

Linguagem SQL para consulta - apelidos de coluna, classificações, eliminando linhas em duplicata e operadores avançados

Prof^a: Miriã da Silveira Coelho Corrêa – M.Sc

Email: miriacoeelho@gmail.com

Sumário

Nome de atributos ambíguos

Apelidos ou variáveis de tupla

LIKE

BETWEEN

IN e NOT IN

IS NULL e IS NOT NULL

Ordem dos resultados da consulta

BD Empresa

- Para os exemplos que serão dados nesta aula, considere o esquema do banco de dados empresa mostrado abaixo:*

FUNCIONÁRIO

pnome	minicial	unome	<u>cpf</u>	datanasc	endereco	sexo	salario	cpf_supervisor	dnr
-------	----------	-------	------------	----------	----------	------	---------	----------------	-----

DEPARTAMENTO

dnome	<u>dnumero</u>	cpf_gerente	data_inicio_gerente	data_criacao
-------	----------------	-------------	---------------------	--------------

LOCALIZACAO_DEP

<u>dnumero</u>	<u>dlocal</u>
----------------	---------------

TRABALHA_EM

<u>fcpf</u>	<u>pnr</u>	horas
-------------	------------	-------

PROJETO

projnome	<u>projnumero</u>	projlocal	dnum
----------	-------------------	-----------	------

DEPENDENTE

<u>fcpf</u>	<u>nome dependente</u>	sexo	datanasc	parentesco
-------------	------------------------	------	----------	------------

Nomes de atributos ambíguos

- *Em SQL, o mesmo nome pode ser usado para dois (ou mais) atributos, desde que estes estejam em tabelas diferentes.*
- *Se isto acontecer, e uma consulta em múltiplas tabelas se referir a dois ou mais atributos com o mesmo nome, é preciso qualificar o nome do atributo com o nome da tabela, para evitar ambiguidade.*

Nomes de atributos ambíguos

- *Em SQL, o mesmo nome pode ser usado para dois (ou mais) atributos, desde que estes estejam em tabelas diferentes.*
- *Se isto acontecer, e uma consulta em múltiplas tabelas se referir a dois ou mais atributos com o mesmo nome, é preciso qualificar o nome do atributo com o nome da tabela, para evitar ambiguidade.*
- *Basta prefixar o nome da tabela ao nome do atributo e separando os dois por um ponto.*

Exemplo

- Suponha que, os atributos *dnr* e *unome* da tabela *FUNCIONARIO* fossem chamados de *dnumero* e *nome*, e o atributo *dnome* de *DEPARTAMENTO* também fosse chamado *nome*.

FUNCIONÁRIO

pnome	minicial	unome	<u>cpf</u>	datanasc	endereco	sexo	salario	cpf_supervisor	dnr
-------	----------	-------	------------	----------	----------	------	---------	----------------	-----



FUNCIONÁRIO

pnome	minicial	nome	<u>cpf</u>	datanasc	endereco	sexo	salario	cpf_supervisor	dnumero
-------	----------	-------------	------------	----------	----------	------	---------	----------------	----------------

DEPARTAMENTO

dnome	<u>dnumero</u>	cpf_gerente	data_inicio_gerente	data_criacao
-------	----------------	-------------	---------------------	--------------



DEPARTAMENTO

nome	<u>dnumero</u>	cpf_gerente	data_inicio_gerente	data_criacao
-------------	----------------	-------------	---------------------	--------------

Exemplo de consulta

- C22: Recuperar o nome e endereço de todos os funcionários que trabalham para o departamento de 'Pesquisa'.*

Funcionário

	pnome	inicial	nome	cpf		sexo	salario	cpf_supervisor	dnumero
1	Jennifer	S	Souza	98765432168		F	43000.00	88866555576	4
2	André	V	Pereira	98798798733		M	25000.00	98765432168	4
3	Jorge	E	Brito	88866555576		M	55000.00		1
4	João	B	Silva	12345678966		M	30000.00	33344555587	5
5	Fernando	T	Wong	33344555587		M	40000.00	88866555576	5
6	Joice	A	Leite	45345345376	...	F	25000.00	33344555587	5
7	Jorge	L	Caio	10000000000		M	26000.00		5
8	Alice	J	Zelaya	99988777767		F	25000.00	98765432168	4
9	Ronaldo	K	Lima	66688444476		M	38000.00	33344555587	

