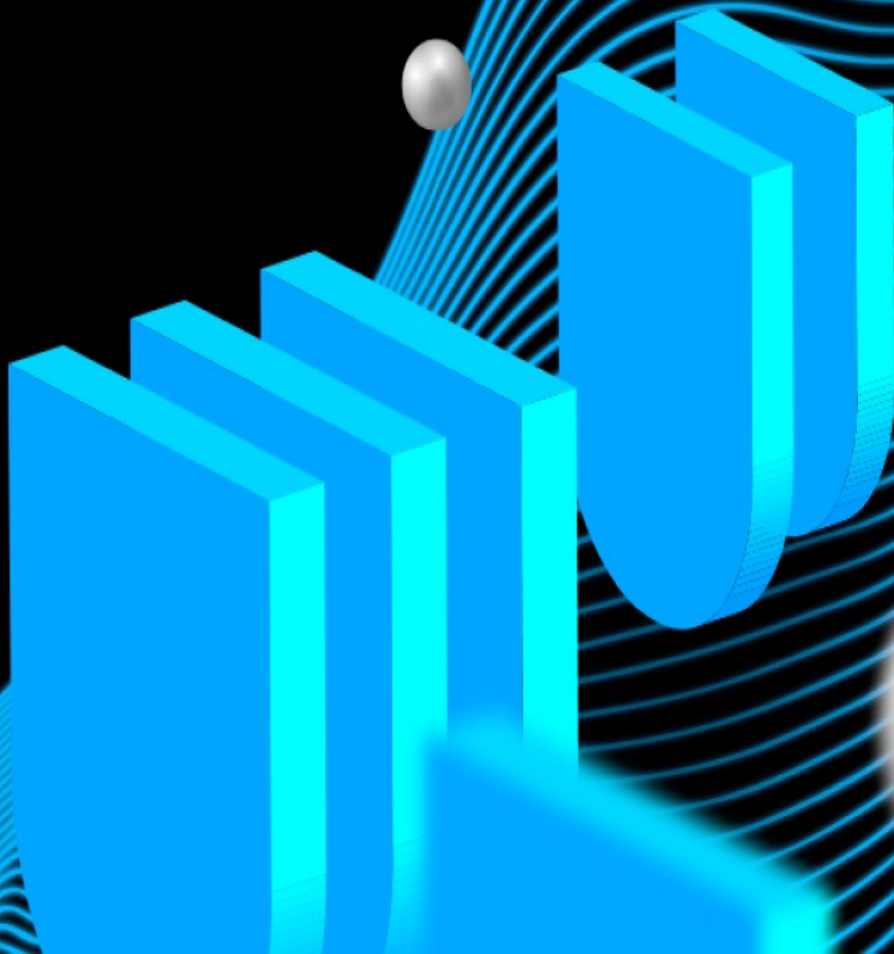




UNITAU
Universidade de Taubaté





SELECT (parte 1)

SELECT



O comando **SELECT** retorna linhas de zero ou mais tabelas.

```
SELECT [ ALL | DISTINCT [ ON ( expressão [, ...] ) ] ]  
      * | expressão [ AS nome_de_saída ] [, ...]  
      [ FROM item_do_from [, ...] ]  
      [ WHERE condição ]  
      [ GROUP BY expressão [, ...] ]  
      [ HAVING condição [, ...] ]  
      [ WINDOW window_name AS ( window_definition ) [, ...] ]  
      [ { UNION | INTERSECT | EXCEPT } [ ALL ] seleção ]  
      [ ORDER BY expressão [ ASC | DESC | USING operador ] [, ...] ]  
      [ OFFSET início { ROW | ROWS } ]  
      [ FETCH { FIRST | NEXT } [ contador ] { ROW | ROWS } { ONLY | WITH TIES } ]  
      [ FOR lock_strength [ OF nome_da_tabela [, ...] ] [NOWAIT | SKIP LOCKED] ]
```


SELECT



Exemplo:

```
SELECT * FROM curso;
```

codigo	 	nome
-----+-----		
0001	 	Tecnologia da Informacao
0002	 	Engenharia da Computacao
0003	 	Sistemas de Informacao

SELECT



O comando **SELECT** pode ser usado também com expressões, sem a necessidade de utilizar nenhuma tabela.

Exemplo:

```
SELECT 4^2, random();
```

?column?	random
16	0.00814559450373054

SELECT



É possível utilizar um alias para as colunas obtidas na seleção.

