

# SQL BATCH

## 16. Atividades

Os comandos e “SQL Batches” que você escrever devem ser salvos no arquivo de nome

```
ra99999999_batch.txt,
```

substituindo o prefixo pelo seu registro acadêmico. Ao final, envie o arquivo ao servidor.

**16.1.** Na figura 4, estão descritos os esquemas das tabelas:

```
tb_pessoas_teste e
tb_pes_amigas_pes.
```

Utilizando os dados das duas tabelas e o operador de concatenação, produza o seguinte texto para cada um dos relacionamentos que existem entre as pessoas.

A pessoa de nome “**AAA**” e código “**P100000**” requisitou amizade da pessoa de nome “**BBB**” e código “**P200000**”.

Os textos em vermelho precisam ser obtidos do banco de dados e os textos em preto são cadeias constantes. As aspas devem aparecer também no resultado.

**16.2.** Crie as tabelas abaixo que servirão para testes do script apresentado como exemplo e também para atividades futuras.

```
create table tb_pessoas_ligadas(nivel int not null,
                                cod_pes char(7) not null,
                                cod_pes_anterior char(7) not null,
                                constraint ct_pessoas_ligadas_pk primary key(cod_pes, cod_pes_anterior))

create table tb_frenteira(cod_pes char(7) not null,
                           constraint ct_frenteira_pk primary key(cod_pes)
)

create table tb_adjacentes_temp(cod_pes char(7) not null,
                                 cod_pes_anterior char(7) not null,
                                 constraint ct_adjacentes_temp_pk primary key(cod_pes, cod_pes_anterior)
)
```

Teste o script do exemplo. Ele está disponível no mesmo endereço onde estão os demais documentos da disciplina com o seguinte nome:

```
recursive_friends_aln.sql.
```

Você não precisa salvar nada referente a este item no arquivo de respostas.

**16.3.** Faça alterações no o script “SQL” deste documento para apresentar a lista com o código e nome dos amigos dos amigos dos amigos de um determinado indivíduo, sem levar em conta a orientação das arestas, ou seja, eles estão a três arestas de distância (nível 3). Ordene os resultados pelo nome das pessoas. Observação: Você deve utilizar a cláusula “distinct” somente quando necessário. Neste caso, por que ela é necessária?

Codifique o script que produz o resultado solicitado e teste para as seguintes pessoas:

P100000 e  
P701110.

Os resultados do primeiro teste devem ser semelhantes aos seguintes:

```
Amigos a distancia 3 de
-----
P100000

cod_pes nome_pes
-----
P101120 Amanda
P101130 Fabiola
P101110 Gian
```

Os resultados do segundo teste devem ser semelhantes aos seguintes:

```
Amigos a distancia 3 de
-----
P701110

cod_pes nome_pes
-----
P701122 Cintia
P701151 Claudio
P701152 Fabrizio
P701142 Frederico
P701131 Goulart
P701133 Jailson
P701134 Jefferson
P701143 João
P700000 Karina
P701153 Leo
P701500 Liliane
P701400 Lucilene
P701121 Luiz
P701124 Marilisa
P701600 Milena
P701123 Milton
P701144 Raul
P701154 Reinaldo
P701141 Samantha
P701200 Tania
P701132 Tulio
P701300 Vanda
```

**16.4** Seja “p” uma determinada pessoa. Criar um “SQL Batch” semelhante ao apresentado no exemplo que encontra todas as pessoas que direta ou indiretamente alcançam a pessoa “p” percorrendo as arestas sempre no sentido da origem para destino. Falando de outro modo, é preciso encontrar todos os vértices do grafo que direta ou transitivamente levam ao vértice informado por meio das arestas orientadas. Na nomenclatura que estabelecemos, equivale a dizer que desejamos obter o conjunto de todos seus **requisitantes diretos e indiretos**. Esta questão é mais simples do que a do exemplo porque você não precisa colocar no resultado o vértice anterior do percurso nem incluir os amigos respondentes diretos e indiretos, mas precisa manter a informação sobre a distância em número de arestas.

O resultado deve ser apresentado da seguinte forma:

Encontrar os predecessores diretos ou indiretos

-----  
P101111

nivel	cod_pes	nome_pes	data_nasc
-----	-----	-----	-----
0	P101111	Cassia	Apr 24 2002
1	P101110	Gian	Mar 25 2003
2	P101100	Jamil	Apr 20 2004
3	P101000	Ludmila	Apr 18 2002
3	P102000	Adriana	Jul 18 2001
3	P103000	Santoro	Dec 9 2004
4	P100000	Sandro	Jan 2 2000
5	P104200	Marco	Feb 18 2002
6	P104000	Tobias	Oct 23 1999

Observações:

- A ordenação deve ser pelo nível e, em segundo lugar, pelo código da pessoa;
- O vértice de nível zero contém os dados da pessoa para quem a pesquisa é realizada;
- O nível aumenta à medida que fica mais distante o alcance transitivo em relação à pessoa de referência;
- Como um mesmo vértice pode ser predecessor de mais que um vértice, há a necessidade da cláusula “**distinct**”.