Departamento

	dnumero	nome	cpf_gerente	data_inicio_gerente
1	1	Matriz	88866555576	1981-06-19
2	3	Compras	33344555587	
3	4	Administração	98765432168	1995-01-01
4	5	Pesquisa	33344555587	1988-05-22

Exemplo de consulta

- *C22: Recuperar o nome e endereço de todos os funcionários que trabalham para o departamento de 'Pesquisa'.*

Para evitar ambiguidade faríamos esta consulta prefixando os nomes de atributo nome e dnumero em C22 para especificar a quais estamos nos referindo, pois os mesmos nomes de atributo são usados nas duas relações. Deste modo teríamos:

Exemplo de consulta

```
SELECT pnome, funcionario.nome, endereco  
FROM funcionario, departamento  
WHERE departamento.nome= 'Pesquisa' AND  
departamento.dnumero= funcionario.dnumero;
```

	pnome	nome	endereco
1	João	Silva	Rua das Flores, 751, São Paulo, SP
2	Fernando	Wong	Rua da Lapa, 34, São Paulo, SP
3	Ronaldo	Lima	Rua Rebouças, 65, Piracicaba, SP
4	Joice	Leite	Avenida Lucas Obes, 74, São Paulo, SP

- Nomes de atributo totalmente qualificados podem ser usados por clareza mesmo que não haja ambiguidade nos nomes de atributo.

Exemplo de consulta

SELECT funcionario.pnome, funcionario.unome,
funcionario.endereco

FROM funcionario, departamento

WHERE departamento.dnome= 'Pesquisa' **AND**
departamento.dnumero= funcionario.dnr;

	pnome	unome	endereco
1	João	Silva	Rua das Flores, 751, São Paulo, SP
2	Fernando	Wong	Rua da Lapa, 34, São Paulo, SP
3	Ronaldo	Lima	Rua Rebouças, 65, Piracicaba, SP
4	Joice	Leite	Avenida Lucas Obes, 74, São Paulo, SP

Exemplo de consulta

- *A ambiguidade de atributos também surge em consultas que se referem à mesma relação duas vezes, como no exemplo a seguir.*
- *C23: Para cada funcionário, recupere o primeiro e o último nome do funcionário e o primeiro e o último nome de seu supervisor imediato.*

Funcionário

	pnome	minicial	unome	cpf	datanasc	endereco		dnr
1	Jorge	L	Caio	10000000000	1955-12-01	Rua Carolina Silva, 832, São Paulo, SP		5
2	João	B	Silva	12345678966	1965-01-09	Rua das Flores, 751, São Paulo, SP		5
3	Fernando	T	Wong	33344555587	1955-12-08	Rua da Lapa, 34, São Paulo, SP		5
4	Joice	A	Leite	45345345376	1972-07-31	Avenida Lucas Obes, 74, São Paulo, SP		5
5	Ronaldo	K	Lima	66688444476	1962-09-15	Rua Rebouças, 65, Piracicaba, SP	...	
6	Jorge	E	Brito	88866555576	1937-11-10	Rua do Horto, 35, São Paulo, SP		1
7	Jennifer	S	Souza	98765432168	1941-06-20	Avenida Artur de Lima, 54, Santo André, SP		4
8	André	V	Pereira	98798798733	1969-03-29	Rua Timbira, 35, São Paulo, SP		4
9	Alice	J	Zelaya	99988777767	1968-01-19	Rua Souza Lima, 35, Curitiba, PR		4

Exemplo de consulta

```
SELECT f.pnome, f.unome, s.pnome, s.unome  
FROM funcionario AS f, funcionario AS s  
WHERE f.cpf_supervisor = s.cpf;
```

	pnome	unome	pnome	unome
1	Alice	Zelaya	Jennifer	Souza
2	Jennifer	Souza	Jorge	Brito
3	André	Pereira	Jennifer	Souza
4	João	Silva	Fernando	Wong
5	Fernando	Wong	Jorge	Brito
6	Ronaldo	Lima	Fernando	Wong
7	Joice	Leite	Fernando	Wong

Apelidos ou variáveis de tupla

- Neste caso, precisamos declarar nomes de relação alternativos f e s , chamados **apelidos** ou **variáveis de tupla**, para a tabela funcionario.
- Um apelido pode vir após a palavra-chave AS, como mostramos em C23, ou pode vir diretamente após o nome da tabela – por exemplo, escrevendo funcionario f , funcionario s na cláusula FROM de C23.

