



ASEBD / CEBDSQL / PSQL

GRSI / TPSI



ateco

Sintaxe do comando SELECT

SELECT Campo1, Campo2, ...

FROM Tabela1, ..., TabelaK

[WHERE Condição]

[GROUP BY ...]

[HAVING ...]

[ORDER BY ...]

Sintaxe do comando SELECT

SELECT Codigo, Localidade

FROM Postal

OU

SELECT * FROM Postal

Codigo	Localidade
1000	LISBOA
1100	LISBOA
1200	LISBOA
1500	LISBOA
2000	SANTARÉM
2300	TOMAR
3000	COIMBRA
4000	PORTO
9000	FUNCHAL

Sintaxe do comando SELECT

SELECT * FROM Pessoa

Id	Nome	Idade	Salario	Telefone	Cod_Postal
42	António Dias	43	74000.00	789654	1500
5	Célia Morais	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
37	Isabel Espada	28	86000.00	NULL	1100
49	José António	17	210000.00	NULL	1500
14	Nascimento Augusto	35	220000.00	456123	2300
25	Paulo Viegas	32	95000.00	NULL	1500

Sintaxe do comando SELECT

SELECT Nome, Id, Idade, Cod_Postal,

Telefone, Salario

FROM Pessoa

Nome	Id	Idade	Cod_Postal	Telefone	Salario
António Dias	42	43	1500	789654	74000.00
Célia Morais	5	26	1100	123456	170000.00
Florinda Simões	32	35	4000	NULL	147000.00
Isabel Espada	37	28	1100	NULL	86000.00
José António	49	17	1500	NULL	210000.00
Nascimento Augusto	14	35	2300	456123	220000.00
Paulo Viegas	25	32	1500	NULL	95000.00

Sintaxe do comando SELECT

SELECT Nome, Idade, Nome

FROM Pessoa

Nome	Idade	Nome
António Dias	43	António Dias
Célia Morais	26	Célia Morais
Florinda Simões	35	Florinda Simões
Isabel Espada	28	Isabel Espada
José António	17	José António
Nascimento Augusto	35	Nascimento Augusto
Paulo Viegas	32	Paulo Viegas

Cláusula WHERE

SELECT ...

FROM ...

WHERE Condição

Operador	Descrição	Exemplo	Resultado
=	Igual a	7=5	FALSE
>	Maior que	7>5	TRUE
<	Menor que	7<5	FALSE
>=	Maior ou igual a	7>=5	TRUE
<=	Menor ou igual a	7<=5	FALSE
<> ou !=	Diferente	7<>5	TRUE

Cláusula WHERE

SELECT *

FROM Pessoa

WHERE Idade = 35

Id	Nome	Idade	Salario	Telefone	Cod_Postal
42	António Dias	43	74000.00	789654	1500
5	Célia Moraes	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
37	Isabel Espada	28	86000.00	NULL	1100
49	José António	17	210000.00	NULL	1500
14	Nascimento Augusto	35	220000.00	456123	2300
25	Paulo Viegas	32	95000.00	NULL	1500



Id	Nome	Idade	Salario	Telefone	Cod_Postal
32	Florinda Simões	35	147000.00	NULL	4000
14	Nascimento Augusto	35	220000.00	456123	2300

Cláusula WHERE

SELECT ...

FROM ...

WHERE Condição

Operador	Exemplo
AND	Cond1 AND Cond2
OR	Cond1 OR Cond2
NOT	NOT Condição

Cláusula WHERE

SELECT Id, Nome, Idade, Salário

FROM Pessoa

WHERE Idade >= 30 **AND** Idade <= 40

Id	Nome	Idade	Salario	Telefone	Cod_Postal
42	António Dias	43	74000.00	789654	1500
5	Célia Morais	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
37	Isabel Espada	28	86000.00	NULL	1100
49	José António	17	210000.00	NULL	1500
14	Nascimento Augusto	35	220000.00	456123	2300
25	Paulo Viegas	32	95000.00	NULL	1500



Id	Nome	Idade	Salario
32	Florinda Simões	35	147000.00
14	Nascimento Augusto	35	220000.00
25	Paulo Viegas	32	95000.00

