

# Modelo Relacional

Tiago Eugenio de Melo  
tmelo@uea.edu.br

Escola Superior de Tecnologia  
Universidade do Estado do Amazonas

15 de Janeiro de 2022

# Modelo Relacional

# Introdução

- Representa DADOS e RELACIONAMENTOS por um conjunto de Tabelas.

# Introdução

- Representa DADOS e RELACIONAMENTOS por um conjunto de Tabelas.
- O modelo relacional utiliza o conceito de relações ou tabelas no lugar de arquivos.

- Representa DADOS e RELACIONAMENTOS por um conjunto de Tabelas.
- O modelo relacional utiliza o conceito de relações ou tabelas no lugar de arquivos.
- As colunas de uma tabela são chamadas de atributos e suas linhas de tuplas.

- Representa DADOS e RELACIONAMENTOS por um conjunto de Tabelas.
- O modelo relacional utiliza o conceito de relações ou tabelas no lugar de arquivos.
- As colunas de uma tabela são chamadas de atributos e suas linhas de tuplas.
- A importância do modelo relacional em nosso curso deve-se ao fato de ser o modelo sobre o qual é baseada a maioria dos SGBDs comerciais disponíveis hoje em dia (MySQL, Oracle e PostgreSQL, por exemplo).

# Exemplo

relação



tuplas



CLIENTE

Nome	Cidade	Rua	Num. Conta
João	Natal	Av. Jacumã	800-X
Maria	Manaus	V8	300-8
Maria	Manaus	V8	350-3
José	Parintins	R. Belém	300-8



atributos

CONTA

Num. Conta	Saldo
800-X	55,00
300-8	1200,00
350-3	450,00



# Projeto de Banco de Dados

# Projeto de Banco de Dados

- “O objetivo básico de um projeto de banco de dados é possibilitar ao usuário obter a informação exata em um limite aceitável de tempo, de maneira a executar sua tarefa dentro da organização.” (Teorey e Fry)

# Projeto de Banco de Dados

- “O objetivo básico de um projeto de banco de dados é possibilitar ao usuário obter a informação exata em um limite aceitável de tempo, de maneira a executar sua tarefa dentro da organização.” (Teorey e Fry)
- “O objetivo de um projeto de banco de dados relacional é gerar um conjunto de esquemas relacionais, que nos permita guardar informações sem redundância desnecessária, além de nos permitir recuperar a informação facilmente.” (Korth e Silberschatz).

# Perigos potenciais de projetos

# Perigos potenciais de projetos

- Repetição de informação

# Perigos potenciais de projetos

- Repetição de informação
  - ▶ Informações repetidas consomem espaço de armazenamento e dificultam a atualização.

# Perigos potenciais de projetos

- Repetição de informação
  - ▶ Informações repetidas consomem espaço de armazenamento e dificultam a atualização.
- Incapacidade de representar parte da informação

# Perigos potenciais de projetos

- Repetição de informação
  - ▶ Informações repetidas consomem espaço de armazenamento e dificultam a atualização.
- Incapacidade de representar parte da informação
  - ▶ Por vezes tem-se que incluir valores nulos.



# Perigos potenciais de projetos

- Repetição de informação
  - ▶ Informações repetidas consomem espaço de armazenamento e dificultam a atualização.
- Incapacidade de representar parte da informação
  - ▶ Por vezes tem-se que incluir valores nulos.
- Perda de informação.

# Perigos potenciais de projetos

- Repetição de informação
  - ▶ Informações repetidas consomem espaço de armazenamento e dificultam a atualização.
- Incapacidade de representar parte da informação
  - ▶ Por vezes tem-se que incluir valores nulos.
- Perda de informação.
- Projetos mal elaborados sugerem a decomposição de esquemas relacionais com muitos atributos.

# Modelo Relacional

- O Modelo Relacional é simples e sua estrutura uniforme é baseada em conceitos da Teoria dos Conjuntos.

- O Modelo Relacional é simples e sua estrutura uniforme é baseada em conceitos da Teoria dos Conjuntos.
- A simplicidade do Modelo Relacional faz com que a representação do mundo real através de seus conceitos seja de certa forma ineficiente o que ocasiona perdas semânticas consideráveis.

- O Modelo Relacional é simples e sua estrutura uniforme é baseada em conceitos da Teoria dos Conjuntos.
- A simplicidade do Modelo Relacional faz com que a representação do mundo real através de seus conceitos seja de certa forma ineficiente o que ocasiona perdas semânticas consideráveis.
- O MER, ao contrário, utiliza conceitos que permitem a representação mais fiel dos objetos do mundo real e dos relacionamentos entre eles.

# Modelo Relacional

- O Modelo Relacional tem sido implementado nos vários SGBDs tendo como LDD/LMD a linguagem SQL.



# Modelo Relacional

- O Modelo Relacional tem sido implementado nos vários SGBDs tendo como LDD/LMD a linguagem SQL.
- O MER é hoje a ferramenta mais usada em projetos de banco de dados.

# Modelo Relacional

- O Modelo Relacional tem sido implementado nos vários SGBDs tendo como LDD/LMD a linguagem SQL.
- O MER é hoje a ferramenta mais usada em projetos de banco de dados.
- Dizemos que o MER é um modelo do nível conceitual, pois possui um forte poder semântico, capaz de capturar conceitos do mundo real com um mínimo de perdas semânticas, facilitando o seu entendimento.

# Modelo Relacional

- O Modelo Relacional tem sido implementado nos vários SGBDs tendo como LDD/LMD a linguagem SQL.
- O MER é hoje a ferramenta mais usada em projetos de banco de dados.
- Dizemos que o MER é um modelo do nível conceitual, pois possui um forte poder semântico, capaz de capturar conceitos do mundo real com um mínimo de perdas semânticas, facilitando o seu entendimento.
- O Modelo Relacional é, por outro lado, um modelo do nível lógico porque é utilizado para representação em computador de conceitos do mundo real.

# Modelo Relacional

# Modelo Relacional

- O Dr. Codd propôs o modelo relacional de sistemas de bancos de dados em 1970.

# Modelo Relacional

- O Dr. Codd propôs o modelo relacional de sistemas de bancos de dados em 1970.
- Ele é a base para o RDBMS (*Relational Database Management System*).

# Modelo Relacional

- O Dr. Codd propôs o modelo relacional de sistemas de bancos de dados em 1970.
- Ele é a base para o RDBMS (*Relational Database Management System*).
- O modelo relacional consiste nos seguintes itens:

- O Dr. Codd propôs o modelo relacional de sistemas de bancos de dados em 1970.
- Ele é a base para o RDBMS (*Relational Database Management System*).
- O modelo relacional consiste nos seguintes itens:
  - ▶ Conjunto de objetos ou relações.



# Modelo Relacional

- O Dr. Codd propôs o modelo relacional de sistemas de bancos de dados em 1970.
- Ele é a base para o RDBMS (*Relational Database Management System*).
- O modelo relacional consiste nos seguintes itens:
  - ▶ Conjunto de objetos ou relações.
  - ▶ Conjunto de operadores para agir sobre as relações.

# Modelo Relacional

- O Dr. Codd propôs o modelo relacional de sistemas de bancos de dados em 1970.
- Ele é a base para o RDBMS (*Relational Database Management System*).
- O modelo relacional consiste nos seguintes itens:
  - ▶ Conjunto de objetos ou relações.
  - ▶ Conjunto de operadores para agir sobre as relações.
  - ▶ Integridade de dados para manter a precisão e consistência dos dados.

- O Dr. Codd propôs o modelo relacional de sistemas de bancos de dados em 1970.
- Ele é a base para o RDBMS (*Relational Database Management System*).
- O modelo relacional consiste nos seguintes itens:
  - ▶ Conjunto de objetos ou relações.
  - ▶ Conjunto de operadores para agir sobre as relações.
  - ▶ Integridade de dados para manter a precisão e consistência dos dados.
- “*A relational model of data for large shared data banks*” .  
Communications of the ACM, 13(6):377-87, June 1970.

# Objetivos - Modelo Relacional

# Objetivos - Modelo Relacional

- Independência de dados.

