



WEBAULA14

- Projeto Físico de Dados
- Referências

QUESTION BANKS

Projeto Físico de Dados

Ao longo da história, foram feitas pesquisas e, dentre as muitas propostas, destacam-se métodos que nos auxiliam a uma estrutura concisa com maior correção e facilidade para lidar com os dados. Para a representação formal dos dados, dentre os modelos propostos, tem destaque o modelo entidade-relacionamento, que é um padrão de modelagem conceitual e ponto de referência para as propostas de modelagem de objetos hoje, como a Unified Modeling Language (UML).

O modelo entidade-relacionamento (MER), também conhecido como diagrama entidaderelacionamento (DER), foi criado por Peter Chen em 1976 (Cardoso, 2008). Peter Chen, por sinal, é uma das formas de representação gráfica bastante comum nas literaturas para ilustrar uma modelagem conceitual.

Atualmente, utiliza-se esse modelo, ou notação gráfica (Peter Chen) como o primeiro passo para a representação da estrutura de um banco de dados. Projetos que excluem esse processo apresentam muitos erros e falhas, que, ao longo de seu desenvolvimento e aplicação, causam inúmeros problemas. É necessário estar atento e não deixar de utilizar esse método, garantindo, assim, uma estrutura sólida, segura e com menor possibilidade de erros no projeto, seja de pequeno, médio ou grande porte (Cardoso, 2008). A partir do MER é possível uma representação da estrutura lógica do projeto a partir de uma visão genérica. Sua estrutura é feita de forma clara e simples, possibilitando representar os dados do mundo real (minimundo) como objetos denominados "entidade" ligados uns aos outros (objetos) por meio de associações denominadas "relacionamento". Daí a origem no nome "entidade-relacionamento".

Agora que você conhece o seu conceito, porque ela deve ser utilizada e o cenário no qual o MER está inserida, é hora de conhecer os elementos básicos que a constituem.

O MER é constituído por:

- Entidades
- Atributos
- Relacionamentos

Antes de abordarmos mais detalhadamente cada um desses elementos básicos, você deve saber como se dá a representação gráfica de cada um deles. Existem muitas notações para construção de MER. Como já citada anteriormente, a notação original foi proposta por Peter Chen e é composta de entidades (retângulos), relacionamentos (losangos), atributos (círculos) e linhas de conexão (linhas) que indicam a ligação de uma entidade em um relacionamento, assim como a associação dos atributos com a entidade. Observe um exemplo na figura 5:

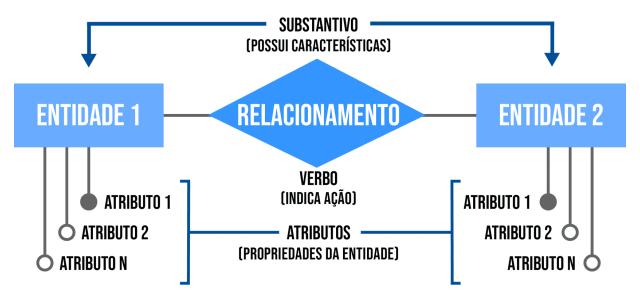


Figura 5: Representação gráfica dos elementos do MER

Vejamos um exemplo: vamos considerar um cenário onde estão envolvidos sócios e títulos de um clube social e esportivo. No exemplo, um sócio é proprietário de um título. Veja como seria a representação gráfica do cenário apresentado na figura 6.

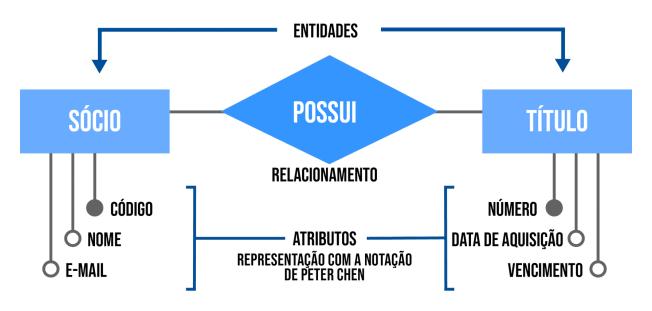


Figura 6: Exemplo de MER

No exemplo apresentado podemos observar a presenta dos elementos que compõe o MER. A seguir vamos explorar em detalhes cada um deles.

