

# Instrução SELECT

A cláusula **select** é obrigatoriamente seguida de uma cláusula **from**. Enquanto a primeira cláusula define quais os dados desejados, a segunda indica de onde serão obtidos.

Segue exemplo:

**Select codprod, dsprod from produto;**

O **select** acima não ordenou os dados nem alterou qualquer característica original, mas trouxe apenas as colunas que foram definidas na lista.

## SELECT DISTINCT

Não mostra eventuais valores repetidos em todas as colunas.

**Select distinct dsprod from produto;**

## 2.2 WHERE

Esta cláusula indica as condições da consulta. As condições melhoram a performance por permitir que o *engine* do banco de dados utilize os índices para organizar a forma de procura.

### Operador Função Exemplo

= compara igualdade: **codprod = 30**

<> Idade diferente de 30: **saldo <> 0**

> Maior (31 para frente): **saldo > 30**

>= maior ou igual (inclui o 30): **saldo >= 30**

< Menor (29 para baixo): **saldo < 30**

<= menor ou igual (inclui o 30): **saldo <= 30**

**Between** Idade que esteja entre 50 e 90 inclusive: **saldo Between 0 And 90**

**Not** Negação de uma condição : **saldo Not Between 0 And 90**

**Is** Compara igualdade entre tipos: **saldo is Null**

**And** Significa que as comparações devem ser todas verdadeiras:

**saldo = 30 And dsprod = "Toalha"**

**Or** Significa apenas uma das comparações precisa ser verdadeira:

**Saldo = 0 Or sldmin < Saldo**

### 2.2.1 Filtragem de Strings

A instrução *like* tem como diferencial o uso dos coringas “%” para qualquer coisa na substituição (similar ao “\*” do DOS), “[” e “]” para definir range. Os exemplos abaixo definem bem estas regras:

**Select \* FROM produto WHERE dsprod LIKE '%a';**  
**-- Terminam com a letra a**

**Select \* FROM produto WHERE dsprod LIKE '%tom%';**  
**--Contém tom em qualquer lugar**

**SELECT \* FROM produto where dsprod LIKE '\_n%';**  
**--Contém a letra n na segunda posição**

**SELECT \* FROM produto where dsprod LIKE '\_\_\_\_n%';**  
**--Contém a letra n na quinta posição**

### 2.2.2 Operador IN

Este operador é uma alternativa a uma extensa lista de *or*, uma vez que sua utilização é definir uma lista dos valores numéricos ou strings desejados.

Veja abaixo a comparação entre um comando com *or* e com *in*:

**Select \* from produto where codgrupo = 10 or codgrupo = 20;**

**Select \* from produto where codgrupo in (10, 20);**

### 2.3 ALIAS

Alias são literalmente apelidos que podemos atribuir a uma tabela, coluna ou o resultado de uma função ou expressão aritmética utilizada em um select.

**Select prvenda as preco, dsprod as Nome from produto;**

**Select cr.codprod as id, cr.dsprod from produto cr;**

### 2.4 ORDER BY

Com esta clausula conseguimos definir em que ordem queremos o retorno dos dados

**Select \* from produto order by dsprod;**

**Select \* from produto order by dsprod desc;**

**Select \* from produto order by dsprod, codprod;**

## 2.5 Retornar número limitado de linhas

Uma das funções que facilitam muito o trabalho com tabelas ordenadas é trazer um numero limitado de registros em uma tabela.

SQL Server: ***Select TOP 10 codprod, dsprod from produto;***

MySQL / PostgreSQL/SQLite: ***Select codprod, dsprod from produto limit 10;***

SQL Server: ***Select codprod, dsprod from produto where rownum <= 10;***

Firebird: ***Select First 10 codprod, dsprod from produto;***

## 3 Agrupando Dados

Estas funções listadas abaixo são utilizadas para totalizar, somar, gerar relatórios, estatísticas e outras funções onde não interessa linha a linha, mas sim um resumo.

Função	Descrição
AVG	Média aritmética
COUNT	Retorna o número de ocorrências de linhas
MAX	Maior valor na coluna
MIN	Menor valor na coluna
SUM	Soma todos os valores da coluna

Deve-se tomar cuidado com o resultado das funções de agregação, com exceção do *count*, pois elas desconsideram os valores nulos. Por exemplo, se houver 20 clientes, destes apenas 16 tem dados e 4 estão nulos. A média será a soma dos valores divididos por 16 e não por 20 como deveria ser, gerando uma média incorreta.

*DICA: Para resolver o problema dos nulos usamos a função isnull(coluna,valor), onde se a coluna definida estiver nula, assume o outro valor.*

***select min(codprod) from produto;***

***select max(codprod) from produto;***

***select min(codprod), max(codprod) from produto;***

***select sum(prvenda) from produto where codgrupo = 187;***

***Select count(\*) from produto where prvenda = 0;***