

Consultas com SQL

Disciplina: Banco de dados

Profa. Dra. Graça Tomazela

A estrutura básica de uma expressão SQL consiste de três cláusulas :
SELECT ,
FROM e
WHERE .

⇒ A cláusula **SELECT** corresponde à operação projeção da álgebra relacional. É usada para listar os atributos desejados no resultado de uma consulta.

⇒ A cláusula **FROM** corresponde à operação produto cartesiano da álgebra relacional. Ela lista as relações a serem examinadas na avaliação da expressão.

⇒ A cláusula **WHERE** corresponde a um predicado de seleção da álgebra relacional. Consiste em um predicado envolvendo atributos de relações que aparecem na cláusula **FROM** .

Sintaxe

```
SELECT nome_atributo1,  
nome_atributo2,...  
FROM nome_tabela1,  
nome_tabela2...  
WHERE condição
```

O comando SELECT seleciona linhas e colunas de tabelas

- SELECT : especifica colunas
- FROM – especifica tabelas
- WHERE – especifica as linhas
- SELECT * recupera todas as colunas
- SELECT sem a cláusula WHERE recupera todas as linhas



Selecionando Colunas Específicas da Tabela

Listar todos os produtos com suas descrições, unidades e valores unitários

```
Select descricao, unidade, valor_unitario
```

```
From produto
```

**Selecionando
todas as
colunas da
Tabela**

Listar todos os dados do vendedor

```
Select *  
From vendedor
```

Usando Literais com a Lista de Atributos

Listar o código e o nome de cada cliente

```
Select  'Código do Cliente', cod_cliente, ' Nome do Cliente',  
nome_cliente  
From cliente
```

**Em cada linha
do resultado da
consulta serão
exibidos os
literais descritos
da seguinte
forma:**

Código do Cliente	290	Nome do Cliente	Gabriel
Código do Cliente	291	Nome do Cliente	Gustavo
Código do Cliente	292	Nome do Cliente	Guilherme
Código do Cliente	293	Nome do Cliente	Renata
Código do Cliente	294	Nome do Cliente	Priscila

Mudando o Cabeçalho das colunas

Listar o número do pedido e o código do cliente da tabela de pedidos

```
Select Pedido = num_pedido, 'Código  
Cliente' = cod_cliente  
from pedido
```

ou

```
Select num_pedido as pedido, cod_cliente as  
'Código Cliente'  
From pedido
```

ou

```
Select num_pedido pedido, cod_cliente  
'Código Cliente'  
From pedido
```

Resultados: sem
alterar o cabeçalho
da coluna X
alterando o
cabeçalho da coluna

	num_pedido	cod_cliente
1	1	2
2	2	2
3	4	10
4	5	17
5	6	1
6	7	6
7	8	18
8	9	13
9	10	10
10	11	15
11	12	7
12	13	12
13	14	9
14	15	14
15	16	7
16	17	11
17	18	7

✓ Consulta executada com êxito.

	Pedido	Código Cliente
3	4	10
4	5	17
5	6	1
6	7	6
7	8	18
8	9	13
9	10	10
10	11	15
11	12	7
12	13	12
13	14	9
14	15	14
15	16	7
16	17	11
17	18	7
18	19	14
19	20	18

✓ Consulta executada com êxito.

Mudando o cabeçalho das colunas, inserindo literal e deixando o aluno louco!!!

```
Select num_pedido Pedido, 'Código do
Cliente' as Exemplo, cod_cliente 'Código do
Cliente'
From pedido
```

Resultados		Mensagens	
	Pedido	Exemplo	Código do Cliente
1	1	Código do Cliente	2
2	2	Código do Cliente	2
3	4	Código do Cliente	10
4	5	Código do Cliente	17
5	6	Código do Cliente	1
6	7	Código do Cliente	6
7	8	Código do Cliente	18
8	9	Código do Cliente	13
9	10	Código do Cliente	10
10	11	Código do Cliente	15
11	12	Código do Cliente	7
12	13	Código do Cliente	12
13	14	Código do Cliente	9
14	15	Código do Cliente	14
15	16	Código do Cliente	7
16	17	Código do Cliente	11
17	18	Código do Cliente	7

✓ Consulta executada com êxito.

