|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| |  |  |  |  | | --- | --- | --- | --- | |  | **Atividade Álgebra Relacional** | |  | | **Disciplina:** Banco de Dados II | **Prof. Me. Guiliano Rangel Alves** | |

Discente: Marcos Antônio Barbosa de Souza

Nessa atividade os alunos deveram desenvolver expressões de álgebra relacional para responder as perguntas feitas sobre o Modelo Relacional da Locadora III MR anexo a atividade.

Como foi trabalhado em sala nos exercícios resolvidos, procurem decompor a perguntas em perguntas menores, dessa forma será mais fácil resolver os exercícios propostos abaixo.

Para resolver os exercícios será necessário utilizar, seleção ( μ ), projeção ( π ), união ( U ), interseção ( ∩ ), diferença de conjuntos ( - ), produto cartesiano ( X ) e funções e operações agregadas.

1. Quais são os filmes (título, ano, tipo(nome) que possuem DVD na locadora?

R: π filme.titulo, filme.ano, tipo.nome (μ filme.tipo\_id = tipo.id (filme X tipo ))

2. Quais são os clientes (nsocio, CPF, nome) que já alugaram algum DVD?

R: π locacao.nsocio, cliente.CPF, cliente.nome (μ cliente.nsocio = locacao.cliente\_nsocio (cliente X locacao))

3. Quais são os clientes (nsocio, CPF, nome) que nunca alugaram um DVD?

R: A ← π nsocio, CPF, nome (μ cliente.nsocio = locacao.cliente\_nsocio (cliente X locacao)) // clientes que alugaram

π nsocio, CPF, nome (cliente - A)

4. Listar os clientes (nsocio, nome) e seus telefones.

R: π cliente.nsocio, cliente.nome, telefone\_cliente.ddd, telefone\_cliente.numero (μ cliente.nsocio = telefone\_cliente.cliente\_nsocio (cliente X telefone\_cliente ))

5. Listar os atores (nome\_real, nome\_de\_ator, data\_de\_nascimento) que já ganharam um óscar.

R: π nome\_real, nome\_de\_ator, data\_de\_nascimento (μ ator.id = atuacao.ator\_id AND atuacao.oscar = TRUE (ator X atuacao ))

6. Listar os filmes (título, ano) juntamente com os atores (nome\_de\_ator) que atuaram no filme (independe de ter ganhado oscar).

R: A ← π titulo, ano, nome\_de\_ator, filme\_id (μ ator.id = atuacao.ator\_id (ator X atuacao ))

π titulo, ano, nome\_de\_ator (μ A.filme\_id = filme.id (A X filme))

7. Quantos clientes a locadora possui?

R: A ← π CPF (cliente )

G sum(A.CPF)

8. Quantos DVDs a locadora possui?

R: G sum(dvd.codigo)

9. Totalizar o número de filmes da locadora por ano (ano, total\_de\_filmes).

R: A ← π ano, id (μ dvd.codigo = filme.id (dvd X filme))  
ano G count (A.id)

10. Quantos filmes cada ator já participou (id, nome\_real,total\_de\_filmes).

R: A ← π ano, ator\_id (μ id = ator\_id (ator X atuacao))  
ano G count(A.ator\_id)

11. Listar o valor total por locação de todas as locações (nome (do cliente), nsocio, data\_hora(locação), valor\_total).

R: A ← π nome, nsocio, data\_hora (μ locacao.cliente\_nsocio = cliente.nsocio (locacao X cliente)) // clientes que locaram  
π nome, nsocio, data\_hora, valor (μ A.nsocio = possui. locacao\_cliente\_nsocio (A X possui))

12. Quantos filmes tem em cada categoria (id, nome, descrição, total\_de\_filmes)?

R: A ← π categoria\_id, filme\_id (μ filme.id = pertence.filme\_id (filme X pertence))  
B ← π categoria\_id, nome, descricao, filme\_id (μ categoria.id = A.categoria\_id (categoria X A))  
categoria\_id, nome, descricao G count(B.filme\_id)

13. Quais DVDs (id, título, ano) nunca foram alugados e nem reservados?

R: A ←π codigo, titulo, ano, filme\_id (μ filme.id = dvd.filme\_id (filme X dvd)) // todos dvd(s)  
B ← π codigo, titulo, ano, filme\_id (μ A.codigo = possui.dvd\_codigo (A X possui)) // dvd(s) locados

C ← π codigo, titulo, ano, filme\_id (μ A.filme\_id = consta.filme\_id (A X consta)) // dvd(s) reservados

π codigo, titulo, ano (A - (B U C))

14. Quais DVDs (id, título, ano) já foram alugados e reservados?

R: A ←π codigo, titulo, ano, filme\_id (μ filme.id = dvd.filme\_id (filme X dvd)) // todos dvd(s)

