

2.

Cliente = (login, password, firstName, lastName, dtAdesao, cCredito, porta, rua, cidade)

Pagamento = (login, pagamento, dtPagamento)

Historial = (login, idFilme, dtVisto)

Filme = (idFilme, titOri, titIng, titPt, lingua, dtEstreia, resumo, localFilme, url, path)

Etiqu_Filme = (idFilme, g nero)

Director = (idDirector, firstName, lastName, dataNasc, nacionalidade)

Actor = (idActor, firstName, lastName, dataNasc, nacionalidade)

FilmeActores = (idFilme, idActor)

FilmeDirectores = (idFilme, idDirector)

Premio_filme = (idFilme, pr mio)

Premio_actor = (idFilme, idActor, pr mio)

Premio_director = (idFilme, idDirector, pr mio)

3.

Cliente

login -> password, firstName, lastName, dt_adesao, c_credito, porta, rua, cidade

Pagamento

login, pagamento -> dt_pagam

Historial

login, idFilme -> dt_visto

Filme

idFilme -> tit_ori, tit_ing, tit_pt, lingua, dt_estreia, resumo, local_fil, url, path

Etiqu_Filme

idFilme -> g nero

Director

idDirector -> firstName, lastName, dataNasc, nacionalidade

Actor

idActor -> firstName, lastName, dataNasc, nacionalidade

FilmeActores

idFilme -> idActor

FilmeDirectores

idFilme -> idDirector

Premio_filme

idFilme -> pr mio

Premio_actor

idFilme, idActor -> prémio

Premio_director

idFilme, idDirector -> prémio

7.

Cliente

super-chave: login

chave candidata: login

chave primária: login

chave estrangeira: N/A

Pagamento

super-chave: login-pagamento. login-pagamento-dtPagamento

chave candidata: login-pagamento

chave primária: login-pagamento

chave estrangeira: login

Historial

super-chave: login

chave candidata: login

chave primária: login

chave estrangeira: login, idFilme

Filme

super-chave: idFilme

chave candidata: idFilme

chave primária: idFilme

chave estrangeira: N/A

Director

super-chave: idDirector

chave candidata: idDirector

chave primaria: idDirector

chave estrangeira: N/A

Actor

super-chave: idActor

chave candidata: idActor

chave primaria: idActor

chave estrangeira: N/A

FilmeDirectores

super-chave: idFilme-idDirector

chave candidata: idFilme-idDirector

chave primária: idFilme-idDirector

chave estrangeira: idFilme, idDirector

FilmeActores

super-chave: idFilme-idActor

chave candidata: idFilme-idActor

chave primária: idFilme-idActor

chave estrangeira: idFilme, idActor

PremioFilme

super-chave: idFilme-prémio

chave candidata: idFilme-prémio

chave primária: idFilme-prémio

chave estrangeira: idFilme

Premio_actor

super-chave: idFilme-idActor-prémio

chave candidata: idFilme-idActor-prémio

chave primária: idFilme-idActor-prémio

chave estrangeira: idFilme, idActor

Premio_director

super-chave: idFilme-idDirector-prémio

chave candidata: idFilme-idDirector-prémio

chave primária: idFilme-idDirector-prémio

chave estrangeira: idFilme, idDirector

EtqFilme

super-chave: idFilme-género

chave candidata: idFilme-género

chave primária: idFilme-género

chave estrangeira: idFilme

10.

a)

```
SELECT titori FROM filme NATURAL INNER JOIN etiqFilme
WHERE genero = 'Thriller'
INTERSECT
SELECT titori FROM filme NATURAL INNER JOIN etiqFilme
WHERE genero = 'Mistério';
```

b)

```
SELECT DISTINCT c.firstname, c.lastname nome
FROM cliente c NATURAL INNER JOIN pagamento p
WHERE p.dtpagamento >= '2015-11-02';
```

c)

```
SELECT DISTINCT TitOri FROM filme NATURAL INNER JOIN filmeActores NATURAL
INNER JOIN actor
WHERE lingua = 'Inglês' and nacionalidade = 'Francesa';
```

d)

```
SELECT COUNT(*)*3 AS total_cobrado
FROM pagamento
WHERE EXTRACT(MONTH FROM dtpagamento) = 11;
```

e)

```
SELECT TitOri FROM filme NATURAL INNER JOIN historial
WHERE DtVisto = '2015-12-02';
```

f)

```
SELECT f.titori AS filmes_vistos_por_hiago93
FROM filme f NATURAL INNER JOIN historial h
WHERE h.login = 'hiago93';
```

g)

```
SELECT firstname, lastname FROM director
EXCEPT
SELECT firstname, lastname FROM director NATURAL INNER JOIN filmeDirectores
NATURAL INNER JOIN etiqFilme
WHERE genero = 'Fantasia';
```

h)

```
WITH numPremiosFilme AS (  
    SELECT f.titori, COUNT(*) AS numPremios  
    FROM filme f NATURAL INNER JOIN premiofilme  
    GROUP BY idfilme  
)
```

```
SELECT *  
FROM numPremiosFilme  
WHERE numPremios = (SELECT MAX(numPremios)  
                    FROM numPremiosFilme);
```

i)

```
WITH numPremiosActores AS (  
    SELECT COUNT(*) AS numPremios  
    FROM premioActor  
    GROUP BY idfilme  
)
```

```
SELECT titori, COUNT(*) AS num_actores_premiados  
FROM premioActor NATURAL INNER JOIN filme  
GROUP BY idFilme, titori  
HAVING COUNT(*) = (SELECT MAX(numPremios)  
                   FROM numPremiosActores);
```

j)

```
SELECT c.firstname, c.lastname, count(idfilme) AS num_filmes_2014  
FROM historial h NATURAL INNER JOIN filme f NATURAL INNER JOIN cliente c  
WHERE EXTRACT(YEAR FROM dtestreia) = '2014'  
GROUP BY h.login, c.firstname, c.lastname  
HAVING count(idfilme) = (SELECT COUNT(idFilme)  
                        FROM filme  
                        WHERE EXTRACT(YEAR FROM dtestreia) = '2014');
```