

1º Teste de Bases de Dados
30/10/2024
duração: 1 hora e 30 minutos (11:00 às 12:30)

Pretende-se desenvolver uma base de dados para gerir a informação de uma rede social de *Fãs de cinema*. Para gerir a rede é necessário representar os dados dos filmes, atores, membros, amigos e gostos com as seguintes relações:

- membro(Nome, Id, País)

Nome - Nome do membro

Id - Identificador do membro, não há 2 membros com o mesmo Id

País - País em que o membro reside

- amigo(Membro, Amigo)

Membro, Amigo - identificadores dos membros amigos

- gosta(Membro, Filme)

Membro - Identificador do membro

Filme - Nome do filme

- filme(Nome, Tipo, Ano, Custo)

Nome - Nome do filme, não há 2 filmes com o mesmo nome

Tipo - Género do filme, pode ser mistério, drama ou etc.

Ano - Ano em que o filme estreou

Custo - Custo da produção do filme

- ator(Nome, País, DNascimento)

Nome - Nome do ator, não há 2 atores com o mesmo nome

País - Nome do País em que o ator nasceu

DNascimento - Ano em que o ator nasceu

- participou(Filme, Ator, Papel)

Filme - Nome do filme

Autor - Nome do autor

Papel - Nome do papel do autor no filme, ex: Batman, Poirot, etc.

Note que um ator pode ter mais do que um papel num filme

1. Indique as chaves candidatas e as chaves estrangeiras de cada relação.

- membro(Nome, Id, País)

Chave candidata - Id

Chave primária - Id

Chave estrangeira - não há

- amigo(Membro, Amigo)
 - Chave candidata - MembroAmigo
 - Chave primária - MembroAmigo
 - Chave estrangeira - Membro e Amigo referenciam membro(Id)
- gosta(Membro, Filme)
 - Chave candidata - MembroFilme
 - Chave primária - MembroFilme
 - Chave estrangeira - Membro referencia membro(Id) e Filme referencia filme(Nome)
- filme(Nome, Tipo, Ano, Custo)
 - Chave candidata - Nome
 - Chave primária - Nome
 - Chave estrangeira - não há
- ator(Nome, País, DNascimento)
 - Chave candidata - Nome
 - Chave primária - Nome
 - Chave estrangeira - não há
- participou(Filme, Ator, Papel)
 - Chave candidata - FilmeAtorPapel
 - Chave primária - FilmeAtorPapel
 - Chave estrangeira - Filme referencia filme(Nome), Ator referencia ator(Nome)

2. Indique as instruções em SQL para criar a tabela das relações: amigo e participou.

```

create table membro(Id char(5) PRIMARY key,
                    nome varchar(50),
                    Pais varchar(50));

create table filme(Nome varchar(50) PRIMARY key,
                   tipo varchar(50),
                   Ano integer,
                   custo integer);

create table gosta(Membro char(5),
                   Filme varchar(50),
                   primary key (Membro,Filme),
                   FOREIGN key (Membro) REFERENCES membro(Id),
                   FOREIGN key (Filme) REFERENCES filme(Nome));

create table ator(Nome varchar(50) primary key, Pais varchar(50),
DNascimento integer);

```

```

create table amigo(
    Membro char(5),
    Amigo char(5),
    Primary key (Membro, Amigo),
    Foreign key (Membro) references membro(Id),
    Foreign key (Amigo) references membro(Id));

create table participou(
    Filme varchar(50),
    Ator varchar(50),
    Papel varchar(50),
    Primary key (Filme, Ator, Papel),
    Foreign key (Filme) references filme(Nome),
    Foreign key (Ator) references ator(Nome));

insert into membro values ('id1', 'Antonio', 'PT');
insert into membro values ('id2', 'Manuel', 'PT');
insert into membro values ('id3', 'Maria', 'PT');
insert into membro values ('id4', 'Antonia', 'ES');
insert into membro values ('id5', 'Luisa', 'ES');

insert into amigo values ('id1','id2');
insert into amigo values ('id1','id3');
insert into amigo values ('id1','id4');
insert into amigo values ('id4','id5');

insert into filme values ('O Padrinho', 'Drama', 1981,100);
insert into filme values ('O Pinoquio', 'Drama', 1971,50);

insert into gosta values ('id2', 'O Padrinho');
insert into gosta values ('id3', 'O Padrinho');
insert into gosta values ('id5', 'O Padrinho');
insert into gosta values ('id2', 'O Pinoquio');

insert into ator values ('Marlon Brando','EU' 1950);
insert into ator values ('Merly Streep','EU' 1960);

insert into participou values ('Marlon Brando', 'O Padrinho', 'Corleone');
insert into participou values ('Merly Streep', 'O Pinoquio', 'Baleia');
insert into participou values ('Merly Streep', 'O Padrinho', 'Mary');