# Objetivos - Modelo Relacional

- Independência de dados.
  - ▶ Ordem.

# Objetivos - Modelo Relacional

- Independência de dados.
  - ▶ Ordem.
  - ▶ Indexação.

# Objetivos - Modelo Relacional

- Independência de dados.
  - ▶ Ordem.
  - ▶ Indexação.
  - ▶ Caminhos de acesso.



# Objetivos - Modelo Relacional

- Independência de dados.
  - ▶ Ordem.
  - ▶ Indexação.
  - ▶ Caminhos de acesso.
- Reduzir inconsistências.

# Objetivos - Modelo Relacional

- Independência de dados.
  - ▶ Ordem.
  - ▶ Indexação.
  - ▶ Caminhos de acesso.
- Reduzir inconsistências.
  - ▶ Regras de projetos (normalização).

# Modelo Relacional - Informal

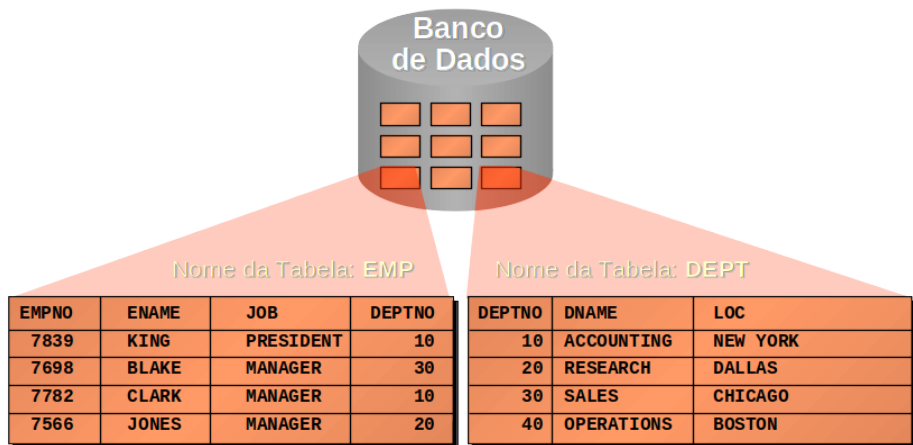
- Um banco de dados relacional é um conjunto de relações ou tabelas bidimensionais, gerenciados por operações relacionais e regidos por restrições de integridade de dados.

- Um banco de dados relacional é um conjunto de relações ou tabelas bidimensionais, gerenciados por operações relacionais e regidos por restrições de integridade de dados.
- Pode ser acessado e modificado executando instruções SQL (*Structured Query Language*).

# Modelo Relacional - Informal

- Um banco de dados relacional é um conjunto de relações ou tabelas bidimensionais, gerenciados por operações relacionais e regidos por restrições de integridade de dados.
- Pode ser acessado e modificado executando instruções SQL (*Structured Query Language*).
- Usa um conjunto de operadores.

# Exemplo



# Relação: Representação Tabular



# Relação: Representação Tabular

- Propriedades:

# Relação: Representação Tabular

- Propriedades:
  - ▶ Cada linha representa uma tupla.

# Relação: Representação Tabular

- Propriedades:
  - ▶ Cada linha representa uma tupla.
  - ▶ Não existe ordenamento entre as tuplas.

# Relação: Representação Tabular

- Propriedades:
  - ▶ Cada linha representa uma tupla.
  - ▶ Não existe ordenamento entre as tuplas.
  - ▶ Não existem linhas duplicadas.

# Relação: Representação Tabular

- Propriedades:
  - ▶ Cada linha representa uma tupla.
  - ▶ Não existe ordenamento entre as tuplas.
  - ▶ Não existem linhas duplicadas.
  - ▶ O relacionamento das colunas com os domínios estabelece-se a princípio dando os nomes dos domínios às colunas.

- Propriedades:
  - ▶ Cada linha representa uma tupla.
  - ▶ Não existe ordenamento entre as tuplas.
  - ▶ Não existem linhas duplicadas.
  - ▶ O relacionamento das colunas com os domínios estabelece-se a princípio dando os nomes dos domínios às colunas.
  - ▶ Não existe ordem entre as colunas, respeitada a propriedade acima.



- Esquema



- Esquema
  - ▶ Definição das tabelas.

- Esquema
  - ▶ Definição das tabelas.
- Instâncias

- Esquema
  - ▶ Definição das tabelas.
- Instâncias
  - ▶ Conjunto de tuplas que compõem as relações.

- Esquema
  - ▶ Definição das tabelas.
- Instâncias
  - ▶ Conjunto de tuplas que compõem as relações.
- Para cada relação define-se, usando a DDL (Linguagem de Definição de Dados):

- Esquema
  - ▶ Definição das tabelas.
- Instâncias
  - ▶ Conjunto de tuplas que compõem as relações.
- Para cada relação define-se, usando a DDL (Linguagem de Definição de Dados):
  - ▶ Nome único.

- Esquema
  - ▶ Definição das tabelas.
- Instâncias
  - ▶ Conjunto de tuplas que compõem as relações.
- Para cada relação define-se, usando a DDL (Linguagem de Definição de Dados):
  - ▶ Nome único.
  - ▶ Atributos.

- Esquema
  - ▶ Definição das tabelas.
- Instâncias
  - ▶ Conjunto de tuplas que compõem as relações.
- Para cada relação define-se, usando a DDL (Linguagem de Definição de Dados):
  - ▶ Nome único.
  - ▶ Atributos.
  - ▶ Restrições de integridade (chave primária, chave estrangeira, valores nulos, etc).

# Restrições de Integridade do Modelo Relacional



# Restrições de Integridade do Modelo Relacional

- Restrições de integridade que se aplicam a todo modelo que se diz conforme ao modelo relacional.

# Restrições de Integridade do Modelo Relacional

- Restrições de integridade que se aplicam a todo modelo que se diz conforme ao modelo relacional.
- Tipos:

# Restrições de Integridade do Modelo Relacional

- Restrições de integridade que se aplicam a todo modelo que se diz conforme ao modelo relacional.
- Tipos:
  - ▶ Integridade de identidade.

# Restrições de Integridade do Modelo Relacional

- Restrições de integridade que se aplicam a todo modelo que se diz conforme ao modelo relacional.
- Tipos:
  - ▶ Integridade de identidade.
  - ▶ Integridade referencial.

# Integridade de Identidade: Conceitos

- Chave candidata

- Chave candidata
  - ▶ Grupo mínimo de atributos tal que a combinação de valores assumida por este grupo corresponde a no máximo uma tupla da relação.

- Chave candidata
  - ▶ Grupo mínimo de atributos tal que a combinação de valores assumida por este grupo corresponde a no máximo uma tupla da relação.
  - ▶ Propriedade mínima.



- Chave candidata
  - ▶ Grupo mínimo de atributos tal que a combinação de valores assumida por este grupo corresponde a no máximo uma tupla da relação.
  - ▶ Propriedade mínima.
  - ▶ Uma relação possui **no mínimo** uma chave candidata.

# Integridade de Identidade: Conceitos

- Chave primária

- Chave primária
  - ▶ Uma entre as chaves candidatas, selecionada pelo projetista como a principal identificação das tuplas de uma relação.

- Chave primária
  - ▶ Uma entre as chaves candidatas, selecionada pelo projetista como a principal identificação das tuplas de uma relação.
  - ▶ Uma relação possui **uma e somente uma** chave primária.

# Integridade de Identidade: Conceitos

- Chave alternativa

- Chave alternativa
  - ▶ Toda chave candidata que não for chave primária.



# Integridade de Identidade: Conceitos

- Valor nulo

- Valor nulo
  - ▶ A inserção de tuplas incompletas pode introduzir valores nulos na base de dados.

- Valor nulo

- ▶ A inserção de tuplas incompletas pode introduzir valores nulos na base de dados.
- ▶ Evitar, sempre que possível.

# Integridade de Identidade: Conceitos

- Integridade de identidade

- Integridade de identidade
  - ▶ Nenhum atributo que participe de uma chave candidata de uma relação pode assumir valor nulo.

