

Bases de Dados, 2023/2024

Universidade do Algarve

Trabalho prático, Parte 4 – SQL

Nota prévia

Esta parte do trabalho vale 40% da nota da componente prática da disciplina.

Árvores geneológicas

Considere o seguinte esquema relacional:

- `Pessoas(id, nome, sexo, dataNasc)`
- `MaeDe(mae, filho)`
- `PaiDe(pai, filho)`

Estas 3 relações/tabelas servem de base para se manter árvores geneológicas de famílias.

A relação `Pessoas` tem informação sobre pessoas. Cada pessoa tem um `id` único que deverá ser um inteiro, um nome que deverá ser uma string com um máximo de 60 caracteres, `sexo` que deverá ser um só carácter ('m' para masculino ou 'f' para feminino), e uma data de nascimento.

As relações `MaeDe` e `PaiDe`, especificam quem é que é mãe de quem, e quem é que é pai de quem. Especificamente, um tuplo (x, y) da relação `MaeDe` significa que a pessoa com o `id` x é mãe da pessoa com o `id` y . Do mesmo modo, um tuplo (x, y) da relação `PaiDe` significa que a pessoa com o `id` x é pai da pessoa com o `id` y .

Esquema e dados

- Crie um ficheiro chamado `esquema.sql` que contenha a definição das tabelas correspondentes a estas 3 relações, bem como a definição de *triggers* e funções que considere apropriados para garantir a consistência dos dados. Por exemplo, não deve ser permitido inserir um tuplo (x, y) na tabela `PaiDe` se x corresponder a um `id` de uma pessoa do sexo feminino. De forma análoga, não deve ser permitido inserir um tuplo (x, y) na tabela `MaeDe` se x corresponder a um `id` de uma pessoa do sexo masculino. Deve também ser garantido que a data de nascimento de um filho(a) seja forçosamente posterior à data de nascimento de seu pai/mãe.

O nome das tabelas e respectivos atributos devem ser exactamente como especificado no esquema relacional apresentado ao início do enunciado.

- Crie um ficheiro chamado `dados.sql` que contenha comandos de SQL para inserir alguns tuplos nas tabelas. Deverá colocar dados suficientes de modo a poder testar as perguntas que se seguem.

Interrogações (queries) em SQL

- Elabore uma interrogação em SQL que retorne uma tabela com 2 colunas: nome da mãe, nome do filho(a), ordenado por ordem alfabética crescente de nome da mãe, e em caso de empate por ordem alfabética crescente de nome do filho(a).
- Pretende-se fazer duas vistas (*views*) chamadas **AvouDe** e **AvohDe** que indicam quem é que é avô de quem, e quem é que é avó de quem. Especificamente, um tuplo (x, y) da vista **AvouDe** significa que a pessoa com o id x é avô da pessoa com o id y . Do mesmo modo, um tuplo (x, y) da vista **AvohDe** significa que a pessoa com o id x é avó da pessoa com o id y . Implemente estas duas vistas em SQL.
- Elabore uma interrogação em SQL que retorne uma tabela com 2 colunas (nome1 e nome2) de tal forma que nome1 seja avô ou avó de nome2. O output deverá vir ordenado por ordem alfabética crescente de nome de avô/avó (nome1) e em caso de empate por ordem alfabética crescente de nome de neto(a) (nome2).
- Elabore uma interrogação em SQL que retorne uma tabela com 2 colunas (nome1 e nome2) de tal forma que nome1 seja irmão ou irmã de nome2. Duas pessoas são irmãos se tiverem o mesmo pai e a mesma mãe. O output deverá vir ordenado por ordem alfabética crescente de nome1. Escreva o código de tal forma que se tuplo (x, y) aparecer no output, não deverá aparecer o tuplo (y, x) .
- Elabore uma interrogação em SQL que retorne uma tabela com 2 colunas (nome1 e nome2) de tal forma que nome1 seja meio-irmão ou meio-irmã de nome2. Duas pessoas são meio-irmãos se tiverem o mesmo pai ou a mesma mãe, mas não ambos. O output deverá vir ordenado por ordem alfabética crescente de nome1. Escreva o código de tal forma que se tuplo (x, y) aparecer no output, não deverá aparecer o tuplo (y, x) .
- Elabore uma interrogação em SQL que retorne uma só coluna com o nome das pessoas que têm mais de 3 filhos(as).
- Elabore uma interrogação em SQL que retorne uma só coluna com o nome das pessoas que têm mais netos(as).

NOTA: Ao resolver as questões acima enunciadas, poderá obviamente criar vistas (views) adicionais e usá-las como suporte para elaborar as suas interrogações.

Prazos e outras formalidades

- **Prazo de entrega:** 8/Dez/2023 até às 23:59
- **Escala de avaliação:** um inteiro de 0..5 (Esta nota será convertida numa escala de 0..20 ao final da disciplina)
- **Penalização de atraso:** -1 valor (na escala de 0..5) por cada dia de atraso (**ATENÇÃO: 1 minuto de atraso já conta como 1 dia de atraso**). Não serão aceites entregas com mais de 2 dias de atraso.
- **Local de entrega:** Tutoria eletrónica.
- **Formato:** A submissão eletrónica tem de ser feita através de 1 único ficheiro ZIP. Esse ficheiro ZIP deverá conter ficheiros com código SQL e respectivos outputs, cujos nomes e conteúdo devem ser os seguintes,
 - `esquema.sql`: Ficheiro com o código SQL que permita criar o esquema da base de dados.
 - `dados.sql`: Ficheiro com comandos SQL para inserir dados nas tabelas da base de dados.
 - `a.sql`, `b.sql`, `c.sql`, ..., `g.sql`: Ficheiros com o código SQL referente às alínea a), b), c), ..., g).

NOTA IMPORTANTE: Os ficheiros `esquema.sql` e `dados.sql` devem poder ser usados por mim para recriar a vossa base de dados. Para além disso, é suposto que os ficheiros correspondentes às vossas queries executem sem erros através da opção `\i` do `psql`. Face ao exposto, devem certificar-se que o vosso código funciona correctamente em PostgreSQL.