SIMBOLOGIA EM MODELAGEM

Profª Carmen Lúcia de B. Costa

Disciplina: Banco de Dados

A modelagem de bancos de dados é necessária para que se diminua ao mínimo possível, a possibilidade de erros e inconsistências no banco de dados.

A modelagem pode ser feita manualmente ou por meio de softwares específicos que podem utilizar a simbologia criada por James Martin o Cross Foot (pé-de galinha).

Levantamento de Dados Para se desenvolver um sistema de banco de dados, não basta apenas ter conhecimento das técnicas de modelagem. É necessário que se conheçam as regras do problema que o sistema se propõe a automatizar. Assim, a fase primordial do desenvolvimento de um sistema de banco de dados é o levantamento de dados. Nesta fase, o projetista do sistema faz a coleta das informações que julgar relevantes ao desenvolvimento do banco de dados por meio de várias técnicas, dentre as quais destacam-se questionários, análise de sistemas anteriores e entrevistas.

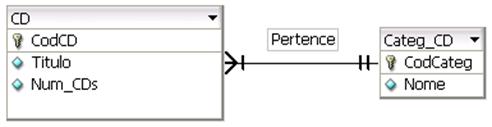
Questionários: Essa técnica tem como ponto positivo o fato de que o usuário ou cliente não tem a necessidade de interromper seus afazeres para atender ao desenvolvedor, podendo respondê-lo no momento que julgar mais apropriado; porém essa “ausência” do desenvolvedor pode fazer com que a pessoa não leve o questionário a sério, respondendo de forma displicente ou simplesmente deixando de responder parte ou totalidade das questões.

Análise de Sistemas Anteriores: Se a empresa já possui um sistema informatizado, a observação de telas de preenchimento, coleta e exibição de dados e de relatórios impressos pode dar ao desenvolvedor informações inerentes ao que se deseja para o nome banco de dados. Entrevistas Sem sombra de dúvida, as entrevistas são o meio mais eficaz de se obterem os dados relevantes ao desenvolvimento de um novo sistema, mas deve-se levar em consideração a necessidade de se entrevistar a pessoa certa. Declaração de Escopo Contém toda a especificação que o usuário deseja de como os dados e procedimentos devem ser trabalhados no sistema. Depois de ter a Declaração de Escopo devidamente avaliada e aprovada por parte dos usuários, passa-se ao processo de Modelagem do Banco de Dados do Sistema. Coad e Yourdom criaram seis regras para se abstraírem objetos de uma declaração de escopo. Uma entidade sempre é representada por um substantivo (Cliente, Produto, Fornecedor, Pedido). Assim, devem-se separar os substantivos constantes na declaração de escopo e aplicar as seguintes regras. • Informação Retida : Será uma entidade somente se existir a informação representada por ela a ser utilizada posteriormente pelo sistema. • Operações Necessárias : A entidade deve conter um conjunto de operações capazes de modificar o valor de seus atributos de alguma forma.

**Notação Crow's Foot (Pé de Galinha)**:

Esta é a notação mais amplamente utilizada dentre os arquitetos de dados. Sua implementação é feita através de retângulos representando as entidades, linhas representando os relacionamentos e traços representando a cardinalidade, onde a cardinalidade “muitos” é representada por um “tridente” aparentando um “pé de galinha”,

ex:

[](https://i.stack.imgur.com/LNkva.jpg)

Use as notações de pés de corvo para ilustrar a relação entre entidades no diagrama de pés de um corvo. As entidades são conectadas por linhas e símbolos em cada extremidade da linha descrevem a cardinalidade da relação entre as entidades.

Como eles são amplamente usados e utilizam apenas quatro símbolos, os diagramas de notação de pés de cavalo ajudam você a comunicar relações complexas em um formato fácil de usar.

Entidades fortes e fracas

* Uma *entidade forte* tem uma chave primária, o que significa que ela pode ser identificada exclusivamente por seus atributos.
* Uma *entidade fraca* é uma entidade que não pode ser identificada exclusivamente por seus atributos. Ela depende da existência de sua entidade pai. Ele deve usar uma chave estrangeira em conjunto com seus atributos para criar uma chave primária. Normalmente, a chave estrangeira é a chave primária de uma entidade forte à que a entidade fraca está relacionada.

A relação entre duas entidades fortes é simplesmente chamada de *relação*.

