

Bacharelado em Sistemas de Informação

Banco de Dados Aula 08 SQL – DML Parte 4

Dr. Diego Buchinger
diego.buchinger@udesc.br

SQL

Structured Query Language

DML – *Data Manipulation Language*
Ordenação, Limite e Agrupamentos

MySQL – Ordenação dos Registros

- Os resultados de uma consulta podem ser ordenados através da cláusula **ORDER BY**
 - ❑ Pode seguir múltiplos critérios, ordenados pela sua precedência

```
SELECT lista_atributos
FROM lista_tabelas
[WHERE condições]
[ORDER BY nome_atrib_1 [ASC|DESC]
{[, nome_atrib_n [ASC|DESC]]};
```

OBS: Se a ordem não for definida (ASC ou DESC), a ordem crescente é utilizada por padrão

- ❑ Ex: mostre o nome e idade dos pacientes, ordenando os resultados pela idade dos pacientes de forma decrescente (mais velhos primeiro), e caso existam idades iguais, ordene estes registros pelo nome, de forma crescente.

```
SELECT nome, idade FROM pacientes
ORDER BY idade DESC, nome ASC;
```

MySQL – Limitando os Registros

- Os resultados de uma consulta podem ser limitados através da cláusula **LIMIT**

```
SELECT lista_atributos
FROM lista_tabelas
[WHERE condições]
[ORDER BY regras]
[LIMIT v1 [,v2]];
```

- Se apenas v_1 for utilizado, ele representa o número de registros que serão exibidos
- Se v_1 e v_2 forem utilizados, v_1 representa quantos registros iniciais devem ser pulados e v_2 representa o número de tuplas

```
SELECT lista_atributos
FROM lista_tabelas
[WHERE condições]
[ORDER BY regras]
[LIMIT qtd OFFSET ini];
```

- No PostgreSQL deve-se utilizar **LIMIT** para indicar o número de tuplas e **OFFSET** para indicar quantos registros iniciais devem ser pulados.

MySQL – Limitando os Registros

- Os resultados de uma consulta podem ser limitados através da cláusula **LIMIT**

- ❑ Ex: Retornar o nome e a idade dos 3 pacientes mais velhos

```
SELECT nome, idade FROM pacientes  
ORDER BY idade LIMIT 3;
```

- ❑ Ex: Retornar o nome, a idade e o salário dos funcionários que recebem o segundo e terceiro maior salário

```
SELECT nome, salario  
FROM funcionarios  
ORDER BY salario DESC  
LIMIT 1, 2;
```

MYSQL

```
SELECT nome, salario  
FROM funcionarios  
ORDER BY salario DESC  
LIMIT 2 OFFSET 1;
```

POSTGRESQL

MySQL – Agrupamento de registros

- Registros podem ser agrupados pela cláusula **GROUP BY**
 - ❑ Agrupamento é feito pelos registros que possuem mesmo valor nas colunas especificadas
 - ❑ Apenas os atributos de agrupamento e as funções de agregação podem aparecer no resultado final da consulta!

```
SELECT lista_atributos
FROM lista_tabelas
[WHERE condições]
[GROUP BY lista_atributos_agrupamento
HAVING condição_para_agrupamento];
```

- ❑ Relembrando algumas funções de agregação:
COUNT, SUM, AVG, MIN, MAX

MySQL – Agrupamento de registros

- Registros podem ser agrupados pela cláusula **GROUP BY**

❖ Listar quantos médicos possuem a mesma especialidade

```
SELECT especialidade, COUNT(*)  
FROM medicos GROUP BY especialidade;
```

❖ Qual é a média de salários pagos aos funcionários oriundos de uma mesma cidade?

```
SELECT cidade, AVG(salario)  
FROM funcionarios GROUP BY cidade;
```

MySQL – Agrupamento de registros

- Registros podem ser agrupados pela cláusula **GROUP BY**
 - ❑ Condições regulares com a cláusula **WHERE** podem ser aplicadas sobre colunas que não fazem parte do agrupamento
 - ❑ Para aplicar condições sobre os grupos que são formados pelo agrupamento deve ser utilizada a cláusula **HAVING**
 - ❑ As condições da cláusula **HAVING** só podem ser definidas sobre os atributos do agrupamento ou sobre funções de agregação
- ❖ Listar as cidades que são origem de pelo menos mais do que um funcionário maior de 30 anos e informar quantos funcionários estão cadastrados nestas cidades

```
SELECT cidade, COUNT(*)  
FROM funcionarios  
WHERE idade > 30  
GROUP BY cidade HAVING COUNT(*) > 1;
```


MySQL – Atualizações usando Consultas

- Comandos de atualização (**INSERT**, **UPDATE** e **DELETE**) podem incluir comandos de consulta
 - ❑ Ex: a média Maria pediu para cancelar (remover) todas as suas consultas após as 17:00.

```
DELETE FROM consultas WHERE hora > '17:00'  
      AND codm = (SELECT codm FROM medicos  
                  WHERE nome = 'Maria');
```

- ❑ Ex: a direção da clínica determinou que deve haver sempre dois médicos por ambulatório, caso contrário o médico não deve ter um ambulatório definido/fixo.

```
UPDATE medicos SET nroa = NULL  
WHERE NOT EXISTS  
      (SELECT * FROM medicos AS m  
        WHERE m.codm != medicos.codm  
          AND m.nroa = medicos.nroa);
```

Exercícios

1. Realize as seguintes consultas:

- a. Mostre os dados de todos os funcionários ordenados pelo salário (decrescente) e pela idade (crescente). Buscar apenas os três primeiros funcionários nesta ordem.
- b. Mostre o nome dos médicos, o número e andar do ambulatório onde eles atendem, ordenado pelo número do ambulatório (crescente)
- c. Mostre o nome do médico e o nome dos pacientes com consulta marcada, ordenado pela data e pela hora (crescente). Buscar apenas as tuplas 3 a 5, nesta ordem.
- d. Mostre os nomes das doenças e a quantidade total de vezes que foram diagnosticadas nas consultas.
- e. Mostre as datas e a quantidade total de consultas em cada data, para horários vespertinos, isto é, após às 12:00 horas.
- f. Mostrar os andares onde existem ambulatórios e o total de leitos (capacidade) por andar.

Exercícios

1. Realize as seguintes consultas:

- g. Mostrar os andares onde existem ambulatórios e a soma de capacidade no andar seja maior ou igual a 100.
- h. Mostrar o nome dos médicos que possuem mais de uma consulta marcada.
- i. Passar todas as consultas da paciente Ana para às 19:00 utilizando um único comando (sem consultar seu codp)
- j. Excluir os pacientes que não possuem consultas marcadas.
- k. Passar todas as consultas do médico Pedro marcadas para o período da manhã para o dia 21/06/2024, no mesmo horário. Utilizar um único comando sem consultar o codm do médico.
- l. O ambulatório 4 (nroa=4) foi transferido para o mesmo andar do ambulatório 1 e sua capacidade é agora o dobro da capacidade do ambulatório de maior capacidade da clínica. Utilizar apenas as informações fornecidas para atualizar, sem consultar e utilizar dados não fornecidos neste enunciado.