



Universidade de São Paulo
Instituto de Ciências Matemáticas e de Computação
Departamento de Ciências de Computação

SCC-0240 – Bases de Dados — Prof. Dr. Caetano Traina Júnior
Primeira Prova – 7 de maio de 2024 – Turma 1

Nome: _____

NUSP: _____

1 Descrição do problema

Uma empresa provê um serviço online para que pessoas anunciem, pesquisem e reservem propriedades para acomodação e meios de hospedagem em localidades de todo o mundo. Ela permite que indivíduos possam alugar no todo ou em parte seus imóveis, incluindo sua própria casa, como uma forma de rendimento extra. A empresa fornece uma plataforma de busca e reservas entre a pessoa que oferece a acomodação e o viajante que procura um local para acomodação em locação. Este trabalho visa criar uma base de dados para armazenar as informações envolvidas nas atividades da empresa.

Dentre essas informações está a descrição das propriedades, que deve estar identificada por seu nome e endereço. Cada propriedade é classificada como sendo uma casa inteira ou um quarto, que pode ser individual ou compartilhado, e sua descrição deve incluir o preço da diária. Cada propriedade pode ter diversas comodidades dentre um conjunto pré-definido (como WiFi, ar-condicionado, cafeteira, etc.), cada um com quantidades variáveis.

A empresa registra o nome, data de nascimento, endereço e email de cada usuário do serviço, e não pode existir dois usuários com o mesmo nome e email. Usuários podem tanto colocar propriedades para locação quanto fazer a locação de propriedades. A disponibilidade de uma propriedade é dada pelo seu proprietário, indicando um período contínuo de disponibilidade pela data inicial e final.

Cada locação tem um código de registro, e se refere somente a uma propriedade e a um locatário, sempre indicando o período de locação pela data de check-in e de check-out. O período de locação deve sempre estar compreendido no período indicado como disponível por seu proprietário, e uma mesma propriedade não pode ser locada por períodos que se sobrepõem. O proprietário possa alterar o valor da diária de suas propriedades sempre que desejar, e quando uma locação é fechada o valor da locação é calculado pelo valor nesse momento.

Cada usuário que coloca uma propriedade para locação tem uma conta bancária onde será depositado o valor da estadia do locatário pelo período da locação, contendo um número de conta e o tipo da conta. Uma conta bancária só pode pertencer a um usuário, mas o usuário pode ser detentor de diversas contas. Por motivos de segurança, a empresa mantém apenas as contas dos usuários que tenham imóveis para locação.

entidade fraca

2 Questões

Sugestão: Use a sintaxe:

```
EXTRACT(<OQue> FROM <Data>)
```

para extrair a componente (DAY ou MONTH ou YEAR) da Data indicada.

Exercício 1:

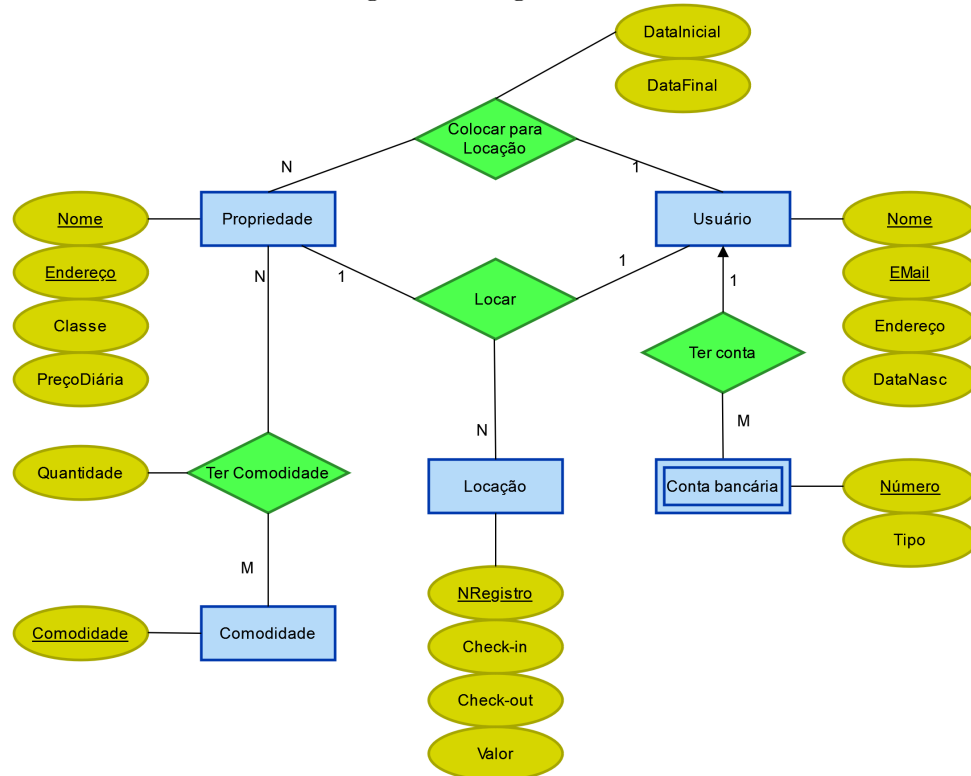
Faça a modelagem E-R da base de dados descrita, usando apenas os conceitos do Modelo Entidade-Relacionamento Básico.