Cláusula WHERE

SELECT Id, Nome, Idade, Salário
FROM Pessoa

WHERE Idade <= 30 **OR** Idade >= 40

OU

SELECT Id, Nome, Idade, Salário
FROM Pessoa

WHERE NOT (Idade > 30 **AND** Idade < 40)

Id	Nome	Idade	Salario	Telefone	Cod_Postal
42	António Dias	43	74000.00	789654	1500
5	Célia Morais	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
37	Isabel Espada	28	86000.00	NULL	1100
49	José António	17	210000.00	NULL	1500
14	Nascimento Augusto	35	220000.00	456123	2300
25	Paulo Viegas	32	95000.00	NULL	1500



Id	Nome	Idade	Salario
42	António Dias	43	74000.00
5	Célia Morais	26	170000.00
37	Isabel Espada	28	86000.00
49	José António	17	210000.00

Cláusula WHERE

- O operador BETWEEN permite especificar intervalos de valores. A sua sintaxe é a seguinte:

SELECT ...

FROM ...

WHERE valor [**NOT**] **BETWEEN** valor 1 **AND** valor2

- Na realidade, as condições seguintes são equivalentes:

valor **>=** valor1 **AND** valor **<=** valor2;

(valor **>=** valor1) **AND** (valor **<=** valor2);

valor **BETWEEN** valor1 **AND** valor2.

Cláusula WHERE

SELECT Id, Nome, Idade, Salário
FROM Pessoa

WHERE Idade >= 30 **AND** Idade <=40

OU

SELECT Id, Nome, Idade, Salário
FROM Pessoa

WHERE Idade **BETWEEN** 30 **AND** 40

Id	Nome	Idade	Salario	Telefone	Cod_Postal
42	António Dias	43	74000.00	789654	1500
5	Célia Morais	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
37	Isabel Espada	28	86000.00	NULL	1100
49	José António	17	210000.00	NULL	1500
14	Nascimento Augusto	35	220000.00	456123	2300
25	Paulo Viegas	32	95000.00	NULL	1500



Id	Nome	Idade	Salario
32	Florinda Simões	35	147000.00
14	Nascimento Augusto	35	220000.00
25	Paulo Viegas	32	95000.00

Cláusula WHERE

SELECT ...

FROM ...

WHERE valor [**NOT**] **IN** (valor1, valor2,... ,valorN)

- As seguintes condições são equivalentes:

valor = 1 **OR** valor = 2 **OR** valor = 3

e

valor **IN** (1,2,3)

Cláusula WHERE

SELECT *

FROM Postal

WHERE Localidade = 'LISBOA' **OR** Localidade = 'TOMAR'

OU

SELECT *

FROM Postal

WHERE Localidade **IN** ('LISBOA', 'TOMAR')

Codigo	Localidade
1000	LISBOA
1100	LISBOA
1200	LISBOA
1500	LISBOA
2300	TOMAR

Cláusula WHERE

SELECT Codigo, Localidade

FROM Postal

WHERE Localidade <> 'LISBOA' **AND** Localidade <> 'TOMAR'

OU

SELECT Codigo, Localidade

FROM Postal

WHERE Localidade **NOT IN** ('LISBOA', 'TOMAR')

Codigo	Localidade
2000	SANTARÉM
3000	COIMBRA
4000	PORTO
9000	FUNCHAL

Cláusula WHERE

SELECT ...

FROM ...

WHERE Campo IS [NOT] NULL

SELECT Nome, Telefone

FROM Pessoa

Nome	Telefone
António Dias	789654
Célia Morais	123456
Florinda Simões	NULL
Isabel Espada	NULL
José António	NULL
Nascimento Augusto	456123
Paulo Viegas	NULL

Cláusula WHERE

SELECT Nome, Telefone

FROM Pessoa

WHERE Telefone **IS NOT NULL**

Nome	Telefone
António Dias	789654
Célia Moraes	123456
Florinda Simões	NULL
Isabel Espada	NULL
José António	NULL
Nascimento Augusto	456123
Paulo Viegas	NULL



