

Aula 08: Comando SELECT: Ordenação, Funções de agregação e agrupamento.

Banco de Dados

Roteiro da Aula

- Comando SELECT:
 - ORDER BY
 - DISTINCT
 - FUNÇÕES DE AGREGAÇÃO
(COUNT,SUM,MIN,MAX,AVG)
 - GROUP BY
 - HAVING

Comando SELECT

- Estudamos que o comando SELECT é usado para consultar o banco de dados e retornar dados recuperados que satisfazem a determinada condição expressa no comando.
- Sua sintaxe é representada da seguinte forma:

```
SELECT <lista de atributos>  
FROM NOME DA TABELA  
WHERE <condições>
```

Ordenação

- Podemos incluir no comando SQL uma cláusula que permita solicitar a ordenação dos resultados fornecidos por um comando SELECT.
- A ordenação é feita da seguinte forma: primeiro os dígitos, depois os caracteres maiúsculos e por último os caracteres minúsculos.
- Representação:

$0 < 1 < \dots < 8 < 9 < \dots < A < B < \dots < Z < \dots < a < b < \dots < z$

Cláusula ORDER BY

- A ordenação pode ser realizada através da cláusula ORDER BY no comando SELECT. Esta cláusula aparece sempre posicionada no final do comando SELECT.
- Veja a sintaxe:

```
SELECT Campo1,Campo2,Campo3  
FROM Tabela  
WHERE Condição  
ORDER BY Campo ASC | DESC
```

- **ASC** indica ordenação **ASC**endente
- **DESC** indica ordenação **DESC**endente

Cláusula ORDER BY

- Situação 01: Selecionar todos os dados da tabela Pessoa, ordenado pela Idade

```
SELECT *  
FROM Pessoa  
ORDER BY Idade
```

- OBS: Quando a ordenação for ASCendente não é necessário incluir a cláusula ASC em ORDER BY, já que a ordenação padrão é ascendente

Cláusula ORDER BY

- Situação 02: Selecionar o Nome e o Salário de todas as pessoas, ordenando o resultado pelo Salário, de tal forma que os maiores salários fiquem no topo da lista.
- Nesta situação usaremos a cláusula DESC de maneira que os maiores salários fiquem no topo da lista.

```
SELECT Nome,Salario  
FROM Pessoa  
ORDER BY Salario DESC
```

Cláusula ORDER BY

- Situação 03: Resolver o mesmo problema anterior, mas só para as pessoas com Cargo de Assistente Administrativo.
- Nesta situação usaremos a cláusula WHERE, antes da ordenação.

```
SELECT Nome, Salario  
FROM Pessoa  
WHERE Cargo = "Assistente Administrativo"  
ORDER BY Salario DESC
```


Cláusula ORDER BY

- É possível ordenar o resultado do comando recorrendo a mais de uma coluna.
- **Situação 04: Selecionar todos os dados da tabela Pessoa, ordenado pela Idade e pelo Salário.**

```
SELECT *  
FROM Pessoa  
ORDER BY Idade,Salario
```

Cláusula ORDER BY

- O comando SELECT permite, além da seleção de colunas, a seleção de expressões ou mesmo a seleção de constantes.
- **Situação 05: Selecionar da tabela Pessoa o Nome e a Idade que irão ter daqui a dois anos. O resultado deverá vir pela nova idade.**

```
SELECT Nome, Idade + 2 as Nova_Idade  
FROM Pessoa  
ORDER BY Idade + 2
```

Cláusula ORDER BY

- É possível também incluir na cláusula de ordenação o número da posição, dentre as colunas a serem apresentadas, da coluna ou colunas pelas quais se pretende ordenar o resultado.
- **Situação 06: Selecionar na tabela Vendas, a Nota Fiscal, o campo Valor, o montante do Imposto (17%) do Valor, e o Valor total com Imposto.**

```
SELECT Nota_Fiscal, Valor, Valor * 0.17 As Imposto, Valor  
+ Valor * 0.17 as Valor_Total  
FROM VENDAS  
ORDER BY 1,4
```

- **OBS: 1 está se referindo ao campo Nota_Fiscal, 4 está se referindo ao valor total da venda(valor + imposto)**

Cláusula DISTINCT

- A cláusula DISTINCT permite eliminar repetições de linhas no resultado de um comando SELECT. A cláusula DISTINCT só pode ser colocada imediatamente depois do SELECT.
- **Situação 07: Selecionar o conjunto das Localidades existente na tabela Postal. OBS: Não exibir localidades iguais.**

```
SELECT DISTINCT Localidade  
FROM Postal
```