A relação entre uma entidade forte e uma entidade fraca é chamada de *relação forte.*

Símbolos na notação de pés de corvo – James Martin

Os diagramas de pés do corvo representam entidades como caixas e relações como linhas entre as caixas. Formas diferentes nas extremidades dessas linhas representam a cardinalidade relativa da relação.

Três símbolos são usados para representar a cardinalidade:

|  |  |  |  |
| --- | --- | --- | --- |
| DESCRIÇÃO | SÍMBOLO | Um anel representa "zero" | O símbolo do anel em Notação de Pés de Corvo. |
| Anel e traço:  Mínimo zero, máximo um (opcional) | O símbolo de Zero ou Um, na notação de pés de cavalo. | Um traço representa "um" | O símbolo de traço em Notação de Pés de Corvo. |
| Traço e traço:  Mínimo um, máximo um (obrigatório) | O símbolo de Um e Somente Um, na notação de pés de cavalo. | O pé de um corvo representa "muitos" ou "infinito" | O símbolo do pé do corvo na Notação de Pés do Corvo. |
| Anel e o pé do canto:  Zero mínimo, máximo de muitos (opcional) | O símbolo de Zero ou Mais, na notação de pés de cavalo. |  |  |
|  |  |  |  |
|  |  |  |  |

Esses símbolos são usados em pares para representar os quatro tipos de cardinalidade que uma entidade pode ter em uma relação. O elemento interno da notação representa o mínimo e o elemento externo (mais próximo da entidade) representa o máximo.

| DESCRIÇÃO | SÍMBOLO |
| --- | --- |
| Anel e traço:  Mínimo zero, máximo um (opcional) | O símbolo de Zero ou Um, na notação de pés de cavalo. |
| Traço e traço:  Mínimo um, máximo um (obrigatório) | O símbolo de Um e Somente Um, na notação de pés de cavalo. |
| Anel e o pé do canto:  Zero mínimo, máximo de muitos (opcional) | O símbolo de Zero ou Mais, na notação de pés de cavalo. |
| Traço e pé de cavalo:  Mínimo um, máximo de muitos (obrigatório) | O símbolo para Um ou Mais, na notação do pé do corvo. |

Dicas para projetar o diagrama

* Identifique todas as entidades de que você precisa. Desenhá-los todos no diagrama.
* Determine quais entidades têm relações entre si e conecte-as. (Nem todas as entidades terão relações. Alguns podem ter várias relações.)
* Cada entidade deve aparecer apenas uma vez no diagrama.

Entidades fortes e fracas

* Uma *entidade forte* tem uma chave primária, o que significa que ela pode ser identificada exclusivamente por seus atributos.
* Uma *entidade fraca* é uma entidade que não pode ser identificada exclusivamente por seus atributos. Ela depende da existência de sua entidade pai. Ele deve usar uma chave estrangeira em conjunto com seus atributos para criar uma chave primária. Normalmente, a chave estrangeira é a chave primária de uma entidade forte à que a entidade fraca está relacionada.

A relação entre duas entidades fortes é simplesmente chamada de *relação*.

A relação entre uma entidade forte e uma entidade fraca é chamada de *relação forte.*

Símbolos na notação de pés de corvo

Os diagramas de pés do corvo representam entidades como caixas e relações como linhas entre as caixas. Formas diferentes nas extremidades dessas linhas representam a cardinalidade relativa da relação.

Três símbolos são usados para representar a cardinalidade:

|  |  |
| --- | --- |
| DESCRIÇÃO | SÍMBOLO |
| Anel e traço:  Mínimo zero, máximo um (opcional) | O símbolo de Zero ou Um, na notação de pés de cavalo. |
| Traço e traço:  Mínimo um, máximo um (obrigatório) | O símbolo de Um e Somente Um, na notação de pés de cavalo. |

Esses símbolos são usados em pares para representar os quatro tipos de cardinalidade que uma entidade pode ter em uma relação. O elemento interno da notação representa o mínimo e o elemento externo (mais próximo da entidade) representa o máximo.

Dicas para projetar o diagrama

* Identifique todas as entidades de que você precisa. Desenhá-los todos no diagrama.
* Determine quais entidades têm relações entre si e conecte-as. (Nem todas as entidades terão relações. Alguns podem ter várias relações.)
* Cada entidade deve aparecer apenas uma vez no diagrama.
* Procure os relacionamentos que você desdoou. Há algum relacionamento redundante? Há algum que não seja desnecessário ou ausente?

