



**UNITAU**  
Universidade de Taubaté





# SELECT (parte 2)



# WHERE



A cláusula **WHERE** permite selecionar os registros que serão retornados na seleção.

**WHERE condicao\_de\_pesquisa**

Exemplo:

```
SELECT * FROM sala WHERE curso='0001';
```

curso	serie	turma
0001	1	A
0001	1	B
0001	2	A
0001	3	A



# WHERE



A cláusula **WHERE** pode ser utilizada para obter os mesmos resultados de um **INNER JOIN**.

**Exemplo:**

```
SELECT curso.nome AS curso, aluno.nome AS aluno, aluno.rg  
FROM curso JOIN aluno ON aluno.curso=curso.codigo;
```

```
SELECT curso.nome AS curso, aluno.nome AS aluno, aluno.rg  
FROM curso,aluno WHERE aluno.curso = curso.codigo;
```

# WHERE



```
SELECT curso.nome AS curso, aluno.nome AS aluno, aluno.rg
FROM curso JOIN aluno ON aluno.curso = curso.codigo;
```

curso	aluno	rg
Tecnologia da Informacao	Ana Lucia	20143531
Tecnologia da Informacao	Luis Claudio	22336362
Tecnologia da Informacao	Marcelo	25343256
Tecnologia da Informacao	Debora	20356328
Tecnologia da Informacao	Fernanda	26344325
Tecnologia da Informacao	Alvaro	21764527
Engenharia da Computacao	Claudio	23336368
Engenharia da Computacao	Andrea	28456474
Engenharia da Computacao	Carla	23636731



# WHERE



```
SELECT curso.nome AS curso, aluno.nome AS aluno, aluno.rg
FROM curso,aluno WHERE aluno.curso = curso.codigo;
```

curso	aluno	rg
Tecnologia da Informacao	Ana Lucia	20143531
Tecnologia da Informacao	Luis Claudio	22336362
Tecnologia da Informacao	Marcelo	25343256
Tecnologia da Informacao	Debora	20356328
Tecnologia da Informacao	Fernanda	26344325
Tecnologia da Informacao	Alvaro	21764527
Engenharia da Computacao	Claudio	23336368
Engenharia da Computacao	Andrea	28456474
Engenharia da Computacao	Carla	23636731

# WHERE



É permitido usar campos de quaisquer tabelas relacionadas na cláusula **FROM**, funções e operadores disponíveis no banco de dados.

## Exemplo:

```
SELECT curso.nome AS curso, aluno.nome AS aluno, aluno.rg  
FROM curso,aluno WHERE curso = '0001' AND aluno.curso =  
curso.codigo;
```

curso		aluno		rg
-----+-----+-----				
Tecnologia da Informacao		Ana Lucia		20143531
Tecnologia da Informacao		Luis Claudio		22336362
Tecnologia da Informacao		Marcelo		25343256
Tecnologia da Informacao		Debora		20356328
Tecnologia da Informacao		Fernanda		26344325
Tecnologia da Informacao		Alvaro		21764527



# Expressões



Além de funções e operadores similares a outras linguagens, a linguagem SQL permite várias expressões que podem ser utilizadas tanto na lista de campos da cláusula **FROM** quanto na condição de seleção da cláusula **WHERE**.



# IN



O operador **IN** verifica se um valor está contido em uma lista de valores.

**Exemplo:**

```
SELECT * FROM curso WHERE codigo IN ( '0001', '0002' );
```

codigo	nome
0001	Tecnologia da Informacao
0002	Engenharia da Computacao

# IN



O operador **IN** também pode ser usado para verificar se um valor está contido no resultado de uma sub consulta.

**Exemplo:**

```
SELECT nome FROM curso WHERE codigo IN ( SELECT curso FROM  
aluno );
```

nome

-----  
Tecnologia da Informacao  
Engenharia da Computacao

# NOT IN



**NOT IN** verifica se um valor não está contido em uma lista de valores ou resultado de uma sub consulta

**Exemplo:**

```
SELECT * FROM curso WHERE codigo NOT IN ( '0001', '0002' );
```

codigo	nome
0003	Sistemas de Informacao



# BETWEEN



O operador **BETWEEN** verifica se um valor está dentro de uma faixa de valores.

