<=	Menor ou igual a
<>	Diferente de

WHERE

- A sintaxe do SELECT com a cláusula WHERE é a seguinte:
 - SELECT <nome dos atributos>
FROM <nome da tabela>
[WHERE <condição>];
- Consulta 9: Selecionar os clientes de Uberlândia.
 - SELECT nome AS Cliente
FROM Cliente
WHERE cidade = “Uberlândia”;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

LIKE

- Quando estamos comparando cadeias de caracteres temos a opção de utilizar o LIKE.
- Neste caso o LIKE não é usado com dados numéricos como int ou float. Para dados numéricos o '=' é o usado.
- Contudo, as cadeia de caracteres onde é usado o LIKE podem conter números, espaços em branco e caracteres especiais como underline e cifrão.

LIKE

- Consulta 10: Selecciona o código do produto que tem o nome igual a 'p1'.
 - SELECT ID_Produto AS CodigoProduto
FROM Produto
WHERE nome LIKE 'p1';

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

LIKE

- Para fazer uma busca de palavra em que conhecemos somente um ou alguns caracteres, utilizamos o LIKE e o símbolo de porcentagem (%), que serve para indicar onde está a parte desconhecida da palavra.
 - % pode ser colocado no início, no fim ou no meio das palavras.

LIKE

- Consulta 11: Seleciona o nome dos clientes que iniciam com a letra 'A'.
 - SELECT nome
FROM Cliente
WHERE nome LIKE 'A%';

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

LIKE

- Se desejarmos que no critério da consulta não conste uma determinada cadeia de caracteres, basta utilizarmos o operador lógico NOT junto ao LIKE.
- Consulta 12: Seleciona os clientes com nomes que não possuem a sílaba 'ro'.
 - SELECT nome
FROM Cliente
WHERE nome NOT LIKE '%ro%';

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

BETWEEN

- Comando utilizado quando precisamos encontrar um valor em um intervalo.
- Consulta 13: Selecione os produtos com valores entre 100 e 500 reais.
 - `SELECT nome`
`FROM Produto`
`WHERE valor BETWEEN 100 AND 500;`

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

BETWEEN

- O comando BETWEEN também pode ser usado com o NOT para excluir casos da consulta.
- Consulta 14: Selecione os produtos com valores que não estão entre 100 e 500 reais.

- SELECT nome

- FROM Produto

- WHERE valor NOT BETWEEN 100 AND 500;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

IN

- O comando IN é utilizado como um comparador quando são apresentadas várias condições, servindo como alternativa para o uso do operador lógico OR relacionando várias condições.
- O comando IN pode ser usado junto ao comando NOT.
- Consulta 15: Selecione os clientes que vivem em Uberlândia ou em São Paulo.
 - SELECT nome
FROM Cliente
WHERE cidade IN (“Uberlândia”, “São Paulo”);

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

IS NULL / IS NOT NULL

- O comando IS NULL serve para verificar se algum atributo tem valor NULL, ou no caso do IS NOT NULL se o atributo não tem valor NULL.
- Consulta 16: Selecione as empresas que possuem cadastrado no banco o número do CNPJ (ou seja, não é igual a NULL).
 - SELECT nome
FROM Empresa
WHERE CNPJ IS NOT NULL;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

Funções agregadas

- AVG(<nome do atributo>): retorna a média de todos os registros do atributo especificado.
- MIN(<nome do atributo>): retorna o valor mínimo dentre todos os registros do atributo especificado.
- MAX(<nome do atributo>) : retorna o valor máximo dentre todos os registros do atributo especificado.
- SUM(<nome do atributo>) : retorna a somatória de todos os registros do atributo especificado.
- COUNT(<*>) : retorna a contagem de registros na tabela.

Funções agregadas

- Consulta 17: Informar a quantidade de clientes cadastrados.
 - SELECT COUNT(*) AS QntClientes
FROM Cliente;
- Consulta 18: Retorne o valor médio dos produtos.
 - SELECT AVG(valor) AS MediaValor
FROM Produto;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

Funções agregadas

- Consulta 19: Retorne o menor valor dos produtos.
 - SELECT MIN(valor) AS MenorValor
FROM Produto;
- Consulta 20: Retorne o maior valor dos produtos.
 - SELECT MAX(valor) AS MaiorValor
FROM Produto;
- Consulta 21: Retorne a soma dos valores dos produtos.
 - SELECT SUM(valor) AS ValorTotal
FROM Produto;

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

HAVING

- Quando precisamos estabelecer condições específicas devemos utilizar o HAVING.
- Com a cláusula WHERE somente as linhas, considerando todos os registros, que atendem os critérios na cláusula são retornadas.
- A cláusula HAVING se aplica às linhas no conjunto de resultado. Somente os registros que atendem os critérios de HAVING são exibidos na saída da consulta.
- A cláusula HAVING é normalmente usada com o GROUP BY quando se deseja restringir o agrupamento.

Having

- Consulta 22: Selecione os produtos com valor acima de 20 reais e entre estes os que tiveram produção menor que 100 unidades.

```
– SELECT nome, total_produzido  
FROM Produto  
WHERE valor > 20  
HAVING total_produzido < 100;
```

Cliente (<u>ID_Cliente</u> , nome, endereco, cidade, telefone, tipo)
Empresa (<u>ID_Empresa</u> , CNPJ, nome, endereco, cidade, telefone)
Cliente_Empresa (<u>ID_Empresa</u> , <u>ID_Cliente</u>)
Produto (<u>ID_Produto</u> , nome, qnt_vendida, valor, total_produzido)
Produto_Empresa(<u>ID_Produto</u> , <u>ID_Empresa</u>)
Produto_Comprado_Cliente(<u>ID_Produto</u> , <u>ID_Cliente</u> , quantidade)

Exercícios

Utilizando a base de dados usada em aula responda usando comandos SQL.

- 1) Dê o comando que liste o nome de cada empresa junto de seu endereço.
- 2) Dê o comando que contabiliza a quantidade de produtos que tem valor menor que 500.
- 3) Dê o comando que liste para cada produto que teve mais de 100 unidades produzidas, o seu nome e valor.
- 4) Dê o comando que liste (sem repetição) os IDs dos produtos que já foram comprados por algum cliente.
- 5) Dê o comando que contabilize a média da quantidade de itens comprados por todos os clientes.

Cliente (ID_Cliente, nome, endereco, cidade, telefone, tipo)

Empresa (ID_Empresa, CNPJ, nome, endereco, cidade, telefone)

Cliente_Empresa (ID_Empresa, ID_Cliente)

Produto (ID_Produto, nome, qnt_vendida, valor, total_produzido)

Produto_Empresa(ID_Produto,ID_Empresa)

Produto_Comprado_Cliente(ID_Produto,ID_Cliente, quantidade)