Exemplo 2....O mundo é louco mesmo!!!

```
Select num_pedido as pedido,  
'Código do Cliente', 'Código do  
Cliente = cod_cliente From pedido
```

OU

```
Select num_pedido as pedido,  
'Código do Cliente', cod_cliente  
'Código do Cliente  
From pedido
```

Resultados		Mensagens	
	pedido	(Nenhum nome de coluna)	Código do Cliente
1	1	Código do Cliente	2
2	2	Código do Cliente	2
3	4	Código do Cliente	10
4	5	Código do Cliente	17
5	6	Código do Cliente	1
6	7	Código do Cliente	6
7	8	Código do Cliente	18
8	9	Código do Cliente	13
9	10	Código do Cliente	10
10	11	Código do Cliente	15
11	12	Código do Cliente	7
12	13	Código do Cliente	12
13	14	Código do Cliente	9
14	15	Código do Cliente	14
15	16	Código do Cliente	7
16	17	Código do Cliente	11
17	18	Código do Cliente	7

✓ Consulta executada com êxito.

Efetuando Operações Aritméticas nas Colunas

Listar os preços dos produtos aumentados de 5%

```
Select nome_produto,  
valor_unitario * 1.05 as 'valor  
aumentado em 5%' from  
produto
```

Selecionando Linhas da Tabela

- A cláusula WHERE especifica quais linhas devem ser recuperadas a partir das condições de pesquisa.
- **Tipo de condições permitidas:**

Operações	Operadores
Relacionais	= > < >= <= <> != !< !>
Faixa de valores	BETWEEN e NOT BETWEEN
Comparação com cadeia de caracteres	LIKE e NOT LIKE
Pertinência a conjuntos	IN e NOT IN
Valores desconhecidos	IS NULL e IS NOT NULL
Combinação de operações	AND, OR
Negações	NOT



Exemplos

Listar o número do pedido , o código do produto e a quantidade dos itens do pedido com quantidade igual a 35

```
Select num_pedido, cod_produto, quantidade  
From item_pedido  
Where quantidade = 35
```

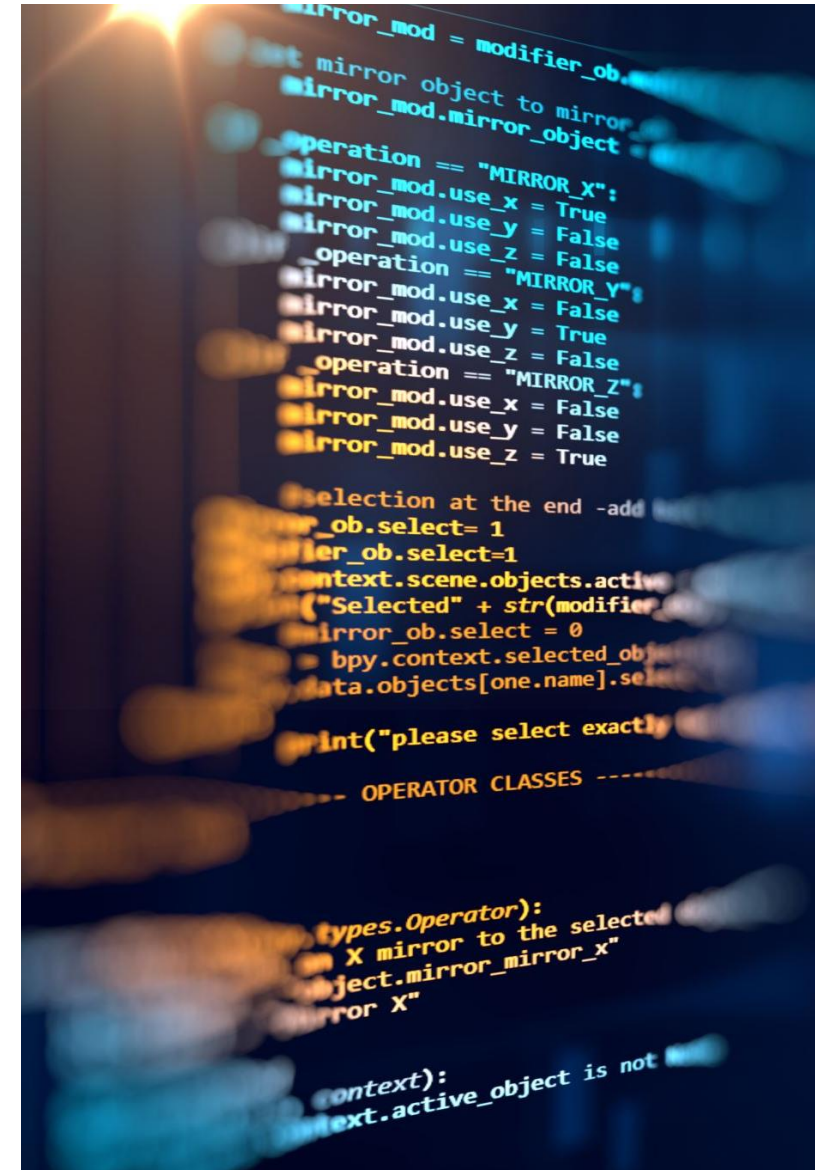
Listar o número do pedido , o código do produto e a quantidade dos itens do pedido com quantidade maior que 35 e menor do que 40

```
Select num_pedido, cod_produto, quantidade  
From item_pedido  
Where quantidade > 35 and quantidade < 40
```