Os objetos ou "coisas" sobre os quais precisamos guardar dados são chamados de entidades. Quando analisamos um problema devemos ficar atentos aos substantivos e nomes que aparecem. Eles podem vir a ser entidades no nosso modelo.

Não considere isso uma regra. Na verdade, torna-se complicado afirmar a existência de "regras" para a definição das entidades, visto que cada cenário (minimundo) pode apresentar particularidades sensíveis ao contexto. No entanto, um bom começo para avaliar cada substantivo ou nome que julgamos ser uma entidade é fazer as seguintes perguntas:

- Há necessidade de guardar informações sobre cada objeto?
- Há mais de um objeto desse tipo?
- É possível identificar cada objeto unicamente?

Voltemos ao cenário ilustrado na figura 6 como exemplo. O cenário envolve um sócio de um clube. Agora vamos responder às perguntas sugeridas:

Há necessidade de guardar informações sobre cada objeto?

Sim!

Há mais de um objeto desse tipo?

Sim!

É possível identificar cada objeto unicamente?

Sim, através do número, por exemplo.

Diante das respostas positivas obtidas, podemos afirmar que sócio é de fato uma entidade que irá compor o modelo. Em caso de uma das perguntas apresentar uma resposta negativa, podemos desconsiderá-la como entidade.

(i) Reflita

Observe que até o momento não estamos dando ênfase às propriedades que caracterizam as entidades. A razão é simples: identificar as entidades é um processo considerado mais complexo. Mas não se preocupe, tão logo você saiba mais sobre atributos, isso também fará parte dos seus estudos.

O próximo passo é conhecer alguns conceitos envolvidos com as entidades. Os mais comumente utilizados são: Especialização, Generalização, Entidade fraca e Entidade forte. Para ajudá-lo(a) a entender melhor esses conceitos vamos mostrar a seguir sua aplicação através de exemplos.

Fazemos uma especialização quando existem atributos que só se aplicam a um subconjunto da entidade. A melhor forma de compreender esse conceito é através de um exemplo. Observe a figura 7.

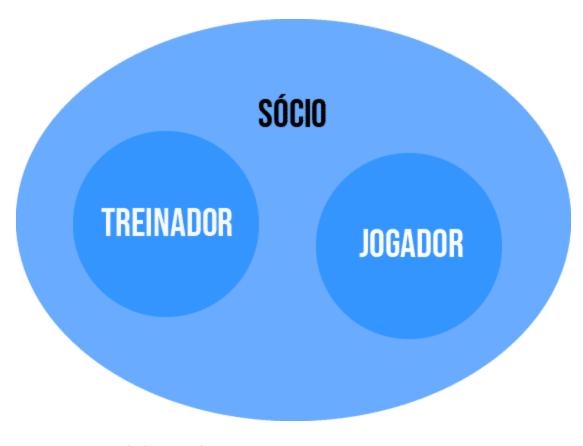


Figura 7: Exemplo de especialização

O clube organiza torneios de voleibol misto. Mas para participar, seja como jogador ou técnico, é preciso ser sócio. Logo temos: um jogador é um sócio, assim como um técnico também. Portanto, uma entidade é considerada subentidade de outra se a primeira é um conjunto da segunda entidade. Ao criar uma subentidade, fazemos uma especialização.

Há também o processo inverso, no qual, ao examinarmos duas ou mais entidades, descobrimos que vários de seus atributos são comuns. Dizemos então que essas entidades são subconjuntos da mesma entidade. Esse processo é conhecido como generalização. Considere o seguinte exemplo ilustrado na figura 8.

