

Sobre comandos SQL para modificar a estrutura de uma tabela, considere as seguintes afirmações:

I. O comando UPDATE é usado para modificar os dados existentes em uma tabela.

II. O comando ALTER é usado para modificar a estrutura de uma tabela.

III. O comando MODIFY é usado para alterar os valores dos dados registrados em uma coluna específica.

Está(ão) correta(s) apenas a(s) afirmação(ões):

- ☐ I, II e III
- ☐ II
- ☒ I e II
- ☐ II e III
- ☐ I e III

Sobre comandos SQL para modificar...

Considere a seguinte relação:

ADDRESS			
ADDRESS_ID	NUMBER(22)	PK	
ADDRESS	VARCHAR2(50)		
ADDRESS2	VARCHAR2(50)	N	
DISTRICT	VARCHAR2(20)		
CITY_ID	NUMBER(22)	FK	
POSTAL_CODE	VARCHAR2(10)	N	
PHONE	VARCHAR2(20)		
LAST_UPDATE	TIMESTAMP		

Assinale a alternativa correta com relação ao registro de telefones (PHONE) nessa relação:

- ☐ O registro está correto e permite o registro de múltiplos telefones para um endereço no campo PHONE da relação ADDRESS
- ☒ Para permitir o registro de múltiplos telefones, deveremos excluir o campo PHONE da tabela ADDRESS e adicionar uma tabela PHONE com uma chave estrangeira ADDRESS_ID com o id do endereço que é relacionado ao telefone
- ☐ Para que o registro fique correto basta alterar para que permita o registro de múltiplos telefones para um endereço no campo PHONE da relação ADDRESS
- ☐ O registro não está correto porque permite o registro de múltiplos telefones para um endereço no campo PHONE da relação ADDRESS
- ☐ Para que o registro fique correto basta alterar para que não permita o registro de múltiplos telefones para um endereço no campo PHONE da relação ADDRESS
- ☐ Para permitir o registro de múltiplos telefones, deveremos excluir o campo PHONE da tabela ADDRESS e adicionar o CAMPO PHONE_ID como chave estrangeira com o ID de uma nova tabela PHONE

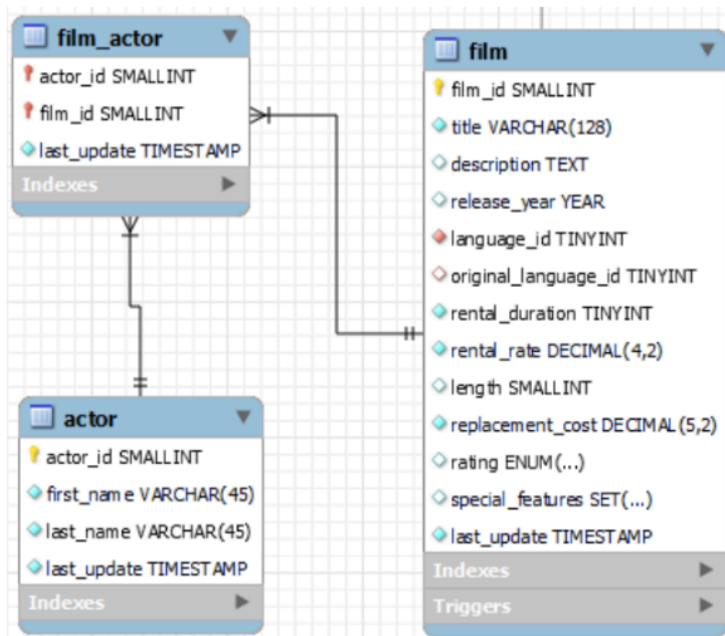
Considere a seguinte relação: (...) Assinale a alternativa correta com relação (...)

