Objetivos:

I. Estrutura do comando select;
II. Funções agrupadoras;
III. Termo having da cláusula select;
IV. Renomear colunas no resultado.

Para reproduzir os exemplos e fazer os exercícios use as cláusulas do arquivo Aula3 - Clausula SQL.txt para criar a tbcurso e colocar os 454 registros.

I. Estrutura do comando select

A cláusula SQL para listar registros pode ter diversos termos e cada termo tem o seu papel na consulta. A seguir tem-se os termos possíveis numa cláusula select:

```
select distinct nomeDaColuna1, nomeDaColuna2, nomeDaColuna3
from nomeDaTabela
where expressão que resulta em verdadeiro/falso
group by nomeDaColuna
having funçãoAgrupadora
order by nomeDaColuna asc/desc
limit numeroDeRegistros
offset numeroDeRegistros;
```

Distinct - evitar registros repetidos no resultado:

O termo distinct é usado para evitar registros repetidos no resultado.

Query Editor | Query History

```
1 select municipio
2 from tbcurso
3 order by municipio asc;
```

Data Output | Explain | Messages

	municipio character varying
1	Adamantina
2	Adamantina
3	Americana
4	Americana
5	Americana
6	Americana
7	Americana
8	Americana

O resultado possui 453 registros

O resultado possui vários registros repetidos

Query Editor | Query History

```
1 select distinct municipio
2 from tbcurso
3 order by municipio asc;
```

Data Output | Explain | Messages

	municipio character varying
1	Adamantina
2	Americana
3	Araçatuba
4	Araraquara
5	Araras
6	Assis
7	Barretos
8	Barueri

O resultado possui 69 registros

O resultado possui 1 registro para cada município

O termo distinct precisa estar imediatamente após o select.

O termo distinct é para evitar registros repetidos no resultado. A consulta a seguir terá 190 registros, apesar de existirem municípios repetidos, não existirá repetição ao considerar as colunas município e turno.

```
select distinct municipio, turno
from tbcurso
order by municipio asc;
```

Temos de usar o termo on para evitar valores repetidos numa determinada coluna. A cláusula a seguir terá 69 registros, pois o termo distinct on(municipio) é usado para evitar valores repetidos somente na coluna municipio.

```
select distinct on (municipio) municipio, turno
from tbcurso
order by municipio asc;
```

Podemos colocar dentro dos parênteses mais de uma coluna, para evitar valores repetidos nessas colunas. No exemplo a seguir serão evitados valores repetidos no par de colunas municipio e turno.

```
select distinct on (municipio,turno) *
from tbcurso
order by municipio asc;
```

Limit e offset:

O termo limit é usado para indicar a quantidade de registros que aceitamos no resultado. No exemplo a seguir a consulta retornaria 454 registros, mas foram retornados apenas os 10 primeiros, pois usamos limit 10.

No segundo exemplo foram retornados 5 registros pelo fato de termos usado limit 5, porém os registros começam na 4ª posição, pois usamos offset 3, ou seja, deslocamos o início em 3 posições.

```
1  select municipio, curso
2  from tbcurso
3  order by municipio asc, curso asc
4  limit 10;
```

Data Output | Explain | Messages | Notifications

	municipio character varying	curso character varying
1	Adamantina	Ciência de Dados
2	Adamantina	Gestão Comercial
3	Americana	Análise e Desenvolvim...
4	Americana	Análise e Desenvolvim...
5	Americana	Gestão Empresarial
6	Americana	Gestão Empresarial
7	Americana	Gestão Empresarial
8	Americana	Jogos Digitais
9	Americana	Logística
10	Americana	Logística

offset 3

limit 5

```
1  select municipio, curso
2  from tbcurso
3  order by municipio asc, curso asc
4  limit 5
5  offset 3;
```

Data Output | Explain | Messages | Notifications

	municipio character varying	curso character varying
1	Americana	Análise e Desenvolvim...
2	Americana	Gestão Empresarial
3	Americana	Gestão Empresarial
4	Americana	Gestão Empresarial
5	Americana	Jogos Digitais

Os termos limit e offset são usados para fazer a paginação de registros. Como exemplo, o Gmail mostra apenas 50 e-mails por página. Considerando que os e-mails estão numa tabela de nome tbmail, então temos as seguintes cláusulas para mostrar as três primeiras páginas de e-mails.

