TÉCNICO LISBOA

Bases de Dados

T08 - Conversão E-A–Relacional

Prof. Daniel Faria
Prof. Flávio Martins

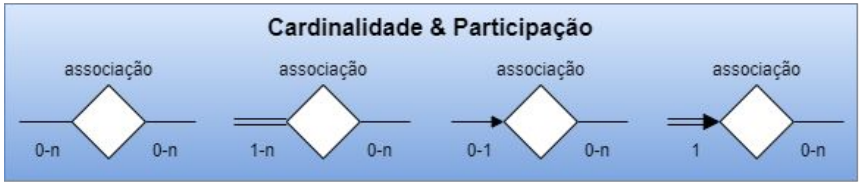
Sumário

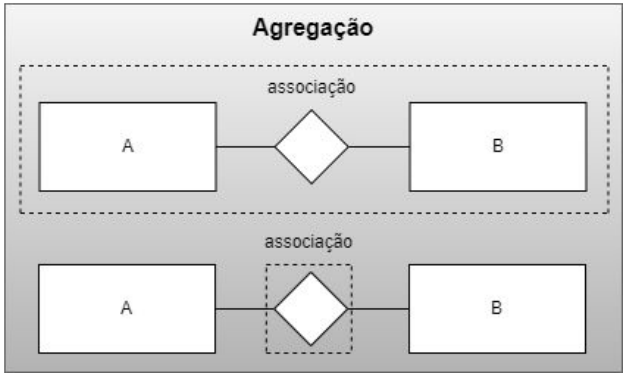
- Recapitulação Breve
- Modelo Relacional
- Conversão E-A–Relacional (Parte I)

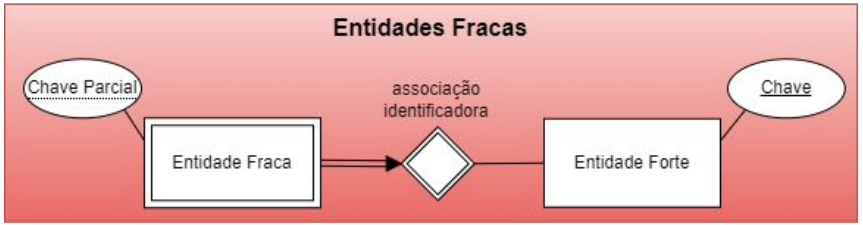
TÉCNICO LISBOA

Recapitulação Breve

Modelo Entidade-Associação







TÉCNICO LISBOA

Erros Comuns

Atributos:

- Atributos deriváveis e mutáveis:
  - e.g. “idade” vs. “data de nascimento”; “duração” vs. “data de início”
- Atributos com sub-atributos representados:
  - e.g. “morada” subdividida em “rua”, “código postal” e “país”
- Atributos que representam associações:
  - e.g. “departamento” em “empregado” para representar a associação “trabalha”
- Atributos com múltiplos valores por instância da entidade:
  - e.g. “cor” em “carro” num sistema de informação de uma marca, em que os clientes podem escolher a cor que querem para carros identificados por “modelo” e “ano”

Erros Comuns

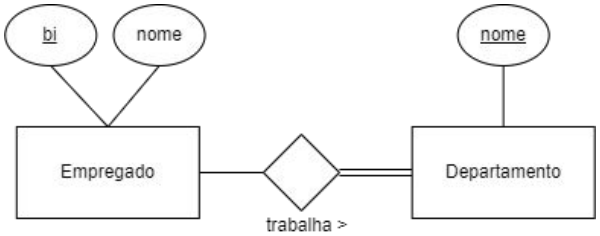
Entidades:

- Entidades sem atributos ou sem chave primária
- Entidades com chave primária mal definida
  - Por exemplo, definição de chave primária composta por dois atributos quando apenas um seria suficiente (único/chave candidata)
- Entidades isoladas no diagrama (i.e. sem associações)
- Entidades sem instâncias/records ou com apenas uma instância/record
  - e.g. “empresa” num diagrama que modela dados de uma única empresa
- Chaves fracas como fortes / Entidades fracas como fortes

Erros Comuns

Associações e Agregações:

- Cardinalidade e participação mal representadas
  - e.g. do lado oposto da associação restringe a entidade errada
- Confundir participação opcional da entidade na associação com ocorrência opcional na associação