São expressões válidas da álgebra relacional (assinale todas as que forem válidas):

- ☒ $\pi \text{ nome, salario } (\sigma \text{ idade} > 40 \text{ (Empregado)})$
- ☐ $\sigma \text{ cpf} = 099.099.099-10 \text{ (} \pi \text{ nome, salario (Empregado))}$
- ☒ $\pi \text{ cpf } (\pi \text{ nome, cpf (Empregado))}$
- ☐ $\pi \text{ dt_nascimento, nome } (\pi \text{ data_nascimento (Cliente))}$
- ☒ $\pi \text{ nome, salario } (\sigma \text{ salario} > 10000 \text{ (Empregado)})$

São expressões válidas da álgebra relacional (assinale todas as que forem válidas):

Considere as seguintes tabelas do Banco de Dados Sakila:



Quais das seguintes instruções SQL apresenta instruções válidas para relacionar os filmes e os atores que fazem parte do elenco:

- I. SELECT title, first_name, last_name FROM film JOIN film_actor ON film.film_id = film_actor.film_id JOIN actor ON film_actor.actor_id = actor.actor_id;
- II. SELECT title, first_name, last_name FROM film_actor JOIN film ON film_actor.film_id = film.film_id JOIN actor ON film_actor.actor_id = actor.actor_id;
- III. SELECT title, first_name, last_name FROM actor JOIN film_actor ON actor.actor_id = film_actor.actor_id JOIN film ON film_actor.film_id = film.film_id;
- IV. SELECT title, first_name, last_name FROM film JOIN actor ON actor.actor_id = film.actor_id;

- ☐ III
- ☐ I e IV
- ☒ I, II e III
- ☐ II, III e IV
- ☐ II e III

Quais das seguintes instruções válidas para relacionar os filmes e os atores que fazem (...)

Considere as seguintes afirmações sobre restrições de integridade em bancos de dados relacionais utilizando o MySQL:

- I. A restrição PRIMARY KEY garante que uma coluna ou conjunto de colunas tenha valores únicos e não nulos.
- II. A restrição FOREIGN KEY garante que um valor em uma coluna de uma tabela referencie um valor válido em uma coluna correspondente de outra tabela.
- III. A restrição UNIQUE garante que todos os valores em uma coluna sejam únicos e não permite valores nulos.
- IV. A restrição CHECK permite definir condições específicas que os valores de uma coluna devem satisfazer.

Está(ão) correta(s) apenas a(s) afirmação(ões):

- ☐ I e II
- ☐ I, II e III
- ☒ I, II e IV
- ☐ II e III
- ☐ III

Considere as seguintes afirmações sobre restrições de integridade em bancos de dados (...)

Indique qual função pode ser utilizada para responder a qual tipo de consulta a um banco de dados relacional:

count	Quantidade de ocorrências de coincidências de um padrão
min	Indicação do cliente mais velho considerando sua data de nas
avg	Média de notas de um aluno
max	Indicação do pedido de maior valor feito para um fornecedor
sum	Total de notas emitidas para um cliente

Indique qual função pode ser utilizada para responder a qual tipo de consulta a um (...)

O seguinte modelo foi desenvolvido para uma loja para registro de itens dos pedidos:

IdItemPedido (PK)	IdPedido (FK)	IdProduto (FK)	Qtde	PrecoItem	TotalItem
-------------------	---------------	----------------	------	-----------	-----------

Considerando as especificações da 1ªFN, 2ªFN e 3ªFN é possível afirmar:

- I. Que a tabela está de acordo com a 1ªFN
- II. Que a tabela está de acordo com a 2ªFN
- III. Que a tabela está de acordo com a 3ªFN
- IV. Que a tabela não atende nem a 1ªFN, nem a 2ªFN e nem 3ªFN

Indique a alternativa que apresenta a(s) afirmação(ões) correta(s):

- ☐ I, II e III
- ☐ II e III
- ☒ I e II
- ☐ I e III
- ☐ IV

O seguinte modelo foi desenvolvido para uma loja para registro de itens dos pedidos:

Sobre as formas normais (NF) em um banco de dados relacional, considere as seguintes afirmações:

