Modelo relacional e comandos SQL para definição de dados (DDL)

Comando SELECT

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>
```

Exemplos:

```
SELECT codigo,aluno,media
FROM NOTAS
WHERE aluno = "Tiago"
```

```
SELECT matricula,nome,responsavel,data_nascimento,cpf,rg,
endereco,codigo_curso,observacoes FROM ALUNOS
WHERE nome = "Camila"
```

```
SELECT * FROM ALUNOS
WHERE nome = "Camila"
```

Obs: o símbolo * na clausula SELECT indica que deverá ser selecionado todos os campos de uma tabela.

Para melhor exemplificar o funcionamento do comando SELECT, veremos as seguintes situações:

Situação 01: Escrever o comando SQL que permite obter o RG, Nome e o Código Postal de todos os clientes registrados no banco de dados.

```
SELECT Rg, Nome, CodigoPostal
FROM Cliente
```

Situação 02: Selecionar todos os dados de todos os pacientes cadastrados no Hospital

Nesta situação usaremos o curinga(*) , ao invés de escrever todos os campos da tabela Paciente no comando SQL.

```
SELECT *
FROM Pacientes
```

Situação 03: Selecionar o ID, Nome, Idade e o Salário de todos os Funcionários com Idade entre 30 e 40 anos

Nesta situação usaremos o operador WHERE, juntamente com operadores lógicos e relacionais.

```
SELECT Id,Nome,Idade,Salario
FROM Funcionario
WHERE Idade >= 30 AND Idade <= 40
```

Situação 04: Selecionar o ID, Nome, Idade e o Salário de todos os Funcionários cuja a idade não está entre 30 e 40 anos.

Nesta situação usaremos o operador WHERE, juntamente com operadores lógicos (AND e NOT) e relacionais.

```
SELECT Id,Nome,Idade,Salario
FROM Funcionario
WHERE Not(Idade >= 30 AND Idade <= 40)
```

Situação 05: Selecionar todos os indivíduos que possuem sobrenome “Silva”

Nesta situação usaremos o operador Like e o curinga “%”

```
SELECT *
FROM Pessoa
WHERE Nome Like "%Silva%"
```

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 ASCendente
- DESC indica ordenação DESCendente

Situação 06: Selecionar todos os dados da tabela Pessoa, ordenado pela Idade.

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

Situação 7: Selecionar todos os dados da tabela Pessoa, ordenado pela Idade e pelo Salário.

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

O comando SELECT também permite agrupar resultados.

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 ...
```

Situação 8: 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
```

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.

Situação 9: 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
```