



BANCO DE DADOS

AULA 2



Prof. Lucas Rafael Filipak

CONVERSA INICIAL

O objetivo desta aula é compreender o funcionamento do modelo de entidade e relacionamento. A ideia é conceitualizar e exemplificar o que seja entidade, atributo, relacionamento e cardinalidade e desenvolver um passo a passo de um estudo de caso da modelagem de um diagrama de entidade e relacionamento (DER). Além disso, pretende-se finalizar a modelagem dos dados com o modelo físico e o modelo lógico, dando uma visão geral e abrangente da utilização dos três modelos. A aula termina com um estudo de caso transformando o modelo ER em modelo relacional.

TEMA 1 – MODELO ENTIDADE E RELACIONAMENTO: ENTIDADE, ATRIBUTO E RELACIONAMENTO

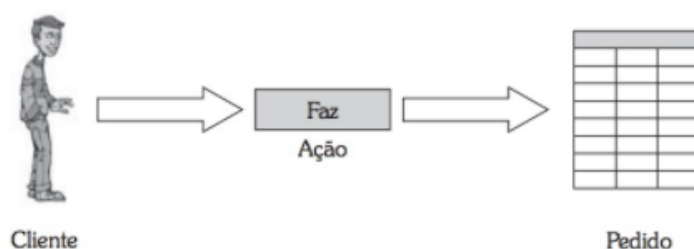
O conceito do modelo de entidade e relacionamento (MER) foi proposto por Peter Chen, nos anos de 1970, e tem como base a perspectiva do mundo real como construído por dois grupos de objetos que formam um negócio: entidades e relacionamentos.

O MER é extremamente útil para mapear, com base em um esquema conceitual, o significado e as interações das empresas reais (Silberschatz; Korth; Sudarshan, 1999).

Em 1976, Chen desenvolveu uma técnica de diagramação capaz de representar o modelo de dados de forma abrangente, que ficou conhecida como *diagrama de entidade e relacionamento* (DER).

Alves (2014, p. 91) diz que as entidades e os relacionamentos possuem uma ligação tão forte que não é possível tratar de um sem mencionar o outro. O que une esses dois componentes é uma ação. É como uma oração formada por um sujeito (entidade), um verbo (ação) e um predicado (relacionamento).

Figura 1 – Componentes de uma entidade-relacionamento



Fonte: Alves, 2014, p. 92.

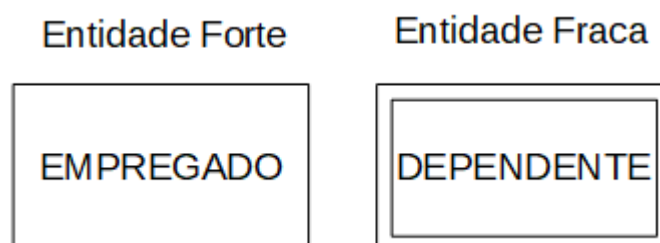
1.1 Entidade

Uma entidade é o objeto básico representado no modelo ER. É uma categoria de elementos relevantes para um negócio, podendo ser, por exemplo, clientes, fornecedores, vendas, pagamentos, entre outros, sobre os quais são realizadas operações para o funcionamento do negócio. Segundo Puga, França e Goya (2013, p. 84), a entidade pode ser caracterizada como:

- Objeto ou fato que deve ter seus dados guardados em um determinado contexto;
- Conjunto de tipo de informação que seja diretamente associado ao domínio de conhecimento analisado;
- Objeto que desempenha um papel específico no sistema;
- Objeto que possui propriedades que o distinguem de outras entidades, podendo ser:
 - ✓ **Objeto concreto:** computador, impressora, veículo, produto, imóvel, etc.;
 - ✓ **Pessoa:** funcionário, cliente, aluno, professor, atendente etc.;
 - ✓ **Evento:** situação em que algo está ocorrendo, ou está planejado para ocorrer. Por exemplo, o agendamento de uma corrida de táxi, o recebimento de uma encomenda etc.

A representação de uma entidade em um modelo entidade-relacionamento (modelo ER) se dá por um retângulo com o nome da entidade escrito dentro. As entidades podem ser classificadas em: *entidades fortes* e *entidades fracas*.

Figura 2 – Notação para representação de entidade forte e entidade fraca



O que diferencia uma entidade forte de uma entidade fraca é o grau de dependência de existência delas. A entidade *empregado*, na Figura 2, não depende de nenhuma outra entidade para existir, por isso é uma entidade forte.