O diagrama E-R está mostrado na figura 1

Destaques:

- Conjunto de Entidades Comodidade: Não é comum ter um CE tendo como atributos apenas sua chave, mas não é proibido. Essa situação é comum quando existe um conjunto de valores

Figura 1: Diagrama E-R



pré-definido mas relativamente grande, e que pode variar com o tempo, tal como é o caso.

- Atributo **Classe**: Esse é um caso semelhante à **Comodidade**, mas nesse caso o conjunto é fixo e pequeno.
- Como o CR **Coloca para Locação** tem cardinalidade 1:N, seus atributos podem ser repassados para o CE que assume a multiplicidade **N**. Mas deixei assim porque semanticamente é mais claro que essas datas se referem ao período de locação, e não propriamente à propriedade.
- o CR **Loca** e o CE **Locação** se referem a um mesmo conceito, que não é mapeado de maneira “cômoda” pelo ME-R básico. Assim, usar um CR ternário é uma solução de compromisso para este caso.
- Não está indicado no diagrama que os períodos de locação não podem se sobrepor, nem que precisam estar contidos no período de disponibilidade da propriedade. Isso deve ser tratado por regras escritas nas aplicações ou como restrições de integridade definidas pelo usuário, mas o diagrama contempla os atributos necessários para que essas regras sejam avaliadas. O mesmo ocorre para o cálculo do **valor** de uma **Locação**.

Exercício 2:

Faça o Mapeamento ME-R→MRel da modelagem obtida no Exercício 1, incluindo as restrições de integridade pertinentes.

Passo 1: Mapear os CE Regulares

Propriedade={Nome, Endereço, Classe, PreçoDiária}
 Usuário={Nome, EMail, Endereço, DataNasc}
 Locação={NRegistro, CheckIn, CheckOut, Valor}
 Comodidade={Comodidade}

Passo 2: Mapear os CE Fracos

ContaBancaria={Número, Tipo, Nome, EMail}

Onde: (Nome, EMail) \xrightarrow{CE} Usuário(Nome, EMail)

Passo 3: Mapear os CR de cardinalidade 1:1

Não tem nenhum.

Passo 4: Mapear os CR de cardinalidade 1:N

Ter conta é o relacionamento total que torna fraco o CE Conta bancária, então já está mapeado.

Colocar para locação deve ser acrescentado ao CE que participa com multiplicidade N:

Propriedade={Nome, Endereço, Classe, PreçoDiária,
NomeUsuário, EMail, DataInicial, DataFinal}

Onde (NomeUsuário, EMail) \xrightarrow{CE} Usuário(Nome, EMail)

Passo 5: Mapear os CR de cardinalidade M:N

TerComodidade={NomeProp, EnderProp, Comodidade, Quantidade}

Onde: (Comodidade) \xrightarrow{CE} (Comodidade(Comodidade))

(NomeProp, EnderProp) \xrightarrow{CE} Propriedade(Nome, Endereço),

Passo 6: Mapear os CR de ordem > 2

Locar={NomeProp, EnderProp, NomeUsuario, EMail, NRegistro}

Onde: (NomeProp, EnderProp) \xrightarrow{CE} Propriedade(Nome, Endereço),

(NomeUsuario, EMail) \xrightarrow{CE} Usuário(Nome, EMail),

(NRegistro) \xrightarrow{CE} Locação(NRegistro)

Portanto, o mapeamento final tem as seguintes relações e restrições de integridade:

Propriedade={Nome, Endereço, Classe, PreçoDiária, NomeUsuário, EMail,
DataInicial, DataFinal}

(NomeUsuário, EMail) \xrightarrow{CE} Usuário(Nome, EMail)

Propriedade={Nome, Endereço, Classe, PreçoDiária}

Usuário={Nome, EMail, Endereço, DataNasc}

Locação={NRegistro, CheckIn, CheckOut, Valor}

Comodidade={Comodidade}

ContaBancaria={Número, Tipo, Nome, EMail}

(Nome, EMail) \xrightarrow{CE} Usuário(Nome, EMail)

TerComodidade={NomeProp, EnderProp, Comodidade, Quantidade}

(Comodidade) \xrightarrow{CE} (Comodidade(Comodidade))