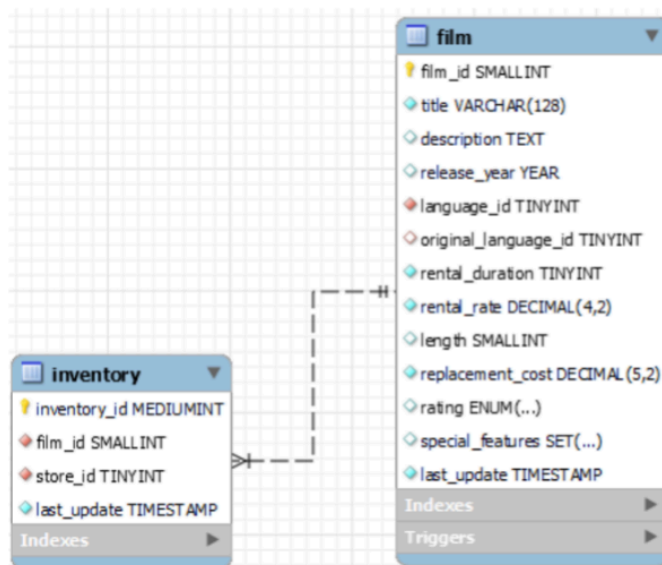
- I. Uma tabela está na 1ª Forma Normal (1NF) quando todos os atributos contêm valores atômicos.
- II. Uma tabela está na 2ª Forma Normal (2NF) quando não possui dependências parciais.
- III. Uma tabela está na 3ª Forma Normal (3NF) quando todos os atributos não chave dependem da chave primária inteira e não dependem transitivamente de outros atributos não chave.

Está(ão) correta(s) apenas a(s) afirmação(ões):

- ☐ III
- ☐ I e II
- ☐ II e III
- ☐ II
- ☒ I, II e III

Sobre as formas normais (NF) em um banco de dados relacional, considere as seguintes

Utilizando as relações indicadas a seguir:



Qual(is) a(s) instrução(ões) adequada(s) para encontrar filmes sem estoque?

- ☒ SELECT film.film_id, title, inventory_id FROM film LEFT JOIN inventory ON inventory.film_id = film.film_id WHERE inventory.film_id IS NULL ORDER BY title
- ☒ SELECT film.film_id, title, inventory_id FROM inventory RIGHT JOIN film ON inventory.film_id = film.film_id WHERE inventory.film_id IS NULL ORDER BY title
- ☐ SELECT film.film_id, title, inventory_id FROM inventory RIGHT JOIN film ON inventory.film_id = film.film_id WHERE film.film_id IS NULL ORDER BY title
- ☐ SELECT film.film_id, title, inventory_id FROM film LEFT JOIN inventory ON inventory.film_id = film.film_id WHERE film.film_id IS NULL ORDER BY title

Utilizando as relações indicadas a seguir: Qual(is) a(s) instrução(ões) adequada(s)

Um dos principais objetivos da modelagem conceitual é evitar redundância e inconsistência no banco de dados. Considerando os princípios da modelagem conceitual, avalie as seguintes afirmativas:

- I. **Normalização:** A normalização é um processo de dividir as relações em várias menores, de modo a eliminar redundância e anomalias de atualização.
- II. **Chave Primária:** A chave primária é um atributo ou conjunto de atributos que identifica unicamente cada tupla em uma relação.
- III. **Chave Estrangeira:** A chave estrangeira é um atributo ou conjunto de atributos que referencia a chave primária de outra relação.
- IV. **Integridade Referencial:** A integridade referencial garante que cada valor de chave estrangeira esteja presente na chave primária da relação referenciada.

Com base na análise das afirmativas, assinale a alternativa que apresenta os princípios que garantem maior **qualidade** para a modelagem conceitual:

- ☐ II, III e IV, apenas.
- ☐ I, II e III, apenas.
- ☐ I, II e IV, apenas.
- ☒ I, II, III e IV
- ☐ Nenhuma das afirmativas garante a qualidade da modelagem conceitual.

Um dos principais objetivos da modelagem conceitual é evitar redundância e (...)