As entidades fracas possuem dependência de existência e/ou identificação e são sempre ligadas a outras tabelas por meio de relacionamentos. Tome-se como exemplo a entidade *dependente*, cuja existência e identificação está vinculada a outra entidade forte, no caso o *empregado*.

1.2 Atributo

Os atributos descrevem as características de uma entidade, podendo ser definidos, segundo Puga, França e Goya (2013, p. 86) como:

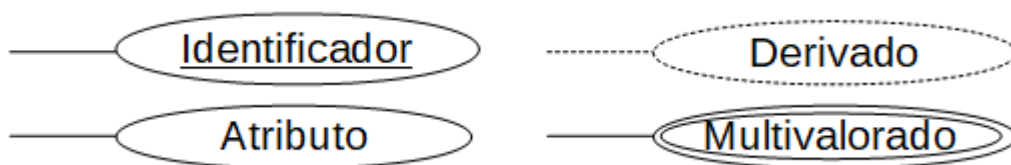
- Informações associadas a uma entidade;
- Características ou propriedades de uma entidade ou relacionamento;
- Descrição, identificação, qualificação ou quantificação de uma entidade.

Entre os exemplos de atributos presentes no modelo ER criado na explicação do esquema conceitual, podem-se citar:

- Nome do funcionário;
- Matrícula do funcionário;
- Ano de fabricação do veículo.

Os atributos são identificados por símbolos, como demonstrado na Figura 3, que são conectados em entidades. Seguindo a notação de Peter Chen, os atributos podem ser:

Figura 3 – Notação dos atributos – Peter Chen



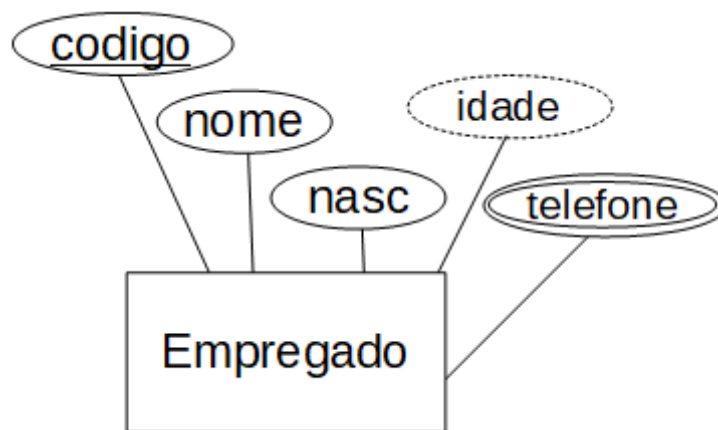
- **Atributo identificador:** permite representar uma identificação única de uma ocorrência em uma entidade. Uma entidade pode possuir mais de um atributo identificador, mas a composição dos atributos deve formar uma identificação única;
- **Atributo não identificado (ou chamado apenas de atributo):** corresponde à maioria dos atributos de uma entidade. Os atributos não identificados poderão conter valores nulos;



- **Atributo multivalorado:** são utilizados para representar os atributos que podem ter mais de uma ocorrência em uma mesma instância de uma entidade. Como exemplo, pode-se pensar num número de telefone. É comum que uma pessoa possua mais de um número de telefone, portanto para esta representação utiliza-se o atributo multivalorado;
- **Atributos derivado:** às vezes, um atributo está relacionado com outro atributo, por exemplo, a idade e data de nascimento de um cliente. Pode-se determinar a sua idade por meio da data de nascimento e da data atual. Atributos como a idade são chamados de *atributos derivados*.

A Figura 4 exemplifica os tipos de atributos utilizados em um DER.

Figura 4 – Exemplos de atributos



Na Figura 4, a entidade *Empregado* tem o atributo *codigo* como atributo identificador (e único para cada pessoa). O *nome* e *nasc* são atributos não identificados (podendo existir nomes e data de nascimento iguais). O atributo *telefone* é um atributo multivalorado (pois o empregado pode possuir mais de um telefone) e o atributo *idade* é um atributo derivado do atributo *nasc*.

