

Aula 7

Data Query Language DQL - Continuação

Faculdade TANDA TANDA



# Linguagem SQL



Continuação da linguagem DQL.



Veja os resultados com linhas repetidas na consulta abaixo:

```
SELECT pais
FROM Cliente;
```

pais

Argentina

Argentina

Austria

Austria

Belgium

Belgium



Podemos eliminar as linhas repetidas aplicando a cláusula DISTINCT:

SELECT DISTINCT <column list>

FROM

SELECT DISTINCT pais FROM Cliente;

pais

Argentina

Austria

Belgium



A cláusula DISTINCT retira repetições de linhas para todas as colunas descritas na declaração SELECT:

SELECT DISTINCT empresa, pais FROM Cliente;

Empresa pais

-----

Empresa AHPOP UK

Empresa AHXHT Mexico

Empresa AZJED Germany

Empresa BSVAR France

Empresa CCFIZ Poland



Muitas vezes queremos visualizar apenas o retorno de algumas linhas e não necessariamente todos os registros de uma tabela. Podemos utilizar a cláusula TOP para isso:

TOP (N) / TOP (N) PERCENT Retorna uma certa quantidade de linhas (ou percentual de linhas) definido.

SELECT top 10 \* FROM exemploSQL SELECT top 10 percent \* FROM exemploSQL



A cláusula WHERE faz o filtro horizontal em uma consulta, ou seja, permite uma redução do número de linhas que retornarão na consulta.

Elemento	Expressão	Descrição
SELECT	<li>sta de seleção&gt;</li>	Define quais as colunas que serão retornadas.
FROM	<tabela de="" origem=""></tabela>	Define a(s) tabela(s) envolvida(s) na consulta.
WHERE	<condição de<br="">pesquisa&gt;</condição>	Filtra as linhas requeridas.
GROUP BY	<agrupar a="" seleção=""></agrupar>	Agrupa a lista requerida (utiliza colunas).
HAVING	<condição agrupamento="" de=""></condição>	Filtra as linhas requeridas, pelo agrupamento.
ORDER BY	<ordem da="" lista=""></ordem>	Ordena o retorno da lista.



Operadores são utilizados para avaliar uma ou mais expressões que retornam os valores possíveis: TRUE, FALSE ou UNKNOWN.

O retorno de dados se dará em todas as tuplas em que a combinação das expressões retornarem TRUE.

#### **Operadores de Comparação Escalar**

SELECT FirstName, LastName,
MiddleName
FROM Person.Person
WHERE ModifiedDate >= `20040101'



Operadores são utilizados para avaliar uma ou mais expressões que retornam os valores possíveis: TRUE, FALSE ou UNKNOWN.

O retorno de dados se dará em todas as tuplas em que a combinação das expressões retornarem TRUE.





Operadores Lógicos são usados para combinar condições na declaração

Retorna somente registros nos quais o primeiro nome seja 'John' **E** o sobrenome, 'Smith'

WHERE FirstName = 'John' AND LastName = 'Smith'



Operadores Lógicos são usados para combinar condições na declaração

Retorna todas as linhas nas quais o primeiro nome seja 'John' **OU** todas nas quais o sobrenome seja 'Smith'

WHERE FirstName = 'John' **OR** LastName = 'Smith'



Operadores Lógicos são usados para combinar condições na declaração

Retorna todas as tuplas nas quais o primeiro nome seja 'John' e o sobrenome NÃO seja 'Smith'

WHERE FirstName = 'John' AND **NOT** LastName = 'Smith'

FirstName	LastName	MiddleName
John	Peoples	NULL
John	Robinson	1
John	Rodman	NULL
John	Rodriguez	М
John	Sandsto	J
John	Taylor	NULL
John	Thomas	W
John	Thomps	1
John	Tippett	NULL



#### Cláusula WHERE Simples

```
SELECT BusinessEntityID AS 'Employee Identification Number', HireDate,
```

VacationHours, SickLeaveHours

FROM HumanResources.Employee

WHERE BusinessEntityID <= 1000

SELECT FirstName, LastName, Phone FROM Person.Person WHERE FirstName = 'John',



#### Cláusula WHERE usando Predicado

Nem sempre usamos operadores de comparação. Em algumas situações, podemos usar outros operadores que são chamados de predicados, simplificando a escrita do script.

Alguns exemplos de Predicado são:

IN, BETWEEN, ANY, SOME, IS, ALL, OR, AND, NOT, EXISTS...

SELECT FirstName, LastName, Phone FROM Person.Person WHERE EmailAddress IS NULL;



• BETWEEN – Restringe dados através de uma faixa de valores possíveis.

SELECT OrderDate, AccountNumber, SubTotal, TaxAmt FROM Sales.SalesOrderHeader WHERE OrderDate BETWEEN '20110801' AND '20110831'

• **BETWEEN** – A mesma lógica do uso de >= AND <=.

SELECT OrderDate, AccountNumber, SubTotal, TaxAmt FROM Sales.SalesOrderHeader WHERE OrderDate >= '20110801' AND OrderDate <= '20110831'

OrderDate	AccountNumber	SubTotal	TaxAmt
2001-08-01 00:00:00.000	10-4020-000018	39677.4848	3174.1988
2001-08-01 00:00:00.000	10-4020-000353	24299.928	1943.9942
2001-08-01 00:00:00.000	10-4020-000206	10295.8366	823.6669
2001-08-01 00:00:00.000	10-4020-000318	1133.2967	90.6637
2001-08-01 00:00:00.000	10-4020-000210	1086.6152	86.9292
2001-08-01 00:00:00.000	10-4020-000164	21923.9352	1753.9148
2001-08-01 00:00:00.000	10-4020-000697	24624.706	1969.9765
2001-08-01 00:00:00.000	10-4020-000191	12286.7218	982.9377

# Atividades extraclasse



 Leitura do arquivo PDF disponibilizado na plataforma