



# Fundamentos de Bacos de Dados

Select e operadores básicos

- 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
-------------	------------------------	------	----------	------------

- A SQL tem uma instrução básica para recuperar informações de um banco de dados: a instrução **SELECT**.
- A SQL permite que uma tabela (relação) tenha duas ou mais tuplas que são idênticas em todos os seus valores de atributo.
- Uma tabela SQL não é um conjunto de tuplas, pois um conjunto não permite dois membros idênticos; em vez disso, ela é um **multiconjunto** de tuplas.

- A forma básica do comando **SELECT** é composta pelas três cláusulas **SELECT**, **FROM** e **WHERE**, e tem a seguinte forma:

**SELECT** <lista atributos>

**FROM** <lista tabelas>

**WHERE** <condição>;

<lista de atributos> é uma lista de nomes de atributos cujos valores devem ser recuperados pela consulta;

<lista tabelas> é uma lista dos nomes de relação exigidos para processar a consulta.

<condição> é uma expressão condicional (booleana) que identifica as tuplas a serem recuperadas pela consulta.

- Em SQL, os operadores básicos de comparação lógicos para comparar valores de atributo entre si e com constantes literais são:

Operador	Significado
=	Igual a
>	Maior do que
>=	Maior do que ou igual
<	Menor do que
<=	Menor do que ou igual
!= ou <>	Diferente de

- C0: Recuperar o primeiro nome e o salário do funcionário em que o salário é maior ou igual a 30000.

## Funcionário

	<b>pnome</b> character varying(15)	<b>minicial</b> character(1)	<b>unome</b> character varying(15)	<b>cpf</b> [PK] character(11)	<b>...</b>	<b>salario</b> numeric(10,2)
<b>1</b>	Jorge	L	Caio	100000000000		26000.00
<b>2</b>	João	B	Silva	12345678966		30000.00
<b>3</b>	Fernando	T	Wong	33344555587		40000.00
<b>4</b>	Joice	A	Leite	45345345376		25000.00
<b>5</b>	Ronaldo	K	Lima	66688444476	<b>...</b>	38000.00
<b>6</b>	Jorge	E	Brito	88866555576		55000.00
<b>7</b>	Jennifer	S	Souza	98765432168		43000.00
<b>8</b>	André	V	Pereira	98798798733		25000.00
<b>9</b>	Alice	J	Zelaya	99988777767		25000.00

- C0: Recuperar o primeiro nome e o salário do funcionário em que o salário é maior ou igual a 30000.

```
SELECT pnome, salario  
FROM funcionario  
WHERE salario >= 30000
```

	pnome character varying(15)	salario numeric(10,2)
1	João	30000.00
2	Fernando	40000.00
3	Jennifer	43000.00
4	Ronaldo	38000.00
5	Jorge	55000.00



- C1: Recuperar a data de nascimento e o endereço do(s) funcionário(s) cujo nome seja 'João B. Silva'.

## Funcionário

	pnome character varying(15)	minicial character(1)	unome character varying(15)	cpf [PK] character(11)	datanasc date	endereco character varying(50)	...
1	Jorge	L	Caio	10000000000	1955-12-01	Rua Carolina Silva, 832, São Paulo, SP	
2	João	B	Silva	12345678966	1965-01-09	Rua das Flores, 751, São Paulo, SP	
3	Fernando	T	Wong	33344555587	1955-12-08	Rua da Lapa, 34, São Paulo, SP	
4	Joice	A	Leite	45345345376	1972-07-31	Avenida Lucas Obes, 74, São Paulo, SP	
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	
7	Jennifer	S	Souza	98765432168	1941-06-20	Avenida Artur de Lima, 54, Santo André, SP	
8	André	V	Pereira	98798798733	1969-03-29	Rua Timbira, 35, São Paulo, SP	
9	Alice	J	Zelaya	99988777767	1968-01-19	Rua Souza Lima, 35, Curitiba, PR	

- C1: Recuperar a data de nascimento e o endereço do(s) funcionário(s) cujo nome seja 'João B. Silva'.

**SELECT** datanasc, endereco

**FROM** funcionario

**WHERE** pnome='João' **AND** minicial='B' **AND**  
unome='Silva';

	<b>datanasc</b> date	<b>endereco</b> character varying(50)
<b>1</b>	1965-01-09	Rua das Flores, 751, São Paulo, SP

- As consultas anteriores envolvem apenas a relação FUNCIONARIO listada na cláusula FROM.
- As consultas selecionam as tuplas FUNCIONARIO individuais que satisfazem a condição da cláusula WHERE, depois projeta o resultado nos atributos pnome, salario, datanasc e endereco respectivamente listados na cláusula SELECT.