Exemplo:

```
SELECT 4^2 AS quadrado, random() AS aleatorio;
```

quadrado	aleatorio
16	0.960566652007401

SELECT



O comando **SELECT** permite que se determine uma lista dos campos a selecionar.

Exemplo:

```
SELECT nome, rg FROM aluno;
```

nome		rg
-----+-----		
Ana Lucia		20143531
Luis Claudio		22336362
Marcelo		25343256
Debora		20356328
Fernanda		26344325
Alvaro		21764527
Claudio		23336368
Andrea		28456474
Carla		23636731
Fernanda		29563735

DISTINCT



O padrão do comando **SELECT** é retornar todas as linhas resultantes da seleção. A cláusula **DISTINCT** elimina as linhas duplicadas do resultado da seleção.

Exemplo:

```
SELECT serie, turma FROM sala;
```

```
SELECT DISTINCT serie, turma FROM sala;
```


DISTINCT



```
SELECT serie, turma FROM sala;
```

serie	turma
1	A
1	B
2	A
3	A
1	A
2	A
3	A

```
SELECT DISTINCT serie, turma FROM sala;
```

serie	turma
1	A
1	B
2	A
3	A

DISTINCT



A cláusula **DISTINCT** pode ser usada em um grupo de colunas específicas.

Exemplo:

```
SELECT DISTINCT ON (curso, turma) curso, turma, nome FROM aluno;
```

curso	turma	nome
0001	A	Ana Lucia
0001	B	Debora
0002	A	Claudio
		Fernanda

DISTINCT



Pode ser usado **ORDER BY** para determinar qual registro será apresentado caso exista mais de um registro com os mesmos valores para as colunas da expressão do **DISTINCT**.

```
SELECT DISTINCT ON (curso, turma) curso, turma, nome FROM  
aluno ORDER BY curso, turma, nome;
```

curso	turma	nome
0001	A	Ana Lucia
0001	B	Alvaro
0002	A	Andrea
		Fernanda

TABLE



Quando se deseja selecionar todos os campos de todos os registros de uma tabela, é possível usar o comando

TABLE [ONLY] tabela [*]

Que é equivalente a

SELECT * FROM tabela

Exemplo:

TABLE curso;

codigo	nome
0001	Tecnologia da Informacao
0002	Engenharia da Computacao
0003	Sistemas de Informacao

JOIN



A cláusula **FROM** permite relacionar registros entre duas tabelas utilizando **JOIN** (junção).

FROM tabela1 CROSS JOIN tabela2

**FROM tabela1 { [INNER] | { LEFT | RIGHT | FULL } [OUTER] }
JOIN tabela2 ON boolean_expression**

**FROM tabela1 { [INNER] | { LEFT | RIGHT | FULL } [OUTER] }
JOIN tabela2 USING (join column list)**

**FROM tabela1 NATURAL { [INNER] | { LEFT | RIGHT | FULL }
[OUTER] } JOIN tabela2**

CROSS JOIN



CROSS JOIN relaciona cada registro da primeira tabela com todos registros da segunda tabela. Se as tabelas possuírem N e M linhas, respectivamente, a tabela resultante terá $N * M$ linhas.

Exemplo:

```
SELECT * FROM curso CROSS JOIN sala;
```


CROSS JOIN



codigo	nome	curso	serie	turma
0001	Tecnologia da Informacao	0001	1	A
0002	Engenharia da Computacao	0001	1	A
0003	Sistemas de Informacao	0001	1	A
0001	Tecnologia da Informacao	0001	1	B
0002	Engenharia da Computacao	0001	1	B
0003	Sistemas de Informacao	0001	1	B
0001	Tecnologia da Informacao	0001	2	A
0002	Engenharia da Computacao	0001	2	A
0003	Sistemas de Informacao	0001	2	A
0001	Tecnologia da Informacao	0001	3	A
0002	Engenharia da Computacao	0001	3	A
0003	Sistemas de Informacao	0001	3	A
0001	Tecnologia da Informacao	0002	1	A
0002	Engenharia da Computacao	0002	1	A
0003	Sistemas de Informacao	0002	1	A
0001	Tecnologia da Informacao	0002	2	A
0002	Engenharia da Computacao	0002	2	A
0003	Sistemas de Informacao	0002	2	A
0001	Tecnologia da Informacao	0002	3	A
0002	Engenharia da Computacao	0002	3	A
0003	Sistemas de Informacao	0002	3	A

INNER JOIN



INNER JOIN cria uma tabela resultante com as combinações em que as linhas das duas tabelas que atendam a condição de junção. A palavra **INNER** é opcional pois é o padrão do SQL.