Apelidos ou variáveis de tupla

- *Também é possível renomear os atributos da relação dentro da consulta em SQL, dando-lhe apelidos.*
- *Por exemplo, se escrevermos:
funcionario AS f(pn, mi, um, cpf, dn, end, sexo, sal, scpf, dnr) na cláusula FROM, pn torna-se um apelido para pnome, mi para minicial, um para unome, e assim por diante.*

Apelidos ou variáveis de tupla

- *Em C23, podemos pensar em f e s como duas cópias diferentes da tabela funcionario;*
- *a primeira, f , representa funcionários no papel de supervisionados ou subordinados;*
- *a segunda, s , representa os funcionários no papel de supervisores.*

Apelidos ou variáveis de tupla

- *Naturalmente, na realidade existe apenas uma relação funcionario, e a condição de junção serve para juntar a própria relação, combinando as tuplas que satisfazem a condição de junção $f.cpf_supervisor = s.cpf$.*

- *SQL normalmente trata uma tabela não como um conjunto, mas como um **multiconjunto**;*
- *tuplas duplicadas podem aparecer mais de uma vez em uma tabela, e no resultado de uma consulta.*
- *A SQL não elimina automaticamente tuplas duplicadas nos resultados das consultas, pelos seguintes motivos:*

- *A eliminação de duplicatas é uma operação dispendiosa. Um modo de implementá-la é classificar as tuplas primeiro e depois eliminar as duplicatas.*
- *O usuário pode querer ver as tuplas duplicadas no resultado de uma consulta.*
- *Quando uma função agregada é aplicada às tuplas, na maioria dos casos não queremos eliminar duplicatas.*

- *Uma tabela SQL com uma chave é restrita a ser um conjunto, uma vez que o valor de chave precisa ser distinto em cada tupla.*
- *Se quisermos eliminar tuplas duplicadas no resultado de uma consulta SQL, usamos a palavra-chave **DISTINCT** na cláusula SELECT, significando que apenas as tuplas distintas deverão permanecer no resultado.*

Exemplo de consulta

- *C24: Recuperar o salário de cada funcionário.*

Funcionário

	pnome	minicial	unome	cpf	datanasc	endereco		dnr
1	Jorge	L	Caio	10000000000	1955-12-01	Rua Carolina Silva, 832, São Paulo, SP		5
2	João	B	Silva	12345678966	1965-01-09	Rua das Flores, 751, São Paulo, SP		5
3	Fernando	T	Wong	33344555587	1955-12-08	Rua da Lapa, 34, São Paulo, SP		5
4	Joice	A	Leite	45345345376	1972-07-31	Avenida Lucas Obes, 74, São Paulo, SP		5
5	Ronaldo	K	Lima	66688444476	1962-09-15	Rua Rebouças, 65, Piracicaba, SP	...	
6	Jorge	E	Brito	88866555576	1937-11-10	Rua do Horto, 35, São Paulo, SP		1
7	Jennifer	S	Souza	98765432168	1941-06-20	Avenida Artur de Lima, 54, Santo André, SP		4
8	André	V	Pereira	98798798733	1969-03-29	Rua Timbira, 35, São Paulo, SP		4
9	Alice	J	Zelaya	99988777767	1968-01-19	Rua Souza Lima, 35, Curitiba, PR		4

Exemplo de consulta

- *C24: Recuperar o salário de cada funcionário.*

SELECT ALL salario
FROM funcionario

	salario
1	43000.00
2	25000.00
3	55000.00
4	30000.00
5	40000.00
6	25000.00
7	26000.00
8	25000.00
9	38000.00

- *Percebemos que existem funcionários com o mesmo salário, logo esse valor aparecerá muitas vezes no resultado da consulta.*
- *Se estivermos interessados apenas em valores de salário distintos, queremos que cada valor apareça apenas uma vez, independentemente de quantos funcionários ganham esse salário. Usando a palavra-chave **DISTINCT**, como em C25.*