Pode haver instâncias de **Empregado** que não estão em *trabalha*, mas todas as instâncias de *trabalha* têm sempre exactamente um **Empregado** e um **Departamento**

Erros Comuns

Restrições de Integridade Textuais:

- Definir textualmente restrições que podem ser modeladas graficamente
- Definir textualmente considerações sobre a funcionalidade do sistema e/ou que não têm manifestação sobre o esquema de dados, e.g.,
  - RI-1: Quando um empregado é removido da base de dados, os seus dependentes também o devem ser
  - RI-2: Um empregado só pode tornar-se gestor de um departamento onde ele trabalhou anteriormente (num esquema de dados que só captura o estado presente)

TÉCNICO LISBOA

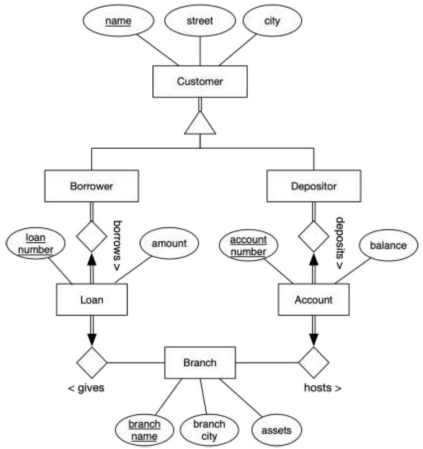
Modelo Relacional

Concepção de Bases de Dados

Especificação de Requisitos

- requisito funcional 1:
- requisito funcional 2:
- …
- restrição de integridade 1:
- restrição de integridade 2:
- …

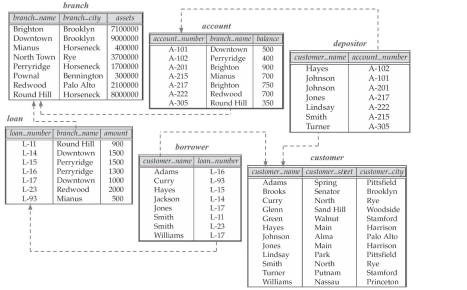
Modelo Conceptual



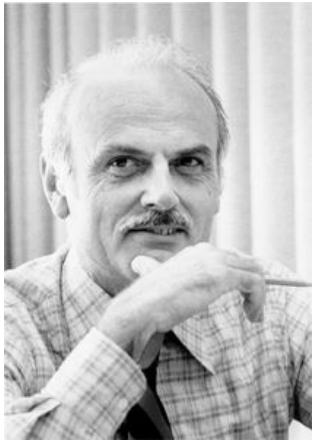
Modelo Relacional

R1 (x, y)
R2 (x, z)

Esquema Relacional (SQL)



Modelo Relacional



E. F. Codd, "A Relational Model of Data for Large Shared Data Banks", *Comm. of the ACM* **13**(6) 1970

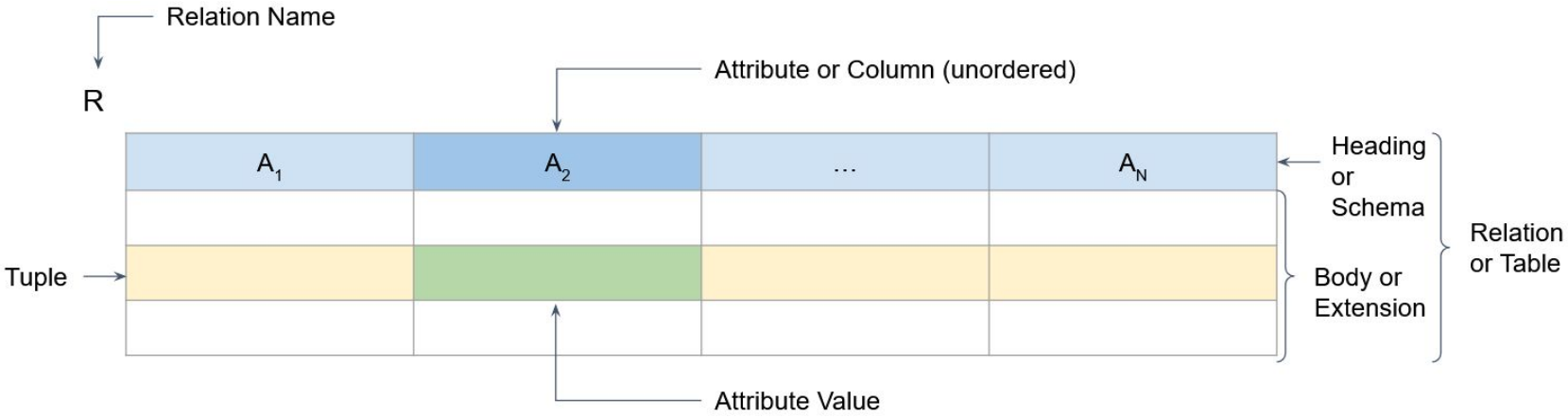
- Baseado em lógica de 1ª ordem: relação é um predicado n-ário
- Raciocínio sobre os dados sob lógica de predicados de dois valores (verdadeiro ou falso)
- Assunção de "mundo fechado": proposições ausentes do corpo de uma relação são falsas
- Operações sobre os dados baseadas na teoria dos conjuntos

