

```

-- 1. Selecionar filmes que possuam artistas com mais do que 1.9m
    de altura.
-- Exibir título do filme, bem como nome do artista e sua altura.
SELECT filme.titulo, ator.nome, ator.altura FROM filme NATURAL
    JOIN elenco NATURAL JOIN ator WHERE ator.altura > 1.9;

-- 2. Para cada país, exibir quantos artistas já atuaram.
-- 0 mesmo artista não pode ser contabilizado mais do que uma
    vez.
SELECT ator.pais, COUNT(DISTINCT ator.idAtor) FROM ator NATURAL
    JOIN elenco GROUP BY ator.pais;

-- 3. Exibir nomes de artistas que não atuaram em nenhum filme.
SELECT ator.nome FROM ator NATURAL LEFT JOIN elenco WHERE
    elenco.idAtor IS NULL;

-- 4. Selecionar título de filmes que foram sequência de outro
    filme.
SELECT filme.titulo FROM filme WHERE filme.idFilmeAnterior IS NOT
    NULL;

-- 5. Exibir nome de diretores que não dirigiram nada entre 2005
    e 2010.
SELECT diretor.nome FROM diretor LEFT JOIN filme ON
    diretor.idDiretor = filme.idDiretor AND filme.ano BETWEEN
    2005 AND 2010 WHERE filme.idFilme IS NULL;

-- 6. Exibir nome de artistas e a bilheteria somada de todos
    seus filmes.
-- Ordenar pela bilheteria.
SELECT ator.nome, SUM(filme.bilheteria) soma FROM ator NATURAL
    JOIN elenco NATURAL JOIN filme GROUP BY ator.idAtor ORDER
    BY soma;

-- 7. Exibir a maior e a menor bilheteria de cada artista.
-- Apenas para artistas que tenha pelo menos dois filmes.
SELECT ator.nome, MAX(filme.bilheteria), MIN(filme.bilheteria) FROM
    ator NATURAL JOIN elenco NATURAL JOIN filme GROUP BY
    ator.idAtor HAVING COUNT(*) >= 2;

-- 8. Para cada diretor, exibir seu nome, nome do artista com
    quem ele trabalhou e a quantidade de trabalhos
    realizados.
-- Observe que cada diretor pode gerar vários agrupamentos, um
    para cada artista com quem ele trabalhou.

```

```

SELECT diretor.nome, ator.nome, COUNT(*) FROM diretor NATURAL
    JOIN filme NATURAL JOIN elenco JOIN ator ON elenco.idAtor
    = ator.idAtor GROUP BY diretor.idDiretor, ator.idAtor;

-- 9. Exibir título de filmes que tiveram participantes do sexo
    feminino.
SELECT filme.titulo FROM filme NATURAL JOIN elenco JOIN ator ON
    elenco.idAtor = ator.idAtor AND ator.sexo = 'F' GROUP BY
    filme.idFilme;

-- 10. Exibir título de filmes que não tiveram participantes do
    sexo feminino.
SELECT filme.titulo FROM filme NATURAL LEFT JOIN elenco LEFT JOIN
    ator ON elenco.idAtor = ator.idAtor AND ator.sexo = 'F'
    GROUP BY filme.idFilme HAVING COUNT(ator.idAtor) = 0;

-- Os próximos exercícios não serão avaliados, mas podem ser
    entregues junto com os demais.

-- 11. Para cada diretor, exibir seu nome e a quantidade de
    artistas do sexo feminino com os quais ele já trabalhou.
SELECT diretor.nome, COUNT(DISTINCT ator.idAtor) FROM diretor
    NATURAL JOIN filme NATURAL JOIN elenco JOIN ator ON
    elenco.idAtor = ator.idAtor AND ator.sexo = 'F' GROUP BY
    diretor.idDiretor;

-- 12. Para cada diretor, exibir seu nome e a quantidade de
    artistas ingleses com os quais ele já trabalhou.
-- Diretores que não trabalharam com artistas ingleses devem
    receber a contagem zero.
SELECT diretor.nome, COUNT(DISTINCT ator.idAtor) FROM diretor
    NATURAL LEFT JOIN filme NATURAL LEFT JOIN elenco LEFT
    JOIN ator ON elenco.idAtor = ator.idAtor AND ator.pais =
    'Inglaterra' GROUP BY diretor.idDiretor;

-- 13. Exibir título de filmes que tiveram participantes tanto do
    sexo masculino quanto feminino.
SELECT filme.titulo FROM filme NATURAL JOIN elenco LEFT JOIN ator
    ator_masculino ON elenco.idAtor = ator_masculino.idAtor
    AND ator_masculino.sexo = 'M' LEFT JOIN ator
    ator_feminino ON elenco.idAtor = ator_feminino.idAtor AND
    ator_feminino.sexo = 'F' GROUP BY filme.idFilme HAVING
    COUNT(ator_masculino.idAtor) > 0 AND COUNT(ator_feminino.idAtor)
    > 0;

```

```
SELECT filme.titulo FROM filme JOIN elenco e1 ON filme.idFilme =  
    e1.idFilme JOIN ator a1 ON e1.idAtor = a1.idAtor JOIN  
    elenco e2 ON filme.idFilme = e2.idFilme JOIN ator a2 ON  
    e2.idAtor = a2.idAtor WHERE a1.sexo = 'M' AND a2.sexo =  
    'F' GROUP BY filme.idFilme;
```

-- 14. Para cada par de artistas que contracenou junto, mostrar o número de filmes em que ambos apareceram.

```
SELECT a1.nome, a2.nome, COUNT(*) FROM filme JOIN elenco e1 ON  
    filme.idFilme = e1.idFilme JOIN ator a1 ON e1.idAtor =  
    a1.idAtor JOIN elenco e2 ON filme.idFilme = e2.idFilme  
    JOIN ator a2 ON e2.idAtor = a2.idAtor WHERE e1.idAtor <  
    e2.idAtor GROUP BY e1.idAtor, e2.idAtor;
```

-- 15. Mostrar títulos do filmes em que tanto Michael Cane quanto Morgan Freeman contracenaram.

```
SELECT filme.titulo FROM filme JOIN elenco e1 ON filme.idFilme =  
    e1.idFilme JOIN ator a1 ON e1.idAtor = a1.idAtor JOIN  
    elenco e2 ON filme.idFilme = e2.idFilme JOIN ator a2 ON  
    e2.idAtor = a2.idAtor WHERE a1.nome = 'Michael Caine' AND  
    a2.nome = 'Morgan Freeman';
```

-- 16. Retorne quantos filmes estão catalogados considerando a primeira letra do título.

```
SELECT LEFT(filme.titulo, 1) primeira_letra, COUNT(*) FROM filme  
    GROUP BY primeira_letra ORDER BY primeira_letra;
```