### Cardinalidade e ordinalidade

Cardinalidade refere-se ao número máximo de vezes que a instância em uma entidade pode ser relacionada a instâncias de outra entidade. Ordinalidade, por sua vez, é o número mínimo de vezes que uma instância em uma entidade pode ser associada a uma instância em uma entidade relacionada.

A cardinalidade e a ordinalidade são representadas pelo estilo de uma linha e sua extremidade, de acordo com o estilo de notação escolhido.



### Chaves

As chaves são uma maneira de categorizar atributos. Diagramas ER ajudam usuários a modelar seus bancos de dados usando várias tabelas que garantem que o banco de dados seja organizado, eficiente e rápido. Chaves são usadas para vincular diversas tabelas em um banco de dados, umas às outras, da maneira mais eficiente possível.

#### Chaves primárias

Chaves primárias são atributos ou combinações de atributos que identificam de modo único apenas uma instância de uma entidade.

#### Chaves estrangeiras

Chaves estrangeiras são criadas toda vez que um atributo refere-se a outra entidade em um relacionamento de um para um ou de um para muitos.



Cada carro só pode ser financiado por um banco, portanto, a chave primária BancoId da tabela Banco é usada como a chave estrangeira FinanciadoPor na tabela Carro. Este BancoID pode ser usado como a chave estrangeira para vários carros.

## Notação de diagramas ER

Enquanto a notação pé de galinha é amplamente aceita como o estilo mais intuitivo, alguns usam a notação OMT, IDEF, Bachman ou UML, de acordo com suas preferências. A notação pé de galinha, no entanto, conta com um formato gráfico intuitivo, sendo a notação de diagramas ER preferida do Lucidchart.

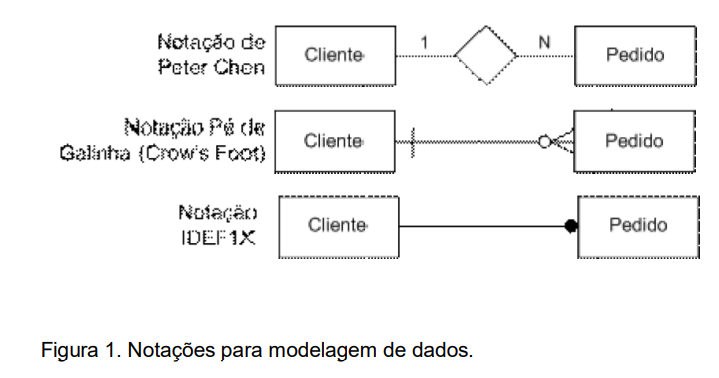
### **Tipos**

|  |  |
| --- | --- |



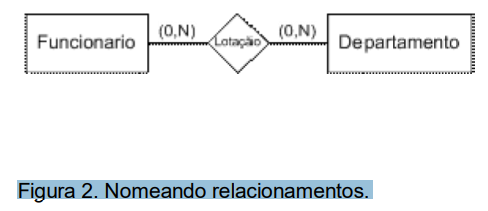
----XXXXXX------XXXXXXX-------XXXXXX--------XXXX

Modelagem de dados: projeto conceitual 1. Sempre faça modelagem de dados, isso ajuda no entendimento do problema e no planejamento de uma solução mais aderente aos seus objetivos. O objetivo do modelo conceitual é a definição do problema, não da solução. 2. Existem diferentes terminologias para modelagem conceitual. Os exemplos da Figura 1 representam diferentes notações para um relacionamento 1:N, sendo as mais comuns de serem implementadas pelas atuais ferramentas CASE. A primeira notação, de Peter Chen, é a mais utilizada para a construção do modelo conceitual enquanto a segunda, do Pé de Galinha, é a mais utilizada para a construção do modelo lógico.