Modelo Relacional

- Base de dados é uma coleção de relações
- Cada relação é um conjunto de tuplos, representado como uma tabela com colunas e linhas
- Álgebra relacional e cálculo relacional para seleccionar dados, com base na teoria dos conjuntos
- Independente de qualquer implementação
    - Pode ser implementado bastante fidedignamente em qualquer SGBD relacional (i.e. que implemente o standard SQL)

Relação

- Conjunto de tuplos n-ários que obedecem a uma especificação de nome e domínio de dados definida num cabeçalho
  - Representada normalmente como uma tabela



TÉCNICO LISBOA

Relação: Definição Formal

- Dado um esquema de atributos $A_1 \ldots A_n$ em que cada Atributo A_i tem um domínio de valores possíveis $D(A_i)$
- Uma Relação R é um conjunto contido no espaço dimensional do esquema de atributos:
  - $R \subseteq D_1 \times \ldots \times D_n$
- Cada elemento $t \in R$ é um n-tuplo:
  - $t = \langle v_1, \ldots, v_n \rangle \mid v_1 \in D(A_1), \ldots, v_n \in D(A_n)$

Relação: Propriedades

- O nome de uma relação deve ser único na base de dados
- O cabeçalho de uma relação é um tuplo que indica o nome e domínio (de dados) de todos os atributos
- Cada atributo deve ter um nome único na relação
- Cada tuplo no corpo de uma relação deve ser único e deve conter um valor para cada atributo, pertencendo ao domínio desse atributo
- O corpo de uma relação é não-ordenado
- Os tuplos são não-ordenados

Relação: Características

- O grau (ou aridade) de uma relação é o seu número de atributos (ou colunas)
- A cardinalidade de uma relação é o seu número de tuplos (ou linhas)
- Duas relações podem partilhar atributos (nominal ou conceptualmente)
    - Atributos comuns são utilizados para interligar relações

Exercício

Quais dos seguintes conjuntos de tuplos não podem ser o corpo de uma relação?

A. {}
B. {⟨⟩}
C. {⟨1⟩}
D. {⟨1⟩, ⟨2⟩}
E. {⟨Alice⟩, ⟨1000⟩}
F. {⟨Alice, 100⟩, ⟨Bob, 200⟩}
G. {⟨Alice, 100⟩, ⟨Bob⟩}
H. {⟨Alice, 100⟩, ⟨Alice, 200⟩}
I. {⟨Alice, 100⟩, ⟨Alice, 100⟩}

Exercício

Qual é o resultado das seguintes expressões?

A. $\{\langle \text{Alice}, 100\rangle, \langle \text{Bob}, 200\rangle\} \cup \{\langle \text{Alice}, 100\rangle\}$

B. $\{\langle \text{Alice}, 100\rangle, \langle \text{Bob}, 200\rangle\} \cap \{\langle \text{Alice}, 200\rangle, \langle \text{Alice}, 100\rangle\}$

Base de Dados Relacional

- Uma base de dados relacional é um conjunto de relações
- Um esquema de base de dados relacional é o conjunto dos esquemas relacionais (ou cabeçalhos) de todas as relações na base de dados
- Esquemas relacionais podem incluir restrições de integridade, que restringem o domínio de dados da relação