1.3 Relacionamento

O relacionamento estabelece uma relação ou associação entre as entidades, com um significado específico do mundo real. Da mesma maneira ocorre no mundo real, onde os objetos não ocorrem de maneira isolada, mas sim se associam ou se vinculam com outros objetos. No modelo ER, os relacionamentos são representados por um losango que relaciona as entidades, podendo ser classificados como relacionamentos fortes ou fracos. Tal como as

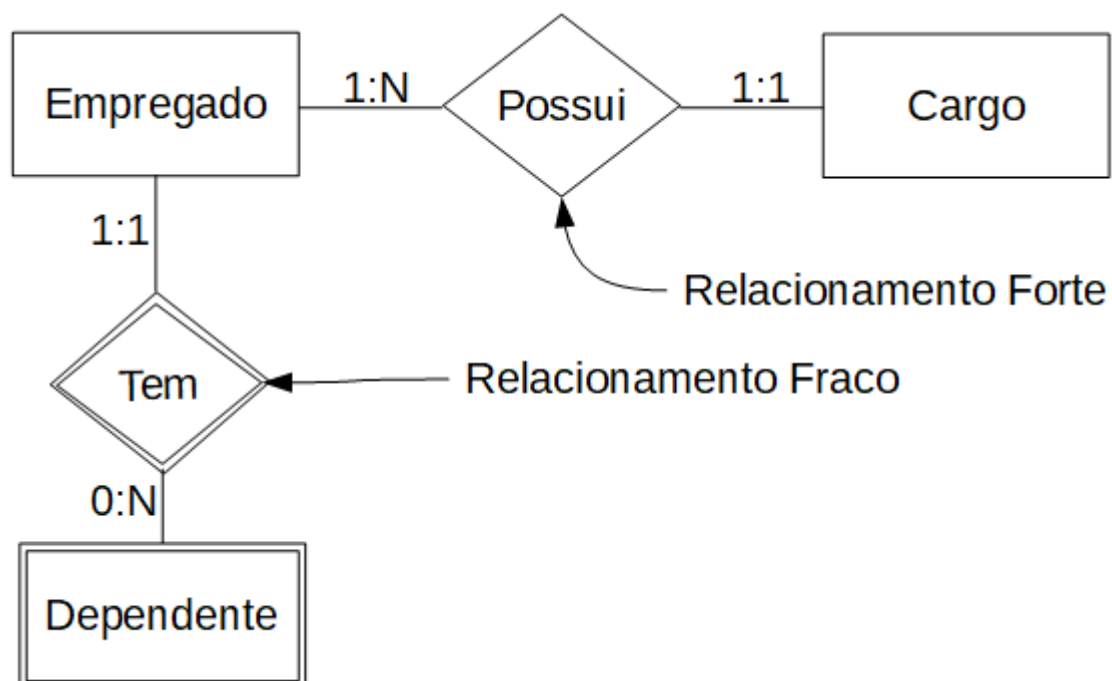


entidades, os relacionamentos também possuem um nome e devem expressar o real significado dentro do contexto modelado.

Há autores que utilizam outra simbologia para representar uma relação. Esse material faz uso da notação de Peter Chen, que utiliza o losango para representar uma relação.

A Figura 5 representa o seguinte cenário: considere que um *Empregado* tem um *Cargo* e que este cargo é uma classificação interna dada pela empresa. Porteiro, auxiliar administrativo, gerente, entre outros, são exemplos de cargos. Um *Empregado* pode ou não ter *Dependente*.

Figura 5 – Notação que representa o relacionamento forte e o relacionamento fraco



Na Figura 5, a entidade *Dependente* é uma entidade fraca. Sempre que existe uma relação entre duas entidades e uma delas é uma entidade fraca, o relacionamento entre as duas será fraco também. No outro relacionamento, também representado na Figura 5, o relacionamento *possui* é representado por um relacionamento forte, pelo fato de não haver dependência de existência ou identificação, pois um empregado não depende do seu cargo para existir ou ser identificado e vice-versa. Observe os seguintes questionamentos:

- Um empregado pode não ter dependentes?
- Um dependente pode ter mais de um empregado associado?



- Determinado empregado pode possuir mais de um dependente?
- Pode existir dependente sem algum empregado associado?

Note que pode haver respostas diferentes para essas perguntas e que cada resposta depende do problema que está sendo modelado. Entretanto, para que se possam expressar essas ideias no modelo, é necessário definir uma propriedade importante do relacionamento: a *cardinalidade*, a qual será tratada na próxima seção.