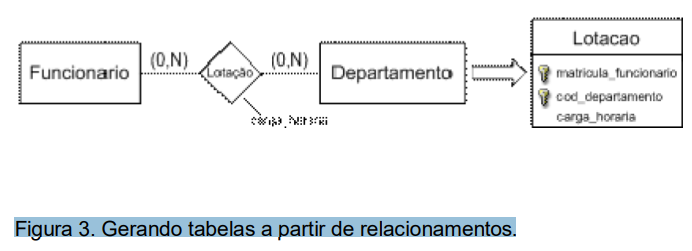


3. Elimine qualquer redundância de dados. Redundâncias permitidas são aquelas relativas a chaves estrangeiras, que fazem referência à chave primária de outra tabela. Por exemplo, não se deve repetir o nome do cliente em seus pedidos, pois ele pode ser facilmente obtido através do relacionamento. 4. Utilize um padrão para dar nomes a entidades. Normalmente, nomes de entidades são no singular. 5. Dê nomes significativos a entidades, atributos e relacionamentos. Nomes que não representam seu real objetivo dificultam a compreensão do modelo.

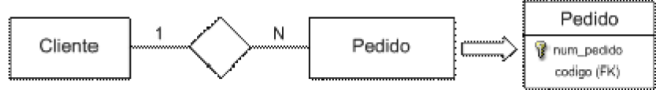
6. Deve-se determinar os relacionamentos e, decidir como é a relação de dependência entre cada entidade é sempre importante. Os tipos de relacionamento podem ser: · 0:1 (mínimo: nenhum – máximo: um); · 0:N (mínimo: nenhum – máximo: muitos); · 1:1 (mínimo: um – máximo: um); · 1:N (mínimo: um – máximo; muitos); · N:N (mínimo: muitos – máximo: muitos). 7. Relacionamentos não obrigatoriamente precisam ter nomes, mas é uma boa prática de modelagem nomeá-los, pois facilita o entendimento. Assim, utilize nomes significativos para os relacionamentos, como apresentado na Figura 2 para o relacionamento Lotação. Figura 2. Nomeando relacionamentos. 8. Relacionamentos N:N ou que possuem atributos normalmente geram novas tabelas no modelo lógico. O nome do relacionamento pode ser utilizado como nome da tabela e deve ser cuidadosamente escolhido, como exemplificado na Figura 3. Figura 3. Gerando tabelas a partir de relacionamentos.



8. Relacionamentos N:N ou que possuem atributos normalmente geram novas tabelas no modelo lógico. O nome do relacionamento pode ser utilizado como nome da tabela e deve ser cuidadosamente escolhido, como exemplificado na Figura 3.



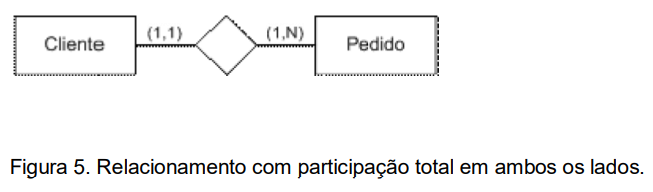
9. Ao dar nomes a atributos determinantes, coloque sempre algo que identifique a entidade que está sendo relacionada. Por exemplo, um atributo chave cod\_cliente é melhor do que simplesmente código. Lembre-se que, no modelo lógico, este atributo será repetido como chave estrangeira nas entidades relacionadas. Percebe-se na Figura 4 que, se o nome da chave primária da entidade Cliente for apenas codigo, quando repetida na entidade Pedido para representar o relacionamento, seu nome não será representativo, ou seja, não se pode observar facilmente a qual entidade está associado. Se o atributo tivesse sido chamado de cod\_cliente, esta associação seria muito mais fácil. Este problema seria mais grave se a entidade Pedido estivesse associada à outra cujo nome de sua chave primária também fosse codigo. Teríamos mais de um atributo com o mesmo nome, que as ferramentas CASE costumam modificar atribuindo uma seqüência numérica, o que dificulta enormemente a compreensão do modelo. 3 Figura 4. Nomeando atributos chave. 10. Procure padronizar os nomes dos atributos de seu diagrama entidade relacionamento. Por exemplo, ao ver o atributo num\_pedido percebe-se facilmente que seu tipo é numérico. A padronização garante um bom entendimento dos atributos perante a equipe de desenvolvimento, garantindo eficiência na fase de desenvolvimento da aplicação. 11. Atenção para entidades desconectadas no diagrama. Podem não ser um problema, mas precisam ser verificadas. Por exemplo, é comum que entidades que representem parâmetros estejam soltas no diagrama. 12. Cuidado com relacionamentos com participação total em ambos os lados de um relacionamento (obrigatoriedade dos dois lados). Isso pode provocar uma situação onde uma entidade dependa da outra recursivamente. Além disso, deve-se estar atentos a situações onde o banco de dados ainda está vazio e precisa-se cadastrar registros em apenas uma das entidades. Observe o exemplo abaixo da Figura 5. Um Cliente possui um ou mais Pedidos e um Pedido é obrigatoriamente de um Cliente. Devido à obrigatoriedade em ambos os lados do relacionamento, esta situação impede que Clientes sejam cadastrados sem Pedidos e que Pedidos sejam feitos sem Clientes.