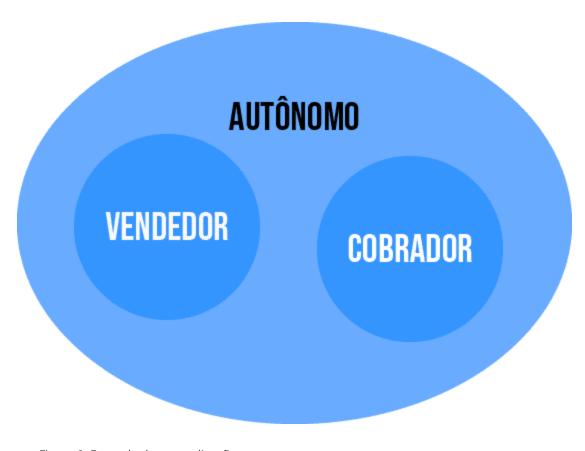
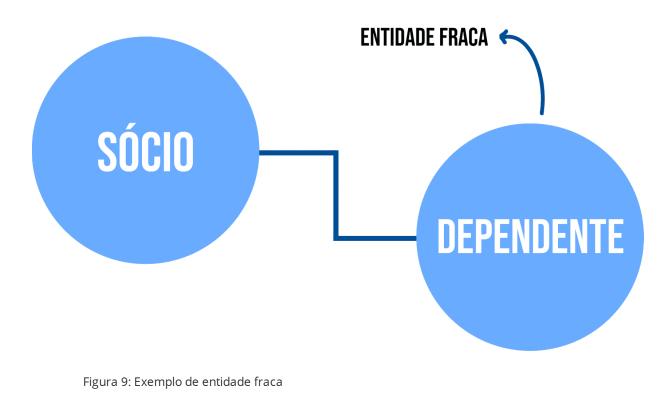


Figura 8: Exemplo de generalização

No exemplo temos: "autônomo" é uma entidade generalizada de vendedor e cobrador. Portanto, uma entidade generalizada contempla atributos comuns a duas ou mais entidades - estas, por sua vez, consideradas subentidades.

Consideramos uma **entidade fraca** quando ela depende de outra para sua existência. Para exemplificar, voltemos ao contexto do clube do qual você é sócio. Porém, agora, considere que você

tenha dependentes, os quais também frequentam o clube. Podemos representar isso por meio da figura 9.



A identificação do dependente (propriedades) está associada à identificação do sócio, obrigatoriamente. Portanto, dependente é uma entidade fraca, pois **depende da existência do sócio, uma vez que não existe dependente sem um sócio**.

Você acabou de conhecer o conceito de entidade fraca. Toda vez que temos uma entidade fraca temos também uma entidade forte. No exemplo utilizado (figura 10), que envolvia sócio e dependentes, a entidade "sócio" executa o papel de entidade forte, conforme podemos observar:

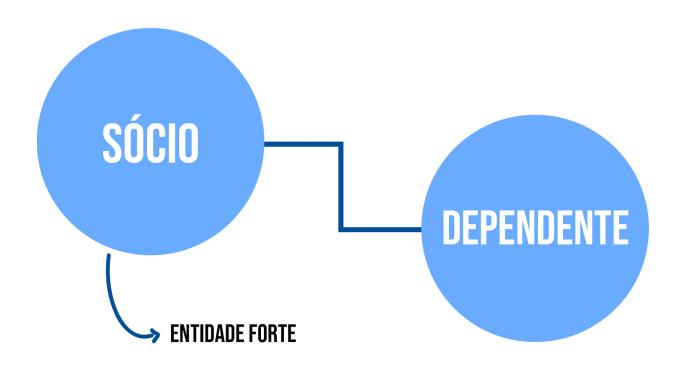


Figura 10: Exemplo de entidade forte

Sempre que existir uma dependência, uma relação entre as entidades envolvidas deverá ser representada. Estamos falando do mesmo contexto envolvido com a entidade fraca, ou seja, dentre as propriedades do dependente haverá uma identificação do sócio, obrigatoriamente. Portanto, sócio é uma entidade forte, pois sem a sua existência não há possibilidade da ocorrência de dependentes.