- Integridade de identidade
  - ▶ Nenhum atributo que participe de uma chave candidata de uma relação pode assumir valor nulo.
  - ▶ A DDL utilizada deve prover algum tipo de mecanismo para definir chaves primárias e candidatas (valores únicos), e para especificar quando um atributo pode ou não aceitar valores nulos.



# Integridade Referencial: Conceitos

- Chave estrangeira (externa)

- Chave estrangeira (externa)
  - ▶ Um ou mais atributos de uma relação R2 cujos valores são necessários para equivaler à chave primária de uma relação R1 (R1 e R2 não necessariamente distintos).

- Chave estrangeira (externa)
  - ▶ Um ou mais atributos de uma relação R2 cujos valores são necessários para equivaler à chave primária de uma relação R1 (R1 e R2 não necessariamente distintos).
  - ▶ Não necessariamente a chave estrangeira participa da chave primária da relação que a contém (R2).

# Integridade Referencial: Conceitos

- Integridade referencial

- Integridade referencial
  - ▶ Se uma relação R2 inclui uma chave estrangeira FK equivalendo à chave primária PK de uma relação R1, então todo valor FK em R2 deve ser:

- Integridade referencial
  - ▶ Se uma relação R2 inclui uma chave estrangeira FK equivalendo à chave primária PK de uma relação R1, então todo valor FK em R2 deve ser:
    - ★ Igual ao valor de PK em alguma tupla de R1, ou



- Integridade referencial

- ▶ Se uma relação R2 inclui uma chave estrangeira FK equivalendo à chave primária PK de uma relação R1, então todo valor FK em R2 deve ser:
  - ★ Igual ao valor de PK em alguma tupla de R1, ou
  - ★ Totalmente nulo.

# Definição de Chave Estrangeira

# Definição de Chave Estrangeira

- Política de rejeição (*DEFAULT*)

# Definição de Chave Estrangeira

- Política de rejeição (*DEFAULT*)
  - ▶ O operação só não é rejeitada se não houver tuplas (chave estrangeira) fazendo referência a uma dada chave primária.

# Definição de Chave Estrangeira

- Política de rejeição (*DEFAULT*)
  - ▶ O operação só não é rejeitada se não houver tuplas (chave estrangeira) fazendo referência a uma dada chave primária.
- Políticas compensatórias (EXPLICITAMENTE DECLARADAS)

# Definição de Chave Estrangeira

- Política de rejeição (*DEFAULT*)
  - ▶ O operação só não é rejeitada se não houver tuplas (chave estrangeira) fazendo referência a uma dada chave primária.
- Políticas compensatórias (**EXPLICITAMENTE DECLARADAS**)
  - ▶ **RESTRICT**: rejeita a atualização ou exclusão de um registro da tabela pai, se houver registros na tabela filha.

# Definição de Chave Estrangeira

- Política de rejeição (*DEFAULT*)
  - ▶ O operação só não é rejeitada se não houver tuplas (chave estrangeira) fazendo referência a uma dada chave primária.
- Políticas compensatórias (**EXPLICITAMENTE DECLARADAS**)
  - ▶ **RESTRICT**: rejeita a atualização ou exclusão de um registro da tabela pai, se houver registros na tabela filha.
  - ▶ **CASCADE**: atualiza ou exclui os registros da tabela filha automaticamente, ao atualizar ou excluir um registro da tabela pai.

# Definição de Chave Estrangeira

- Política de rejeição (*DEFAULT*)
  - ▶ O operação só não é rejeitada se não houver tuplas (chave estrangeira) fazendo referência a uma dada chave primária.
- Políticas compensatórias (**EXPLICITAMENTE DECLARADAS**)
  - ▶ **RESTRICT**: rejeita a atualização ou exclusão de um registro da tabela pai, se houver registros na tabela filha.
  - ▶ **CASCADE**: atualiza ou exclui os registros da tabela filha automaticamente, ao atualizar ou excluir um registro da tabela pai.
  - ▶ **SET NULL**: o valor da chave estrangeira é ajustada para valor NULO ao atualizar ou excluir o registro da tabela pai.



# Definição de Chave Estrangeira

- Política de rejeição (*DEFAULT*)
  - ▶ O operação só não é rejeitada se não houver tuplas (chave estrangeira) fazendo referência a uma dada chave primária.
- Políticas compensatórias (**EXPLICITAMENTE DECLARADAS**)
  - ▶ **RESTRICT**: rejeita a atualização ou exclusão de um registro da tabela pai, se houver registros na tabela filha.
  - ▶ **CASCADE**: atualiza ou exclui os registros da tabela filha automaticamente, ao atualizar ou excluir um registro da tabela pai.
  - ▶ **SET NULL**: o valor da chave estrangeira é ajustada para valor NULO ao atualizar ou excluir o registro da tabela pai.
  - ▶ **SET DEFAULT**: define o valor da coluna na tabela filha, como o valor definido como *default* para ela ao excluir ou atualizar um registro na tabela pai.

# Qual é a política?

```
CREATE TABLE departamento  
(cod INT NOT NULL,  
nome char(15) NOT NULL,  
ramal CHAR(4),  
PRIMARY KEY (cod));
```

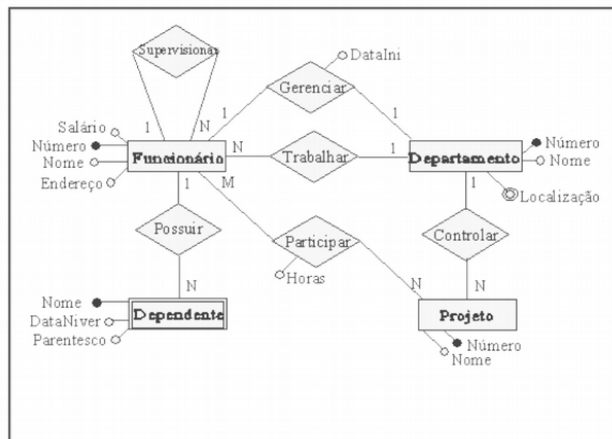
```
CREATE TABLE empregado  
(cpf CHAR(8) NOT NULL,  
nome CHAR(60) NOT NULL,  
salario MONEY,  
dep INT,  
gerente CHAR(8),  
PRIMARY KEY (cpf),  
FOREIGN KEY dep REFERENCES departamento(cod),  
FOREIGN KEY gerente REFERENCES empregado(cpf));
```

# Qual é a política?

```
CREATE TABLE empregado
(cpf CHAR(8) NOT NULL,
nome CHAR(60) NOT NULL,
salario MONEY,
dep INT,
gerente CHAR(8),
PRIMARY KEY (cpf),
FOREIGN KEY dep REFERENCES departamento(cod)
ON DELETE SET NULL
ON UPDATE CASCADE
FOREIGN KEY gerente REFERENCES empregado(cpf)
ON DELETE SET NULL);
```

# Mapeamento ER para Relacional

# ER para Relacional



# Passo 1

# Passo 1

- Mapear todos os conjuntos de entidades regulares (que não são fracas).

# Passo 1

- Mapear todos os conjuntos de entidades regulares (que não são fracas).
- Para cada conjunto de entidade E no esquema ER cria-se uma relação R que inclui todos os atributos de E.



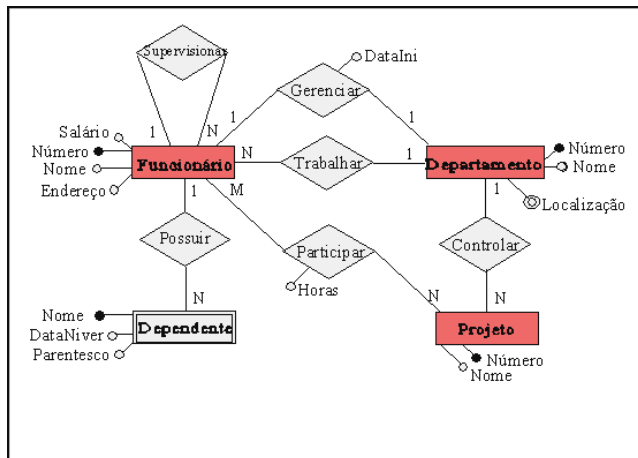
# Passo 1

- Mapear todos os conjuntos de entidades regulares (que não são fracas).
- Para cada conjunto de entidade E no esquema ER cria-se uma relação R que inclui todos os atributos de E.
- Caso exista atributo composto, inclua todos os atributos elementares que compõem o atributo composto.