Cláusula para mostrar a 1ª página de e-mails	Cláusula para mostrar a 2ª página de e-mails	Cláusula para mostrar a 3ª página de e-mails

`select *`
`from tbmail`
`limit 50`
`offset 0;`
	`select *`
	`from tbmail`
	`limit 50`
	`offset 50;`
		`select *`
		`from tbmail`
		`limit 50`
		`offset 100;`

Group by:

O termo group by é usado para agrupar o resultado evitando valores repetidos. Em um primeiro momento, o termo group by faz o mesmo papel do termo distinct, mas o termo group by tem outras aplicações. A cláusula a seguir terá 69 registros pelo fato de termos agrupado pela coluna município.

```
select municipio
from tbcurso
group by municipio
order by municipio asc;
```

A cláusula a seguir gera erro, pois incluímos no resultado (à direita do termo select) uma coluna que não existe no termo group by. À direita do termo select só podem estar as colunas usadas à direita do termo group by.

```
select municipio, curso
from tbcurso
group by municipio
order by municipio asc;
```

A cláusula a seguir possui 333 registros. Ela não gera erro pelo fato de terem as mesmas colunas no select e group by. Neste exemplo, o resultado foi agrupado pelas colunas municipio e curso, podemos agrupar usando qualquer quantidade de colunas.

```
select municipio, curso
from tbcurso
group by municipio, curso
order by municipio asc;
```

Ao usar group by – à direita do select só poderão estar as colunas incluídas no group by

II. Funções agrupadoras

Primeiramente é necessário entender o conceito de função:

- Uma função é um bloco de código que é executado ao ser invocado;

- Uma função possui um nome e para ser chamada é necessário fornecer o nome da função seguido de um par de parênteses. No exemplo a seguir, now é o nome da função:

```
select now();
```

- Uma função retorna algo, a função now retorna a data e horário atual;
- Uma função pode receber parâmetros dentro dos parênteses. No exemplo a seguir, a função length recebe como parâmetro um texto e retorna a quantidade de caracteres desse texto:

```
select length('Ana');
```

- A função length recebe somente um texto como parâmetro. Cada função possui a lista de parâmetros definidos. Para saber a lista e ordem dos parâmetros é necessário consultar o help do PostgreSQL. Como exemplo, a função replace recebe três textos como parâmetros. Os parâmetros são separados por vírgula. A função replace retornará o texto Anx Mxrix, ou seja, a letra a será substituída por x no texto Ana Maria:

```
select replace('Ana Maria','a','x');
```

Existem as seguintes funções agrupadoras:

```
select count(*), sum(vaga), avg(vaga), min(vaga), max(vaga)
from tbcurso;
```

Data Output | Explain | Messages | Notifications

count bigint	sum bigint	avg numeric	min integer	max integer
454	17415	38.3590308370044053	20	80

- count(): recebe como parâmetro o nome de uma coluna ou * e retorna a quantidade de registros do resultado. No exemplo anterior, a função count(*) retornou a quantidade de registros da tbcurso;
- sum(): recebe como parâmetro o nome de uma coluna que possui valor numérico e retorna a soma dos valores contidos nos registros. No exemplo anterior, a função sum(vaga) retornou a soma de todos os valores existentes na coluna vaga;

  Este exemplo apresenta erro pelo fato da coluna municipio possuir texto. A função sum só poderá receber valores numéricos:

```
select sum(municipio)
from tbcurso;
```

- avg(): recebe como parâmetro o nome de uma coluna que possui valor numérico e retorna o valor médio dos valores contidos nos registros. Avg é abreviação de average. No exemplo anterior, a função avg(vaga) retornou a média de todos os valores existentes na coluna vaga;

  Este exemplo apresenta erro pelo fato da coluna municipio possuir texto. A função avg só poderá receber valores numéricos:

```
select avg(municipio)
from tbcurso;
```

- min() e max(): recebem como parâmetro o nome de uma coluna que possui valor numérico e retorna, respectivamente, o menor e maior valor contido nos registros. No exemplo anterior, as funções min(vaga) e max(vaga) retornou, respectivamente, o menor e maior valor existente na coluna vaga.