10. Procure padronizar os nomes dos atributos de seu diagrama entidade relacionamento. Por exemplo, ao ver o atributo num\_pedido percebe-se facilmente que seu tipo é numérico. A padronização garante um bom entendimento dos atributos perante a equipe de desenvolvimento, garantindo eficiência na fase de desenvolvimento da aplicação.

11. Atenção para entidades desconectadas no diagrama. Podem não ser um problema, mas precisam ser verificadas. Por exemplo, é comum que entidades que representem parâmetros estejam soltas no diagrama.

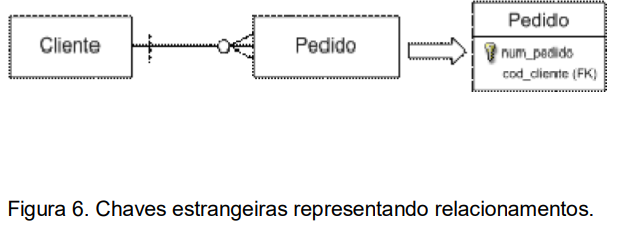
12. Cuidado com relacionamentos com participação total em ambos os lados de um relacionamento (obrigatoriedade dos dois lados). Isso pode provocar uma situação onde uma entidade dependa da outra recursivamente. Além disso, deve-se estar atentos a situações onde o banco de dados ainda está vazio e precisa-se cadastrar registros em apenas uma das entidades. Observe o exemplo abaixo da Figura 5. Um Cliente possui um ou mais Pedidos e um Pedido é obrigatoriamente de um Cliente. Devido à obrigatoriedade em ambos os lados do relacionamento, esta situação impede que Clientes sejam cadastrados sem Pedidos e que Pedidos sejam feitos sem Clientes.

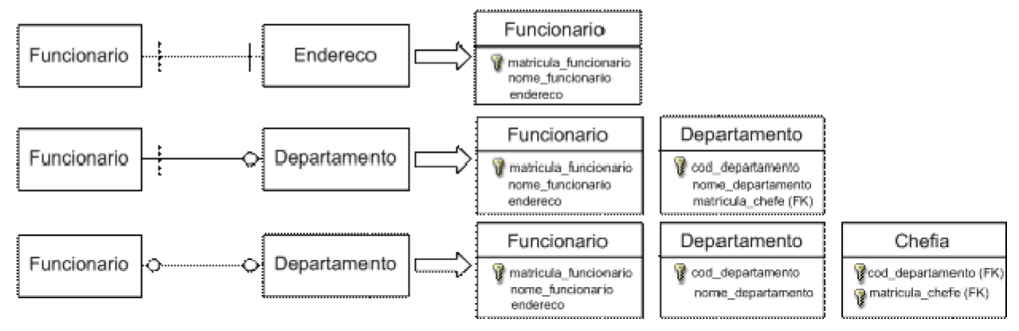


Modelagem de dados: projeto lógico 13. Muitas vezes costuma-se “pular” a etapa de modelo conceitual de dados e passa-se diretamente para o modelo lógico. Em grande parte pode-se creditar isso a muitas ferramentas CASE, que partem diretamente do modelo lógico. Considera-se isso perigoso. A modelagem conceitual foi criada para atender às necessidades do usuário, sem limitações tecnológicas. Desta forma, no primeiro modelo, não se deve preocupar com chaves primárias e estrangeiras. As chaves são necessárias uma vez que é assim que os SGBDs trabalham, mas isso não faz parte das necessidades do usuário. O mesmo pode ser dito com relação ao tipo e tamanho de dados, e ainda mais do mapeamento de relacionamentos 1:1 e n:n. O objetivo do projeto lógico é a definição da solução. 14. Vale lembrar que nem todo modelo lógico de dados é a cópia fiel do modelo conceitual de dados. 15. Toda entidade do modelo conceitual vira uma tabela no modelo lógico. 16. Deve-se relacionar diretamente cada coluna ao escopo da tabela: se um campo descreve o assunto de uma tabela diferente, este campo deve pertencer à outra tabela. 17. Elimine colunas repetidas no modelo: se uma informação se repete em diversas tabelas, é um indício de que existem colunas desnecessárias. Deve-se buscar a tabela que faça mais sentido para conter a coluna em questão.