Assinale todas as expressões válidas segundo a Álgebra Relacional:

- ☐ π dt_nascimento, nome (π nome (Cliente))
- ☒ π nome, cpf (σ idade > 40 (Empregado))
- ☒ σ cpf = 099.099.099-10 (π nome, cpf (Empregado))
- ☐ σ data_nascimento > 01.01.2001 (π nome, cpf (Empregado))
- ☒ σ data_nascimento > 01.01.2001 (π nome, data_nascimento (Empregado))
- ☐ π data_nascimento (π nome, cpf (Empregado))

Assinale todas as expressões válidas segundo a Álgebra Relacional:

Considere as seguintes afirmações sobre comandos SQL:

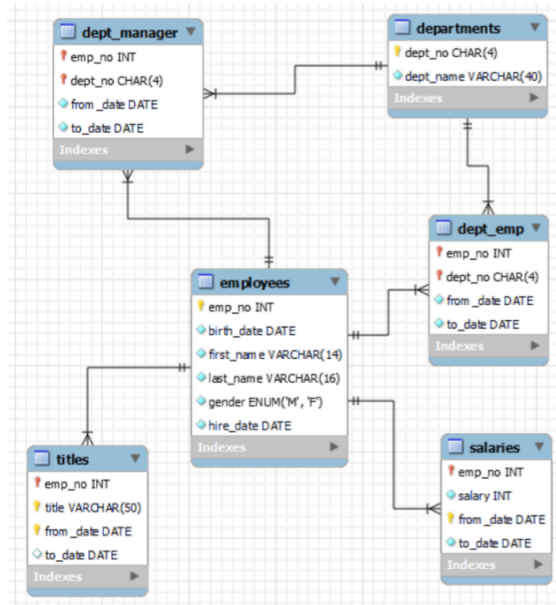
- I. O comando DELETE remove todos os registros de uma tabela, mas mantém a estrutura da tabela.
- II. O comando DROP remove todos os registros e a estrutura da tabela do banco de dados.
- III. O comando TRUNCATE remove todos os registros de uma tabela, mas mantém a estrutura da tabela.

Está(ão) correta(s) apenas a(s) afirmação(ões):

- ☐ III
- ☐ II
- ☐ I
- ☐ I e III
- ☒ I, II e III

Considere as seguintes afirmações sobre comandos SQL:

No seguinte modelo de dados, os salários são registrados anualmente, desde a contratação ou promoção:



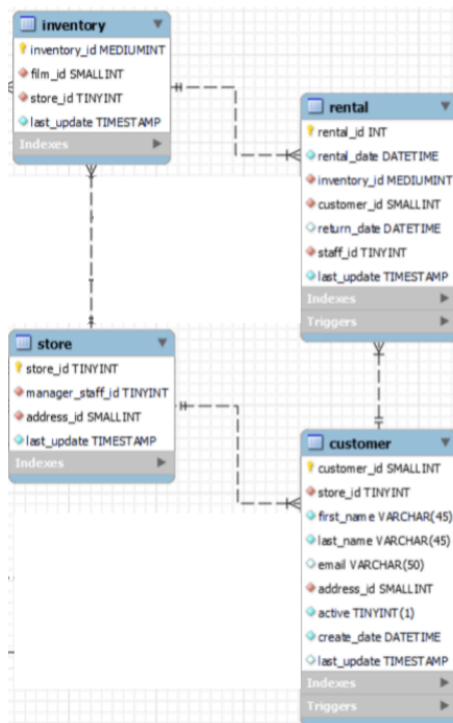
Considerando esse modelo, qual a instrução SQL adequada para:

Identificar os salários pagos por departamento no ano de 2023.