Nome	Telefone
António Dias	789654
Célia Moraes	123456
Nascimento Augusto	456123

Cláusula WHERE

SELECT *

FROM Postal

WHERE Localidade LIKE '_IS%'

Codigo	Localidade
1000	LISBOA
1100	LISBOA
1200	LISBOA
1500	LISBOA
2000	SANTARÉM
2300	TOMAR
3000	COIMBRA
4000	PORTO
9000	FUNCHAL



Codigo	Localidade
1000	LISBOA
1100	LISBOA
1200	LISBOA
1500	LISBOA

Cláusula WHERE

WildCard	Significado
%	Qualquer conjunto de zero ou mais caracteres
_ (underscore)	Um carácter qualquer

Cláusula WHERE

Pesquisa	Descrição	Exemplo
'[abc]%'	Qualquer string começada obrigatoriamente por a, b ou c.	altivo, bola, cão
'[!abc]%'	Qualquer string não começada por a, b ou c.	dália, floral
'[0-9]%'	String começada por um dígito e terminada pelo carácter a, b ou c.	999778a, 8b
'[a-zA-Z]_'	String formada unicamente por dois caracteres. O primeiro é um carácter alfabético minúsculo ou maiúsculo.	A1, aa,a_
'[!0-9]%'	O primeiro carácter não pode ser dígito	Abc, a23d
'AI[m-r]%'	String começada por 'AI'. O terceiro carácter terá de estar entre os caracteres 'm' e 'r' (m,n,o,p,q,r). Depois pode aparecer qualquer string. (Dimensão mínima 3).	Alpendre, Alm, Alpinista123
'AI[m-r]%'	String começada por 'AI'. O terceiro carácter terá de estar entre os caracteres 'm' e 'r' (m,n,o,p,q,r). Depois pode aparecer qualquer string. No entanto o último carácter não pode ser numérico. (Dimensão mínima 4).	Alpendre

Cláusula WHERE

Precedência dos operadores

Operador	escritção
()	Parêntesis
* /	Multiplicação / Divisão
+ -	Adição / Subtracção
NOT	
AND	
OR	



Cláusula WHERE

SELECT *
FROM Pessoa
WHERE Idade <= 27 **OR**
 Idade >= 30 **AND**
 Telefone **IS NULL**

OU

SELECT *
FROM Pessoa
WHERE Idade <= 27 **OR**
 (Idade >= 30 **AND** Telefone **IS NULL**)

Id	Nome	Idade	Salario	Telefone	Cod_Postal
42	António Dias	43	74000.00	789654	1500
5	Célia Morais	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
37	Isabel Espada	28	86000.00	NULL	1100
49	José António	17	210000.00	NULL	1500
14	Nascimento Augusto	35	220000.00	456123	2300
25	Paulo Viegas	32	95000.00	NULL	1500



Id	Nome	Idade	Salario	Telefone	Cod_Postal
5	Célia Morais	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
49	José António	17	210000.00	NULL	1500
25	Paulo Viegas	32	95000.00	NULL	1500

Ordenação ORDER BY

SELECT Campo1 , Campo2 , . . . , CampoN, *

FROM Tabela1 , . . . , TabelaN

[WHERE Condição]

[GROUP BY ...]

[HAVING ...]

[ORDER BY Campo **[ASC DESC]** , Campo **[ASC DESC]** , ...]

Ordenação ORDER BY

SELECT *
FROM Pessoa
ORDER BY Idade

Id	Nome	Idade	Salario	Telefone	Cod_Postal
49	José António	17	210000.00	NULL	1500
5	Célia Morais	26	170000.00	123456	1100
37	Isabel Espada	28	86000.00	NULL	1100
25	Paulo Viegas	32	95000.00	NULL	1500
32	Florinda Simões	35	147000.00	NULL	4000
14	Nascimento Augusto	35	220000.00	456123	2300
42	António Dias	43	74000.00	789654	1500

Ordenação ORDER BY

```
SELECT Nome, Salario  
FROM Pessoa  
WHERE Telefone IS NOT NULL  
ORDER BY Salario DESC
```

Nome	Salario
Nascimento Augusto	220000.00
Célia Moraes	170000.00
António Dias	74000.00