Exemplo de consulta

- *C25: Recuperar os valores de salário distintos de cada funcionário.*

Funcionário

	pnome	minicial	unome	cpf	datanasc	endereco		dnr
1	Jorge	L	Caio	10000000000	1955-12-01	Rua Carolina Silva, 832, São Paulo, SP		5
2	João	B	Silva	12345678966	1965-01-09	Rua das Flores, 751, São Paulo, SP		5
3	Fernando	T	Wong	33344555587	1955-12-08	Rua da Lapa, 34, São Paulo, SP		5
4	Joice	A	Leite	45345345376	1972-07-31	Avenida Lucas Obes, 74, São Paulo, SP		5
5	Ronaldo	K	Lima	66688444476	1962-09-15	Rua Rebouças, 65, Piracicaba, SP	...	
6	Jorge	E	Brito	88866555576	1937-11-10	Rua do Horto, 35, São Paulo, SP		1
7	Jennifer	S	Souza	98765432168	1941-06-20	Avenida Artur de Lima, 54, Santo André, SP		4
8	André	V	Pereira	98798798733	1969-03-29	Rua Timbira, 35, São Paulo, SP		4
9	Alice	J	Zelaya	99988777767	1968-01-19	Rua Souza Lima, 35, Curitiba, PR		4

Exemplo de consulta

- *C25: Recuperar os valores de salário distintos de cada funcionário.*

SELECT DISTINCT salario
FROM funcionario

	salario
▶	26000.00
	30000.00
	40000.00
	25000.00
	38000.00
	55000.00
	43000.00

- *Em geral, uma consulta `SELECT DISTINCT` elimina duplicatas, enquanto uma consulta com `SELECT ALL` não elimina. A cláusula `SELECT` sem `ALL` ou `DISTINCT` é equivalente a `SELECT ALL`.*

- *Este operador é usado para executar pesquisas especiais;*
- *Cadeias parciais são especificadas usando dois caracteres reservados:*
 - *% substitui um número qualquer de zero ou mais caracteres;*
 - *o sublinhado (_) substitui um único caractere.*

Exemplo de consulta

- *C12: Recuperar todos os funcionários cujo endereço esteja em São Paulo, SP.*

Funcionário

	pnome	minicial	unome	cpf	datanasc	endereco		dnr
1	Jorge	L	Caio	10000000000	1955-12-01	Rua Carolina Silva, 832, São Paulo, SP		5
2	João	B	Silva	12345678966	1965-01-09	Rua das Flores, 751, São Paulo, SP		5
3	Fernando	T	Wong	33344555587	1955-12-08	Rua da Lapa, 34, São Paulo, SP		5
4	Joice	A	Leite	45345345376	1972-07-31	Avenida Lucas Obes, 74, São Paulo, SP		5
5	Ronaldo	K	Lima	66688444476	1962-09-15	Rua Rebouças, 65, Piracicaba, SP	...	
6	Jorge	E	Brito	88866555576	1937-11-10	Rua do Horto, 35, São Paulo, SP		1
7	Jennifer	S	Souza	98765432168	1941-06-20	Avenida Artur de Lima, 54, Santo André, SP		4
8	André	V	Pereira	98798798733	1969-03-29	Rua Timbira, 35, São Paulo, SP		4
9	Alice	J	Zelaya	99988777767	1968-01-19	Rua Souza Lima, 35, Curitiba, PR		4

Exemplo de consulta

- *C12: Recuperar todos os funcionários cujo endereço esteja em São Paulo, SP.*

SELECT *pnome, unome*

FROM *funcionario*

WHERE *Endereco* **LIKE** *'%São Paulo, SP';*

	pnome	unome
1	André	Pereira
2	Jorge	Brito
3	João	Silva
4	Fernando	Wong
5	Joice	Leite
6	Jorge	Caio

- *Para ilustrar a correspondência dos padrões considere os seguintes exemplos:*
 - *'Maria%': localiza qualquer string começando com "Maria";*
 - *'%ria%': localiza qualquer string que contém "ria";*
 - *'_ _ _': localiza qualquer string com exatamente 3 caracteres;*
 - *'_ _ _%': localiza qualquer string com pelo menos 3 caracteres.*

Exemplo de consulta

- *C13: Encontrar todos os funcionários em que o primeiro nome tem dois caracteres quaisquer seguido de ice.*