- ☐ select d.dept_name, s.salary
from salaries s
INNER JOIN employees e ON s.emp_no = e.emp_no
INNER JOIN dept_emp de ON de.emp_no = e.emp_no
INNER JOIN departments d ON d.dept_no = de.dept_no
WHERE s.from_date > "2023-01-01" and s.to_date < "2023-12-31"
having d.dept_name;
- ☐ select d.dept_name, s.salary
from salaries s
INNER JOIN employees e ON s.emp_no = e.emp_no
INNER JOIN dept_emp de ON de.emp_no = e.emp_no
INNER JOIN departments d ON d.dept_no = de.dept_no
WHERE s.from_date > "2023-01-01" and s.to_date < "2023-12-31"
group by d.dept_name;
- ☒ select d.dept_name, sum(s.salary)
from salaries s
INNER JOIN employees e ON s.emp_no = e.emp_no
INNER JOIN dept_emp de ON de.emp_no = e.emp_no
INNER JOIN departments d ON d.dept_no = de.dept_no
WHERE s.from_date > "2023-01-01" and s.to_date < "2023-12-31";
- Resposta correta:
- ☐ select d.dept_name, sum(s.salary)
from salaries s
INNER JOIN employees e ON s.emp_no = e.emp_no
INNER JOIN dept_emp de ON de.emp_no = e.emp_no
INNER JOIN departments d ON d.dept_no = de.dept_no
WHERE s.from_date > "2023-01-01" and s.to_date < "2023-12-31"
group by d.dept_name;

No seguinte modelo de dados, os salários são registrados anualmente, desde a (...)

Considere o seguinte conjunto de tabelas:



Qual o comando SQL para identificar quais os ids dos filmes (film_id) que foram alugados por um cliente com email joao@email.com.br na loja em que o gerente possui manager_staff_id 2?

- ☒

```
SELECT i.film_id
FROM inventory i
JOIN rental r ON i.inventory_id = r.inventory_id
JOIN customer c ON r.customer_id = c.customer_id
JOIN store s ON i.store_id = s.store_id
WHERE c.email = 'joao@email.com.br'
AND s.manager_staff_id = 2;
```
- ☐

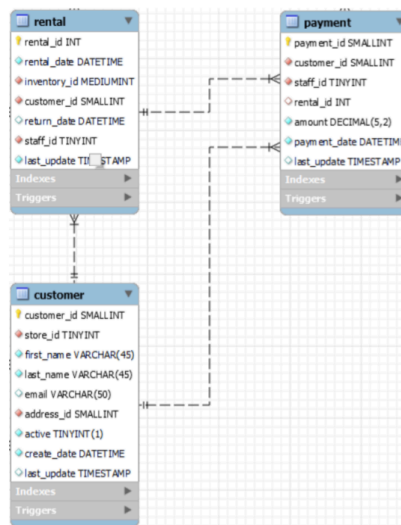
```
SELECT r.inventory_id
FROM inventory i
JOIN rental r ON i.inventory_id = r.inventory_id
JOIN customer c ON r.customer_id = c.customer_id
JOIN store s ON r.store_id = s.store_id
WHERE c.email = 'joao@email.com.br'
AND s.manager_staff_id = 2;
```
- ☐

```
SELECT r.inventory_id
FROM rental r
JOIN inventory i ON r.inventory_id = i.inventory_id
JOIN customer c ON r.customer_id = c.customer_id
JOIN store s ON i.store_id = s.store_id
WHERE c.email = 'joao@email.com.br'
AND s.manager_staff_id = 2;
```
- ☐

```
SELECT i.film_id
FROM inventory i
JOIN rental r ON i.inventory_id = r.inventory_id
JOIN customer c ON r.customer_id = c.customer_id
JOIN store s ON c.store_id = s.store_id
WHERE c.email = 'joao@email.com.br'
AND s.manager_staff_id = 2;
```

Qual o comando SQL para identificar quais os ids dos filmes (film_id) que foram (...)