Ordenação ORDER BY

- **SELECT ***
- **FROM** Comissao
- **WHERE** Id < 40
- **ORDER BY** Id, Valor **DESC**

Id	Id_Msg	Valor
14	10	10500
14	100	3750
14	60	2600
14	70	400
25	10	2500
25	30	370
37	30	14230
37	10	5500
37	50	120
37	40	20

Eliminar repetições DISTINCT

SELECT Localidade
FROM Postal

SELECT DISTINCT Localidade
FROM Postal

Localidade
LISBOA
LISBOA
LISBOA
LISBOA
SANTARÉM
TOMAR
COIMBRA
PORTO
FUNCHAL



Localidade
COIMBRA
FUNCHAL
LISBOA
PORTO
SANTARÉM
TOMAR

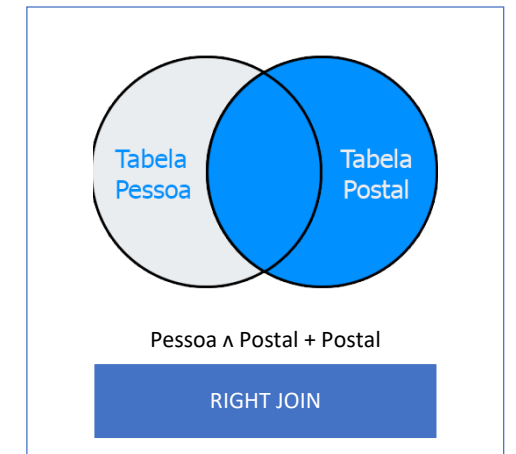
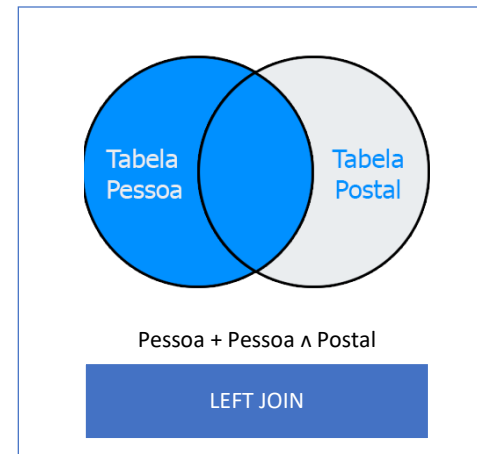
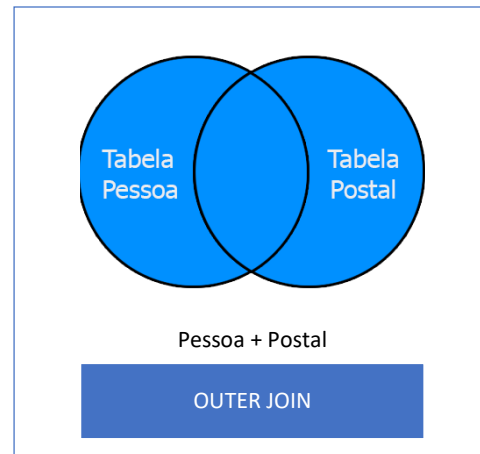
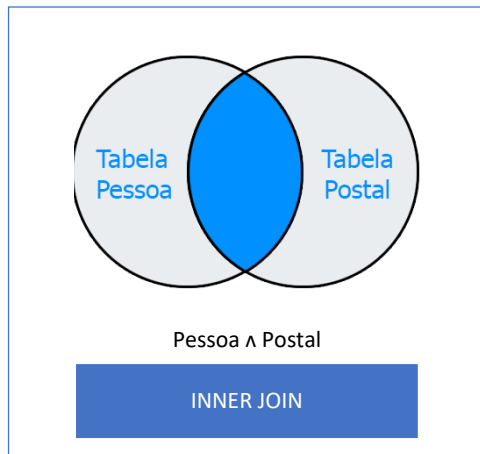
Funções de agregação (JOINS)