Quando necessário especificar a que tabela o campo pertence, utiliza-se o nome da tabela e um ponto antes do nome do campo.

A condição **ON boolean_expression** seleciona as combinações que resultarem em verdadeiro ao avaliar a expressão.

INNER JOIN



Exemplo:

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

ou

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


INNER JOIN



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

codigo	curso	aluno	rg
0001	Tecnologia da Informacao	Ana Lucia	20143531
0001	Tecnologia da Informacao	Luis Claudio	22336362
0001	Tecnologia da Informacao	Marcelo	25343256
0001	Tecnologia da Informacao	Debora	20356328
0001	Tecnologia da Informacao	Fernanda	26344325
0001	Tecnologia da Informacao	Alvaro	21764527
0002	Engenharia da Computacao	Claudio	23336368
0002	Engenharia da Computacao	Andrea	28456474
0002	Engenharia da Computacao	Carla	23636731

LEFT OUTER JOIN



LEFT OUTER JOIN cria uma tabela resultante com as combinações em que as linhas das duas tabelas que atendam a condição de junção, seguidas das linhas da primeira tabela que não atendem a condição de junção.

Em todos os tipos de **OUTER JOIN**, a palavra **OUTER** é opcional. Os campos referentes a tabela que não atende a condição de junção terão valor nulo.

Exemplo:

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

ou

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


LEFT OUTER JOIN



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

codigo	curso	aluno	rg
0001	Tecnologia da Informacao	Ana Lucia	20143531
0001	Tecnologia da Informacao	Luis Claudio	22336362
0001	Tecnologia da Informacao	Marcelo	25343256
0001	Tecnologia da Informacao	Debora	20356328
0001	Tecnologia da Informacao	Fernanda	26344325
0001	Tecnologia da Informacao	Alvaro	21764527
0002	Engenharia da Computacao	Claudio	23336368
0002	Engenharia da Computacao	Andrea	28456474
0002	Engenharia da Computacao	Carla	23636731
0003	Sistemas de Informacao		

RIGHT OUTER JOIN



RIGHT OUTER JOIN cria uma tabela resultante com as combinações em que as linhas das duas tabelas que atendam a condição de junção, seguidas das linhas da segunda tabela que não atendem a condição de junção.

Exemplo:

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

ou

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

RIGHT OUTER JOIN



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

codigo	curso	aluno	rg
0001	Tecnologia da Informacao	Ana Lucia	20143531
0001	Tecnologia da Informacao	Luis Claudio	22336362
0001	Tecnologia da Informacao	Marcelo	25343256
0001	Tecnologia da Informacao	Debora	20356328
0001	Tecnologia da Informacao	Fernanda	26344325
0001	Tecnologia da Informacao	Alvaro	21764527
0002	Engenharia da Computacao	Claudio	23336368
0002	Engenharia da Computacao	Andrea	28456474
0002	Engenharia da Computacao	Carla	23636731
		Fernanda	29563735

FULL OUTER JOIN



FULL OUTER JOIN cria uma tabela resultante com as combinações em que as linhas das duas tabelas que atendam a condição de junção, seguidas das linhas que não atendem a condição de junção de cada uma das tabelas.

Exemplo:

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

ou

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


FULL OUTER JOIN



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

codigo	curso	aluno	rg
0001	Tecnologia da Informacao	Ana Lucia	20143531
0001	Tecnologia da Informacao	Luis Claudio	22336362
0001	Tecnologia da Informacao	Marcelo	25343256
0001	Tecnologia da Informacao	Debora	20356328
0001	Tecnologia da Informacao	Fernanda	26344325
0001	Tecnologia da Informacao	Alvaro	21764527
0002	Engenharia da Computacao	Claudio	23336368
0002	Engenharia da Computacao	Andrea	28456474
0002	Engenharia da Computacao	Carla	23636731
		Fernanda	29563735
0003	Sistemas de Informacao		

USING



A condição **USING** simplifica a escrita quando a condição de junção é a igualdade de colunas que tem o mesmo nome nas duas tabelas.

```
FROM tab1 JOIN tab2 USING ( col1, col2, ... )
```

Equivale a:

```
FROM tab1 JOIN tab2 ON tab1.col1=tab2.col1 AND  
tab1.col2=tab2.col2 AND ...
```

Quando é utilizada a condição **ON**, as colunas duplicadas aparecem duas vezes na tabela resultante. Quando é utilizada a condição **USING**, as colunas duplicadas aparecem apenas uma vez na tabela resultante.