# Passo 1

- Mapear todos os conjuntos de entidades regulares (que não são fracas).
- Para cada conjunto de entidade E no esquema ER cria-se uma relação R que inclui todos os atributos de E.
- Caso exista atributo composto, inclua todos os atributos elementares que compõem o atributo composto.
- Escolha um dos atributos chave de E como chave primária para a relação R.

# Identificação das entidades regulares (fortes)



# Mapear conjuntos de entidades regulares

# Mapear conjuntos de entidades regulares

- Funcionario = {FNumero, FNome, FEndereco, FSalario}

# Mapear conjuntos de entidades regulares

- Funcionario = {FNumero, FNome, FEndereco, FSalario}
- Departamento = {DNumero, DNome}

# Mapear conjuntos de entidades regulares

- Funcionario = {FNumero, FNome, FEndereco, FSalario}
- Departamento = {DNumero, DNome}
- Projeto = {PNumero, PNome}

# Passo 1 - Observações



# Passo 1 - Observações

- O atributo “Localização” não foi representado na relação “Departamento” pois é um atributo multi-valorado e será tratado no passo 7.

# Passo 1 - Observações

- O atributo “Localização” não foi representado na relação “Departamento” pois é um atributo multi-valorado e será tratado no passo 7.
- O conjunto de entidade “Dependente” é um conjunto de entidade fraco e será tratado no seguinte, o passo 2.

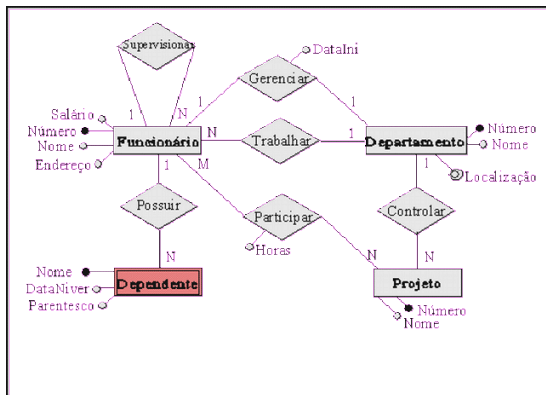
# Passo 2

- Para cada entidade fraca  $F$  no esquema ER cria-se uma relação  $R$  formada por todos os atributos de  $F$  mais os atributos que são chave das entidades envolvidas nos relacionamentos com  $F$ .

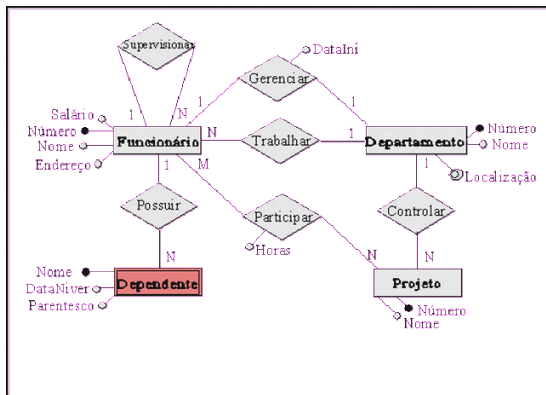
## Passo 2

- Para cada entidade fraca  $F$  no esquema ER cria-se uma relação  $R$  formada por todos os atributos de  $F$  mais os atributos que são chave das entidades envolvidas nos relacionamentos com  $F$ .
- A chave de  $R$  é a chave de  $F$  concatenada com as chaves das entidades envolvidas com  $F$ .

# Mapear conjuntos de entidades fracas

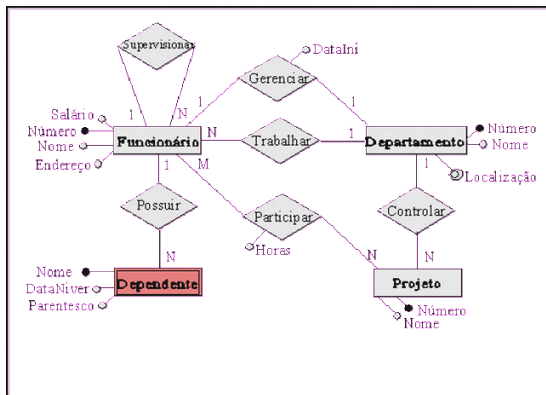


# Mapear conjuntos de entidades fracas



Segundo passo:

# Mapear conjuntos de entidades fracas



Segundo passo:

- Dependente = {DNome, FNúmero, DataNiver, Parentesco}



# Passo 3 - Mapear Conjuntos de Relacionamentos Binário 1:1

## Passo 3 - Mapear Conjuntos de Relacionamentos Binário 1:1

- O mapeamento de conjuntos de relacionamentos (CR) nem sempre resultam em uma nova relação, como é o caso dos conjuntos de relacionamentos binários 1:1. Deve-se identificar os conjuntos de entidades S e T que participam do relacionamento.

# Passo 3 - Mapear Conjuntos de Relacionamentos Binário 1:1

- O mapeamento de conjuntos de relacionamentos (CR) nem sempre resultam em uma nova relação, como é o caso dos conjuntos de relacionamentos binários 1:1. Deve-se identificar os conjuntos de entidades S e T que participam do relacionamento.
- Uma das entidades poderá ser escolhida. Se uma entidade participar de forma total no relacionamento, esta deverá ser escolhida.

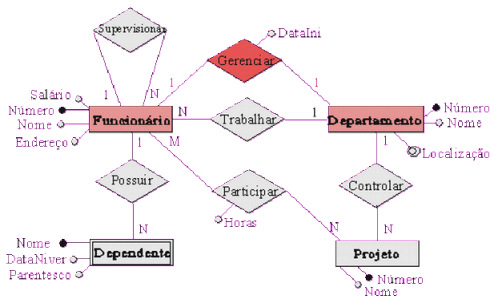
## Passo 3 - Mapear Conjuntos de Relacionamentos Binário 1:1

- O mapeamento de conjuntos de relacionamentos (CR) nem sempre resultam em uma nova relação, como é o caso dos conjuntos de relacionamentos binários 1:1. Deve-se identificar os conjuntos de entidades S e T que participam do relacionamento.
- Uma das entidades poderá ser escolhida. Se uma entidade participar de forma total no relacionamento, esta deverá ser escolhida.
- Acrescenta-se a entidade escolhida os atributos do relacionamento e ainda os atributos chaves da outra entidade.

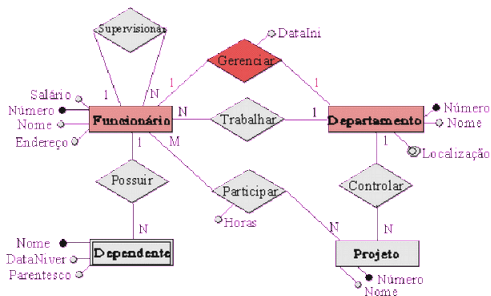
## Passo 3 - Mapear Conjuntos de Relacionamentos Binário 1:1

- O mapeamento de conjuntos de relacionamentos (CR) nem sempre resultam em uma nova relação, como é o caso dos conjuntos de relacionamentos binários 1:1. Deve-se identificar os conjuntos de entidades S e T que participam do relacionamento.
- Uma das entidades poderá ser escolhida. Se uma entidade participar de forma total no relacionamento, esta deverá ser escolhida.
- Acrescenta-se a entidade escolhida os atributos do relacionamento e ainda os atributos chaves da outra entidade.
- Deve-se ressaltar que os atributos chave que foram acrescentados à entidade escolhida são incluídos como atributos não chave nesta relação.