Exemplo:

```
SELECT nome,rg FROM aluno WHERE rg BETWEEN '21000000' AND  
'23999999';
```

nome	rg
Luis Claudio	22336362
Alvaro	21764527
Claudio	23336368
Carla	23636731

# BETWEEN



O operador **BETWEEN** exige que o primeiro valor seja menor ou igual ao segundo. Para fazer a comparação mesmo que o primeiro valor seja maior que o segundo, deve ser usado **BETWEEN SYMMETRIC**.

Exemplo:

```
SELECT nome,rg FROM aluno WHERE rg BETWEEN '23999999' AND '21000000';
```

```
nome | rg
-----+-----
(0 rows)
```

```
SELECT nome,rg FROM aluno WHERE rg BETWEEN SYMMETRIC '23999999' AND '21000000';
```

```
      nome      |      rg
-----+-----
 Luis Claudio   | 22336362
 Alvaro         | 21764527
 Claudio        | 23336368
 Carla          | 23636731
```

# EXISTS



**EXISTS** verifica se uma sub consulta retorna pelo menos um registro.

**Exemplo:**

```
SELECT * FROM curso WHERE EXISTS ( SELECT 1 FROM aluno WHERE  
curso=curso.codigo );
```

codigo	nome
0001	Tecnologia da Informacao
0002	Engenharia da Computacao



# NOT EXISTS



**NOT EXISTS** verifica se uma sub consulta não retorna nenhum registro.

**Exemplo:**

```
SELECT * FROM curso WHERE NOT EXISTS ( SELECT 1 FROM aluno  
WHERE curso=curso.codigo );
```

codigo	nome
0003	Sistemas de Informacao

# ALL



**ALL** verifica se um valor atende a uma comparação com todos os registros retornados por uma sub consulta.

**Exemplo:**

```
SELECT nome, salario FROM funcionario WHERE salario > ALL  
( SELECT salario FROM funcionario WHERE funcao='4' );
```

nome	salario
Claudia	4500.00
Sandro	3800.00
Luana	3800.00
Nanci	4600.00

# ANY/SOME



**ANY/SOME** verifica se um valor atende a uma comparação com algum dos registros retornados por uma sub consulta.

**Exemplo:**

```
SELECT nome,salario,funcao FROM funcionario WHERE salario <
ANY ( SELECT salario FROM funcionario WHERE funcao = '2' );
```

nome	salario	funcao
Andre	1200.00	2
Pedro	1200.00	2
Andreia	1300.00	2
Luis	1300.00	2



# CASE



**CASE** permite a seleção entre múltiplas opções.

**Exemplo:**

```
SELECT nome, CASE WHEN serie='1' THEN 'primeiro' WHEN  
serie='2' THEN 'segundo' WHEN serie='3' THEN 'terceiro' ELSE  
'nao matriculado' END FROM aluno;
```

nome		case
-----+-----		
Ana Lucia		primeiro
Luis Claudio		primeiro
Marcelo		primeiro
Debora		primeiro
Fernanda		primeiro
Alvaro		primeiro
Claudio		primeiro
Andrea		primeiro
Carla		segundo
Fernanda		nao matriculado

# LIKE



**LIKE** compara uma cadeia de caracteres com um padrão. O caractere percentagem ( % ) corresponde a qualquer cadeia com zero ou mais caracteres.

**Exemplo:**

```
SELECT * FROM aluno WHERE nome LIKE 'An%';
```

matricula	nome	rg	curso	serie	turma
1	Ana Lucia	20143531	0001	1	A
8	Andrea	28456474	0002	1	A

# LIKE



O caractere sublinhado (    ) corresponde a qualquer caractere.

**Exemplo:**

```
SELECT * FROM aluno WHERE rg LIKE '2_3%';
```

matricula	nome	rg	curso	serie	turma
2	Luis Claudio	22336362	0001	1	A
3	Marcelo	25343256	0001	1	A
4	Debora	20356328	0001	1	B
5	Fernanda	26344325	0001	1	B
7	Claudio	23336368	0002	1	A



# Operadores para valores nulos



**O valor nulo ( **NULL** ) é considerado como desconhecido e o uso de operadores lógicos e de comparação tradicionais com valor nulo tem como resultado o valor nulo, o que pode ter consequências inesperadas se não houver maior atenção na elaboração das consultas.**

