

Conversões de arquivos de mídia

Entidade-Relacionamento (E-R) é um modelo de dados conceitual de alto nível cujos conceitos foram projetados para estar o mais próximo possível da visão que o usuário tem dos dados, não se preocupando em representar como esses dados estarão realmente armazenados. Tem como ideologia básica a percepção do mundo real, representando-o por meio de uma coleção de objetos chamados de entidade.

A abordagem E-R foi criada em 1976 por Peter Chen. Ela pode ser considerada um padrão para modelagem conceitual, estudada no material específico. Neste capítulo, veremos os conceitos básicos referentes à abordagem E-R, bem como uma contextualização para utilização aplicada a jogos desta modelagem.

1. Entidade

Nada mais é do que um conjunto de objetos do mundo real, sendo eles concretos ou abstratos, sobre os quais se deseja manter informações no banco de dados. Um exemplo de objeto concreto aplicado a jogos pode ser uma pessoa, o jogador; já um objeto abstrato pode ser o personagem que o usuário escolhe.

Essas entidades são representadas por um retângulo, como pode ser visto na figura 1.



Figura 1 – Representação gráfica de entidades

A imagem apresenta dois retângulos com um texto no centro de cada um identificando a entidade. À esquerda, a entidade jogador e, à direita, a entidade personagem.

2. Atributos

São propriedades (características) que identificam as entidades. Cada entidade possui uma coleção de elementos de dados. Um domínio de valores é associado a cada atributo de uma entidade. Esses domínios podem ser um conjunto de números inteiros, números reais, cadeias de caracteres ou qualquer outro tipo de valor que o atributo possa assumir.

Os atributos são representados apenas pelo seu nome ligado à entidade por uma linha reta. Isso pode ser visto na figura 2, onde estão representados alguns atributos das entidades jogador e personagem para um jogo MMO, além do domínio em que um jogador precisa fazer o *login* com uma senha e a escolha de um personagem de uma determinada raça com uma habilidade específica.

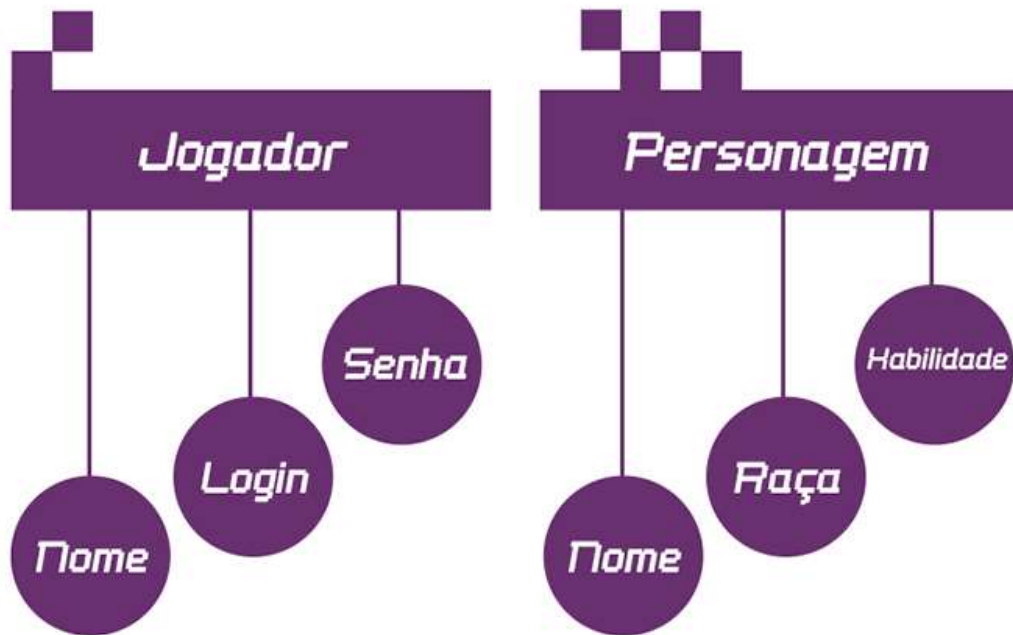


Figura 2 – Exemplo de atributos de entidades

A imagem apresenta dois retângulos, conforme descrito acima, representando as entidades jogador e personagem. Em cada retângulo, estão conectados os atributos, com uma linha com pequenos círculos na ponta. Cada círculo contém textos no centro identificando-os. A entidade jogador tem os atributos nome, *login* e senha. A entidade personagem tem os atributos nome, raça e habilidade.