Operações baseada em faixas de valores

Sintaxe

```
SELECT  
nome_atributo1, nome_atributo2....  
  
FROM tabela  
WHERE expressão  
[NOT] BETWEEN expressão AND  
expressão
```



Exemplo

Listar o código e a descrição dos produtos que tenham o valor unitário na faixa R\$0,32 até R\$2,00

```
Select cod_produto, descricao_produto
```

```
From produto
```

```
Where valor_unitario between $0.32 and $2.00
```

```
Select cod_produto, descricao_produto
```

```
From produto
```

```
Where valor_unitario >=$0.32 and  
valor_unitario <= $2.00
```

Exemplo

Select cod_produto, descricao_produto

From produto

Where valor_unitario not between \$0.32 and \$2.00

Select cod_produto, descricao_produto

From produto

Where valor_unitario < \$0.32 OR valor_unitario > \$2.00

Operações de pertinência a Conjuntos (Listas)

Sintaxe SELECT nome_atributo1, nome_atributo2....

FROM tabela

(lista de valores) WHERE [NOT] expressão [NOT] IN

Exemplo

Listar os vendedores que têm a faixa de comissão A ou B

Select nome_vendedor

From vendedor

Where faixa_comissao in ('A', 'B')

Select nome_vendedor

From vendedor

Where faixa_comissão = 'A' or faixa_comissão= 'B'

Exemplos

Listar os vendedores que têm a faixa de comissão diferente A e B

```
Select nome_vendedor
```

```
From vendedor
```