Agora vamos falar sobre os atributos. Os atributos são as propriedades das entidades. Para Machado (2014), os atributos representam propriedades elementares de uma entidade ou relacionamento. Cada

atributo está associado a um domínio particular, que é um conjunto de valores válidos para o atributo. Podemos afirmar também que os atributos caracterizam as entidades, dando sentido à sua existência.

Na MER a denominação é "atributo". Em breve, após conhecer o modelo físico de dados e, por consequência, a efetivação das estruturas projetadas em um SGBD, você vai descobrir que "atributo" é sinônimo de "coluna" ou "campo" em uma estrutura de banco de dados. Mas por ora vamos continuar a utilizar o termo **atributo**.



(i) Dica

Não há entidade sem atributo. Mesmo entidades com um único atributo são consideradas entidades. Mas atenção, é muito raro a existência de entidade com apenas um atributo.

Vejamos alguns exemplos de atributos. Para a entidade "sócio", entre outros, temos:

- número do sócio
- data de nascimento
- endereço de e-mail

Os atributos podem ser classificados de acordo com suas características. Alguns, por sua vez, podem ser classificados em mais de um tipo. Isso pouco vai influenciar o processo de modelagem de dados. Mesmo assim, vamos conhecer os principais tipos de atributos:

Atributo simples: É o mais comum. Não possui qualquer característica especial. Ex.: gênero, data de nascimento, etc.

- Atributo composto: Pode ser dividido em partes. Ex.: nome (prenome, nome intermediário, sobrenome), endereço (logradouro, tipo de logradouro, bairro, cidade, estado e CEP), etc.
- Atributo derivado: Derivado de operações envolvendo outros atributos. Ex.: subtotal do item NF, total da NF, etc.
- **Atributo determinante:** Valor único para cada atributo instanciado. É sério candidato a chave primária da tabela. Ex.: código de identificação, número do CPF, número do PIS.

Os objetos do mundo real (entidades) não existem sozinhos. Eles se associam uns aos outros. Essa associação é denominada relacionamento.



A resposta é simples: os relacionamentos representados no MER serão responsáveis pelas regras de integridade referencial das estruturas propostas e, consequentemente, exercerão um papel importante na garantia da consistência dos dados que serão armazenados.

A importância dos relacionamentos estabelecidos no MER não pode ser avaliada simplesmente a partir da explanação acima. Você vai conhecer na prática a sua importância a partir do momento em que as estruturas conceituais são efetivadas juntamente com as regras (relacionamentos) e os dados passam a ser manipulados.

Antes de iniciarmos na exploração dos tipos de relacionamentos existentes no MER é fundamental que resgatemos dois conceitos já apresentados no primeiro ciclo de nossa disciplina: chave primária e chave estrangeira.

O papel das chaves: a chave primária e a chave estrangeira são fundamentais para o modelo relacional. A partir deste ponto em que vamos discutir projetos de banco de dados relacionais, esses conceitos serão frequentemente utilizados. Por isso, vejamos:

- Chave primária: é a combinação de um ou mais atributos que identificam unicamente uma tabela. O valor deve ser único para cada registro (linha).
- Chave estrangeira: é a combinação de um ou mais atributos que referenciam outra tabela. Possui o valor da chave primária da tabela referenciada.

(i) Reflita

Com esses conceitos é possível afirmar que a chave primária e a chave estrangeira são os elementos que estabelecem o relacionamento entre as entidades. Lembre-se disso toda vez que falarmos em chave, seja ela primária ou estrangeira.