Restrições de Integridade

Os tipos de restrições de integridade contemplados no modelo relacional são:

- Restrições de domínio (ou especificações de tipo de dados)
- Restrições de chave e unicidade
- Restrições de integridade referencial (ou de chave estrangeira)

Restrições de Domínio

- Indicações de tipo de dados (string, int, etc) são frequentemente omitidas na declaração do esquema relacional
  - São posteriormente declarados na implementação em SQL
- Deve no entanto declarar-se restrições de domínio que transcendam o tipo de dados, e.g.:

```
account(acct_num, balance, branch_name)
    (balance >= 0)
```

Restrições de Chave e Unicidade

- **Chave candidata**: um conjunto mínimo (*minimal*) de Atributos que identifica univocamente cada instância de uma Relação
  - Pode haver vários (conjuntos de) Atributos que cumprem o critério
  - A **chave primária** é escolhida de entre eles e representada sublinhando os Atributos que dela fazem parte
    - A que optimiza a referência a um tuplo; normalmente só um atributo do tipo numérico
  - As restantes chaves candidatas são declaradas como únicas

Restrições de Chave e Unicidade

- Exemplo:

```
student(snum, name, login, course)
    UNIQUE(login)
    UNIQUE(name, course)
```

  - *Name* e *course* não são individualmente chaves candidatas, portanto (*name*, *course*) é um conjunto mínimo

Restrições de Integridade Referencial

- A interligação de relações no modelo racional baseia-se no uso de atributos partilhados, e.g.:

```
account(account_number, branch_name, balance)
depositor(customer_name, account_number)
```

  - É essencial para a consistência dos dados que todos os "account_number" em "depositor" estejam também presentes em "account", e que não possa haver alterações aos dados que violem isso
  - Para isso usamos uma restrição de chave estrangeira (foreign key)

Restrições de Integridade Referencial

- Restrição de chave estrangeira (foreign key):

```
account(account_number, branch_name, balance)
depositor(customer_name, account_number)
    account_number: FK(account)
```

- Impede:
  - Remoção de linhas de “account” cujo “account_number” esteja em “depositor”
  - Adição de linhas em “depositor” cujo “account_number” não esteja em “account”
  - Actualização de valores de “account_number” apenas em “depositor” ou apenas em “account” que existam em ambas

Restrições de Integridade Referencial

- Restrições de chave estrangeira podem não ser “estrangeiras”, i.e. podem referenciar a própria tabela, e.g.:

```
student(sid, name, partner)
    partner: FK(student.sid)
```

  - Obriga a que todos os estudantes tenham um parceiro?

NULLs

- NULL indica que o valor é desconhecido ou não aplicável
- Embora não contemplado no modelo relacional original, o NULL foi adicionado mais tarde por Codd, e é uma parte integral do standard SQL
    - A lógica deixa de ser de dois valores e passa a ser de três valores (true, false, unknown): comparar NULL com qualquer coisa (mesmo outro NULL) devolve *unknown*

Restrições de Preenchimento (NOT NULL)

- Restrições de chave primária impedem NULLs (os atributos que fazem parte da chave primária têm sempre de ser preenchidos)
- Mas restrições de unicidade e de chave estrangeira não impedem NULLs
  - Um atributo que é único ou chave estrangeira pode ser NULL sem violar a restrição de integridade (útil para casos como o de auto-referência opcional de uma tabela)
- Podemos exigir o preenchimento com uma restrição NOT NULL, e.g.:

```
employee(ssn, name, did)
    did: FK(department) NOT NULL
```