Vamos, agora, ver os diversos tipos de atributos que existem.

Compostos: os atributos compostos podem ser divididos em partes menores, ou subpartes, as quais representariam atributos básicos, mais simples, com significados independentes. Por exemplo, um atributo endereço pode ser subdividido em rua, cidade, estado e CEP. Poderíamos também dividir o atributo rua em número, nome-rua e número-apartamento.

Simples: são chamados também por atributos atômicos. Eles não são divisíveis.

Monovalorados: são atributos que possuem apenas um valor para uma entidade em particular. Por exemplo, a idade é um atributo monovalorado para uma entidade funcionário.

Multivalorado: são atributos que possuem um ou mais valores. Por exemplo, o atributo idioma de uma entidade menu pode conter os valores válidos para os idiomas em que o menu é apresentado.

Armazenado: em geral, todos os atributos são armazenados.

Derivado: alguns atributos podem ter uma relação entre si. Por exemplo, idade e data de nascimento de uma pessoa. O atributo idade é preenchido a partir do cálculo da data de nascimento com a data atual.

Nulo: em alguns casos, uma entidade pode não necessitar de um valor aplicável a um de seus atributos. Por exemplo, no atributo composto número-apartamento, visto acima, apenas definiremos valores para esse campo quando a entidade funcionário em particular morar em um prédio. A representação de um atributo sem valor é colocarmos um valor especial *null*. *Null* também pode ser utilizado quando não conhecemos o valor de um atributo, por exemplo, quando é desconhecida a data de nascimento de uma pessoa.

(Adaptado de: ELMASRI & NAVATHE, 2011, págs. 39-41).

3. Relacionamento

Um relacionamento é uma associação entre uma ou várias entidades. O símbolo que representa o relacionamento no modelo E-R é o losango, e as entidades relacionadas são ligadas por linhas. Um exemplo de relacionamento pode ser observado na figura 3.



Figura 3 – Relacionamento entre duas entidades

Dois retângulos representando as entidades ligados por uma linha com um losango no centro, com o texto no seu interior indicando o tipo de relacionamento que as entidades possuem.

4. Cardinalidade de relacionamentos

Uma propriedade importante dos relacionamentos é a especificação de quantas ocorrências de uma entidade podem estar associadas a uma determinada ocorrência de outra entidade.

Existem dois tipos de cardinalidades.

Máximas: especificam o valor máximo de ocorrências de uma entidade A com relação a uma outra entidade B.

Mínimas: número mínimo de ocorrências de uma entidade A com relação a uma outra entidade B.

4.1 Cardinalidade máxima

Um para um (1:1): uma ocorrência de A está associada, no máximo, a uma ocorrência de B, e uma ocorrência em B está associada a, no máximo, uma ocorrência em A.

Como exemplo de um relacionamento “um para um”, vamos imaginar que para iniciar o *game* de um jogador (*single player*) seja necessário criar uma sessão. Logo, o jogador pode possuir apenas uma sessão e essa sessão pode ser possuída por apenas um jogador, como pode ser visto na figura 4.



Figura 4 – Exemplo de relacionamento um para um (1:1)

Um para muitos (1:N): uma ocorrência de A está associada a várias ocorrências de B, porém uma ocorrência de B deve estar associada, no máximo, à uma ocorrência em A.

Um exemplo de um relacionamento “um para muitos” pode ser descrito como o cenário de jogo *Massively Multiplayer Online* (MMO), em que um jogador possui uma sessão e uma sessão é possuída por muitos jogadores. Esse exemplo pode ser observado na figura 5.