# Operadores para valores nulos



```
SELECT matricula, nome, serie FROM aluno WHERE (matricula>6);
```

matricula	nome	serie
7	Claudio	1
8	Andrea	1
9	Carla	2
10	Fernanda	

```
SELECT matricula, nome FROM aluno WHERE (matricula>6) AND (serie!='2');
```

matricula	nome
7	Claudio
8	Andrea

# Operadores para valores nulos



```
SELECT matricula, nome, curso FROM aluno;
```

matricula	nome	curso
1	Ana Lucia	0001
2	Luis Claudio	0001
3	Marcelo	0001
4	Debora	0001
5	Fernanda	0001
6	Alvaro	0001
7	Claudio	0002
8	Andrea	0002
9	Carla	0002
10	Fernanda	

```
SELECT matricula, nome FROM aluno WHERE NOT (curso='0001');
```

matricula	nome
7	Claudio
8	Andrea
9	Carla



# Operadores para valores nulos



```
SELECT * FROM aluno WHERE curso = NULL;  
matricula | nome | rg | curso | serie | turma  
-----+-----+-----+-----+-----+-----  
(0 rows)
```

```
SELECT * FROM aluno WHERE curso != NULL;  
matricula | nome | rg | curso | serie | turma  
-----+-----+-----+-----+-----+-----  
(0 rows)
```

```
SELECT nome FROM curso WHERE codigo NOT IN ( SELECT curso  
FROM aluno );  
nome  
-----  
(0 rows)
```

# COALESCE



**COALESCE** retorna o primeiro valor não nulo de uma lista de valores.

**Exemplo:**

```
SELECT nome, COALESCE(curso, 'nao matriculado') FROM aluno;
```

<b>nome</b>	<b>coalesce</b>
Ana Lucia	0001
Luis Claudio	0001
Marcelo	0001
Debora	0001
Fernanda	0001
Alvaro	0001
Claudio	0002
Andrea	0002
Carla	0002
Fernanda	nao matriculado

# COALESCE



```
SELECT nome FROM curso WHERE codigo NOT IN ( SELECT  
COALESCE(curso, '') FROM aluno );
```

nome

-----  
**Sistemas de Informacao**



# NULLIF



**NULLIF** retorna nulo se dois valores são iguais ou o primeiro valor se não forem iguais.

**Exemplo:**

```
SELECT nome, NULLIF( turma, 'B' ) FROM aluno;
```

nome	nullif
Ana Lucia	A
Luis Claudio	A
Marcelo	A
Debora	
Fernanda	
Alvaro	
Claudio	A
Andrea	A
Carla	A
Fernanda	

# Operadores para valores nulos



Para realizar operações lógicas e comparações com um tipo de dado qualquer sem ter o valor nulo como resultado, devem ser usados os operadores:

**expression IS NULL**

**expression IS NOT NULL**

**expression IS DISTINCT FROM expression**

**expression IS NOT DISTINCT FROM expression**

# IS NULL



Para verificar se um valor é nulo, deve ser usado o operador **IS NULL**.

**Exemplo:**

```
SELECT nome, curso FROM aluno WHERE curso IS NULL;
```

nome		curso
-----	+	-----
Fernanda		



# IS NOT NULL



Para verificar se um valor não é nulo, deve ser usado o operador **IS NOT NULL**.

**Exemplo:**

```
SELECT nome, curso FROM aluno WHERE curso IS NOT NULL;
```

nome	curso
-----+-----	
Ana Lucia	0001
Luis Claudio	0001
Marcelo	0001
Debora	0001
Fernanda	0001
Alvaro	0001
Claudio	0002
Andrea	0002
Carla	0002

# IS DISTINCT FROM



Para verificar se um valor é diferente de outro valor mesmo que os valores possam ser nulos, deve ser usado o operador **IS DISTINCT FROM**.

Exemplo:

```
SELECT nome, curso FROM aluno WHERE curso != '0001';
```

nome	curso
Claudio	0002
Andrea	0002
Carla	0002

```
SELECT nome, curso FROM aluno WHERE curso IS DISTINCT FROM '0001';
```

nome	curso
Claudio	0002
Andrea	0002
Carla	0002
Fernanda	



# IS NOT DISTINCT FROM



Para verificar se um valor é igual a outro valor mesmo que os valores possam ser nulos, deve ser usado o operador **IS NOT DISTINCT FROM**.