  Apesar de ser recomendado que as funções min e max recebam valores numéricos. Elas podem receber texto, neste caso elas retornam, respectivamente ,o 1º e último texto considerando a ordem alfabética:

```
select min(municipio), max(municipio)
from tbcurso;
```

Data Output | Explain | Messages | Notifications

min text	max text
Adamantina	Taubaté

Observações:

- Constitui erro chamar uma função agrupadora que recebe um valor numérico passando o asterisco. O exemplo a seguir apresentará erro:

```
select sum(*)
from tbcurso;
```

- Constitui erro chamar uma função agrupadora que recebe um valor numérico passando uma coluna que possui textos. O exemplo a seguir apresentará erro:

```
select sum(municipio)
from tbcurso;
```

O termo group by cria subconjuntos e as funções agrupadoras atuam nesses subconjuntos. No exemplo a seguir, o group by criará quatro subconjuntos, um subconjunto para cada turno (EaD, matutino, noturno e vespertino) e a função count será aplicada em cada um desses subconjuntos. Por este motivo, obteve-se que existem 47 cursos EaD nas Fatecs.

```
select turno, count(*)
from tbcurso
group by turno
order by turno asc;
```

turno character varying	count bigint
EaD	47
matutino	155
noturno	213
vespertino	39

III. Termo having da cláusula select

O termo having é aplicado em subconjuntos para filtrar o resultado. No exemplo ao lado, a expressão `count(*) < 100` será aplicada após a função `count(*)` atuar nos subconjuntos.

```
select turno, count(*)
from tbcurso
group by turno
having count(*) < 100
order by turno asc;
```

	turno character varying	count bigint
1	EaD	47
2	vespertino	39

Em outras palavras, o termo having é equivalente ao termo where, pois usamos expressões que resultam em valores true/false. Porém, funções agrupadoras não podem ser usadas no termo where. O exemplo ao lado gera erro:

```
select turno, count(*)
from tbcurso
where count(*) < 100
group by turno
order by turno asc;
```

IV. Renomear colunas no resultado

O comando as é usado para renomear uma coluna no resultado. No exemplo a seguir as colunas foram renomeadas no resultado do select para Quantidade e Total de vagas.

Observe que o texto precisa estar obrigatoriamente entre aspas duplas.

```
select count(*) as "Quantidade", sum(vaga) as "Total de vagas"
from tbcurso;
```

Quantidade bigint	Total de vagas bigint
454	17415

Exercícios

Para fazer os exercícios use as cláusulas do arquivo Aula7 - Clausula SQL.txt para criar a tbcurso e colocar os 454 registros.

Exercício 1: Fazer uma consulta para listar a quantidade de cursos por unidade. Apresente o resultado ordenado em ordem alfabética. Renomear as colunas para Fatec e Quantidade.

O resultado terá 75 registros.

Dica: use o termo group by e função agrupadora count.

	Fatec character varying	Quantidade bigint
1	Fatec Adamantina	2
2	Fatec Americana - Ministro Ralph Biasi	12
3	Fatec Araçatuba - Prof. Fernando Amaral de Almeida Prado	3
4	Fatec Araraquara	3
5	Fatec Araras	2
6	Fatec Assis	2
7	Fatec Baixada Santista - Rubens Lara	11
8	Fatec Barretos	1
9	Fatec Barueri - Padre Danilo José de Oliveira Ohl	9
10	Fatec Bauru	7

Exercício 2: Alterar a consulta do Exercício 1 para mostrar o resultado ordenado em ordem decrescente de quantidade.

O resultado terá 75 registros.

Dica: ordene pela quantidade.

	Fatec character varying	Quantidade bigint
1	Fatec São Paulo	23
2	Fatec Sorocaba - José Crespo Gonzales	14
3	Fatec Zona Leste	12
4	Fatec Americana - Ministro Ralph Biasi	12
5	Fatec Baixada Santista - Rubens Lara	11
6	Fatec Osasco - Pref. Hirant Sanazar	11
7	Fatec Mauá	10
8	Fatec São José dos Campos - Prof. Jes...	10
9	Fatec Tatuí - Prof. Wilson Roberto Ribeir...	9
10	Fatec Barueri - Padre Danilo José de Oli...	9

Exercício 3: Alterar a consulta do Exercício 2 para mostrar apenas o registro que possui a maior quantidade.