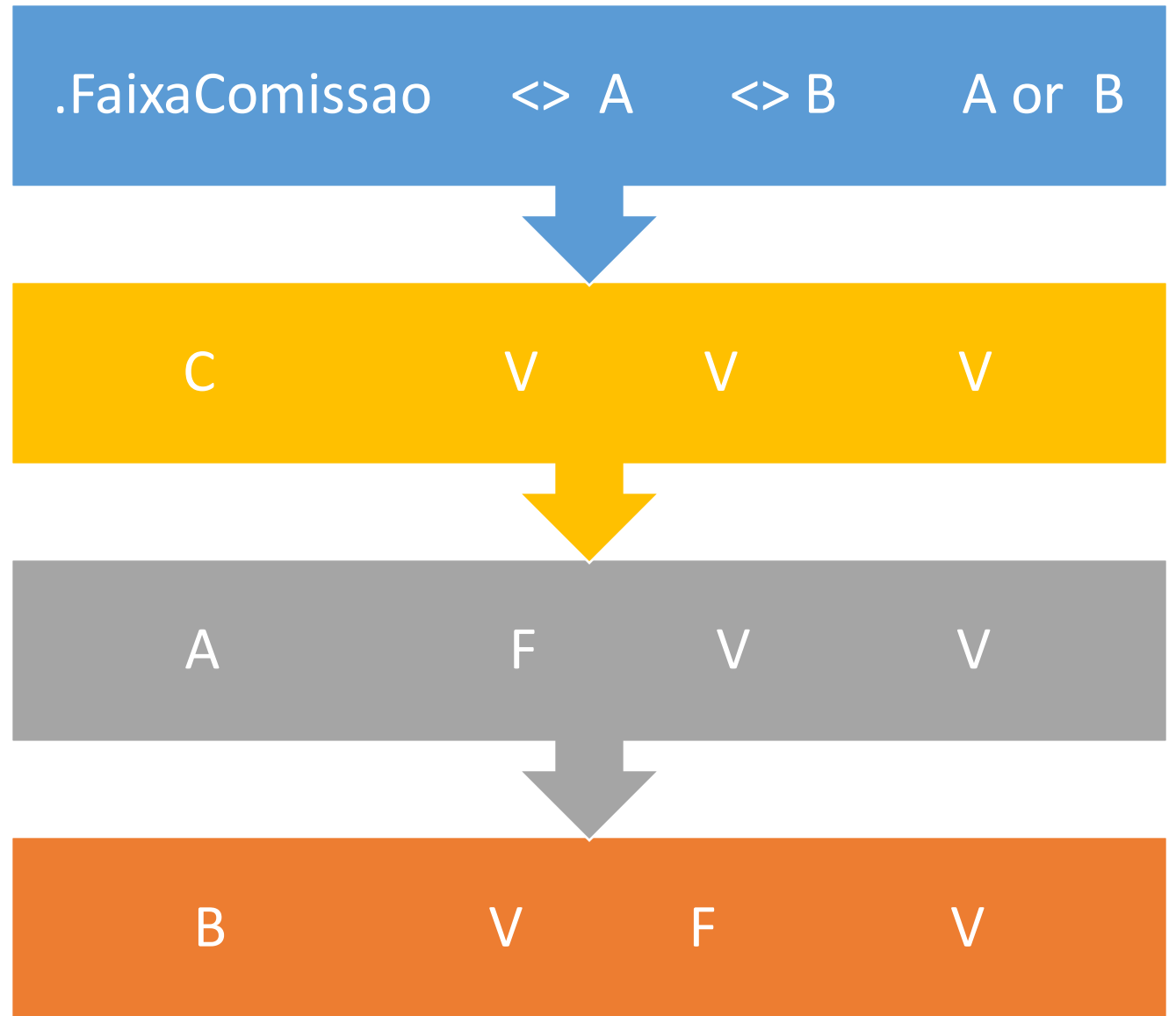
```
Where faixa_comissao not in ('A', 'B')
```

```
Select nome_vendedor
```

```
From vendedor
```

```
Where faixa_comissao <> 'A' and  
faixa_comissao <> 'B'
```

Tabela verdade



Comparações entre Cadeias de Caracteres

Sintaxe SELECT nome_atributo1, Nome_atributo2....

FROM tabela

WHERE expressão [NOT] LIKE '*cadeia de caracteres*'

Listar todos os produtos que tenham a sua unidade começando por K

Select cod_produto, descricao_produto

From produto

Where descrição like 'k%'



Operador Like


O operador LIKE é utilizado para comparações em cadeias de caracteres. Os *padrões* são descritos usando os seguintes caracteres especiais :

por cento (%) - O caractere % substitui qualquer subcadeia.

sublinhado (_) - O caractere _ substitui qualquer caractere.

Caracteres entre colchetes ([]) - Qualquer caractere dentro da faixa ou conjunto especificado. (SQL Server)

[^] - Qualquer caractere que não esteja dentro da faixa ou conjunto especificado. (SQL Server)



Exemplos



Expressão	Resultado
LIKE 'BR%'	Nomes que comecem com BR
LIKE '%een'	Nomes que terminem com een
LIKE '%en%'	Nomes que tenham a sequência en em qualquer posição
LIKE '_en'	Nomes de três letras terminando por en
LIKE '[CK]%'	Nomes que comecem com C ou K
LIKE '[S-V]ing'	Nomes de quatro letras que terminem ing e comecem com uma letra de S a V
LIKE 'M[^c]%'	Nomes que comecem com M e não tenham a letra c como segundo caracter

Exemplos

Select nome from vendedor
Where nome like 'Jo%'

Select descricao from produto
Where descricao like '%lo'

Select nome from vendedor
Where nome like '%ri%'

Select descricao, unidade from produto
Where unidade like '_g'



Funções Agregadas

Sintaxe

SELECT função_agregada1 ([ALL/ DISTINCT]
expressão)....