Funções de Agregação

- As funções de agregação (funções estatísticas) têm por objetivo obter informações sobre o conjunto de linhas especificados na cláusula **WHERE** ou sobre grupos de linhas indicados na cláusula **GROUP BY**.
- As funções disponíveis são:
 - COUNT: número de linhas
 - MAX: o maior valor da coluna
 - MIN: o menor valor da coluna
 - SUM: soma de todos os valores da coluna
 - AVG: média de todos os valores da coluna

Função COUNT

- Situação 08: Quantos Funcionários existem e quantos têm telefone fixo.

```
SELECT COUNT(Nome) AS Total_Funcionario, COUNT(Telefone)  
as Total_Funcionario_Com_Telefone  
FROM Funcionarios
```

Função COUNT

- Situação 09: Quantos Funcionários existem em nosso Banco de Dados

```
SELECT COUNT(*) AS Total  
FROM Funcionarios
```

Função MIN E MAX

- Situação 10: Qual o maior salário e a menor idade dos funcionários da empresa

```
SELECT MAX(Salario) AS Maior_Salario, MIN(Idade) AS  
Menor_Idade  
FROM Funcionarios
```


Função MIN E MAX

- **Situação 11: Qual o menor e maior valor de comissão superior a 1000 e inferior a 10000 na tabela Venda**

```
SELECT MAX(Comissao) AS Maior_Comissao, MIN(Comissao) AS  
Menor_Comissao  
FROM Venda  
WHERE Comissao >= 1000 AND Comissao <= 10000
```

Função SUM

- A função SUM retorna a soma de uma determinada coluna.
- **Situação 12: Qual é o valor total das comissões a pagar?**

```
SELECT SUM(Comissao) AS Total_Comissao  
FROM Venda
```

Função AVG

- A função AVG retorna a média dos valores de uma determinada coluna.
- Situação 13: Qual é o salário dos Funcionários com mais de 40 anos.

```
SELECT AVG(Salario) AS Media_Salario  
FROM Funcionario  
WHERE Idade > 40
```

Agrupamento

- As cláusulas de agrupamento estão relacionadas com as funções de agregação, e são úteis no tratamento de informações de forma agrupada.
- As cláusulas GROUP BY e HAVING fazem parte do comando SELECT e são representadas de acordo com a sintaxe a seguir:

```
SELECT Campos  
FROM Tabela  
WHERE Condição  
GROUP BY ...  
HAVING ...
```

Cláusula GROUP BY

- A cláusula GROUP BY divide o resultado de um SELECT em um grupo de resultados que serão processados pelas funções de agregação.
- **Situação 14: Mostrar a quantidade de CARROS vendidos agrupados pelo Modelo.**

```
SELECT Modelo_Carro, Count(Nota_Fiscal) AS  
Quantidade_Carros_Vendidos  
FROM Vendas  
GROUP BY Modelo_Carro
```

Cláusula GROUP BY

- Situação 15: Mostrar para cada funcionário o valor total das vendas realizadas.

```
SELECT Nome, Sum(Valor) AS Total_Vendas  
FROM Vendas  
GROUP BY Nome
```

Cláusula **HAVING**

- A cláusula **HAVING** faz restrições ao nível dos grupos que são processados. É comum surgir a dúvida sobre **WHERE** ou **HAVING**.
- A diferença entre **HAVING** e **WHERE** é que a cláusula **WHERE** é usada para restringir os registros a serem considerados na seleção.
- A Cláusula **HAVING** restringe os grupos que foram formados depois da aplicação da cláusula **WHERE**.

Cláusula HAVING

- Situação 16: Mostrar para cada funcionário o valor total das vendas superiores a 80000

```
SELECT Nome, Sum(Valor) AS Total_Vendas  
FROM Vendas  
GROUP BY Nome  
HAVING Sum(Valor) > 80000
```


Cláusula GROUP BY

- **Situação 17: Selecionar a quantidade de cada produto vendido em uma compra, ordenados em ordem decendente.**

```
SELECT Codigo_Produto,Count(Codigo_Produto) AS QUANTIDADE  
FROM Compra  
GROUP BY Codigo_Produto  
ORDER BY 2 DESC
```

Próxima Aula

- Exercícios Práticos em Sala de Aula