Considerando as relações a seguir:



Coloque os componentes da instrução em ordem de forma a que a instrução SQL possa ser adequadamente executada para obter o valor médio das locações cuja soma total seja acima de 100.00 pagas por usuário em ordem decrescente de valor médio:

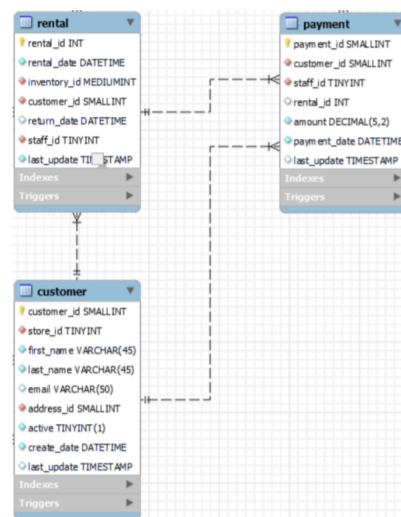
select c.first_name, c.last_name, avg(p.amount) as valor from
 customer c INNER JOIN rental r USING(customer_id)

INNER JOIN payment p USING(rental_id)
 GROUP BY c.first_name, c.last_name
 HAVING sum(p.amount) > 100.00
 ORDER by valor

desc

Coloque os componentes da instrução em ordem de forma a que a instrução

Considerando as relações a seguir:



Coloque os componentes da instrução em ordem de forma

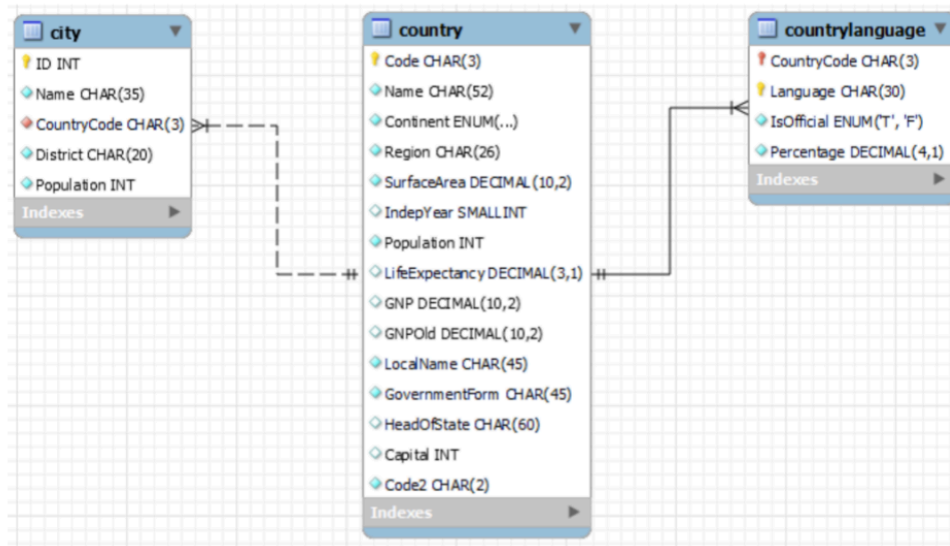
Coloque os componentes da instrução em ordem de forma a que a instrução SQL possa ser adequadamente executada para obter o total de locações acima de 100.00 pagas por usuário em ordem decrescente do valor total de locações:

select c.first_name, c.last_name, sum(p.amount) as valor from
 customer c INNER JOIN rental r USING(customer_id)

INNER JOIN payment p USING(rental_id)
 GROUP BY c.first_name, c.last_name
 HAVING sum(p.amount) > 100.00
 ORDER by valor

desc

O modelo a seguir tem informações sobre países, como expectativa de vida (LifeExpectancy), População total (Population), Área (SurfaceArea) e produto nacional bruto (GNP).



Qual das alternativas apresenta a instrução SQL correta para apresentar o nome do Produto Nacional Bruto do país que tem o maior Produto Nacional Bruto?