Funcionário

	pnome	minicial	unome	cpf	datanasc	endereco	dnr
1	Jorge	L	Caio	100000000000	1955-12-01	Rua Carolina Silva, 832, São Paulo, SP	5
2	João	B	Silva	12345678966	1965-01-09	Rua das Flores, 751, São Paulo, SP	5
3	Fernando	T	Wong	33344555587	1955-12-08	Rua da Lapa, 34, São Paulo, SP	5
4	Joice	A	Leite	45345345376	1972-07-31	Avenida Lucas Obes, 74, São Paulo, SP	5
5	Ronaldo	K	Lima	66688444476	1962-09-15	Rua Rebouças, 65, Piracicaba, SP	...
6	Jorge	E	Brito	88866555576	1937-11-10	Rua do Horto, 35, São Paulo, SP	1
7	Jennifer	S	Souza	98765432168	1941-06-20	Avenida Artur de Lima, 54, Santo André, SP	4
8	André	V	Pereira	98798798733	1969-03-29	Rua Timbira, 35, São Paulo, SP	4
9	Alice	J	Zelaya	99988777767	1968-01-19	Rua Souza Lima, 35, Curitiba, PR	4

Exemplo de consulta

- *C13: Encontrar todos os funcionários em que o primeiro nome tem dois caracteres quaisquer seguido de ice.*

```
SELECT pnome, unome  
FROM funcionario  
WHERE pnome LIKE '__ice';
```

	pnome	unome
1	Alice	Zelaya
2	Joice	Leite

ESCAPE

ESCAPE

- *Além disso, precisamos de uma regra para especificar apóstrofos ou aspas simples (' ') se eles tiverem de ser incluídos em uma cadeia, pois são usados para iniciar e terminar cadeias.*
- *Se um apóstrofo (') for necessário, ele será representado como dois apóstrofos consecutivos (''), de modo que não será interpretado como o término da cadeia.*

BETWEEN

- *Outro operador de comparação que pode ser usado é o **BETWEEN**.*
- *Usado para recuperar linhas com base em uma faixa de valores.*

Exemplo de consulta

- *C14: Recuperar todos os funcionários no departamento 5 cujo salário esteja entre R\$30000 e R\$40000.*

Funcionário

	pnome	minicial	unome		salario		dnr
1	Jorge	L	Caio		26000.00		5
2	João	B	Silva		30000.00		5
3	Fernando	T	Wong		40000.00		5
4	Joice	A	Leite		25000.00		5
5	Ronaldo	K	Lima		38000.00		
6	Jorge	E	Brito	...	55000.00	...	1
7	Jennifer	S	Souza		43000.00		4
8	André	V	Pereira		25000.00		4
9	Alice	J	Zelaya		25000.00		4

Exemplo de consulta

- *C14: Recuperar todos os funcionários no departamento 5 cujo salário esteja entre R\$30000 e R\$40000.*

SELECT *

FROM funcionario

WHERE (salario **BETWEEN** 30000 **AND** 40000) **AND**
dnr=5;

	pnome	inicial	unome	cpf	datanasc	endereco	sexo	salario	cpf_supervisor	dnr
1	João	B	Silva	12345678966	1965-01-	Rua das Flores, 75	M	30000.00	33344555587	5
2	Fernando	T	Wong	33344555587	1955-12-	Rua da Lapa, 34,	S M	40000.00	88866555576	5
3	Ronaldo	K	Lima	66688444476	1962-09-	Rua Rebouças, 65,	M	38000.00	33344555587	5

BETWEEN

- A condição (salario **BETWEEN** 30000 **AND** 40000) em C14 é equivalente à condição ((salario >= 30.000) **AND** (salario <= 40.000)).

- 101001010100111101000010010111010010 110101010101110100004100001010010100

Exemplo de consulta

- *C15: Recuperar o nome e endereço de todos os funcionários que trabalham para o departamento de 'Pesquisa'.*

Funcionário

	pnome	minicial	unome	cpf	datanasc	endereco	dnr
1	Jorge	L	Caio	100000000000	1955-12-01	Rua Carolina Silva, 832, São Paulo, SP	5
2	João	B	Silva	12345678966	1965-01-09	Rua das Flores, 751, São Paulo, SP	5
3	Fernando	T	Wong	33344555587	1955-12-08	Rua da Lapa, 34, São Paulo, SP	5
4	Joice	A	Leite	45345345376	1972-07-31	Avenida Lucas Obes, 74, São Paulo, SP	5
5	Ronaldo	K	Lima	66688444476	1962-09-15	Rua Rebouças, 65, Piracicaba, SP	...
6	Jorge	E	Brito	88866555576	1937-11-10	Rua do Horto, 35, São Paulo, SP	1
7	Jennifer	S	Souza	98765432168	1941-06-20	Avenida Artur de Lima, 54, Santo André, SP	4
8	André	V	Pereira	98798798733	1969-03-29	Rua Timbira, 35, São Paulo, SP	4
9	Alice	J	Zelaya	99988777767	1968-01-19	Rua Souza Lima, 35, Curitiba, PR	4