Siga a estratégia explicada na figura 5, de forma análoga ao script do exemplo. Utilize as tabelas auxiliares abaixo cujos nomes contêm o sufixo “1” para distinguir das utilizadas no exemplo explicativo. Elas possuem estruturas mais simples porque não contemplam informações sobre os caminhos, ou seja, do passo imediatamente anterior. Como primeira parte deste exercício, crie as seguintes tabelas:

```
create table tb_pessoas_ligadas1(nivel int not null,
                                cod_pes char(7) not null,
                                constraint ct_pessoas_ligadas1_pk primary key(cod_pes)
)
create table tb_frenteiral(cod_pes char(7) not null,
                           constraint ct_frenteiral_pk primary key(cod_pes)
)
create table tb_adjacentes_temp1(cod_pes char(7) not null,
                                  constraint ct_adjacentes_temp1_pk primary key(cod_pes)
)
```

Este script tem riscos potenciais causados por erros de codificação que produzem um “loop” infinito. Os impactos desta condição sobre um ambiente de produção corporativo podem ser catastróficos do ponto de vista da continuidade e estabilidade das operações. As possíveis consequências para o nosso ambiente simplificado do laboratório são:

- a) Consumo excessivo de “CPU” tornando o sistema muito lento para você e seus colegas, podendo, até mesmo, impedir que eles possam trabalhar;
- b) Esgotamento dos recursos de armazenamento.

Portanto, preste atenção!

Codifique o script que produz o resultado solicitado e teste para as seguintes pessoas:

```
'P701110',  
'P104100' e  
'P302000'.
```

Confira os seus resultados com os mostrados a seguir.

## Exemplo 1

Encontrar os predecessores diretos ou indiretos

-----  
P701110

nivel	cod_pes	nome_pes	data_nasc
0	P701110	Agnes	May 28 2000
1	P701100	Helena	Feb 21 2002
2	P701000	Renato	Oct 21 2003
3	P700000	Karina	Sep 30 2003
3	P701131	Goulart	Jul 11 2002
4	P701130	Ruth	May 11 2004
4	P701154	Reinaldo	Aug 21 2003
5	P701150	Izabel	Dec 26 2000

## Exemplo 2

Encontrar os predecessores diretos ou indiretos

-----  
P104100

nivel	cod_pes	nome_pes	data_nasc
0	P104100	Andressa	Nov 9 2001
1	P104000	Tobias	Oct 23 1999
2	P100000	Sandro	Jan 2 2000
3	P104200	Marco	Feb 18 2002

### Exemplo 3

Encontrar os predecessores diretos ou indiretos

-----  
P302000

nivel	cod_pes	nome_pes	data_nasc
0	P302000	Romulo	Oct 1 1999
1	P300000	Heraldo	Jun 26 2003

**16.5.** Considere uma pessoa que fez requisições de amizade a uma ou várias pessoas; e algumas ainda não responderam. Escreva um script que verifica se o requisitante que fez as solicitações pendentes de resposta (aresta verde) possui os respondentes indecisos (podem ser vários requisições não atendidas) entre seus amigos diretos e indiretos, independentemente de seguir por caminhos de **requisitantes ou respondentes**. É claro que os amigos diretos não fazem muito sentido no contexto desta questão porque, neste caso, já haveria uma solitação anterior atendida ou uma inversa igualmente atendida de forma direta. Teste o seu script com os seguintes códigos:

```
'P701120',  
'P501000' e  
'P705100'.
```

Obs: Utilize o script do exemplo e acrescente a verificação. Como a pessoa pode ter várias dessas situações, a resposta deverá apresentar uma linha por respondente indeciso. Vide o exemplo:

Encontrar os "amigos indecisos" entre os amigos diretos ou indiretos

-----  
P701120

cod_pes	nome_pes	data_nasc	nivel
P104200	Marco	Feb 18 2002	NULL
P700000	Karina	Sep 30 2003	3

Neste caso, a pessoa de código “P701120” fez convites para “P104200” e “P700000” que ainda não responderam. A primeira pessoa não está no seu círculo de amigos diretos ou indiretos enquanto a segunda é uma pessoa amiga que está a três arestas de distância. Como sugestão, utilize um “sub-select” escalar na “select\_list”.