- ☐ SELECT ci.Name, MAX(ci.Population), co.Name, cl.Language, co.GNP
FROM
city ci
INNER JOIN country co ON co.code = ci.CountryCode
INNER JOIN countryLanguage cl ON co.Code = cl.CountryCode
WHERE co.GNP = (SELECT MAX(GNP) FROM country)
AND cl.IsOfficial = "T"
- ☐ SELECT ci.Name, ci.Population, co.Name, cl.Language, co.GNP
FROM
city ci
INNER JOIN country co ON co.code = ci.CountryCode
INNER JOIN countryLanguage cl ON co.Code = cl.CountryCode
WHERE co.GNP = (select max(GNP) from country)
AND cl.IsOfficial = "T"
ORDER BY ci.Population DESC
- ☒ SELECT ci.Name, ci.Population, co.Name, cl.Language, co.GNP
FROM
city ci
INNER JOIN country co ON co.code = ci.CountryCode
INNER JOIN countryLanguage cl ON co.Code = cl.CountryCode
WHERE co.GNP = (select max(GNP) from country)
AND cl.IsOfficial = "T"
ORDER BY ci.Population DESC
LIMIT 1
- ☐ SELECT ci.Name, ci.Population, co.Name, cl.Language, max(co.GNP)
FROM
city ci
INNER JOIN country co ON co.code = ci.CountryCode
INNER JOIN countryLanguage cl ON co.Code = cl.CountryCode
WHERE cl.IsOfficial = "T"
ORDER BY ci.Population DESC
LIMIT 1

O modelo a seguir tem informações sobre países, como expectativa de vida

Considere a seguinte relação:

IdCliente	NomeCompleto	CPF	DataCadastro	Telefones
11	Ana Lima	033.033.330-12	12/01/2023	11-90909-0909, 11-98980-9898

Considerando essa relação, é possível afirmar:

- ☐ Ela está de acordo com a 2FN (atende a 2ª forma normal)
- ☐ Para que seja normalizada, sem perda de informação, basta excluir a tabela telefone.
- ☐ Ela está de acordo com a 3FN (atende a 3ª forma normal)
- ☐ Ela está de acordo com a 1FN (atende a 1ª forma normal)
- ☒ Para que seja normalizada, sem perda de informação, deve ser criada uma tabela telefones, com chave estrangeira idCliente e com a inclusão de um registro para cada telefone.

Considere a seguinte relação: Considerando essa relação, é possível afirmar:

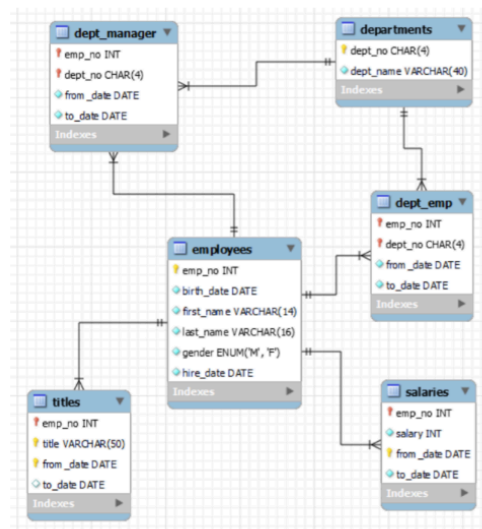
Considere a seguinte relação:

idCliente	NomeCompleto	CPF	DataCadastro	Documentos
12	Carla Silva	022.022.022-11	12/01/2023	12.090.901-1, OAB-560.989-3

- ☒ Para que seja normalizada, sem perda de informação, deve ser criada uma tabela documentos, com chave estrangeira idCliente e com a inclusão de um registro para tipo de documento e número de documento.
- ☐ Ela está de acordo com a 1FN (atende a 1ª forma normal)
- ☐ Ela está de acordo com a 3FN (atende a 3ª forma normal)
- ☐ Ela está de acordo com a 2FN (atende a 2ª forma normal)
- ☐ Para que seja normalizada, sem perda de informação, basta excluir a coluna documento.