Departamento

	dnumero	dnome	cpf_gerente	data_inicio_gerente
1	1	Matriz	88866555576	1981-06-19
2	3	Compras	33344555587	
3	4	Administração	98765432168	1995-01-01
4	5	Pesquisa	33344555587	1988-05-22

Exemplo de consulta

- *C2: Recuperar o nome e endereço de todos os funcionários que trabalham para o departamento de 'Pesquisa'.*

SELECT pnome, unome, endereco
FROM funcionario, departamento
WHERE dnome='Pesquisa' AND dnumero = dnr;

	pnome	unome	endereco
1	João	Silva	Rua das Flores, 751, São Paulo, SP
2	Fernando	Wong	Rua da Lapa, 34, São Paulo, SP
3	Ronaldo	Lima	Rua Rebouças, 65, Piracicaba, SP
4	Joice	Leite	Avenida Lucas Obes, 74, São Paulo, SP

Exemplo de consulta

- *C15: Recuperar o nome e endereço de todos os funcionários que trabalham para o departamento de 'Pesquisa'.*

```
SELECT pnome, unome, endereco  
FROM funcionario  
WHERE dnr IN (SELECT dnumero  
FROM departamento  
WHERE dnome='Pesquisa');
```

	pnome	unome	endereco
1	João	Silva	Rua das Flores, 751, São Paulo, SP
2	Fernando	Wong	Rua da Lapa, 34, São Paulo, SP
3	Ronaldo	Lima	Rua Rebouças, 65, Piracicaba, SP
4	Joice	Leite	Avenida Lucas Obes, 74, São Paulo, SP

IS NULL

- *A SQL permite o uso de valores nulos para indicar a ausência de informações sobre o valor de um atributo.*
- *Podemos usar a palavra-chave especial NULL em um predicado para testar a presença de um valor nulo.*

Exemplo de consulta

- C16: Recuperar o nome e o número do departamento dos empregados que ainda não possuem supervisor.*

Funcionário

	pnome	minicial	unome	cpf		sexo	salario	cpf_supervisor	dnr
1	Jennifer	S	Souza	98765432168		F	43000.00	88866555576	4
2	André	V	Pereira	98798798733		M	25000.00	98765432168	4
3	Jorge	E	Brito	88866555576		M	55000.00		1
4	João	B	Silva	12345678966		M	30000.00	33344555587	5
5	Fernando	T	Wong	33344555587		M	40000.00	88866555576	5
6	Joice	A	Leite	45345345376	...	F	25000.00	33344555587	5
7	Jorge	L	Caio	10000000000		M	26000.00		5
8	Alice	J	Zelaya	99988777767		F	25000.00	98765432168	4
9	Ronaldo	K	Lima	66688444476		M	38000.00	33344555587	

Exemplo de consulta

- *C16: Recuperar o nome e o número do departamento dos empregados que ainda não possuem supervisor.*

```
SELECT pnome,dnr  
FROM funcionario  
WHERE cpf_supervisor IS NULL;
```

	pnome	dnr
1	Jorge	1
2	Jorge	5

- O predicado **IS NOT NULL** testa a ausência de uma valor nulo.
- C17: Recuperar o nome e o número do departamento dos empregados que possuem supervisor.

Funcionário

	pnome	minicial	unome	cpf		sexo	salario	cpf_supervisor	dnr
1	Jennifer	S	Souza	98765432168		F	43000.00	88866555576	4
2	André	V	Pereira	98798798733		M	25000.00	98765432168	4
3	Jorge	E	Brito	88866555576		M	55000.00		1
4	João	B	Silva	12345678966		M	30000.00	33344555587	5
5	Fernando	T	Wong	33344555587		M	40000.00	88866555576	5
6	Joice	A	Leite	45345345376	...	F	25000.00	33344555587	5
7	Jorge	L	Caio	100000000000		M	26000.00		5
8	Alice	J	Zelaya	99988777767		F	25000.00	98765432168	4
9	Ronaldo	K	Lima	66688444476		M	38000.00	33344555587	

IS NOT NULL

- *C17: Recuperar o nome e o número do departamento dos empregados que possuem supervisor.*

```
SELECT pnome,dnr  
FROM funcionario  
WHERE cpf_supervisor IS NOT NULL;
```

	pnome	dnr
1	Jennifer	4
2	André	4
3	João	5
4	Fernando	5
5	Joice	5
6	Alice	4
7	Ronaldo	