Exemplo:

```
SELECT nome, curso FROM aluno WHERE curso = (SELECT curso
FROM aluno WHERE matricula=10);
```

```
nome | curso
-----+-----
(0 rows)
```

```
SELECT nome, curso FROM aluno WHERE curso IS NOT DISTINCT
FROM (SELECT curso FROM aluno WHERE matricula=10);
```

```
nome | curso
-----+-----
Fernanda |
```



# Operadores para valores nulos



Para valores lógicos que podem ser nulos podem ser usados os operadores:

**expression IS UNKNOWN**

**expression IS NOT UNKNOWN**

**expression IS TRUE**

**expression IS NOT TRUE**

**expression IS FALSE**

**expression IS NOT FALSE**

# IS UNKNOWN / IS NOT UNKNOWN



Para verificar se um valor lógico é nulo / não é nulo pode ser usado o operador **IS UNKNOWN** / **IS NOT UNKNOWN**.

```
SELECT matricula, nome FROM aluno WHERE (curso='0001') IS UNKNOWN;
```

matricula	nome
10	Fernanda

```
SELECT matricula, nome FROM aluno WHERE (curso='0001') IS NOT UNKNOWN;
```

matricula	nome
1	Ana Lucia
2	Luis Claudio
3	Marcelo
4	Debora
5	Fernanda
6	Alvaro
7	Claudio
8	Andrea
9	Carla



# IS TRUE / IS NOT TRUE



Para verificar se um valor lógico é verdadeiro / não é verdadeiro pode ser usado o operador **IS TRUE** / **IS NOT TRUE**.

```
SELECT nome, curso FROM aluno WHERE NOT (curso='0001');
```

nome	curso
Claudio	0002
Andrea	0002
Carla	0002

```
SELECT nome, curso FROM aluno WHERE (curso='0001') IS NOT TRUE;
```

nome	curso
Claudio	0002
Andrea	0002
Carla	0002
Fernanda	



# IS FALSE / IS NOT FALSE



Para verificar se um valor lógico é falso / não é falso pode ser usado o operador **IS FALSE / IS NOT FALSE**.

```
SELECT nome, curso FROM aluno WHERE NOT (curso!='0002');
```

nome	curso
Claudio	0002
Andrea	0002
Carla	0002

```
SELECT nome, curso FROM aluno WHERE (curso='0002') IS NOT FALSE;
```

nome	curso
Claudio	0002
Andrea	0002
Carla	0002
Fernanda	

# Tipos Compostos



As condições da seleção podem utilizar tipos compostos.

**Exemplo:**

```
SELECT nome, curso, serie, turma FROM aluno WHERE  
(serie,turma) = ('1','A');
```

nome	curso	serie	turma
Ana Lucia	0001	1	A
Luis Claudio	0001	1	A
Marcelo	0001	1	A
Claudio	0002	1	A
Andrea	0002	1	A

```
SELECT nome, curso, serie, turma FROM aluno WHERE  
(serie,turma) IN (('1','B'),('2','A'));
```

nome	curso	serie	turma
Debora	0001	1	B
Fernanda	0001	1	B
Alvaro	0001	1	B
Carla	0002	2	A



# Subconsultas



Subconsultas ( subqueries ) são consultas dentro de uma outra consulta.

Geralmente, onde é permitida uma expressões em um comando SQL, pode ser utilizadas uma subconsulta na expressão.

**Exemplo:**

```
SELECT nome, rg, ( SELECT nome FROM curso WHERE codigo =  
aluno.curso ) AS curso FROM aluno;
```

nome	rg	curso
Ana Lucia	20143531	Tecnologia da Informacao
Luis Claudio	22336362	Tecnologia da Informacao
Marcelo	25343256	Tecnologia da Informacao
Debora	20356328	Tecnologia da Informacao
Fernanda	26344325	Tecnologia da Informacao
Alvaro	21764527	Tecnologia da Informacao
Claudio	23336368	Engenharia da Computacao
Andrea	28456474	Engenharia da Computacao
Carla	23636731	Engenharia da Computacao
Fernanda	29563735	



# Subconsultas



Subconsultas também podem ser utilizadas na condição da clausula **WHERE**.

**Exemplo:**

```
SELECT nome,secao FROM funcionario WHERE ( SELECT diretoria  
FROM secao WHERE secao.codigo=funcionario.secao ) = 'PES';
```

nome		secao
-----+-----		
Claudia		ADM
Marta		ADM
Ana		FPG
Luana		FPG
Pedro		ADM
Andreia		FPG

# Subconsultas



Também é possível utilizar subconsultas na clausula FROM.