Observem as seguintes tabelas

Id	Nome	Idade	Salario	Telefone	Cod_Postal	Codigo	Localidade
42	António Dias	43	74000.00	789654	1500	1000	LISBOA
5	Célia Morais	26	170000.00	123456	1100	1100	LISBOA
32	Florinda Simões	35	147000.00	NULL	4000	1200	LISBOA
37	Isabel Espada	28	86000.00	NULL	1100	1500	LISBOA
49	José António	17	210000.00	NULL	1500	2000	SANTARÉM
14	Nascimento Augusto	35	220000.00	456123	2300	2300	TOMAR
25	Paulo Viegas	32	95000.00	NULL	1500	3000	COIMBRA
						4000	PORTO
						9000	FUNCHAL



JOINS



CROSS JOIN

Produto cartesiano de tabelas

SELECT *

FROM Pessoa, Postal

Id	Nome	Idade	...	cod postal	código	localidade
42	António Dias	43		1500	1000	LISBOA
5	Célia Morais	26		1100	1000	LISBOA
32	Florinda Simões	35		4000	1000	LISBOA
37	Isabel Espada	28		1100	1000	LISBOA
49	José António	17		1500	1000	LISBOA
14	Nascimento Augusto	35		2300	1000	LISBOA
25	Paulo Viegas	32		1500	1000	LISBOA
42	António Dias	43		1500	1100	LISBOA
5	Célia Morais	26		1100	1100	LISBOA
32	Florinda Simões	35		4000	1100	LISBOA
37	Isabel Espada	28		1100	1100	LISBOA
49	José António	17		1500	1100	LISBOA
14	Nascimento Augusto	35		2300	1100	LISBOA
25	Paulo Viegas	32		1500	1100	LISBOA
42	António Dias	43		1500	1200	LISBOA
5	Célia Morais	26		1100	1200	LISBOA
32	Florinda Simões	35		4000	1200	LISBOA
37	Isabel Espada	28		1100	1200	LISBOA
49	José António	17		1500	1200	LISBOA
14	Nascimento Augusto	35		2300	1200	LISBOA
25	Paulo Viegas	32		1500	1200	LISBOA
42	António Dias	43		1500	1500	LISBOA
5	Célia Morais	26		1100	1500	LISBOA
32	Florinda Simões	35		4000	1500	LISBOA
37	Isabel Espada	28		1100	1500	LISBOA
49	José António	17		1500	1500	LISBOA
14	Nascimento Augusto	35		2300	1500	LISBOA
25	Paulo Viegas	32		1500	1500	LISBOA
42	António Dias	43		1500	2000	SANTARÉM
5	Célia Morais	26		1100	2000	SANTARÉM
32	Florinda Simões	35		4000	2000	SANTARÉM
37	Isabel Espada	28		1100	2000	SANTARÉM
49	José António	17		1500	2000	SANTARÉM
14	Nascimento Augusto	35		2300	2000	SANTARÉM
...

EQUI JOIN = INNER JOIN

SELECT *
FROM Pessoa, Postal
WHERE Cod_Postal = Codigo

Id	Nome	Idade	Salario	Telefone	Cod_Postal	Codigo	Localidade
42	António Dias	43	74000.00	789654	1500	1500	LISBOA
5	Célia Morais	26	170000.00	123456	1100	1100	LISBOA
32	Florinda Simões	35	147000.00	NULL	4000	4000	PORTO
37	Isabel Espada	28	86000.00	NULL	1100	1100	LISBOA
49	José António	17	210000.00	NULL	1500	1500	LISBOA
14	Nascimento Augusto	35	220000.00	456123	2300	2300	TOMAR
25	Paulo Viegas	32	95000.00	NULL	1500	1500	LISBOA

SELECT *
FROM Pessoa **INNER JOIN** Postal
ON Pessoa.Cod_Postal = Postal.Codigo

Funções de Agregação

Função	Descrição
COUNT	Devolve o número de linhas.
MAX	Devolve o maior valor da coluna.
MIN	Devolve o menor valor da coluna.
SUM	Devolve soma de todos os valores da coluna.
AVG	Devolve a média de todos os valores da coluna.