TEMA 2 – CARDINALIDADE

A cardinalidade é a quantificação de um relacionamento determinada com base nas regras de negócios, mostrando, em termos quantitativos, como os dados são associados uns aos outros. Em outras palavras, a cardinalidade é um número que expressa o comportamento (número de ocorrências) de determinada entidade associada a uma ocorrência da entidade em questão por meio do relacionamento. Para Alves (2014, p. 98), “cardinalidade é uma restrição para um relacionamento binário que determina quantas vezes uma entidade pode participar de um relacionamento”.

Um relacionamento sempre possui dois sentidos: o de ida e o de volta. A cardinalidade de uma relação é definida em cada um dos sentidos do relacionamento, representada por (x, y) em que x é a cardinalidade mínima e y é a máxima. Cada uma tem um objetivo:

- Mínima: orientar a obrigatoriedade ou opcionalidade do relacionamento;
- Máxima: define a quantidade máxima que um elemento pode estar associado no relacionamento.

O relacionamento entre as entidades deve ser criado e validado de acordo com as regras que sustentam o negócio, permitindo manter a integridade do modelo de dados.

Figura 6 – Relacionamento entre entidades





A posição correta da cardinalidade é do lado oposto à sua entidade correspondente. Na Figura 6, a cardinalidade **0:N** faz referência à entidade *Empregado*, já a cardinalidade **(1:1)** faz referência à entidade *Dependente*. Isso significa que:

- Uma ocorrência de empregado pode não estar associada a uma ocorrência de dependente ou pode estar associada a várias ocorrências dele (o empregado pode não possuir dependentes ou pode possuir vários);
- Uma ocorrência de dependente está associada a apenas uma ocorrência de empregado (o dependente possui apenas um empregado responsável).

Na prática, para as cardinalidades máximas, costumamos distinguir dois tipos: **1** (um) e **N** (cardinalidades maiores que 1). Já para as cardinalidades mínimas, costumamos distinguir dois tipos: **0** (zero) e **1** (um). Na tabela 1, são apresentadas as possíveis cardinalidades que podem ser atribuídas em um relacionamento.

Tabela 1 – Graus de cardinalidade

Cardinalidade	Descrição
1:1	Um elemento de uma entidade se relaciona com um elemento de outra entidade
1:N ou N:1	Um elemento de uma entidade pode se relacionar com mais de um elemento de outra entidade
N:N	Vários elementos de uma entidade podem se relacionar com vários elementos de outra entidade e vice-versa
0:1	Um elemento de uma entidade pode se relacionar a nenhuma ou uma ocorrência de outra entidade
0:N	Um elemento de uma entidade pode se relacionar a nenhuma ou muitas ocorrências de outra entidade

A Figura 7 representa outro exemplo de DER. O DER é composto por entidades, relacionamentos e atributos. Nas Figura 6 e 7, não foram representados os atributos para não poluir o exemplo, mas eles sempre devem aparecer.

- Um motorista dirige um veículo;
- Um veículo é dirigido por um motorista.



Figura 7 – Exemplo de entidade e relacionamento (DER)



TEMA 3 – ESTUDO DE CASO

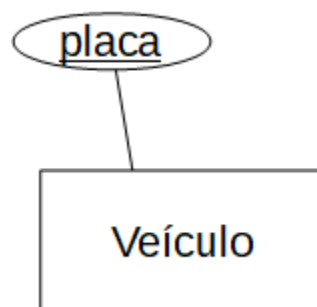
O estudo de caso contempla a criação de um DER para armazenar informações sobre veículos. Após a análise de requisitos, sabe-se que:

- a. O veículo possui sempre uma placa única em todo o país;
- b. O sistema tem um cadastro de pessoas com CPF, nome e data de nascimento;
- c. O veículo possui sempre um responsável legal por ele;
- d. O veículo é sempre de uma marca e de um modelo;
- e. O veículo precisa manter o histórico desta responsabilidade (propriedade).