# Mapear Conjuntos de Relacionamentos Binário 1:1

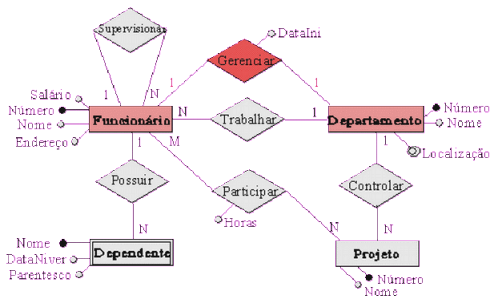


# Mapear Conjuntos de Relacionamentos Binário 1:1



Terceiro passo:

# Mapear Conjuntos de Relacionamentos Binário 1:1

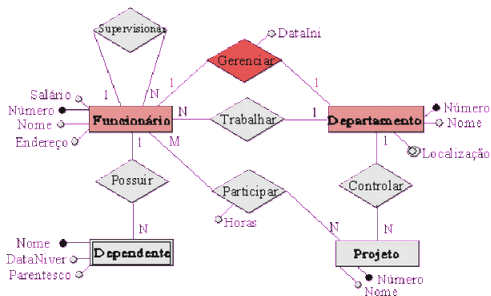


Terceiro passo:

- Departamento = { DNumero, DNome, DataIni, FNumero }



# Mapear Conjuntos de Relacionamentos Binário 1:1



Terceiro passo:

- Departamento = {DNumero, DNome, DataIni, FNumero}

Observação: Os atributos FNumero e DataIni são adicionados à relação “Departamento”, definida no primeiro passo. Note que o atributo FNumero é adicionado como um atributo não chave na relação “Departamento”.

# Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

## Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

- Os conjuntos de relacionamento binário regular (não fraco) 1:N também não são representados como novas relações.

## Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

- Os conjuntos de relacionamento binário regular (não fraco) 1:N também não são representados como novas relações.
- Primeiro identifica-se o conjunto de entidades que participa da relação com cardinalidade N que será chamada de S e o outro conjunto de entidade chamada de T.

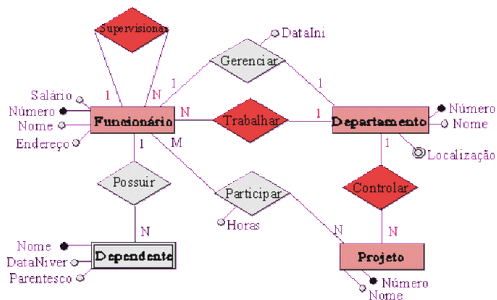
## Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

- Os conjuntos de relacionamento binário regular (não fraco) 1:N também não são representados como novas relações.
- Primeiro identifica-se o conjunto de entidades que participa da relação com cardinalidade N que será chamada de S e o outro conjunto de entidade chamada de T.
- Os atributos do conjunto de relacionamentos são acrescentados no conjunto de entidade S, ou seja, o conjunto de entidade com cardinalidade N.

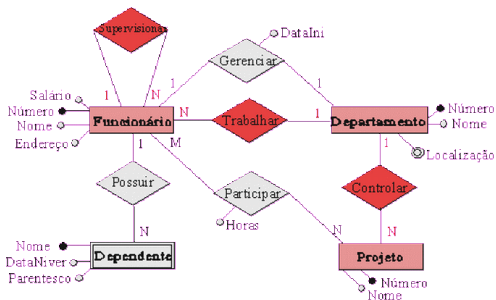
## Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

- Os conjuntos de relacionamento binário regular (não fraco) 1:N também não são representados como novas relações.
- Primeiro identifica-se o conjunto de entidades que participa da relação com cardinalidade N que será chamada de S e o outro conjunto de entidade chamada de T.
- Os atributos do conjunto de relacionamentos são acrescentados no conjunto de entidade S, ou seja, o conjunto de entidade com cardinalidade N.
- Os atributos chave da relação que mapeia o conjunto de entidade que participa com cardinalidade 1, representado por T, são também acrescentados no conjunto de entidade S como atributos não chave.

# Mapear Conjuntos de Relacionamentos Binário 1:N



# Mapear Conjuntos de Relacionamentos Binário 1:N



Neste passo são mapeados os relacionamentos: supervisionar, trabalhar e controlar.



# Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

# Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

Quarto passo:

## Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

Quarto passo:

- Funcionario = {FNumero, FNome, FEndereco, SSalario, SuperNumero, DNumero}

## Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

Quarto passo:

- Funcionario = {FNumero, FNome, FEndereco, SSalario, SuperNumero, DNumero}
- Projeto = {PNumero, PNome, DNumero}

## Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

Quarto passo:

- Funcionario = {FNumero, FNome, FEndereco, SSalario, SuperNumero, DNumero}
- Projeto = {PNumero, PNome, DNumero}
- Avaliando o relacionamento supervisionar, nota-se que este relacionamento não possui atributos, logo, apenas o atributo SuperNumero foi adicionado a relação “Funcionario”, definida no primeiro passo.

## Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

Quarto passo:

- $\text{Funcionario} = \{\underline{\text{FNumero}}, \text{FNome}, \text{FEndereco}, \text{SSalario}, \underline{\text{SuperNumero}}, \underline{\text{DNumero}}\}$
- $\text{Projeto} = \{\underline{\text{PNumero}}, \text{PNome}, \underline{\text{DNumero}}\}$
- Avaliando o relacionamento supervisionar, nota-se que este relacionamento não possui atributos, logo, apenas o atributo SuperNumero foi adicionado a relação “Funcionario”, definida no primeiro passo.
- O mesmo acontece com o relacionamento trabalhar; apenas o atributo DNumero foi adicionado a relação “Funcionario”.

# Passo 4 - Mapear Conjuntos de Relacionamentos Binários Regulares 1:N

Quarto passo:

- $\text{Funcionario} = \{\underline{\text{FNumero}}, \text{FNome}, \text{FEndereco}, \text{SSalario}, \underline{\text{SuperNumero}}, \underline{\text{DNumero}}\}$
- $\text{Projeto} = \{\underline{\text{PNumero}}, \text{PNome}, \underline{\text{DNumero}}\}$
- Avaliando o relacionamento supervisionar, nota-se que este relacionamento não possui atributos, logo, apenas o atributo SuperNumero foi adicionado a relação “Funcionario”, definida no primeiro passo.
- O mesmo acontece com o relacionamento trabalhar; apenas o atributo DNumero foi adicionado a relação “Funcionario”.
- No relacionamento controlar, que também não possui atributos, foi adicionado o atributo DNumero na relação “Projeto” que já havia sido definida no primeiro passo.

# Passo 5 - Mapear Conjuntos de Relacionamentos Binário N:N



## Passo 5 - Mapear Conjuntos de Relacionamentos Binário N:N

- Para cada relacionamento binário N:N cria-se uma nova relação.

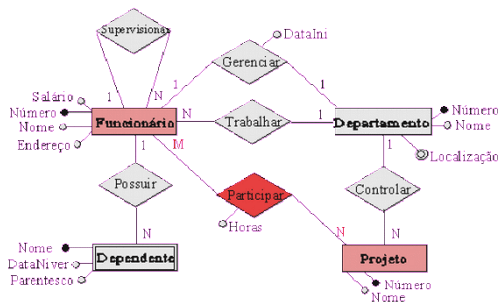
## Passo 5 - Mapear Conjuntos de Relacionamentos Binário N:N

- Para cada relacionamento binário N:N cria-se uma nova relação.
- Os atributos da relação são os atributos do conjunto de relacionamento juntamente com os atributos chave das relações que mapeiam os conjuntos de entidades envolvidas.

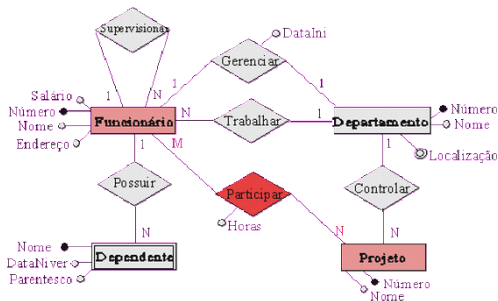
# Passo 5 - Mapear Conjuntos de Relacionamentos Binário N:N

- Para cada relacionamento binário N:N cria-se uma nova relação.
- Os atributos da relação são os atributos do conjunto de relacionamento juntamente com os atributos chave das relações que mapeiam os conjuntos de entidades envolvidas.
- A chave da relação é a concatenação dos atributos chaves das relações que mapeiam os conjuntos de entidades envolvidas.

