Aula 3 – Funções agrupadoras

Modelagem de Banco de Dados – Prof. Marcelo Sudo

**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 Aula7 - 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

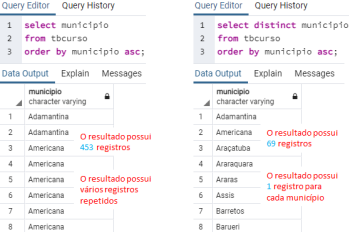
order by nomeDaColuna asc/desc

limit numeroDeRegistros

offset numeroDeRegistros;

**Distinct - evitar registros repetidos no resultado:**

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



O termo distinct precisa estar imediatamente após o select.

1

Aula 3 – Funções agrupadoras

Modelagem de Banco de Dados – Prof. Marcelo Sudo

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

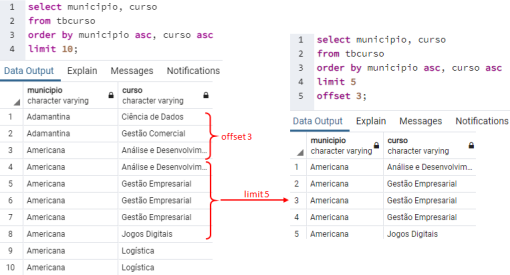
order by municipio asc;

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.2

Aula 3 – Funções agrupadoras

Modelagem de Banco de Dados – Prof. Marcelo Sudo

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

select \*

from tbmail

limit 50

offset 0;

**Group by:**

Cláusula para mostrar a 2ª página de e mails

select \*

from tbmail

limit 50

offset 50;

Cláusula para mostrar a 3ª página de e mails

select \*

from tbmail

limit 50

offset 100;

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;

3

Aula 3 – Funções agrupadoras

Modelagem de Banco de Dados – Prof. Marcelo Sudo

∙ 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;

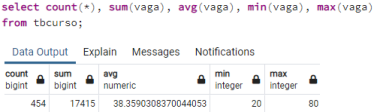
∙ Uma função pode receber parâmetros dentro dos parênteses. No exemplo a seguir, a função length recebe como parâmetro um texto e retorna a quantidade de caracteres desse texto:

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:



∙ 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:

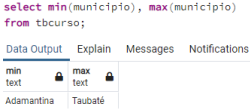
select avg(municipio)

from tbcurso;

4

Aula 3 – Funções agrupadoras

Modelagem de Banco de Dados – Prof. Marcelo Sudo

∙ 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: 

Observações:

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

select sum(\*)

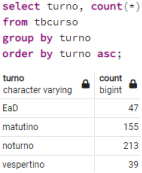
from tbcurso;

∙ 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.



**III. Termo having da cláusula select**

5

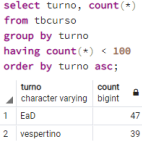
Aula 3 – Funções agrupadoras

Modelagem de Banco de Dados – Prof. Marcelo Sudo

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.

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:

**IV. Renomear colunas no resultado**

****

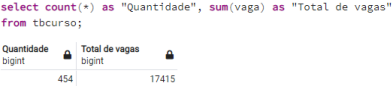
select turno, count(\*) from tbcurso

where count(\*) < 100 group by turno

order by turno asc;

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.



**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.

6

Aula 3 – Funções agrupadoras

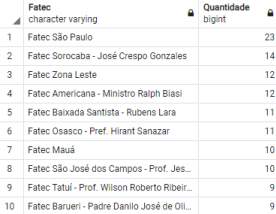
Modelagem de Banco de Dados – Prof. Marcelo Sudo



**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.



**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.

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

7

O resultado terá 1 registro. Dica: use o termo offset.

Aula 3 – Funções agrupadoras

Modelagem de Banco de Dados – Prof. Marcelo Sudo



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

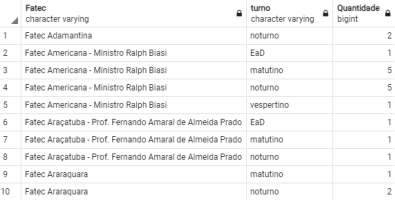
O resultado terá 9 registros.

Dica: use o termo having.



**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.

8

Aula 3 – Funções agrupadoras

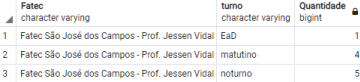
Modelagem de Banco de Dados – Prof. Marcelo Sudo

**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.



**Exercício 8:** Alterar a consulta do Exercício 6 para listar somente o resultado de São José dos Campos. O resultado terá 3 registros.



**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.



**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.

9