```

3. Indique as expressões em SQL e em álgebra relacional que lhe permitem obter a seguinte informação:

- (a) Que amigos do membro 'id1' é que gostam de 'O padrinho'?

$$\pi_{(Amigo)}(\sigma_{(amigo.Membro='id1' \wedge Filme='OPadrinho' \wedge gosta.Membro=Amigo)}(amigoXgosta))$$

```

select Amigo
from amigo, gosta
where amigo.Membro= 'id1' and Filme='O Padrinho' and gosta.Membro=Amigo

```

- (b) Qual é o nome dos membros que não gostam de 'O Pinóquio' ?

$$\pi_{(Nome)}(membro)-\pi_{(Nome)}(\sigma_{(Id=Membro \wedge Filme='OPinquito')}(membroXgosta))$$

```

select Nome
from membro
except
select Nome
from membro, gosta
where Id=Membro and Filme='O Pinóquio'

```

- (c) Quais são os atores que participaram em 'O padrinho' e em 'O Pinóquio'?

$$\pi_{(Ator)}(\sigma_{(Filme='OPadrinho')}(participou)) \cap \pi_{(Ator)}(\sigma_{(Filme='OPinquito')}(participou))$$

```

select Ator
from participou
where Filme = 'O Padrinho'
intersection
select Ator
from participou
where Filme = 'O Pinóquio'

```

- (d) Para cada membro indique os atores dos filmes de que gosta.

$$\pi_{(Membro,Ator)}(gosta \bowtie participou)$$

```

select Membro, Ator
from gosta natural inner join participou

```

- (e) Quantos atores participaram em filmes no papel de 'Poirot'?

$$\mathcal{G}_{count(Ator)}(\sigma_{(Papel='Poirot')}(participou))$$

```

select count(Ator)
from participou
where Papel='Poirot'

```

- (f) Que membros são amigos de membros que gostam de filmes em que participou o 'Marlon Brando'?

$$\pi_{(amigo.Membro)}(\sigma_{(Ator='MarlonBrando' \wedge Amigo=gosta.Membro \wedge gosta.Filme=participou.Filme)}(amigoXgostaXparticipou))$$

```

select amigo.Membro
from amigo, gosta, participou
where Ator= 'Marlon Brando' and Amigo=gosta.Membro and
gosta.Filme = participou.Filme

```

- (g) Para cada filme indique o número de membros que residem em Portugal e gostam do filme.

$$Filme \mathcal{G}_{count}(ID)(\sigma_{(ID=Membro \wedge País=Portugal)}(membro X gosta))$$

```

select Filme, count(ID)
from membro, gosta
where ID=Membro and País = Portugal

```

- (h) Quais são os membros que gostam do filme mais caro?

$$\begin{aligned} \text{caro} &\leftarrow \mathcal{G}_{(max(custo) as custo)}(filme) \\ \text{filme_caro} &\leftarrow \pi_{(Nome as Filme)}(filme \bowtie caro) \\ \pi_{(Membro)}(gosta \bowtie filme_caro) \end{aligned}$$

```

with caro as (select max(custo) as custo from filme)
with filme\_caro as (select Nome as Filme from filme natural inner join caro)
select Membro
from gosta natural inner join filme\_caro

```

- (i) Qual é o ator que participou em todos os filmes em que o 'Marlon Brando' participou?

$$\pi_{(Ator,Filme)}(participou) \div \pi_{(Filme)}(\sigma_{(Ator='MarlonBrando')}(participou))$$

```

select Ator
from participou as a
where not exists (select Filme
                  from participou
                  where ator = 'Marlon Brando'
                  except
                  select Filme
                  from participou as b
                  where b.ator = a.ator)

```