Exemplo:

```
SELECT * FROM sala JOIN aluno ON sala.curso=aluno.curso AND  
sala.serie=aluno.serie AND sala.turma=aluno.turma;  
SELECT * FROM sala JOIN aluno USING (curso,serie,turma);
```


USING



```
SELECT * FROM sala JOIN aluno ON sala.curso=aluno.curso AND  
sala.serie=aluno.serie AND sala.turma=aluno.turma;
```

curso	serie	turma	matricula	nome	rg	curso	serie	turma
0001	1	A	1	Ana Lucia	20143531	0001	1	A
0001	1	A	2	Luis Claudio	22336362	0001	1	A
0001	1	A	3	Marcelo	25343256	0001	1	A
0001	1	B	4	Debora	20356328	0001	1	B
0001	1	B	5	Fernanda	26344325	0001	1	B
0001	1	B	6	Alvaro	21764527	0001	1	B
0002	1	A	7	Claudio	23336368	0002	1	A
0002	1	A	8	Andrea	28456474	0002	1	A
0002	2	A	9	Carla	23636731	0002	2	A

USING



```
SELECT * FROM sala JOIN aluno USING (curso,serie,turma);
```

curso	serie	turma	matricula	nome	rg
0001	1	A	1	Ana Lucia	20143531
0001	1	A	2	Luis Claudio	22336362
0001	1	A	3	Marcelo	25343256
0001	1	B	4	Debora	20356328
0001	1	B	5	Fernanda	26344325
0001	1	B	6	Alvaro	21764527
0002	1	A	7	Claudio	23336368
0002	1	A	8	Andrea	28456474
0002	2	A	9	Carla	23636731

FULL OUTER JOIN



NATURAL é uma forma abreviada de **USING**, usando como condição de junção a igualdade de todas as colunas que tem o mesmo nome nas duas tabelas.

```
SELECT * FROM sala NATURAL JOIN aluno;
```

curso	serie	turma	matricula	nome	rg
0001	1	A	1	Ana Lucia	20143531
0001	1	A	2	Luis Claudio	22336362
0001	1	A	3	Marcelo	25343256
0001	1	B	4	Debora	20356328
0001	1	B	5	Fernanda	26344325
0001	1	B	6	Alvaro	21764527
0002	1	A	7	Claudio	23336368
0002	1	A	8	Andrea	28456474
0002	2	A	9	Carla	23636731

Exercícios



Usando JOIN, obtenha os seguintes dados das tabelas:

- a) código e descrição da diretoria, código e descrição da seção, para as diretorias que tenham seção
- b) nome do funcionário, descrição da função e salário do funcionário, para para as funções que tenham funcionário
- c) código e nome do banco, agencia, numero e descrição da conta para os bancos que tenham conta
- d) código e descrição da seção, nome, salario e data de nascimento do funcionário, para as seções que tenham funcionários
- e) código e descrição da seção, nome, salario e data de nascimento do funcionário, mesmo para as seções sem funcionários
- f) nome do fabricante, modelo, cor e ano do automóvel mesmo para os fabricantes sem automóvel
- g) nome do funcionário, descrição da função e salário do funcionário, mesmo para as funções sem funcionário

Exercícios



- h) código e descrição da seção, nome, salario e data de nascimento do funcionário, mesmo para as seções sem funcionários ou funcionários sem seção
- i) código e nome do banco, código, agencia, numero e descrição da conta mesmo os bancos sem conta ou conta sem banco
- j) descrição da diretoria, descrição da seção e nome do funcionário, para as diretorias/secoes que tenham funcionários
- k) data da venda, nome da revenda, nome do cliente, modelo do automovel para as revendas, clientes e automoveis com vendas
- l) nome do fabricante, modelo e ano do automóvel e nome do cliente para fabricantes, automóveis e clientes com vendas
- m) descrição da diretoria, descrição da seção e nome do funcionário, mesmo para as diretorias/seções que não tenham funcionários

Exercícios



- n) descrição da diretoria, descrição da seção e nome do funcionário, mesmo para as diretorias que não tenham seções/funcionários
- o) descrição da diretoria, descrição da seção e nome do funcionário, mesmo para as diretorias que não tenham seções/funcionários ou diretoria/secao que não tenham funcionários
- p) descrição da diretoria, descrição da seção e nome do funcionário, mesmo para os funcionários sem seção
- q) descrição da seção, nome do funcionário, descrição da função, salario do funcionário, mesmo para funcionários sem seção