Forma	Descrição
COUNT (*)	Devolve o número de linhas que resulta de um comando SELECT.
COUNT (COLUNA)	Devolve o número de ocorrências na coluna diferentes de NULL.
COUNT (DISTINCT COLUNA)	Devolve o número de ocorrências (sem repetições) na coluna.

COUNT

```
SELECT COUNT(*) AS Tot1,  
        COUNT(Telefone) AS Tot2,  
FROM Pessoa
```

Id	Nome	Idade	Salario	Telefone	Cod_Postal
42	António Dias	43	74000.00	789654	1500
5	Célia Morais	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
37	Isabel Espada	28	86000.00	NULL	1100
49	José António	17	210000.00	NULL	1500
14	Nascimento Augusto	35	220000.00	456123	2300
25	Paulo Viegas	32	95000.00	NULL	1500



Tot1	Tot2
7	3

MAX / MIN

SELECT

MAX(Salario) AS 'Gordinho',

MIN(Salario) AS 'Magrinho'

FROM Pessoa

Id	Nome	Idade	Salario	Telefone	Cod_Postal
42	António Dias	43	74000.00	789654	1500
5	Célia Morais	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
37	Isabel Espada	28	86000.00	NULL	1100
49	José António	17	210000.00	NULL	1500
14	Nascimento Augusto	35	220000.00	456123	2300
25	Paulo Viegas	32	95000.00	NULL	1500



Gordinho	Magrinho
220000	74000

MAX / MIN

SELECT MIN(Nome) AS Primeiro
FROM Pessoa

OU

SELECT TOP(1) Nome AS Primeiro
FROM Pessoa
ORDER BY Nome

Id	Nome	Idade	Salario	Telefone	Cod_Postal
42	António Dias	43	74000.00	789654	1500
5	Célia Morais	26	170000.00	123456	1100
32	Florinda Simões	35	147000.00	NULL	4000
37	Isabel Espada	28	86000.00	NULL	1100
49	José António	17	210000.00	NULL	1500
14	Nascimento Augusto	35	220000.00	456123	2300
25	Paulo Viegas	32	95000.00	NULL	1500



Primeiro
Antonio Dias

SUM

```
SELECT  
    SUM(Valor) AS 'Comissões'  
FROM Comissao
```

Comissões
42480

```
SELECT SUM(Valor*0.23) AS IVA  
FROM Comissao
```

IVA
9770,40

Agrupamento (GROUP BY)

```
SELECT Id, SUM(Valor) AS Total  
FROM Comissao  
GROUP BY Id
```

Id	Valor
14	10500
14	3750
14	400
14	2600
25	2500
25	370
37	20
37	14230
37	5500
37	120
42	20
42	170
49	2300



Id	Total
14	17250
25	2870
37	19870
42	190
49	2300

Agrupamento (HAVING)

```
SELECT Id, SUM(Valor) AS Total
FROM Comissao
--WHERE Valor > 1000
GROUP BY Id
HAVING SUM(Valor) > 1000
```

```
SELECT Id, SUM(Valor) AS Total
FROM Comissao
WHERE Valor > 1000
GROUP BY Id
--HAVING SUM(Valor) > 1000
```

Id	Valor
14	10500
14	3750
14	400
14	2600
25	2500
25	370
37	20
37	14230
37	5500
37	120
42	20
42	170
49	2300



HAVING SUM(Valor) > 1000

Id	Total
14	17250
25	2870
37	19870
49	2300

WHERE Valor > 1000

Id	Total
14	16850
25	2500
37	19730
49	2300

Agrupamento (HAVING)

- Servem ambos para restringir o conjunto de registos a apresentar.
- Utiliza-se a clausula WHERE sempre que se pretende restringir os registos a considerar na seleção.
- A clausula HAVING serve para restringir os grupos que foram formados depois e aplicada a restrição da clausula WHERE.

Id	Valor
14	10500
14	3750
14	400
14	2600
25	2500
25	370
37	20
37	14230
37	5500
37	120
42	20
42	170
49	2300



HAVING SUM(Valor) > 1000

Id	Total
14	17250
25	2870
37	19870
49	2300

WHERE Valor > 1000

Id	Total
14	16850
25	2500
37	19730
49	2300