B ← π codigo, titulo, ano, filme\_id, possui.sequencia (μ A.codigo = possui.dvd\_codigo (A X possui)) // dvd(s) locados

C ← π codigo, titulo, ano, filme\_id, consta.sequencia (μ A.filme\_id = consta.filme\_id (A X consta)) // dvd(s) reservados

π codigo, titulo, ano (μ B.filme\_id = C.filme\_id (B X C))

15. Quais DVDs (id, título, ano) já foram reservados mas nunca foram alugados?

R: A ← π id, titulo, ano (μ filme.id = dvd.filme\_id (filme X dvd))

B ← π id, titulo, ano (μ A.dvd.codigo = possui.dvd\_codigo (A X possui)) // alugados

C ← μ filme.id = consta.filme\_id (filme X consta)

D ← π id, titulo, ano (μ C. filme\_id = reserva.cliente\_nsocio (C X reserva)) // reservados

π id, titulo, ano (D - B)

16. Qual o(s) ator(es) (id, nome\_real) que nunca ganhou um óscar?

R: A ← π filme\_id, ator\_id, oscar (μ filme.id = atuacao.filme\_id (filme X atuacao))

π ator\_id, nome\_real (μ A.ator\_id = ator.id AND oscar = FALSE (A X ator))

17. Qual o(s) DVD(s) (id, título, ano) mais alugados da locadora?

R: A ←π codigo, titulo, ano, filme\_id (μ filme.id = dvd.filme\_id (filme X dvd)) // todos dvd(s)

B ← π codigo, titulo, ano, sequencia (μ A.dvd.codigo = possui.dvd\_codigo (A X possui)) // alugados

codigo, titulo, ano G max(sequencia)(B)

18. Qual o(s) DVD(s) (id, título, ano) menos alugado da locadora?

R: A ←π codigo, titulo, ano (μ filme.id = dvd.filme\_id (filme X dvd)) // todos dvd(s)

B ← π codigo, titulo, ano, sequencia (μ A.dvd.codigo = possui.dvd\_codigo (A X possui)) // alugados

codigo, titulo, ano G min(sequencia)(B)

19. Qual o(s) tipo(s) (id,nome,valor) de filme mais caro?

R: A ←π codigo, titulo, ano, filme\_id (μ filme.id = dvd.filme\_id (filme X dvd)) // todos dvd(s)

B ← π id, titulo, ano, valor (μ A.dvd.codigo = possui.dvd\_codigo (A X possui)) // alugados

id, titulo, valor G max(valor)(B)

20. Listar os dados das locações (nome(cliente), data\_hora(locação), sequencia(possui), data\_devolucao(possui), valor(possui), título(filme), ano(filme) e nome(tipo).

R: A ← π nome, locacao\_data\_hora, sequencia, data\_devolucao, valor, dvd\_codigo (μ possui.locacao\_cliente\_nsocio = cliente.nsocio (possui X cliente)) // locações por cliente

B ← π nome, locacao\_data\_hora, sequencia, data\_devolucao, valor, filme\_id (μ A.dvd\_codigo = dvd.codigo (A X dvd)) // Locação por DVD(s)

C ← π nome, locacao\_data\_hora, sequencia, data\_devolucao, valor, titulo, ano, tipo\_id (μ B.filme\_id = filme.id (B X filme)) // DVD(s) por Filme

π nome, locacao\_data\_hora, sequencia, data\_devolucao, valor, titulo, ano, nome (μ C.tipo\_id = tipo.id (C X tipo)) // Filme por Tipo

21. Qual o(s) filme(s) (id, título, ano) foi mais alugado?

R: A ←π filme\_id, titulo, ano, codigo (μ filme.id = dvd.filme\_id (filme X dvd)) // todos dvd(s)

B ← π filme\_id, titulo, ano, sequencia (μ A.dvd.codigo = possui.dvd\_codigo (A X possui)) // alugados

filme\_id, titulo, ano G max(sequencia)(B)

22. Qual o(s) filme(s) (id, título, ano) mais lucrativo para a locadora (verificar o filme que somando as locações tem o maior valor)?

R: A ←π filme\_id, titulo, ano, codigo (μ filme.id = dvd.filme\_id (filme X dvd)) // todos dvd(s)

B ← π filme\_id, titulo, ano, valor, locacao\_cliente\_nsocio, locacao\_data\_hora (μ A.dvd.codigo = possui.dvd\_codigo (A X possui)) // alugados

C ← filme\_id, titulo, ano G sum(valor) as total (B) // nº total de locações

D ← G max (total) as maior ( C ) // maior locação

π filme\_id, titulo, ano (μ D.total = C.maior (D X C)) // o mais lucrativo