# Mapear Conjuntos de Relacionamentos Binário N:N

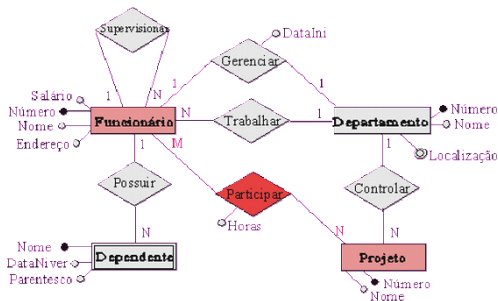


# Mapear Conjuntos de Relacionamentos Binário N:N



Quinto passo:

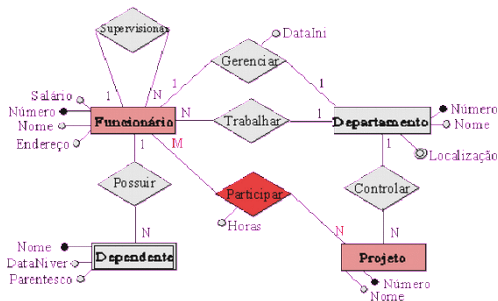
# Mapear Conjuntos de Relacionamentos Binário N:N



Quinto passo:

O relacionamento “participar” que envolve os conjuntos de entidades “Funcionario” e “Projeto” é avaliado neste passo.

# Mapear Conjuntos de Relacionamentos Binário N:N



Quinto passo:

O relacionamento “participar” que envolve os conjuntos de entidades “Funcionario” e “Projeto” é avaliado neste passo.

- Participar = { FNumero, PNumero, horas }

## Passo 6 - Mapear Conjuntos de Relacionamentos $n$ -ário, $n > 2$



## Passo 6 - Mapear Conjuntos de Relacionamentos n-ário, $n > 2$

- Para conjuntos de relacionamentos n-ário,  $n > 2$  sempre considera-se que possuem cardinalidade vários:vários:vários. Para cada conjunto de relacionamento (CR) será criada uma nova relação cujos atributos próprios são os do CR (se existirem) e cuja chave é formada pelos atributos chave das relações que mapeiam os conjuntos de entidades (CE's) envolvidos.

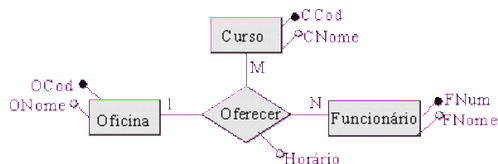
## Passo 6 - Mapear Conjuntos de Relacionamentos $n$ -ário, $n > 2$

- Para conjuntos de relacionamentos  $n$ -ário,  $n > 2$  sempre considera-se que possuem cardinalidade vários:vários:vários. Para cada conjunto de relacionamento (CR) será criada uma nova relação cujos atributos próprios são os do CR (se existirem) e cuja chave é formada pelos atributos chave das relações que mapeiam os conjuntos de entidades (CE's) envolvidos.
- Os CR's de ordem maior que três são tratados da mesma maneira que os ternários. Seu mapeamento cria uma relação para cada CR e esta relação possui: os atributos do CR como atributos próprios e, como chave, os atributos concatenados de cada relação que mapeia os CE's envolvidos.

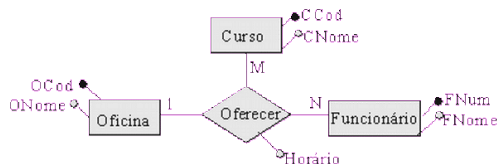
## Passo 6 - Mapear Conjuntos de Relacionamentos $n$ -ário, $n > 2$

- Para conjuntos de relacionamentos  $n$ -ário,  $n > 2$  sempre considera-se que possuam cardinalidade vários:vários:vários. Para cada conjunto de relacionamento (CR) será criada uma nova relação cujos atributos próprios são os do CR (se existirem) e cuja chave é formada pelos atributos chave das relações que mapeiam os conjuntos de entidades (CE's) envolvidos.
- Os CR's de ordem maior que três são tratados da mesma maneira que os ternários. Seu mapeamento cria uma relação para cada CR e esta relação possui: os atributos do CR como atributos próprios e, como chave, os atributos concatenados de cada relação que mapeia os CE's envolvidos.
- Como a modelagem utilizada para ilustrar os passos anteriores não tem relacionamento  $n$ -ário, com  $n > 2$ , escolheu-se um exemplo particular que será apresentado a seguir.

## Passo 6 - Mapear Conjuntos de Relacionamentos n-ário, $n > 2$

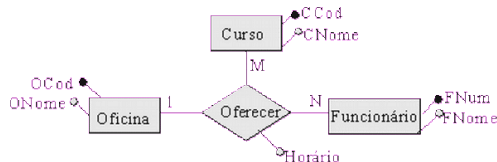


## Passo 6 - Mapear Conjuntos de Relacionamentos n-ário, $n > 2$



Sexto passo:

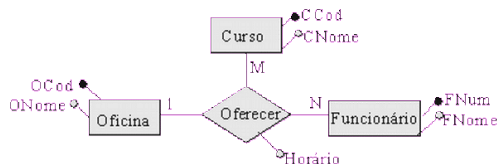
## Passo 6 - Mapear Conjuntos de Relacionamentos n-ário, $n > 2$



Sexto passo:

O relacionamento “oferecer” que envolve os conjuntos de entidades “Oficina”, “Curso” e “Funcionario” é avaliado neste passo.

## Passo 6 - Mapear Conjuntos de Relacionamentos n-ário, $n > 2$



Sexto passo:

O relacionamento “oferecer” que envolve os conjuntos de entidades “Oficina”, “Curso” e “Funcionario” é avaliado neste passo.

- Oferecer = {OCod, CCod, FNum, Horário}

# Passo 7 - Mapear Atributos Multi-valorados



## Passo 7 - Mapear Atributos Multi-valorados

Existem duas maneiras de mapear atributos multi-valorados:

## Passo 7 - Mapear Atributos Multi-valorados

Existem duas maneiras de mapear atributos multi-valorados:

- 1 A primeira maneira não se leva em conta o conhecimento adicional sobre o atributo que está sendo mapeado. Para cada atributo multi-valorado cria-se uma nova relação que tem como chave os atributos chave da relação a qual pertencia juntamente com o atributo multi-valorado tomado como um atributo simples (mono-valorado).

## Passo 7 - Mapear Atributos Multi-valorados

Existem duas maneiras de mapear atributos multi-valorados:

- 1 A primeira maneira não se leva em conta o conhecimento adicional sobre o atributo que está sendo mapeado. Para cada atributo multi-valorado cria-se uma nova relação que tem como chave os atributos chave da relação a qual pertencia juntamente com o atributo multi-valorado tomado como um atributo simples (mono-valorado).
- 2 A segunda forma de mapear atributos multi-valorados leva-se em conta o conhecimento adicional sobre o atributo que está sendo mapeado. Em alguns casos é possível determinar a quantidade de ocorrências de valores nos atributos.