O segundo exemplo produz o seguinte resultado.

```
Encontrar os "amigos indecisos" entre os amigos diretos ou indiretos
-----
P501000

cod_pes nome_pes data_nasc nivel
-----
P302300 Suzanne Jul 28 2003 NULL
```

O terceiro exemplo produz o seguinte resultado.

```
Encontrar os "amigos indecisos" entre os amigos diretos ou indiretos
-----
P705100

cod_pes nome_pes data_nasc nivel
-----
P600000 Eliane Apr 14 2003 NULL
P701140 Turmalino May 24 2000 5
P701151 Claudio Dec 18 1999 5
P701600 Milena Jun 11 2003 4
```

**16.6.** Escreva um script que verifica se um requisitante que fez uma solicitação que foi rejeitada (arestas marrons) possui respondentes “pouco amigos” (podem ser vários respondentes) entre as pessoas que estão no conjunto de **requisitantes diretos ou indiretos conforme a orientação das arestas**. É claro que os requisitantes diretos não fazem muito sentido no contexto desta questão porque, neste caso, já haveria uma requisição inversa direta atendida, ou seja, existe um aresta que os liga diretamente em que ele é o respondente. Teste o seu script com os seguintes códigos:

```
'P101130' e
'P701114'.
```

Obs: Utilize o script do exercício 16.2 e acrescente a verificação. Como a pessoa pode ter várias dessas situações, a resposta deverá apresentar uma linha por respondente “pouco amigo”. Vide o exemplo:

```
Encontrar "os poucos amigos" entre os amigos requisitantes indiretos
-----
P101130

cod_pes nome_pes data_nasc nivel
-----
P102000 Adriana Jul 18 2001 2
P302100 Odilon Oct 11 2001 NULL
```

Neste caso, a pessoa de código “P101130” convidou “P102000” e “P302100” e ambos recusaram. O primeiro deles está entre os amigos recursivamente requisitantes na distância 2, mas o segundo não faz parte dessa lista. Utilize um “sub-select” escalar na “select\_list”.

O segundo teste deste exercício produz o seguinte resultado:

```
Encontrar "os poucos amigos" entre os amigos requisitantes indiretos
-----
P701114

cod_pes nome_pes data_nasc nivel
-----
P302200 Alessandro Aug 10 2001 NULL
```

**16.7.** Altere o script do exemplo que está detalhado neste roteiro para apresentar a lista de todos os amigos diretos e indiretos (pela nossa nomenclatura, inclui os requisitantes e respondentes) que estão a maior distância de uma determinada pessoa. O resultado deve mostrar a distância em que eles se encontram na lista de amigos e o vizinho imediatamente anterior no caminho, ou seja, o formato do resultado deve ser o mesmo do exemplo deste roteiro, mas restrito aos dados do nível mais alto ordenados pelo código dos amigos.

Codifique o script que produz o resultado solicitado e teste para as seguintes pessoas:

```
'P701110' e
'P701000'.
```

O resultado do primeiro teste é o seguinte:

```
Encontrar os amigos diretos ou indiretos mais distantes
-----
P701110

nivel cod_pes nome data_nasc cod_pes_anterior nome_pes_anterior
-----
5 P704010 Cleonice Apr 7 2000 P704000 Nataly
5 P705100 Amelia Mar 4 2002 P705000 Lorenzo
```

O resultado do segundo teste é o seguinte:

```
Encontrar os amigos diretos ou indiretos mais distantes
-----
P701000

nivel cod_pes nome data_nasc cod_pes_anterior nome_pes_anterior
-----
3 P701111 Thalita Oct 28 1998 P701110 Agnes
3 P701112 Gustavo Oct 22 2003 P701110 Agnes
3 P701113 Fabiano Oct 28 2004 P701110 Agnes
3 P701114 Ciomara May 14 2002 P701110 Agnes
3 P701121 Luiz Feb 8 2002 P701120 Francisca
3 P701122 Cintia Jul 21 2000 P701120 Francisca
3 P701123 Milton May 12 2000 P701120 Francisca
3 P701124 Marilisa Nov 20 2000 P701120 Francisca
3 P701132 Tulio Jul 7 1999 P701130 Ruth
3 P701133 Jailson Jun 23 2001 P701130 Ruth
```

3 P701134	Jefferson	Mar 20 2002	P701130	Ruth
3 P701141	Samantha	Oct 30 2001	P701140	Turmalino
3 P701142	Frederico	Dec 11 2002	P701140	Turmalino
3 P701143	João	Aug 4 2004	P701140	Turmalino
3 P701144	Raul	Apr 21 2000	P701140	Turmalino
3 P701151	Claudio	Dec 18 1999	P701150	Izabel
3 P701152	Fabrizio	May 26 2002	P701150	Izabel
3 P701153	Leo	Oct 9 2002	P701150	Izabel
3 P704010	Cleonice	Apr 7 2000	P704000	Nataly
3 P705100	Amelia	Mar 4 2002	P705000	Lorenzo

Até a próxima aula!  
 Prof. Satoshi Nagayama.