Muito bem, resgatamos o conceito de chaves e possivelmente você ainda não está convencido do porquê precisamos projetar as estruturas de dados? Pois bem, o dado é parte fundamental em um sistema computacional. Uma estrutura de dados planejada e aperfeiçoada permite o(a) analista/projetista desenhar os processos, a interface do usuário, os relatórios e demais recursos relacionados ao desenvolvimento da aplicação sempre que necessário. Portanto, projetar as estruturas de armazenamento dos dados é um passo fundamental no desenvolvimento de sistemas computacionais, além de favorecer o entendimento do problema a ser solucionado.

É chegado o momento de tratarmos dos tipos de relacionamentos e as cardinalidades (ou conectividades). Os relacionamentos apresentam características e tipologias. Basicamente existem três tipos de relacionamentos. O primeiro, e mais comum, é conhecido como binário, pois envolve exclusivamente duas entidades. Veja a representação gráfica na figura 11.



Figura 11: Exemplo de relacionamento binário

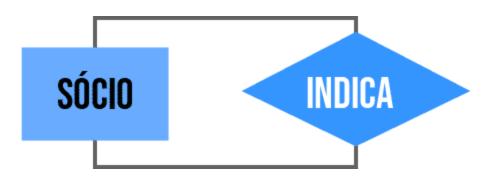


Figura 12: Exemplo de relacionamento recursivo binário

O segundo é conhecido como recursivo binário ou autorrelacionamento, pois envolve uma entidade, sendo que esta exerce o papel de duas. Veja a representação gráfica da figura 12.

O terceiro grupo de relacionamento é caracterizado pela ocorrência de três entidades, portanto, é conhecido como ternário. Não é muito comum sua ocorrência e, portanto, para exemplificar vamos considerar o cenário no qual uma empresa tem funcionários que atuam em diferentes projetos com habilidades específicas a cada projeto. Algo assim: João trabalha no projeto A como programador de sistemas, e no projeto B como analista de qualidade. Agora veja e representação da figura 13.

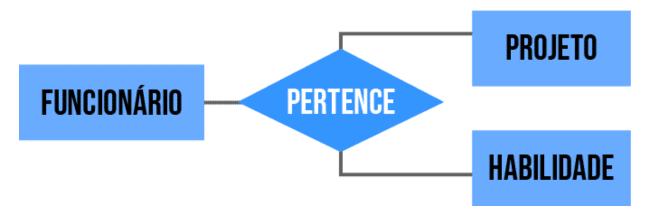


Figura 13: Exemplo de relacionamento ternário



O relacionamento ternário apresenta uma particularidade. Tão logo você conheça os tipos de conectividades (ou cardinalidades) dos relacionamentos, será mais fácil seu entendimento e aplicação. Aguarde, estamos chegando lá!

Vamos agora conhecer os três tipos de conectividades, ou cardinalidades, termo utilizado em algumas literaturas. Os tipos de conectividades representam a forma como uma entidade se associa a outra. Observe na figura 14 como se aplica cada um dos tipos.

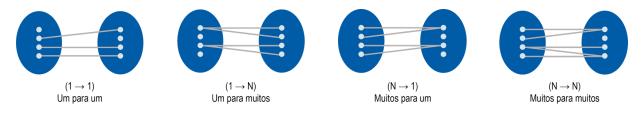


Figura 14: Tipos de conectividades

(i) Dica

O tipo de conectividade um-para-muitos e muitos-para-um são equivalentes. Isso porque a diferença entre eles limita-se a posição dos conjuntos de dados (entidades) na representação gráfica. Trataremos ambos como sendo um-para-muitos.

Agora que conhecemos os tipos de conectividades, vamos entender como aplica-los. Mas antes mesmo de convidar você a conhecer os tipos de conectividades, vamos entender o como é o processo investigativo para descobrirmos qual o tipo que se aplica a cada associação. Podemos utilizar a seguinte expressão:

Um(a) [ENTIDADE A] [RELACIONAMENTO] [preposição] **quantos(as)** [ENTIDADE B]?