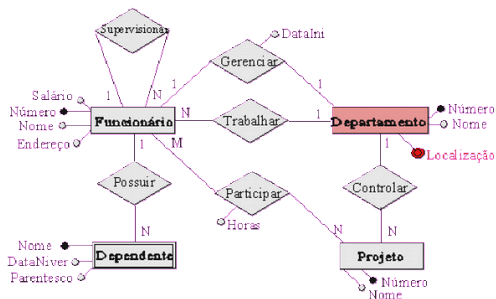
## Passo 7 - Mapear Atributos Multi-valorados

Existem duas maneiras de mapear atributos multi-valorados:

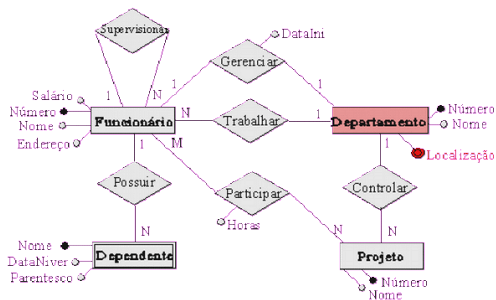
- 1 A primeira maneira não se leva em conta o conhecimento adicional sobre o atributo que está sendo mapeado. Para cada atributo multi-valorado cria-se uma nova relação que tem como chave os atributos chave da relação a qual pertencia juntamente com o atributo multi-valorado tomado como um atributo simples (mono-valorado).
- 2 A segunda forma de mapear atributos multi-valorados leva-se em conta o conhecimento adicional sobre o atributo que está sendo mapeado. Em alguns casos é possível determinar a quantidade de ocorrências de valores nos atributos.

Quando isso acontece e essa quantidade é pequena, pode-se instanciar essa quantidade de atributos como mono-valorados na mesma relação que mapeia o conjunto de entidade ou conjunto de relacionamento ao qual o atributo multi-valorado está associado.

# Passo 7 - Mapear Atributos Multi-valorados

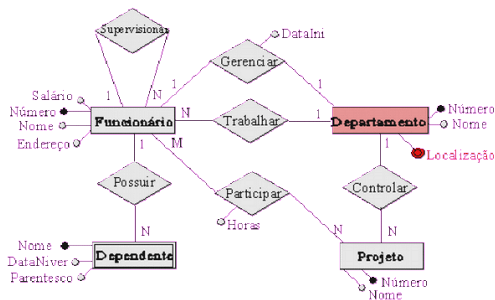


# Passo 7 - Mapear Atributos Multi-valorados



Sétimo passo:

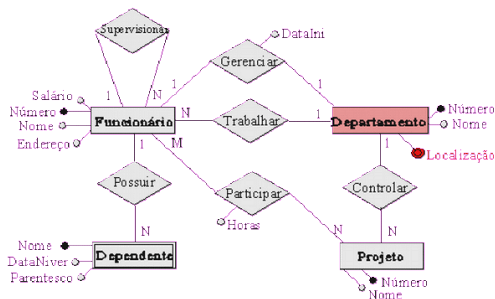
## Passo 7 - Mapear Atributos Multi-valorados



Sétimo passo:

O atributo “Localizacao” é transformado em uma nova relação.

## Passo 7 - Mapear Atributos Multi-valorados



Sétimo passo:

O atributo “Localizacao” é transformado em uma nova relação.

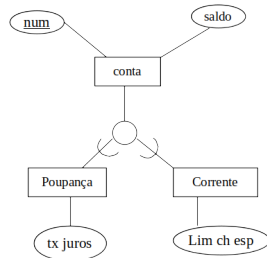
- LocalDep = {DNumero, Localizacao}



# Passo 8 - Mapear Generalização/Especialização

## Passo 8 - Mapear Generalização/Especialização

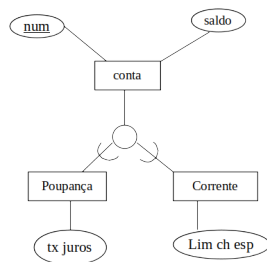
Existem duas maneiras de transformar uma generalização em tabelas:



## Passo 8 - Mapear Generalização/Especialização

Existem duas maneiras de transformar uma generalização em tabelas:

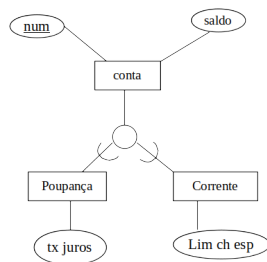
- 1 Criar a tabela para o conjunto de entidades de nível superior. Para cada conjunto de entidades de nível inferior, criar uma tabela que inclua uma coluna para cada um dos atributos daquele conjunto de entidades mais uma coluna para cada atributo da chave primária do conjunto de entidades de nível superior.



## Passo 8 - Mapear Generalização/Especialização

Existem duas maneiras de transformar uma generalização em tabelas:

- 1 Criar a tabela para o conjunto de entidades de nível superior. Para cada conjunto de entidades de nível inferior, criar uma tabela que inclua uma coluna para cada um dos atributos daquele conjunto de entidades mais uma coluna para cada atributo da chave primária do conjunto de entidades de nível superior.

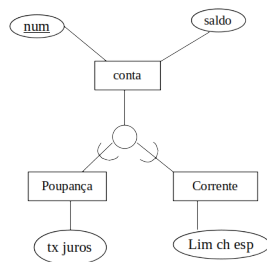


- Conta = {num, saldo}

## Passo 8 - Mapear Generalização/Especialização

Existem duas maneiras de transformar uma generalização em tabelas:

- 1 Criar a tabela para o conjunto de entidades de nível superior. Para cada conjunto de entidades de nível inferior, criar uma tabela que inclua uma coluna para cada um dos atributos daquele conjunto de entidades mais uma coluna para cada atributo da chave primária do conjunto de entidades de nível superior.

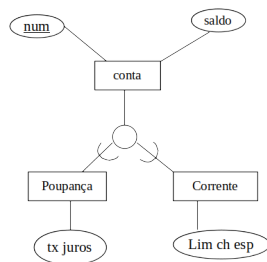


- Conta = {num, saldo}
- Poupança = {num, tx\_juros}

## Passo 8 - Mapear Generalização/Especialização

Existem duas maneiras de transformar uma generalização em tabelas:

- 1 Criar a tabela para o conjunto de entidades de nível superior. Para cada conjunto de entidades de nível inferior, criar uma tabela que inclua uma coluna para cada um dos atributos daquele conjunto de entidades mais uma coluna para cada atributo da chave primária do conjunto de entidades de nível superior.



- Conta = {num, saldo}
- Poupança = {num, tx\_juros}
- Corrente = {num, lim\_ch\_esp}

# Passo 8 - Mapear Generalização/Especialização

## Passo 8 - Mapear Generalização/Especialização

Existem duas maneiras de transformar uma generalização em tabelas:



## Passo 8 - Mapear Generalização/Especialização

Existem duas maneiras de transformar uma generalização em tabelas:

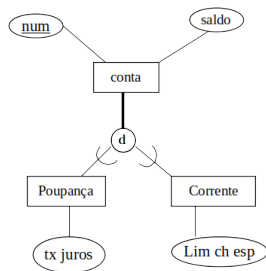
- 2 Se a generalização é **mutuamente exclusiva** e **total**, então uma outra representação alternativa é possível.

## Passo 8 - Mapear Generalização/Especialização

Existem duas maneiras de transformar uma generalização em tabelas:

- Se a generalização é **mutuamente exclusiva** e **total**, então uma outra representação alternativa é possível.

Para cada conjunto de entidades de nível inferior, cria-se uma tabela que inclua uma coluna para cada um dos atributos do conjunto de entidades mais uma coluna para cada atributo de conjunto de entidades de nível superior

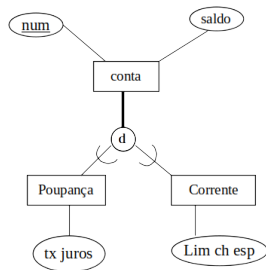


## Passo 8 - Mapear Generalização/Especialização

Existem duas maneiras de transformar uma generalização em tabelas:

- 2 Se a generalização é **mutuamente exclusiva** e **total**, então uma outra representação alternativa é possível.

Para cada conjunto de entidades de nível inferior, cria-se uma tabela que inclua uma coluna para cada um dos atributos do conjunto de entidades mais uma coluna para cada atributo de conjunto de entidades de nível superior



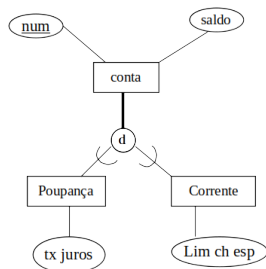
- Poupança = {num, saldo, tx\_juros}

## Passo 8 - Mapear Generalização/Especialização

Existem duas maneiras de transformar uma generalização em tabelas:

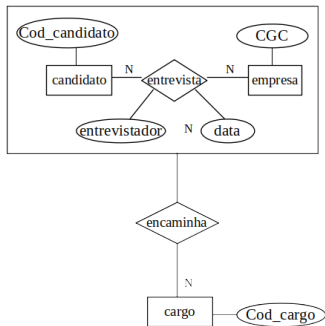
- Se a generalização é **mutuamente exclusiva** e **total**, então uma outra representação alternativa é possível.