Considere a seguinte relação: Para que seja normalizada, sem perda de informação

Observe o modelo:



Avalie de acordo com o modelo qual(is) das seguintes instruções pode(m) ser utilizada(s) para identificar a(s) contratação(ões) mais recente(s):

- I. SELECT first_name, last_name, hire_date FROM employees WHERE hire_date = (SELECT min(hire_date) FROM employees)
- II. SELECT first_name, last_name FROM employees ORDER BY hire_date desc LIMIT 1
- III. SELECT first_name, last_name FROM employees WHERE hire_date = (SELECT hire_date from employees order by hire_date desc LIMIT 1)

☐ I e III

☒ II e III

Resposta correta: III

☐ II

☐ I

☐ III

Observe o modelo: Avalie de acordo com o modelo qual(is) das seguintes

Resumo Geral: Modelagem de Dados, Normalização e Consultas SQL

1. Normalização de Dados

- 1ª Forma Normal (1FN):

- Requer que todos os atributos sejam atômicos (sem múltiplos valores em um único campo).
- Exemplo: Campos como `Telefones` ou `Documentos` com valores separados por vírgula violam a 1FN.
- Solução: Criar tabelas separadas (ex: `Telefones`, `Documentos`) com chaves estrangeiras.

- 2ª Forma Normal (2FN):

- Exige que a tabela esteja na 1FN e que todos os atributos não-chave dependam totalmente da chave primária.
- Elimina dependências parciais (ex: atributos que dependem apenas de parte de uma chave composta).

- 3ª Forma Normal (3FN):

- Exige que a tabela esteja na 2FN e que não haja dependências transitivas (atributos não-chave não podem depender de outros atributos não-chave).

2. Comandos SQL para Modificação de Estrutura

- DELETE: Remove registros, mas mantém a estrutura da tabela. Não reinicia contadores de auto-incremento.
- TRUNCATE: Remove todos os registros rapidamente, mantém a estrutura e reinicia contadores.
- DROP: Remove a tabela completamente (dados + estrutura).

3. Álgebra Relacional

- Operações válidas:

- Projeção (π): Seleciona colunas específicas.
- Seleção (σ): Filtra linhas com base em condições.
- Regra: Não se pode projetar ou filtrar por atributos que não estão na relação atual.

4. Consultas SQL Avançadas

- Exemplo 1: Encontrar filmes sem estoque:

```
SELECT film_id FROM film  
LEFT JOIN inventory ON film.film_id = inventory.film_id  
WHERE inventory.film_id IS NULL;
```

- Exemplo 2: Calcular médias com agregação:

```
SELECT departamento, AVG(salario)  
FROM funcionarios  
GROUP BY departamento  
HAVING SUM(salario) > 10000;
```

- Exemplo 3: Contratações mais recentes:

```
SELECT first_name, last_name  
FROM employees  
WHERE hire_date = (SELECT MAX(hire_date) FROM employees);
```

5. Chaves e Integridade Referencial

- Chave Primária (`PRIMARY KEY`): Identifica registros unicamente (não nula e única).
 - Chave Estrangeira (`FOREIGN KEY`): Garante que um valor referencie uma chave primária em outra tabela.
 - Integridade Referencial: Assegura que chaves estrangeiras sempre apontem para valores válidos.
-

6. Resumo das Melhores Práticas

1. Normalize tabelas até a 3FN para evitar redundâncias.
 2. Use JOIN para relacionar tabelas e WHERE/HAVING para filtrar.
 3. Para consultas complexas, combine subconsultas com operações de agregação (`SUM`, `AVG`, `MAX`).
 4. Evite campos calculados em tabelas normalizadas (ex: `TotalItem`). Calcule-os em consultas.
 5. Prefira TRUNCATE para limpar tabelas inteiras (é mais eficiente que `DELETE`).
-

Exemplo Prático de Normalização

Antes (Violando 1FN):

idCliente	Documentos
-----	-----
12	RG:123, OAB:456

Depois (Normalizado):

- Tabela `Clientes`:

idCliente	Nome
-----	-----
12	Carla Silva

- Tabela `Documentos`:

idDocumento	idCliente	Tipo	Número
-----	-----	-----	-----
1	12	RG	123
2	12	OAB	456