- C3: Selecionar todos os Cpf's de FUNCIONARIO

## Funcionário

	pnome character varying(15)	minicial character(1)	unome character varying(15)	cpf [PK] character(11)	datanasc date	endereco character varying(50)	...
1	Jorge	L	Caio	10000000000	1955-12-01	Rua Carolina Silva, 832, São Paulo, SP	
2	João	B	Silva	12345678966	1965-01-09	Rua das Flores, 751, São Paulo, SP	
3	Fernando	T	Wong	33344555587	1955-12-08	Rua da Lapa, 34, São Paulo, SP	
4	Joice	A	Leite	45345345376	1972-07-31	Avenida Lucas Obes, 74, São Paulo, SP	
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	
7	Jennifer	S	Souza	98765432168	1941-06-20	Avenida Artur de Lima, 54, Santo André, SP	
8	André	V	Pereira	98798798733	1969-03-29	Rua Timbira, 35, São Paulo, SP	
9	Alice	J	Zelaya	99988777767	1968-01-19	Rua Souza Lima, 35, Curitiba, PR	

- C3: Selecionar todos os Cpf's de FUNCIONARIO

```
SELECT cpf  
FROM funcionario
```

	cpf character(11)
1	98765432168
2	98798798733
3	88866555576
4	12345678966
5	33344555587
6	45345345376
7	10000000000
8	99988777767
9	66688444476

- O asterisco (\*) é utilizado para recuperar os valores de todos os atributos.
- Para recuperar todos os valores de atributo das tuplas selecionadas, não precisamos listar todos os nomes de atributos especificamente em SQL; basta especificar um asterisco (\*), que significa todos os atributos.

- C4: Recuperar todos os valores de atributo de qualquer FUNCIONARIO que trabalha no DEPARTAMENTO número 5.

## Funcionário

	pnome character varying(50)	minicial character(1)	unome character varying(50)	cpf [PK] character(11)	datanasc date	endereço character varying(50)	...	dnr integer
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

- C4: Recuperar todos os valores de atributo de qualquer FUNCIONARIO que trabalha no DEPARTAMENTO número 5.

```
SELECT *
FROM funcionario
WHERE Dnr=5;
```

	pnome character varying(15)	inicial character(1)	unome character varying(15)	cpf character(11)	datanasc date	endereco character varying(50)	sexo character(1)	salario numeric(10,2)	cpf_supervisor character(11)	dnr integer
1	João	B	Silva	12345678966	1965-01-09	Rua das Flores, 751, São Paulo, SP	M	30000.00	33344555587	5
2	Fernando	T	Wong	33344555587	1955-12-08	Rua da Lapa, 34, São Paulo, SP	M	40000.00	88866555576	5
3	Ronaldo	K	Lima	66688444476	1962-09-15	Rua Rebouças, 65, Piracicaba, SP	M	38000.00	33344555587	5
4	Joice	A	Leite	45345345376	1972-07-31	Avenida Lucas Obes, 74, São Paulo, SP	F	25000.00	33344555587	5



- SQL permite o uso de aritmética nas consultas.
- Os operadores estão listados de acordo com sua precedência.
- Ou seja, divisão e multiplicação têm preferência sobre adição e a subtração.

Operador	Descrição
/	Dividir
*	Multiplicar
-	Subtrair
+	Adicionar

- C5: Calcular a remuneração mensal dos funcionários e depois adicionar o bônus de 50.

## Funcionário

	pnome character varying(15)	minicial character(1)	unome character varying(15)	...	salario numeric(10,2)	...	dnr integer
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

- C5: Calcular a remuneração mensal dos funcionários e depois adicionar o bônus de 50

**SELECT** pnome, unome, salario/ 12+50 **AS**  
 Salario\_Mensal\_Com\_Bonus  
**FROM** funcionario

	pnome character varying(15)	unome character varying(15)	salario_mensal_com_bonus numeric
1	Jennifer	Souza	3633.33333333333333
2	André	Pereira	2133.33333333333333
3	Jorge	Brito	4633.33333333333333
4	João	Silva	2550.00000000000000
5	Fernando	Wong	3383.33333333333333
6	Joice	Leite	2133.33333333333333
7	Jorge	Caio	2216.66666666666667
8	Alice	Zelaya	2133.33333333333333
9	Ronaldo	Lima	3216.66666666666667

- C6: Adicionar o bônus de 50 e depois calcular a remuneração mensal.

## Funcionário

	pnome character varying(15)	minicial character(1)	unome character varying(15)	...	salario numeric(10,2)	...	dnr integer
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

- C6: Adicionar o bônus de 50 e depois calcular a remuneração mensal.

```
SELECT pnome, unome, (salario+50)/12 AS  
Salario_Mensal_Com_Bonus  
FROM funcionario
```

	pnome character varying(15)	unome character varying(15)	salario_mensal_com_bonus numeric
1	Jennifer	Souza	3587.5000000000000000
2	André	Pereira	2087.5000000000000000
3	Jorge	Brito	4587.5000000000000000
4	João	Silva	2504.1666666666666667
5	Fernando	Wong	3337.5000000000000000
6	Joice	Leite	2087.5000000000000000
7	Jorge	Caio	2170.8333333333333333
8	Alice	Zelaya	2087.5000000000000000
9	Ronaldo	Lima	3170.8333333333333333