Onde:

ENTIDADE A → coisa/objeto que faz o papel do sujeito na oração;

RELACIONAMENTO → verbo que indica a ação;

Preposição → estabelece a conexão (ex. de, a, com, por, para, até, em);

ENTIDADE B → coisa/objeto que sofre a ação (relacionamento) provocada pelo sujeito.

É necessário que você faça a substituição dos elementos entre colchetes e em seguida realizar o questionamento. É possível que pequenas adequações sejam necessárias em razão dos verbos. Faça sem problemas. Fique certo de que isto não irá interferir no resultado de sua análise.



(i) Dica

A resposta ao questionamento deverá sempre ser adicionada ao lado da entidade que sofreu a ação (pergunta)!

A seguir vamos exemplificar o uso da expressão acima. Antes, é importante destacar que inicialmente iremos utilizar a notação gráfica do Peter Chen para representarmos o MER. Oportunamente falaremos mais sobre outros tipos de notações gráficas.

Considere parte de um cenário em que temos funcionário e cônjuge. A figura 15 ilustra como seria a representação gráfica para um exemplo de conectividade um-para-um.





Vejamos a aplicação da expressão que você aprendeu anteriormente:

Um(a) [ENTIDADE A] [RELACIONAMENTO] [preposição] quantos(as) [ENTIDADE B]?

Substituindo os termos:

Um FUNCIONÁRIO é CASADO com quantos CÔNJUGES?

Inverte-se a pergunta:

Um CÔNJUGE é CASADO com quantos FUNCIONÁRIOS?

A reposta de cada uma das questões feitas deve ser atribuída no lado da entidade que sofre a ação, ou seja, que foi questionada. Assim, temos um exemplo de conectividade um-para-um.



Você está lembrado do tipo de relacionamento recursivo binário? Pois bem, aqui é cenário em que poderíamos avaliar a possibilidade de utilizarmos. Para isso teríamos que considerar que funcionário e cônjuge poderiam estar caracterizados como pessoa e, neste caso, seria possível a utilização deste tipo de relacionamento. Fica a dica. Iremos discutir isso nas atividades práticas.

Este tipo de conectividade (um-para-um) é o menos comum dentre os três que você irá conhecer. Em geral é utilizado quando uma ocorrência de uma entidade equivalente a um registro em uma tabela, se relaciona uma única vez com uma ocorrência de outra entidade. O exemplo está indicando que um funcionário estará relacionado com apenas um cônjuge, assim como um cônjuge também estará relacionada apenas a um funcionário. Nesse tipo de conectividade a ligação (chave) poderá ser adicionado a qualquer um dos lados, porém é fundamental o bom senso. No exemplo acima é recomendado adicionar nas propriedades da entidade cônjuge.



A sugestão de incluir da chave de ligação nas propriedades do cônjuge se justifica pelo fato de que, não necessariamente, todo funcionário terá um cônjuge. Porém, todo cônjuge, uma vez que é cônjuge, terá um funcionário associado.

Passamos agora a conectividade um-para-muitos. O tipo de conectividade um-para-muitos é o mais utilizado entre entidades de um MER. Em geral, é utilizado quando um registro de uma entidade se relaciona mais de uma vez com registros de outras entidades. A figura 16 apresenta parte de um cenário onde temos departamentos e funcionários.



Figura 16: Exemplo de conectividade um-para-muitos

Vejamos a aplicação da expressão que você aprendeu anteriormente:

Um(a) [ENTIDADE A] [RELACIONAMENTO] [preposição] quantos(as) [ENTIDADE B]?

Substituindo os termos:

Um DEPARTAMENTO lota quantos FUNCIONÁRIOS?

Inverte-se a pergunta:

Um FUNCIONÁRIO é lotado em quantos DEPARTAMENTOS?



Se invertemos a posição das entidades, ou seja, à esquerda funcionário e a direita departamento, temos o exemplo transformado em muitos-para-um. Portanto, não há diferença entre um-para-muitos e muitos-para-um, exceto o fato da posição gráfica deles. Assim, o exemplo utilizado é adequado a ambos os casos.