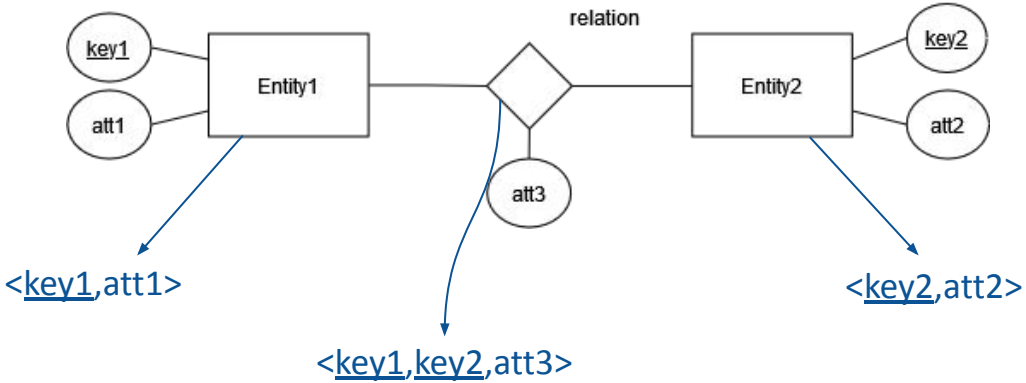
Outras Restrições

- Tal como em E-A, há restrições de integridade que não podem ser expressas no modelo relacional mas que devem ser capturadas textualmente, e.g.
  - Empregados só podem trabalhar em departamentos em que o seu superior também trabalha
  - Um aluno só se poder inscrever a uma cadeira num semestre se não tiver atingido o limite de créditos esse semestre
- Algumas destas restrições serão posteriormente implementáveis em SQL usando Assertions, Triggers e Stored Procedures (que veremos mais à frente)

TÉCNICO LISBOA

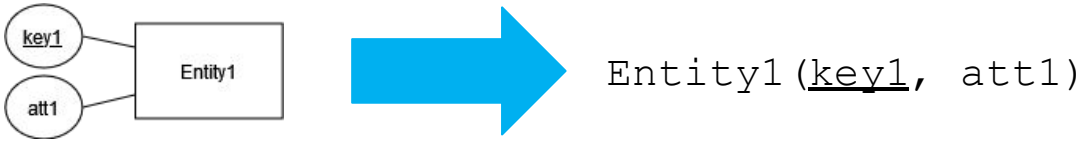
Conversão E-A–Relacional (Parte I)

Conversão E-A–Relacional



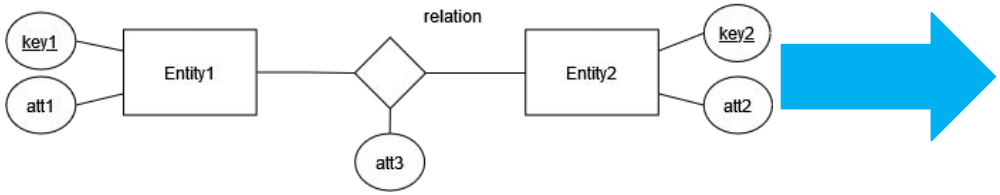
- Se entidades e associações em E-A ambos representam tuplos de atributos com chaves, não podemos convertê-los diretamente em relações?
  - Em geral, podemos, mas nem sempre é a solução ótima

Entidades



- Uma Entidade deve geralmente ser convertida diretamente numa relação com:
  - Os mesmos atributos (pelo menos)
  - A mesma chave primária
  - As mesmas declarações de unicidade

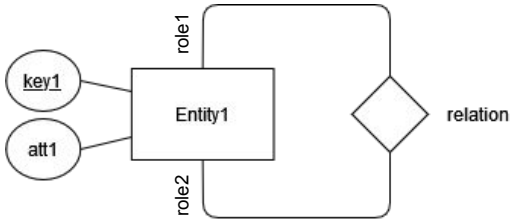
Associações




```
relation(key1,key2,att3)
   key1: FK(Entity1)
   key2: FK(Entity2)
```

- Uma Associação sem restrições de cardinalidade ou participação deve também ser convertida diretamente numa relação com:
  - A combinação de chaves primárias das entidades que liga como chave primária
  - Chaves estrangeiras para as entidades que liga
  - Atributos descritivos que tenha

Associações

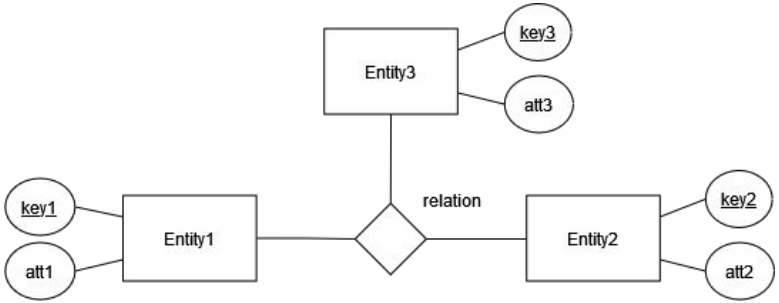



```
relation(role1,role2)
   role1: FK(Entity1.key1)
   role2: FK(Entity1.key1)
```