Exemplo de consulta

- *C18: Recuperar todas as informações dos funcionários cujo Unome não seja Souza.*

Funcionário

	pnome	minicial	unome	cpf		sexo	salario	cpf_supervisor	dnr
1	Jennifer	S	Souza	98765432168		F	43000.00	88866555576	4
2	André	V	Pereira	98798798733		M	25000.00	98765432168	4
3	Jorge	E	Brito	88866555576		M	55000.00		1
4	João	B	Silva	12345678966		M	30000.00	33344555587	5
5	Fernando	T	Wong	33344555587		M	40000.00	88866555576	5
6	Joice	A	Leite	45345345376	...	F	25000.00	33344555587	5
7	Jorge	L	Caio	100000000000		M	26000.00		5
8	Alice	J	Zelaya	99988777767		F	25000.00	98765432168	4
9	Ronaldo	K	Lima	66688444476		M	38000.00	33344555587	

- *C18: Recuperar todas as informações dos funcionários cujo Unome não seja Souza.*

SELECT *

FROM funcionario

WHERE *unome* **NOT IN** ('Souza');

	pnome	inicial	unome	cpf	datanasc		sexo	cpf_supervisor	dnr
1	João	B	Silva	12345678966	1965-01-09		M	33344555587	5
2	Fernando	T	Wong	33344555587	1955-12-08		M	88866555576	5
3	Alice	J	Zelaya	99988777767	1968-01-19		F	98765432168	4
4	Ronaldo	K	Lima	66688444476	1962-09-15	■ ■ ■	M	33344555587	5
5	Joice	A	Leite	45345345376	1972-07-31		F	33344555587	5
6	André	V	Pereira	98798798733	1969-03-29		M	98765432168	4
7	Jorge	E	Brito	88866555576	1937-11-10		M		1

Operadores lógicos

- *Combina resultado de duas condições de componentes para produzir um único resultado. Segue abaixo os operadores em ordem de precedência.*

Operador	Significado
NOT	Retorna verdadeiro (true) se a condição seguinte for falsa (false).
AND	Retorna verdadeiro (true) se as condições componentes forem verdadeira (true).
OR	Retorna verdadeiro (true) se cada condição de componentes for verdadeira (true).

Exemplo de consulta

- C19: Recuperar todas as informações dos funcionários que possuem salário menor que 30000 ou maior que 50000.*

Funcionário

	pnome	minicial	unome	cpf		sexo	salario	cpf_supervisor	dnr
1	Jennifer	S	Souza	98765432168		F	43000.00	88866555576	4
2	André	V	Pereira	98798798733		M	25000.00	98765432168	4
3	Jorge	E	Brito	88866555576		M	55000.00		1
4	João	B	Silva	12345678966		M	30000.00	33344555587	5
5	Fernando	T	Wong	33344555587		M	40000.00	88866555576	5
6	Joice	A	Leite	45345345376	...	F	25000.00	33344555587	5
7	Jorge	L	Caio	10000000000		M	26000.00		5
8	Alice	J	Zelaya	99988777767		F	25000.00	98765432168	4
9	Ronaldo	K	Lima	66688444476		M	38000.00	33344555587	

Exemplo de consulta

- C19: Recuperar todas as informações dos funcionários que possuem salário menor que 30000 ou maior que 50000.*

SELECT *

FROM funcionario

WHERE salario NOT BETWEEN 30000 **AND** 50000;

	pnome	inicial	unome	cpf		sexo	salario	cpf_supervisor	dnr
1	André	V	Pereira	98798798733		M	25000.00	98765432168	4
2	Jorge	E	Brito	88866555576		M	55000.00		1
3	Joice	A	Leite	45345345376		F	25000.00	33344555587	5
4	Jorge	L	Caio	10000000000	■ ■ ■	M	26000.00		5
5	Alice	J	Zelaya	99988777767		F	25000.00	98765432168	4

Ordem dos resultados da consulta

- A *SQL* permite que o usuário ordene as tuplas no resultado de uma consulta pelos valores de um ou mais dos atributos que aparecem, usando a cláusula **ORDER BY**.
- A ordem padrão é ordem crescente de valores. Podemos especificar a palavra-chave **DESC** se quisermos ver o resultado em uma ordem decrescente de valores. A palavra-chave **ASC** pode ser usada para especificar a ordem crescente explicitamente.