O resultado terá 1 registro.

Dica: use o termo limit.

	Fatec character varying	Quantidade bigint
1	Fatec São Paulo	23

Exercício 4: Alterar a consulta do Exercício 3 para mostrar apenas o registro que possui a segunda maior quantidade.

O resultado terá 1 registro.

Dica: use o termo offset.

	Fatec character varying	Quantidade bigint
1	Fatec Sorocaba - José Crespo Gonzales	14

Exercício 5: Alterar a consulta do Exercício 1 para mostrar apenas as unidades que possuem exatamente três cursos.

O resultado terá 9 registros.

Dica: use o termo having.

	Fatec character varying	Quantidade bigint
1	Fatec Araçatuba - Prof. Fernando Amaral de Almeida Prado	3
2	Fatec Araraquara	3
3	Fatec Capão Bonito	3
4	Fatec Itapira - Ogari de Castro Pacheco	3
5	Fatec Itatiba	3
6	Fatec Jacareí - Prof. Francisco de Moura	3
7	Fatec Pompéia - Shunji Nishimura	3
8	Fatec São Roque	3
9	Fatec Sumaré	3

Exercício 6: Fazer uma consulta para listar a quantidade de cursos por turno e unidade. Apresente o resultado ordenado em ordem alfabética. Renomear as colunas para Fatec e Quantidade.

O resultado terá 211 registros.

	Fatec character varying	turno character varying	Quantidade bigint
1	Fatec Adamantina	noturno	2
2	Fatec Americana - Ministro Ralph Biasi	EaD	1
3	Fatec Americana - Ministro Ralph Biasi	matutino	5
4	Fatec Americana - Ministro Ralph Biasi	noturno	5
5	Fatec Americana - Ministro Ralph Biasi	vespertino	1
6	Fatec Araçatuba - Prof. Fernando Amaral de Almeida Prado	EaD	1
7	Fatec Araçatuba - Prof. Fernando Amaral de Almeida Prado	matutino	1
8	Fatec Araçatuba - Prof. Fernando Amaral de Almeida Prado	noturno	1
9	Fatec Araraquara	matutino	1
10	Fatec Araraquara	noturno	2

Exercício 7: Alterar a consulta do Exercício 6 para listar somente as unidades que possuem cinco cursos no mesmo turno.

O resultado terá 7 registros.

	Fatec character varying	turno character varying	Quantidade bigint
1	Fatec Americana - Ministro Ralph Biasi	matutino	5
2	Fatec Americana - Ministro Ralph Biasi	noturno	5
3	Fatec Botucatu	noturno	5
4	Fatec Mauá	noturno	5
5	Fatec Osasco - Pref. Hirant Sanazar	noturno	5
6	Fatec São José dos Campos - Prof. Jessen Vidal	noturno	5
7	Fatec Zona Leste	matutino	5

Exercício 8: Alterar a consulta do Exercício 6 para listar somente o resultado de São José dos Campos.

O resultado terá 3 registros.

	Fatec character varying	turno character varying	Quantidade bigint
1	Fatec São José dos Campos - Prof. Jessen Vidal	EaD	1
2	Fatec São José dos Campos - Prof. Jessen Vidal	matutino	4
3	Fatec São José dos Campos - Prof. Jessen Vidal	noturno	5

Exercício 9: Alterar a consulta do Exercício 8 para listar a quantidade total de vagas por turno.

O resultado terá 3 registros.

Dica: use a função sum.

	Fatec character varying	turno character varying	Quantidade bigint
1	Fatec São José dos Campos - Prof. Jessen Vidal	EaD	20
2	Fatec São José dos Campos - Prof. Jessen Vidal	matutino	160
3	Fatec São José dos Campos - Prof. Jessen Vidal	noturno	200

Exercício 10: Fazer uma consulta para listar a quantidade total de vagas por turno considerando todas as unidades e cursos.

O resultado terá 4 registros.

turno character varying	Quantidade bigint
EaD	940
matutino	6310
noturno	8605
vespertino	1560