Os dados levantados pela análise de requisito geralmente são em tópicos, pois o cliente vai repassando as informações conforme a conversa com o analista vai acontecendo. Para um melhor entendimento, será elaborado o DER item a item. Na Figura 8, são representadas as informações descritas no tópico (a):

- a. O veículo possui sempre uma placa única em todo o país;

Figura 8 – Entidade veículo com atributo *placa*



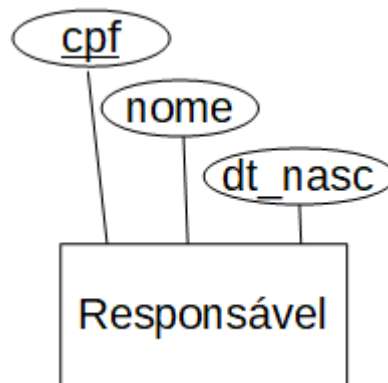
Cria-se a entidade *Veículo* para representar um objeto concreto, e o atributo *placa* para representar a característica que um *Veículo* possui. Pode-se definir esse atributo como identificador, considerando que a placa do veículo é única.



Na Figura 9, são representadas as informações descritas no tópico (b):

- b. O sistema tem um cadastro de pessoas (responsável) com CPF, nome e data de nascimento;

Figura 9 – Entidade responsável com seus atributos

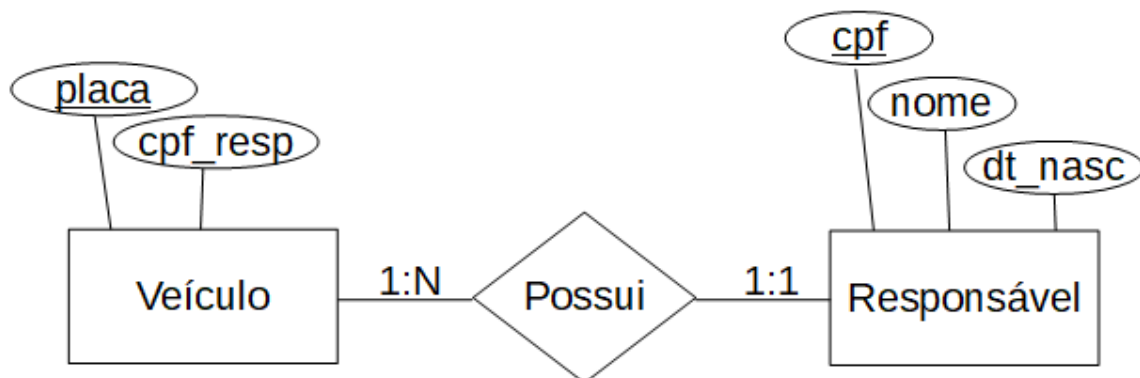


Cria-se a entidade *Responsável* utilizando *cpf* como atributo identificador, e *nome* e *dt_nasc* como atributos não identificados.

Na Figura 10, são representadas as informações descritas no tópico (c), as quais conectam as definições feitas nos tópicos (a) e (b):

- c. O veículo possui sempre um responsável legal por ele.

Figura 10 – Relacionamento entre as entidades *veículo* e *responsável*



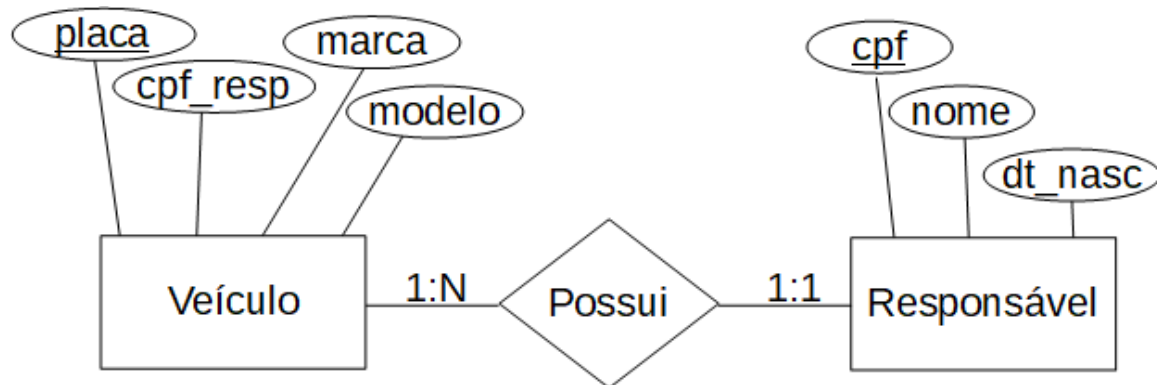
Na entidade *Veículo*, adiciona-se o *cpf_resp* como um atributo não identificado. Considerando as especificações da análise de requisitos, foi adicionada a cardinalidade: um responsável pode ter vários veículos (1:N) e um veículo só pode ter um responsável (1:1).