- O mesmo vale para auto-associações sem restrições
  - Com a ressalva de que não podemos definir dois atributos com o mesmo nome (quando há papéis na relação, o usual é usar o nome dos papéis)
  - Restrições como irreflexividade têm de ser expressas em texto, e.g.:

```
IC-1: role1 must be different from role2 in relation
```

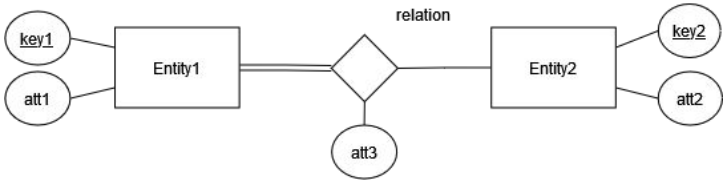
Associações




```
relation(key1,key2,key3)
    key1: FK(Entity1)
    key2: FK(Entity2)
    key3: FK(Entity3)
```

- O mesmo vale também para associações ternárias

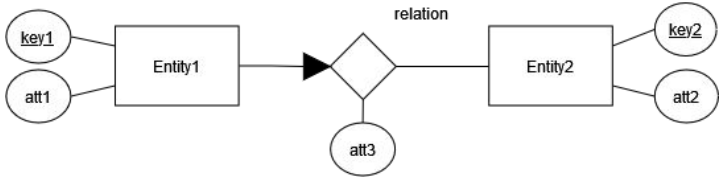
Associações Obrigatórias




```
Entity1(key1,att1)
Entity2(key2,att2)
relation(key1,key2,att3)
    key1: FK(Entity1)
    key2: FK(Entity2)
IC-1: any key1 in Entity1
must exist in relation
```

- Precisamos de uma restrição de integridade textual para caso de participação obrigatória (sem limite de cardinalidade)
  - Podíamos definir uma FK de Entity1 para relation, mas ter FKs circulares é má ideia!

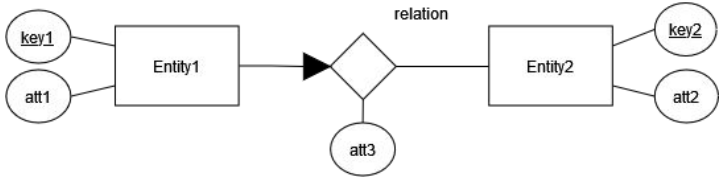
Associações 1-N Opcionais




```
Entity1(key1,att1)
Entity2(key2,att2)
relation(key1,key2,att3)
    key1: FK(Entity1)
    key2: FK(Entity2)
```

- **Opção 1:**
  - Representar a associação como relação, com **chave apenas composta pela chave da entidade cuja cardinalidade está restringida**
    - Cada key1 só pode ocorrer uma vez em relation

Associações 1-N Opcionais



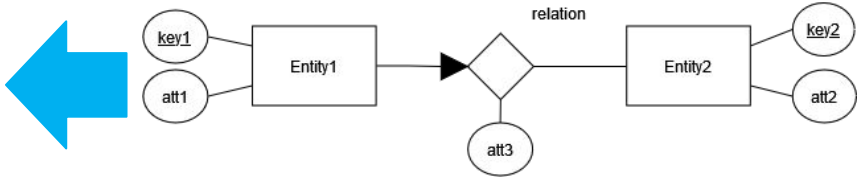

```
Entity1(key1,att1,key2,att3)
    key2: FK(Entity2)

Entity2(key2,att2)
```

- **Opção 2:**
  - Representar a associação adicionando os atributos necessários à entidade cuja cardinalidade está restringida, incluindo uma chave estrangeira para a outra entidade

Associações 1-N Opcionais

Opção 1

```
Entity1(key1,att1)
Entity2(key2,att2)
relation(key1,key2,att3)
   key1: FK(Entity1)
   key2: FK(Entity2)
```