Da mesma forma, a reposta de cada uma das questões feitas deve ser atribuída no lado da entidade que sofre a ação, ou seja, que foi questionada. Ao substituirmos os termos temos, um departamento lota quantos funcionários? A resposta é muitos, portanto devemos colocar o "n" no lado do funcionário. Um funcionário é lotado em quantos departamentos? A resposta é "1" (um), portanto devemos colocar o "1" no lado do departamento. Veja que nessa pergunta é desprezado o fato de conter um "n" que indica muitos ao lado do funcionário, ou seja a pergunta investigativa independe da resposta anterior. Assim, temos um exemplo de conectividade um-para-muitos.

É importante destacar que houve a necessidade de adequar o uso do verbo para que a pergunta tivesse mais sentido. Isso é comum em razão do uso da linguagem natural no exercício da atividade.

Em se tratando da ligação (chave) temos que neste tipo de conectividade a regra é precisa e objetiva. A chave deverá ser adicionada ao lado "n" (muitos) do relacionamento. No caso do exemplo

apresentado, estamos falando de adicionar uma chave nas propriedades da entidade "funcionário".



(i) Reflita

No exemplo utilizado, você imagina por que devemos incluir a chave de ligação no lado do funcionário? Vejamos. Como o departamento pode lotar mais de um funcionário, se tivermos que criar chaves nas propriedades da entidade departamento quantas seriam? Uma, duas, três, cinco, não sabemos, pois há um indicativo de que pode ser "n" (muitos). Já ao considerarmos que um funcionário está lotado em apenas "1" (um) departamento, logo fica fácil estabelecer a ligação entre as entidades acrescentando uma chave nas propriedades da entidade funcionário.

Por fim, temos a conectividade muitos-para-muitos. O tipo muitos-para-muitos é comum em um MER. Geralmente é utilizado quando vários registros de uma entidade se relacionam com vários registros da outra entidade. Algumas literaturas afirmam que este tipo de relacionamento não existe, pois para que ele possa ser concebido é necessária a geração de uma terceira entidade caracterizando assim dois relacionamentos um-para-muitos entre elas, que ganha o sentido de muitos-para-muitos. No entanto, devemos sim considerar este tipo, pois quando o modelo foi concebido, este é um princípio básico da criação. Além disso, ao assumirmos dois relacionamentos com conectividade um-para-muitos, algumas propriedades são perdidas e podem comprometer as regras de integridade referencial. Oportunamente falaremos mais sobre isso.



Figura 17: Exemplo de conectividade muitos-para-muitos

Vejamos a aplicação da expressão já conhecida:

Um(a) [ENTIDADE A] [RELACIONAMENTO] [preposição] quantos(as) [ENTIDADE B]?

Substituindo os termos:

Um ALUNO cursa quantas DISCIPLINAS?

Inverte-se a pergunta:

Uma DISCIPLINA é cursada por quantos ALUNOS?

Substituindo os termos temos, um aluno cursa quantas disciplinas? A resposta é muitas, portanto devemos colocar um "n" ao lado da entidade disciplina. Com os devidos ajustes invertesse a pergunta. Uma disciplina é cursada por quantos alunos? A resposta é muitas, portanto devemos colocar um "n" também ao lado do aluno. Veja que nesta pergunta também é desprezado o fato de conter um "n" ao lado da entidade disciplina, ou seja, novamente a pergunta investigativa independe da resposta anterior.

Em uma conectividade muitos-para-muitos não é possível adicionar um elo (chave) em cada uma das propriedades das entidades envolvidas no relacionamento, à luz do que foi feito na conectividade tipo um-para-muitos. Neste caso a regra é outra.