Exemplo de consulta

- *C20: Recuperar o primeiro nome e o salário do funcionário em que o salário é maior ou igual a 30000, ordenado pelo primeiro nome.*

Funcionário

	pnome	minicial	unome	cpf		sexo	salario	cpf_supervisor	dnr
1	Jennifer	S	Souza	98765432168		F	43000.00	88866555576	4
2	André	V	Pereira	98798798733		M	25000.00	98765432168	4
3	Jorge	E	Brito	88866555576		M	55000.00		1
4	João	B	Silva	12345678966		M	30000.00	33344555587	5
5	Fernando	T	Wong	33344555587		M	40000.00	88866555576	5
6	Joice	A	Leite	45345345376	...	F	25000.00	33344555587	5
7	Jorge	L	Caio	10000000000		M	26000.00		5
8	Alice	J	Zelaya	99988777767		F	25000.00	98765432168	4
9	Ronaldo	K	Lima	66688444476		M	38000.00	33344555587	

Exemplo de consulta

- *C20: Recuperar o primeiro nome e o salário do funcionário em que o salário é maior ou igual a 30000, ordenado pelo primeiro nome.*

```
SELECT pnome, salario  
FROM funcionario  
WHERE salario >= 30000  
ORDER BY pnome;
```

	pnome	salario
1	Fernando	40000.00
2	Jennifer	43000.00
3	João	30000.00
4	Jorge	55000.00
5	Ronaldo	38000.00

Exemplo de consulta

- C21: Recuperar o nome e endereço de todos os funcionários que trabalham para o departamento de 'Pesquisa'. Ordene pelo primeiro nome de maneira descendente.*

Funcionário

	pnome	minicial	unome	cpf		sexo	salario	cpf_supervisor	dnr
1	Jennifer	S	Souza	98765432168		F	43000.00	88866555576	4
2	André	V	Pereira	98798798733		M	25000.00	98765432168	4
3	Jorge	E	Brito	88866555576		M	55000.00		1
4	João	B	Silva	12345678966		M	30000.00	33344555587	5
5	Fernando	T	Wong	33344555587		M	40000.00	88866555576	5
6	Joice	A	Leite	45345345376	...	F	25000.00	33344555587	5
7	Jorge	L	Caio	10000000000		M	26000.00		5
8	Alice	J	Zelaya	99988777767		F	25000.00	98765432168	4
9	Ronaldo	K	Lima	66688444476		M	38000.00	33344555587	

Departamento

	dnumero	dnome	cpf_gerente	data_inicio_gerente
1	1	Matriz	88866555576	1981-06-19
2	3	Compras	33344555587	
3	4	Administração	98765432168	1995-01-01
4	5	Pesquisa	33344555587	1988-05-22

Exemplo de consulta

- *C21: Recuperar o nome e endereço de todos os funcionários que trabalham para o departamento de 'Pesquisa'. Ordene pelo primeiro nome de maneira descendente.*

```
SELECT pnome, unome, endereco  
FROM funcionario, departamento  
WHERE dnome='Pesquisa' AND dnumero = dnr  
ORDER BY pnome DESC;
```

	pnome	unome	endereco
1	Jorge	Caio	Rua Carolina Silva, 832, São Paulo, SP
2	Joice	Leite	Avenida Lucas Obes, 74, São Paulo, SP
3	João	Silva	Rua das Flores, 751, São Paulo, SP
4	Fernando	Wong	Rua da Lapa, 34, São Paulo, SP

Resumo das consultas

- *Uma simples consulta em SQL pode consistir em até quatro cláusulas, mas apenas as duas primeiras – SELECT e FROM – são obrigatórias. As cláusulas são especificadas na seguinte ordem, com aquelas entre colchetes [...] sendo opcionais:*

SELECT <lista atributos>

FROM <lista tabelas>

[WHERE <condição>]

[ORDER BY <lista atributos>]

Resumo das consultas

- A cláusula *SELECT* lista os atributos a serem recuperados, e a cláusula *FROM* especifica todas as relações (tabelas) necessárias na consulta simples.
- A cláusula *WHERE* identifica as condições para selecionar as tuplas dessas relações, incluindo condições de junção, se necessário.

Resumo das consultas

- *ORDER BY* especifica uma ordem para exibir os resultados de uma consulta. Duas cláusulas adicionais, *GROUP BY* e *HAVING*, serão descritas nas próximas aulas.