18. Dimensione corretamente os tipos de dados em função de seus conteúdos para diminuir o espaço de armazenamento. 19. Campos de texto com tamanho variável tendem a consumir menos espaço de armazenamento. 20. Defina corretamente a obrigatoriedade para atributos das entidades de forma a retratar o objetivo da entidade. Por exemplo, o nome do cliente deve ser obrigatório, pois não faz sentido cadastrar um cliente sem seu nome. Muitas vezes, preocupase apenas com a obrigatoriedade para os atributos chave, mas esta questão é importante para todos os atributos, tomandose o devido cuidado de não impor restrições demais que impeçam que novos registros possam ser inseridos. 21. Elimine atributos derivados ou calculados: não é recomendado armazenar o resultado de cálculos nas tabelas. O correto é que o cálculo seja gerado sob demanda, normalmente em uma consulta. 22. Toda tabela deve ter uma chave primária, que pode ser simples ou composta. A chave primária é o identificador do registro e deve ser única dentro de uma tabela. 23. Ao determinar a chave primária, deve-se escolher, em cada tabela, quais colunas formarão a chave primária. Para uma coluna ser candidata à chave primária, deverá atender aos principais requisitos: · Deverá ser a menor possível; · O seu valor deverá ser diferente de vazio ou zero (not null); · Deverá ser de preferência numérica; · O seu valor deverá ser único para toda a tabela. 24. Chaves estrangeiras devem corresponder a chaves primárias da relação associada ou ser nulas, quando não for um campo obrigatório.

25. Crie chaves cegas (Blind Key) toda vez que não for possível identificar a chave primária, ou quando esta for muito complexa, composta por muitos atributos. Esses tipos de atributos geralmente são conhecidos por atributos falsos, ou seja, não fazem parte de forma explícita da regra de negócio, porém foram criados para garantir flexibilidade ou integridade ao modelo de dados desenvolvido. Por exemplo, um funcionário pode ter diversos dependentes, que possuem nome e data de nascimento. Nenhum destes atributos, nem mesmo a concatenação deles, garante uma chave primária única. Uma solução é a criação de um novo atributo determinante, como cod\_dependente. 26. Refine os relacionamentos uma vez que alguns destes poderão gerar novos elementos que complementarão o modelo, sendo estes: · Relação de Herança (mutuamente exclusivo ou não): por exemplo, numa hierarquia de Funcionario contendo especializações Gerente e Engenheiro, um determinado funcionário pode ser Gerente e Engenheiro ao mesmo tempo?; · Entidade Associativa: gerada quando informações que necessitam ser armazenadas não podem ser colocadas numa das entidades do relacionamento. Acontece em situações de relacionamentos com atributos ou relacionamentos N:N. 27. Relacionamentos são representados por chaves estrangeiras, ou Foreign Key (FK), atributos correspondentes à chave primária de outra relação, estabelecendo a base para a integridade referencial. No exemplo da Figura 6, o atributo cod\_cliente, chave primária da tabela Cliente, foi repetido na tabela Pedido para representar o relacionamento. Desta forma, um Pedido possui um cliente que necessariamente deve estar presente na tabela Cliente (através de seu código).





| DESCRIÇÃO | SÍMBOLO |
| --- | --- |
| Anel e traço:  Mínimo zero, máximo um (opcional) | O símbolo de Zero ou Um, na notação de pés de cavalo. |
| Traço e traço:  Mínimo um, máximo um (obrigatório) | O símbolo de Um e Somente Um, na notação de pés de cavalo. |
| Anel e o pé do canto:  Zero mínimo, máximo de muitos (opcional) | O símbolo de Zero ou Mais, na notação de pés de cavalo. |
| Traço e pé de cavalo:  Mínimo um, máximo de muitos (obrigatório) | O símbolo para Um ou Mais, na notação do pé do corvo. |