(NomeProp, EnderProp) \xrightarrow{CE} Propriedade(Nome, Endereço),

Locar={NomeProp, EnderProp, NomeUsuario, EMail, NRegistro}

(NomeProp, EnderProp) \xrightarrow{CE} Propriedade(Nome, Endereço),

(NomeUsuario, EMail) \xrightarrow{CE} Usuário(Nome, EMail),

(NRegistro) \xrightarrow{CE} Locação(NRegistro)

Nota: A relação Comodidade potencialmente pode ser eliminada para as consultas de leitura, pois ela não contém informações que não estejam na relação TerComodidade. No entanto, ela é importante nas operações de atualização, pois qualquer valor a ser colocado na relação TerComodidade tem que estar na relação Comodidade (via a chave estrangeira correspondente).

Exercício 3:

Mostre o Nome e Endereço de todas as propriedades colocadas para locação por exatamente 10 dias, junto com o Nome e Endereço de seus respectivos proprietários.

- Escreva o comando SQL correspondente;

```
SELECT P.Nome, P.Endereço, U.Nome, U.Endereço
FROM Propriedade P, Usuário U
WHERE (P.NomeUsuário, P.EMail)=(U.Nome, U.EMail)
AND DataFinal=DataInicial+10;
```

- Escreva a expressão correspondente em álgebra relacional.

$$\pi_{\{P.Nome, P.Endereco, U.Nome, U.Endereco\}} \left(\sigma_{(DataFinal=DataInicial+10)} \left(\begin{matrix} (NomeUsuario=Nome \wedge EMail=EMail) \\ Propriedade \bowtie Usuario \end{matrix} \right) \right)$$

Note que a comparação `DataFinal=DataInicial+10` envolve dois atributos da mesma tabela, portanto corresponde à uma seleção σ .

Exercício 4:

Mostre o Nome e Endereço de todos os quartos compartilhados com 'WiFi' disponíveis para locação. Caso esses quartos tenham tido alguma locação com início neste ano, inclua também essas locações indicando quem loca esse quarto e o período respectivo à cada locação.

```
SELECT P.Nome, P.Endereço, L.Nomeusuário, L.EMail, LR.CheckIn, LR.CheckOut
FROM Propriedade P LEFT OUTER JOIN Locar L
ON (P.Nome, P.Endereço)=(L.NomeProp, EnderProp)
JOIN Locação LR
ON L.NRegistro=LR.NRegistro
JOIN TerComodidade C
ON (P.Nome, Endereço)=(C.NomeProp, C.EnderProp)
WHERE C.Comodidade~* 'garagem'
AND EXTRACT(YEAR FROM LR.CheckIn)=2024
AND P.Classe ~* 'quarto compartilhado';
```

Aqui existem muitas variações possíveis. O que é importante:

- Tem que ter uma junção externa da **Propriedade** com o relacionamento ternário **Locar**, porque é pedido para listar todas as propriedades que atendem às aquelas condições. Sempre que existir uma locação, ela deve ser acrescentada, mas não haver locação não é impeditivo para que a **Propriedade** exista e seja listada;
- A classe da propriedade é 'quarto compartilhado' (ou algum outro padrão de texto equivalente);
- Quando existe uma tupla em **Locar**, deve-se buscar a data de **Check-In** nessa relação para verificar se ela começou este ano;
- Não é necessário usar a relação **Usuário**, pois a identificação dele já está na relação **Locar** (mas pode usar se quiser).
 - O mesmo vale para as relações **TerComodidade** e **Comodidade**.
- A comparação da data armazenada em **CheckIn** envolve o ano. Essa solução é correta, mas qualquer maneira usada para indicar isso é válido para a prova.
- Se um relacionamento existir em **Locar**, isso garante que a entidade **Locação** existe, portanto a junção interna é usada.

- Se uma **Propriedade** não tiver a **Comodidade** 'WiFi', a propriedade não deve ser listada, portanto a junção da **Propriedade** com a **Comodidade** deve ser interna também.

Exercício 5:

Liste o rendimento obtido por cada **Propriedade**.

```
SELECT P.Nome, P.Endereço, SUM(LR.Valor)
FROM Propriedade P JOIN Locar L
    ON (P.Nome, P.Endereço)=(L.NomeProp, EnderProp)
JOIN Locação LR
    ON L.NRegistro=LR.NRegistro
GROUP BY P.Nome, P.Endereço;
```

O que é importante aqui:

- A consulta é centrada no CE Ternário **Locar**, envolvendo os CE **Propriedade** e **Locação**. Nenhuma informação adicional do CE **Usuário** é necessária, portanto a respectiva relação não é usada.
- O agrupamento é feito para cada **Propriedade**, identificado por sua chave primária.

Boa Prova!