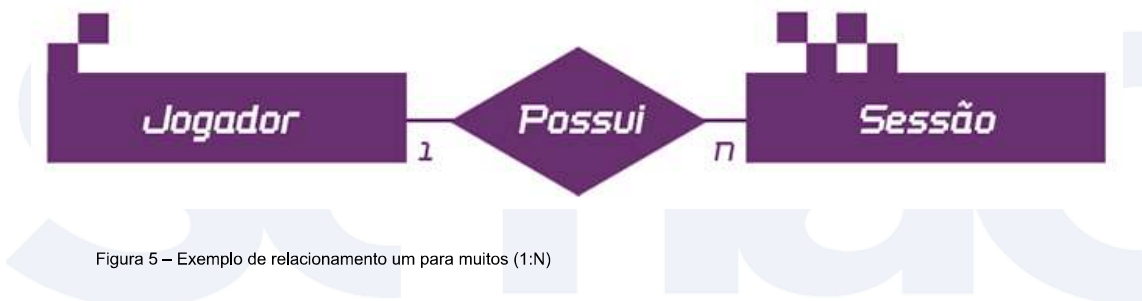


Figura 5 – Exemplo de relacionamento um para muitos (1:N)

Muitos para muitos (N:N): uma ocorrência de A está associada a qualquer número de ocorrências de B, e uma ocorrência em B está associada a qualquer número de ocorrências em A.

Para exemplificar o relacionamento “muitos para muitos”, vamos imaginar o cenário do jogo MMO, porém abordando o relacionamento entre jogadores e personagens, em que um jogador possui muitos personagens e um personagem pode ser escolhido por mais de um jogador. Esse exemplo pode ser visto na figura 6.



Figura 6 – Exemplo de relacionamento muitos para muitos (N:N)

4.2 Cardinalidade mínima

Representa a participação mínima de uma entidade em um relacionamento. É também chamada de restrição de participação ou restrição de dependência de existência.

Consiste nos valores a seguir.

Zero: participação opcional, existência não dependente. Representa uma associação opcional.

Um ou muitos: participação obrigatória, existência dependente. Representa uma associação obrigatória.

Um exemplo de associação opcional entre entidades pode ser visto na figura 7. O exemplo mostra a cardinalidade mínima entre as entidades quanto ao relacionamento de pertinência entre a entidade jogador e o *ranking*. No exemplo de relacionamento opcional mostrado na figura 7, a cardinalidade (0,N) faz referência ao *ranking*, já a cardinalidade (1,1), faz referência ao jogador. Isto pode ser traduzido como:

Uma ocorrência de jogador pode não estar associada a uma ocorrência de ranking ou pode estar associada a uma ocorrência dele (determinado jogador pode ou não pertencer ao ranking dos 10+).

Uma ocorrência de ranking está associada a apenas uma ocorrência de jogador.



Figura 7 – Exemplo de um relacionamento opcional

5. Integridade referencial

Em resumo, integridade referencial é um conceito de banco de dados que garante que todos os relacionamentos propostos entre tabelas no modelo E-R serão respeitados, dando a certeza de que os dados de um banco de dados estarão íntegros. Esses relacionamentos são baseados nas definições de uma chave primária e uma chave estrangeira, além de regras pré-definidas para a manipulação dessas chaves.

Sempre que for necessário alterar, inserir ou apagar um registro em uma entidade, os relacionamentos deverão ser validados por meio das chaves estrangeiras das tabelas relacionadas.

Chave candidata

Chave candidata é um atributo ou um conjunto de atributos de uma tabela que identifica uma única linha. A chave primária é extraída a partir do conjunto de chaves candidatas de uma tabela.

Chave única

Essa chave é definida por um marcador de unicidade, garantindo que um atributo ou um conjunto de atributos não podem ser duplicados na mesma tabela.

Chave primária

A chave primária de uma tabela é uma das chaves candidatas que melhor identifica uma tabela. Os critérios para escolher a chave primária são:

- ◆ O conteúdo da coluna ou do conjunto de colunas deve sempre existir;
- ◆ O valor da chave não deve poder ser alterado.