Exemplo:

```
SELECT funcionario.nome,funcionario.salario,foo.descricao AS
diretoria FROM funcionario, ( SELECT secao.codigo,
diretoria.descricao FROM secao,diretoria WHERE
diretoria.codigo=secao.diretoria ) AS foo WHERE
foo.codigo=funcionario.secao AND funcionario.salario>3500;
```

nome	salario	diretoria
Claudia	4500.00	Pessoal
Sandro	3800.00	Financeira
Luana	3800.00	Pessoal
Nanci	4600.00	Producao



# LATERAL



A cláusula **LATERAL** pode ser utilizada com subconsultas no **FROM** para que a subconsulta possa fazer referencia a colunas das tabelas que precedam a subconsulta.

Sem a cláusula **LATERAL** a subconsulta será avaliada independentemente das tabelas precedentes.

**Exemplo:**

```
SELECT funcionario.nome,funcionario.salario,foo.descricao AS
diretoria FROM funcionario, LATERAL ( SELECT
diretoria.descricao FROM secao,diretoria WHERE
diretoria.codigo=secao.diretoria AND
secao.codigo=funcionario.secao) AS foo WHERE
funcionario.salario>3500;
```

nome	salario	diretoria
Claudia	4500.00	Pessoal
Sandro	3800.00	Financeira
Luana	3800.00	Pessoal
Nanci	4600.00	Producao



# Exercícios



**Sem utilizar JOIN, obtenha os seguintes dados das tabelas:**

- 01) matrícula, nome e salário dos funcionários da seção 'MAN'**
- 02) modelo, ano e país dos automóveis de cor preta**
- 03) nome, data de início e de final da participação dos participantes do projeto 02**
- 04) modelo, cor e ano dos automóveis fabricados no Brasil em 2007**
- 05) nome dos analistas do projeto 01**
- 06) modelo, cor e ano dos automóveis fabricados no Brasil que foram comprados em Jan/2010**
- 07) data e valor dos pagamentos de cartão de crédito feitos entre Fev/2012 e Mar/2012**
- 08) modelos dos automóveis fabricados na Argentina**
- 09) nome dos programadores que desenvolvem para Android**
- 10) matrícula, nome dos funcionários e descrição da seção dos funcionários que tenham seção**

# Exercícios



- 11) nome do fabricante, modelo, ano, cor do automóvel para os fabricantes que tem automóvel**
- 12) descrição do produto, data e quantidade das saídas de estoque do dia 25/03/2010**
- 13) modelo, cor e ano dos automóveis vendidos em Fev/2010**
- 14) matrícula, nome e descrição da seção de todos os funcionários**
- 15) nome do participante, fase e plataforma ( se não for relativo a uma plataforma específica ,identificar como “Neutro” ) dos participantes do projeto 01**
- 16) matrícula, nome e descrição da diretoria de todos os funcionários**
- 17) nome do banco (identificar como “Local” se não foi realizado em um banco ), data e valor para os débitos realizados em Abr/2012**
- 18) matrícula, nome dos funcionários e descrição da função dos funcionários sem seção**
- 19) nome dos participantes da organização que não tem superior**



# Exercícios



- 20) matrícula, nome dos funcionários, descrição da função, descrição da seção e descrição da diretoria dos funcionários que tem seção**
- 21) descrição do projeto, nome do participante e descrição da plataforma para os programadores**
- 22) nome do fabricante, modelo e ano do automóvel e nome do cliente para fabricantes, automóveis e clientes com vendas**
- 23) matrícula, nome e salário dos funcionários das seções 'MAN', 'ADM', 'FPG' e que tenham salário maior ou igual a \$3500 ( usar IN )**
- 24) descrição das seções que tem funcionários com salário maior que \$4000 (usar IN)**
- 25) nome e data de nascimento dos funcionários que nasceram entre 74 e 76 (usar BETWEEN)**
- 26) código e descrição das diretorias que tenham funcionários**
- 27) descrição dos projetos que foram desenvolvidos para Android**
- 28) nome dos clientes que compraram carros mexicanos**



# Exercícios



**29) modelo, ano e cor dos automóveis que não foram vendidos**

**30) descrição dos produtos que não tem entrada na segunda quinzena de Jan/2010**

**31) nome e data de nascimento do funcionário mais velho (usar ALL)**

**32) nome e grau de experiência do funcionário (supervisores e técnicos nível II são considerados “experientes”, técnicos nível I são considerados “intermediários”, auxiliares e trainees são considerados “iniciantes”)**

**33) código e descrição das seções que possuem técnicos nível II mas não possuem técnicos nível I**

**34) matrícula, nome e descrição da seção dos funcionários que não pertencem a diretoria de pessoal**