Para cada conjunto de entidades de nível inferior, cria-se uma tabela que inclua uma coluna para cada um dos atributos do conjunto de entidades mais uma coluna para cada atributo de conjunto de entidades de nível superior

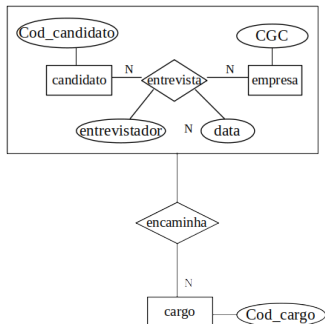


- Poupança = {num, saldo, tx\_juros}
- Corrente = {num, saldo, lim\_ch\_esp}

## Passo 9 - Mapear Agregação

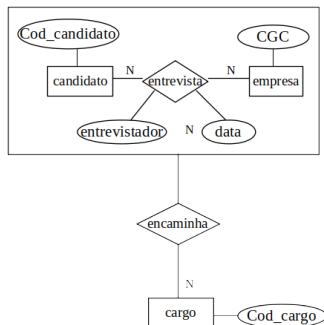


## Passo 9 - Mapear Agregação



Nono passo:

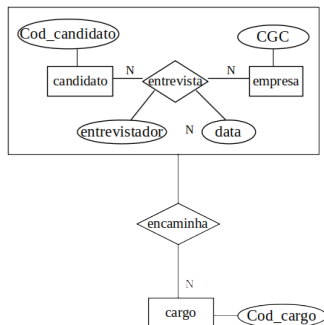
## Passo 9 - Mapear Agregação



Nono passo:

A transformação de agregação em tabela é bastante direta. Considere o exemplo acima. A tabela para o relacionamento “entrevista” inclui uma coluna para cada atributo do relacionamento, uma para a chave primária de “candidato” e uma para “empresa”.

## Passo 9 - Mapear Agregação



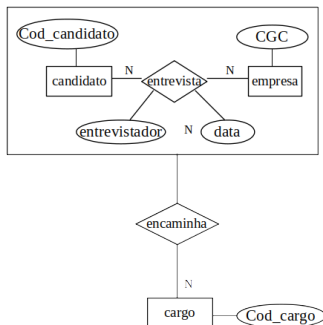
Nono passo:

A transformação de agregação em tabela é bastante direta. Considere o exemplo acima. A tabela para o relacionamento “entrevista” inclui uma coluna para cada atributo do relacionamento, uma para a chave primária de “candidato” e uma para “empresa”.

- `entrevista = {cod_candidato, CGC, entrevistador, data}`



## Passo 9 - Mapear Agregação



Nono passo:

A transformação de agregação em tabela é bastante direta. Considere o exemplo acima. A tabela para o relacionamento “entrevista” inclui uma coluna para cada atributo do relacionamento, uma para a chave primária de “candidato” e uma para “empresa”.

- entrevista = {cod\_candidato, CGC, entrevistador, data}
- encaminha = {cod\_candidato, CGC, cod\_cargo}

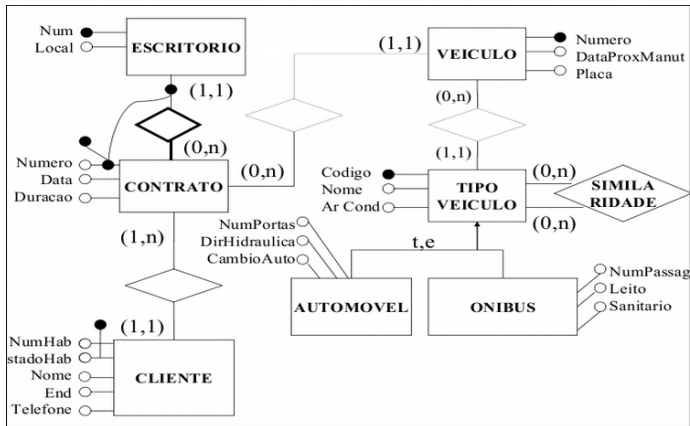
# Exercícios

# Questões para revisão

- Qual é a diferença entre LDD e LDM? Dê um exemplo prático de cada.
- Qual é a propriedade do Modelo Relacional que garante que não existem linhas duplicadas nas tabelas?
- Explique a diferença entre esquema e instância. Dê um exemplo de cada.
- Qual é a diferença entre chave candidata e chave primária. Dê um exemplo de cada.
- O valor 0 (zero) é considerado um valor nulo em um banco de dados? Justifique a sua resposta.
- Ocorre uma violação das propriedades do Modelo Relacional se uma chave estrangeira tiver um valor nulo? Justifique a sua resposta.

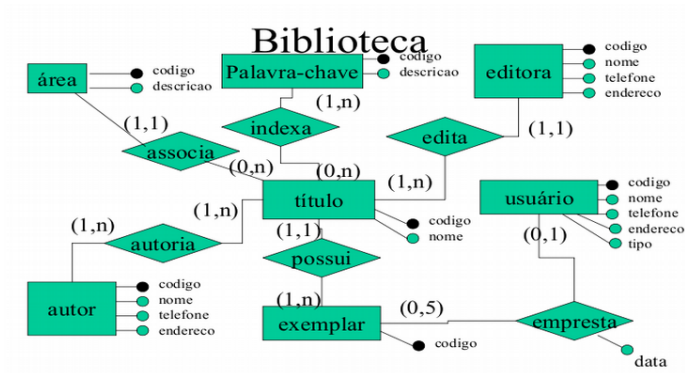
# Exercícios

Faça o mapeamento para o Modelo Relacional.



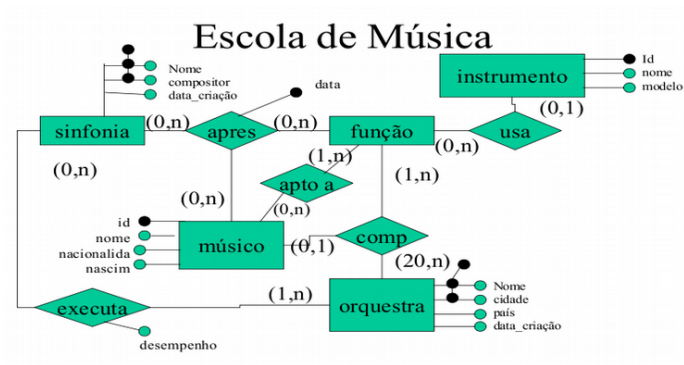
# Exercícios

Faça o mapeamento para o Modelo Relacional.



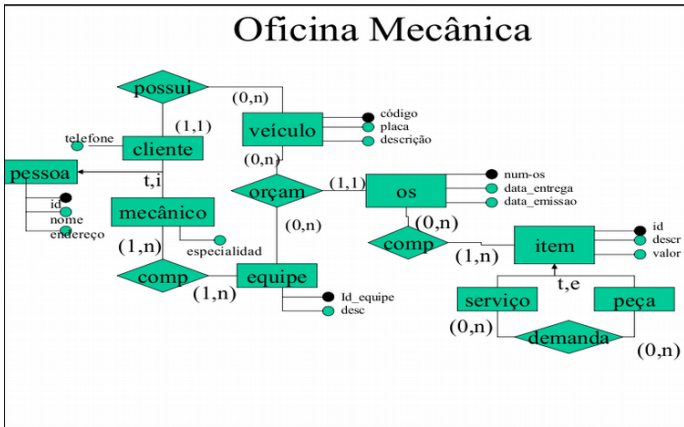
# Exercícios

Faça o mapeamento para o Modelo Relacional.



# Exercícios

Faça o mapeamento para o Modelo Relacional.



# Exercícios

Faça o mapeamento para o Modelo Relacional.