FROM nome_tabela

WHERE condição

Função	Parâmetros	descrição
AVG	([ALL/ DISTINCT] expressão)	Média de valores na coluna especificada, todos ou distintos
COUNT	([ALL/ DISTINCT] expressão)	Número de valores na coluna, todos ou distintos
COUNT	(*)	Número de linhas selecionadas
MAX	(expressão)	Maior valor na coluna
MIN	(expressão)	Menor valor na coluna
SUM	([ALL/ DISTINCT] expressão)	Somatório de valores na coluna, todos ou distintos

Exemplos

MAX, MIN

Listar o menor e o maior salário de vendedor

```
Select min(salario_fixo) AS 'MENOR SALARIO',  
max(salario_fixo) AS 'MAIOR SALARIO'  
From vendedor
```

SUM

Mostrar a quantidade total pedida para o produto de código '78'

```
Select SUM (quantidade),  
From item_pedido  
Where cod_produto = 78
```



Exemplos

AVG

Qual a média dos salários fixos dos vendedores ?

```
Select avg(salario_fixo) AS MEDIA_SALARIO  
From vendedor
```

COUNT

Quantos vendedores ganham acima de R\$ 2.500,00 de salário fixo

```
Select count (*) from vendedor  
Where salario_fixo > $2500
```

Utilizando as cláusulas **GROUP BY** e **HAVING**

Sintaxe

SELECT nome_atributo1,
nome_atributo2,...

FROM nome_tabela1,
nome_tabela2...

WHERE condição

[GROUP BY expressão,....]

[HAVING condição]

Exemplos

Listar QUANTOS produtos cada pedido contém

```
Select num_pedido, total_produtos = count (*)  
From item_pedido  
Group by num_pedido
```

Listar quantos pedidos (num_pedido, total de produtos) têm mais do que três produtos e que foram solicitados em quantidade maior que 100

```
Select num_pedido, total_produtos = COUNT (*)  
From item_pedido  
Where quantidade > 100  
Group by num_pedido  
Having count(*) > 3
```



Funções de data

SQL Server

FUNÇÃO DE DATA	TIPO DE RETORNO	DESCRIÇÃO
DATEADD(datepart, number, datetime)	Datetime	Produz uma data somando um intervalo a uma data especificada
DATEDIFF(datepart, datetime1, datetime2)	Int	Retorna a diferença entre duas datas, de acordo com o datepart especificado
DATENAME (datepart, datetime)	Varchar	Retorna o nome do datepart correspondente à data especificada
DATEPART(datepart, datetime)	Int	Retorna um inteiro representando a datepart especificada da data especificada
GETDATE()	datetime	Retorna a data e a hora correntes do sistema

Datepart	Abreviação	Valores
Year	YY	1753-9999
Quarter	QQ	1-4
Month	MM	1-12
Dayofyear	DY	1-366
Day	DD	1-31
Week	WEEK	1-53
Weekday	DW	1-7
Hour	HH	0-23
Minute	MI	0-59
Second	SS	0-59
millisecond	MS	0-999

Exemplos

```
Select num_pedido  
From pedido  
Where datepart(weekday, data_entrega) = 3  
And datepart(yy, data_entrega) = 2015
```

```
Select num_pedido  
From pedido  
Where datepart(day, data_entrega) = 12 and  
datepart(mm, data_entrega) = 2  
and datepart(yy, data_entrega) = 2009
```

Exemplos

Select *

From pedido

Where datediff(mm, getdate(), data_entrega) > 1

Select dateadd(day, 14, data_entrega), num_pedido

From pedido

Select data_entrega+ 14, num_pedido

From pedido

Exemplos

```
Select dateadd(mm, 5, data_entrega),  
num_pedido
```

```
From pedido
```

```
Select data_entrega+ 150,  
num_pedido
```

```
From pedido
```





Exemplos

```
SELECT YEAR(DATA_ENTREGA)  
FROM PEDIDO
```

```
SELECT  
MONTH(DATA_ENTREGA)  
FROM PEDIDO
```

```
SELECT DAY(DATA_ENTREGA)  
FROM PEDIDO
```