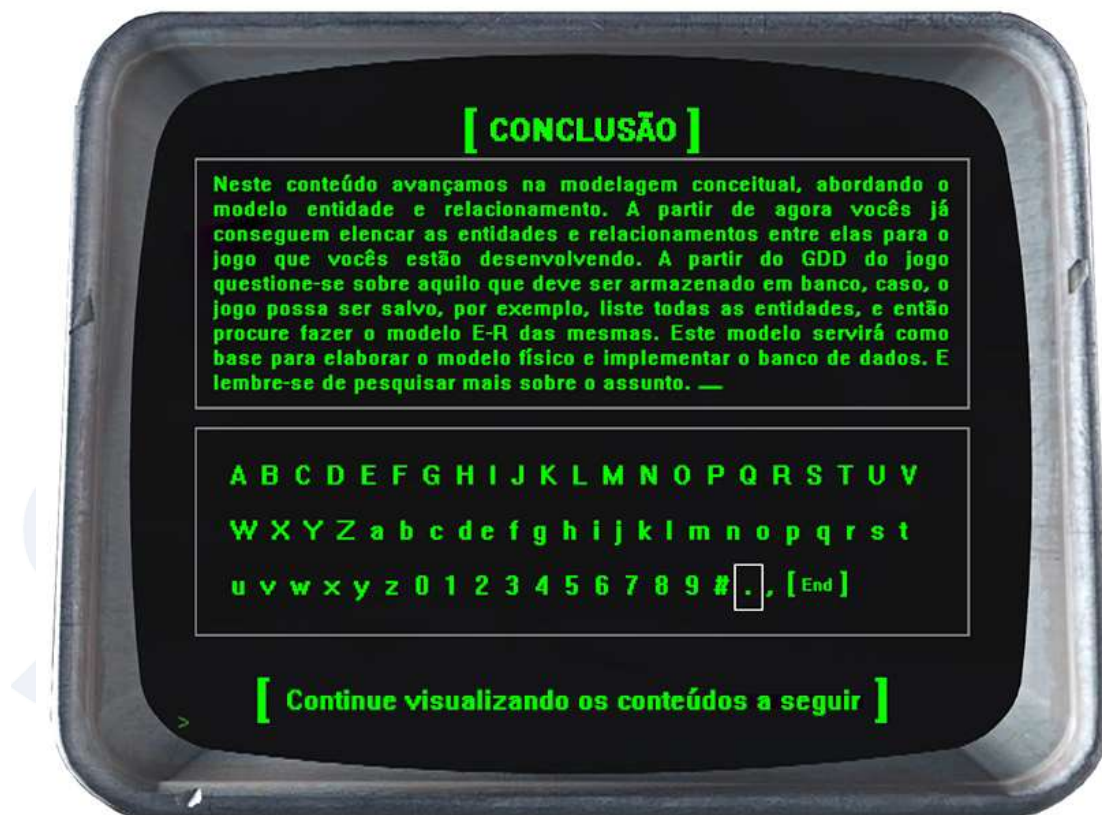
A chave primária tem as seguintes características:

- ◆ Apenas uma chave primária por tabela;
- ◆ Pode ser composta por mais de uma coluna;
- ◆ Não pode permitir valores nulos.

Chave estrangeira

A chave estrangeira se refere ou é relacionada com alguma chave primária de uma tabela, podendo ser inclusive da mesma tabela. A chave estrangeira tem as seguintes características:

- ◆ Uma tabela pode possuir zero, uma ou diversas chaves estrangeiras;
- ◆ Pode ser composta de uma ou mais colunas contando que ela seja exatamente igual à chave primária que está relacionada (tipo de dado e tamanho);
- ◆ Pode permitir valores nulos.



Conclusão Neste conteúdo avançamos na modelagem conceitual, abordando o modelo entidade e relacionamento. A partir de agora vocês já conseguem elencar as entidades e relacionamentos entre elas para o jogo que vocês estão desenvolvendo. A partir do GDD do jogo questione-se sobre aquilo que deve ser armazenado em banco, caso, o jogo possa ser salvo, por exemplo, liste todas as entidades, e então procure fazer o modelo E-R das mesmas. Este modelo servirá como base para elaborar o modelo físico e implementar o banco de dados. E lembre-se de pesquisar mais sobre o assunto. Continue visualizando os conteúdos a seguir.

Bibliografia

ELMASRI, R.; NAVATHE, S.B. **Sistemas de banco de dados**. Pearson Brasil, 2011.
 HEUSER, Carlos Alberto. **Projeto de Bancos de Dados**. Porto Alegre: Sagra Luzzatto, 2009.