O exemplo está indicando que um aluno cursa uma ou mais disciplinas e que uma disciplina por sua vez também pode ser cursada por um ou mais alunos. O elo (chave) deve ser adicionado a uma nova estrutura uma entidade de ligação à qual não irá conter apenas um, mas sim ambos os elos (chaves) de cada uma das entidades envolvidas no relacionamento. Você imagina por quê? Vejamos! Como não podemos criar chaves nas propriedades da entidade aluno e nem na entidade disciplina, pois não sabemos quantos seriam necessárias (muitos-para-muitos), a solução é a criação de uma nova estrutura que servirá de ligação.

(i) Dica

Mais detalhes sobre esta nova estrutura de ligação criada neste tipo de conectividade abordaremos no conteúdo sobre a derivação do modelo conceitual para o projeto físico.

Outro elemento importante da modelagem entidade-relacionamento diz respeito a aspecto de condicionalidade. A condicionalidade nada mais é do que a análise da necessidade ou não de uma associação (relacionamento) de fato existir. Estamos falando da obrigatoriedade no relacionamento.

Para alguns esse elemento é um aspecto que não interfere no projeto de banco de dados. Mas é um tremendo engano pensar assim. A obrigatoriedade no relacionamento será fundamental no momento da criação do projeto físico do banco de dados, pois ele indicará se um determinado campo chave terá ou não seu preenchimento obrigatório quando do cadastramento dos dados.

Vamos a um exemplo. Observe a figura 18 onde é apresentado parte de um cenário em que um departamento lota vários funcionários.



Figura 18: Exemplo de obrigatoriedade no relacionamento

A representação gráfica indica que todo funcionário obrigatoriamente deve estar lotado em um departamento. Portanto, a ligação (chave) adicionado nas propriedades da entidade "funcionário"

deverá ser preenchido (eis a obrigatoriedade), e esta, por sua vez, deverá conter um valor correspondente na entidade "departamento".



(i) Dica

A obrigatoriedade só tem sentido em uma relação do tipo "um-para-muitos". Quanto avançarmos para as atividades práticas isso ficará evidenciado.

O oposto da obrigatoriedade, por natureza óbvia é o relacionamento de não obrigatoriedade. Ele é bastante comum em relacionamentos no MER. Para muitos, essa é a forma de omitir uma informação desconhecida ou imprecisa. Mas atenção, não se utilize deste subterfúgio para ignorar o aspecto da obrigatoriedade, pois do contrário, os controles sobre de preenchimento ou não do(s) campo(s) chave(s) terão que ser feitos exclusivamente na camada a aplicação. A figura 19 ilustra um exemplo com o mesmo cenário apresentado no caso anterior.



Figura 19: Exemplo de não obrigatoriedade no relacionamento

No exemplo acima, a representação gráfica indica que um funcionário não obrigatoriamente deve estar lotado em um departamento. Portanto, a ligação (chave) adicionado nas propriedades da entidade "funcionário" não necessariamente deverá ser preenchido (eis a não obrigatoriedade) e poderá ser nulo (sem valor definido). A não obrigatoriedade também só tem sentido em uma relação do tipo "um para muitos".

Encerramento

Agora que você conhece os elementos envolvidos no MER, suas especificidades e particularidades, é necessário avançarmos para o processo de normalização. A normalização é uma técnica relevante na obtenção de modelos de dados ideal, ou seja, que atenda na sua totalidade as necessidades do usuário considerando o cenário (minimundo).

CONTINUE

Referências

Referências

ALVES, William Pereira. Banco de dados. São Paulo: Erica, 2014

CARDOSO, Vírginia M. Sistemas de banco de dados. São Paulo: Saraiva, 2008.

CHEN, Peter. Gerenciando banco de dados: a abordagem entidade-relacionamento para projeto lógico. São Paulo, McGraw-Hill, 1990.

HEUSER, Carlos Alberto. Projeto de banco de dados. Porto Alegre: Bookman, 2011.

MACHADO, Felipe Nery Rodrigues. Banco de dados: projeto e implementação.3. São Paulo: Erica, 2014.

Referências de Imagens

Divisão de Modalidades de ensino (DME), Fundação Universidade Regional de Blumenau (FURB), 2019a - o.