Opção 2

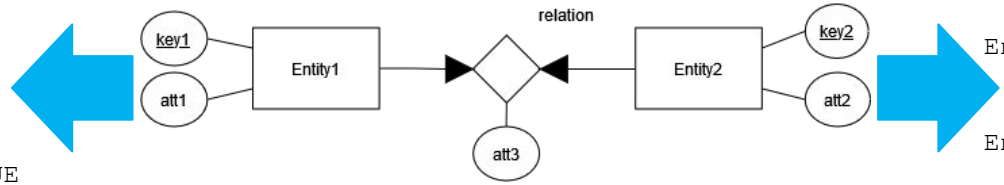
```
Entity1(key1,att1,key2,att3)
   key2: FK(Entity2)

Entity2(key2,att2)
```

- Opção 1: Evita NULLs, sendo preferível se muitas instâncias da entidade restrita não participam na associação e/ou se a associação tem muitos atributos
- Opção 2: Evita criar uma relação, sendo preferível se a maior parte das instâncias da entidade restrita participam na associação e/ou se a associação não tem atributos

Associações 1-1 Opcionais

Opção 1

```
Entity1(key1,att1)
Entity2(key2,att2)
relation(key1,key2,att3)
   key1: FK(Entity1)
   key2: FK(Entity2) UNIQUE
```

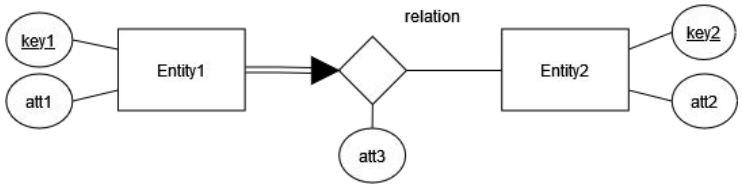


Opção 2

```
Entity1(key1,att1,key2,att3)
   key2: FK(Entity2) UNIQUE

Entity2(key2,att2)
```

- Opção 1: Semelhante a 1-N, mas podemos escolher qualquer uma das chaves estrangeiras como chave primária, e temos de declarar a outra como única
- Opção 2: Semelhante a 1-N, mas podemos incorporar a associação em qualquer das duas entidades (escolher que participa mais) e temos de declarar a chave estrangeira como única

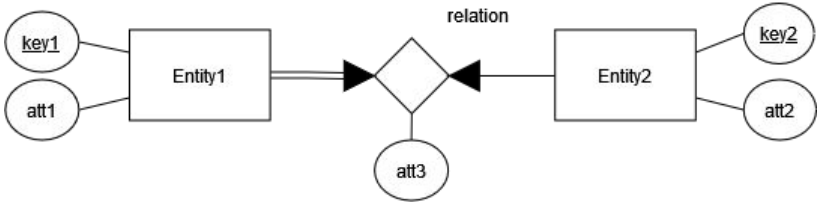
Associações 1-N Obrigatórias




```
Entity1(key1,att1,key2,att3)
    key2: FK(Entity2) NOT NULL

Entity2(key2,att2)
```

- Neste caso não há desvantagem em incorporar a associação na entidade cuja cardinalidade está restringida dado que a participação é obrigatória
- Notar que é necessário explicitar o NOT NULL

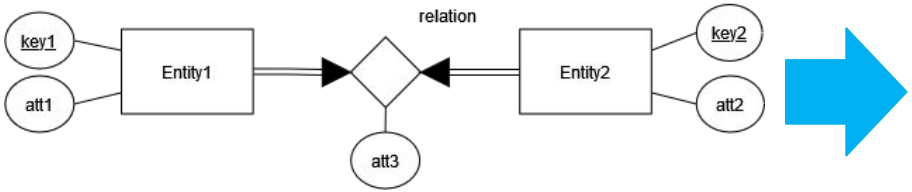
Associações 1-1 Obrigatórias




```
Entity1(key1,att1,key2,att3)
    key2: FK(Entity2) NOT NULL
    UNIQUE(key2)

Entity2(key2,att2)
```

- Igual ao caso anterior, mas temos de acrescentar unique na chave estrangeira

Associações 1-1 Obrigatórias




```
Entity1(key1,att1,key2,att3)
    key2: FK(Entity2) NOT NULL
    UNIQUE (key2)

Entity2(key2,att2)

IC-1: any key2 in Entity2 must
exist in Entity1
```

- **Opção 1:**
  - Como antes mais restrição de integridade para participação
- **Opção 2:**
  - Fundir as entidades e associação

TÉCNICO LISBOA