Na Figura 11, são representadas as informações descritas no tópico (d):

- d. O veículo é sempre de uma marca e de um modelo;



Figura 11 – Entidade *veículo* com os atributos

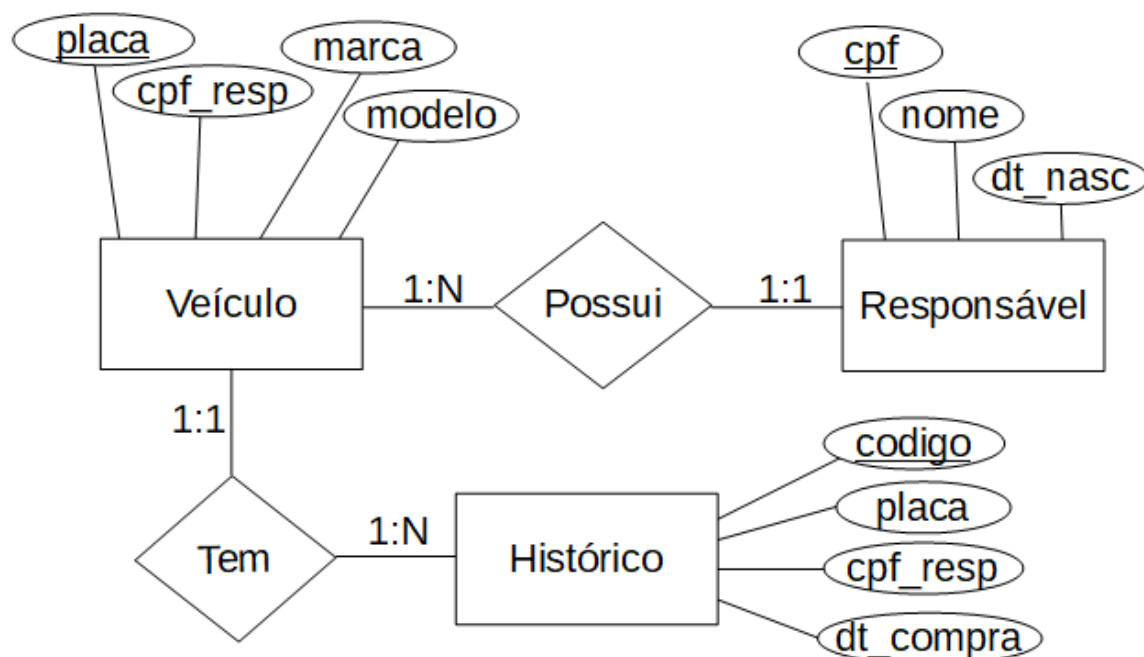


Considerando que *marca* e *modelo* se referem a novas características que a entidade *Veículo* pode ter, esses dois atributos foram conectados à entidade *Veículo* como atributos não identificados.

Para concluir, na Figura 12, são representadas as informações descritas no tópico (e):

- e. O veículo precisa manter o histórico desta responsabilidade (propriedade);

Figura 12 – DER completo



Cria-se a entidade *Histórico*. O atributo *cpf_resp* identifica de quem era o veículo na *dt_compra*. Com todos esses atributos, armazena-se o histórico do veículo. Foi definida a cardinalidade como uma ocorrência: o histórico tem apenas um veículo (1:1) e um veículo pode ter vários históricos (1:N).



TEMA 4 – MODELOS: LÓGICO, FÍSICO E RELACIONAL

4.1 Modelo Lógico

O modelo lógico de dados reflete as propriedades necessárias para a tradução do modelo conceitual, de maneira que seja possível a descrição dos elementos capazes de serem interpretados por um Sistema Gerenciador de Banco de Dados (SGBD), tais como o detalhamento dos atributos, chaves de acesso, integridade referencial e normalização. O modelo lógico é independente de *hardware*, ou seja, qualquer alteração (escolha de um computador, sistema operacional diferente etc.) não afetará no modelo lógico.

Dentro do modelo lógico, algumas regras e restrições começam a ser implementadas no desenho, informando as colunas, que são únicas e que não podem ser duplicadas, representadas anteriormente no DER pelos atributos identificadores, aqui chamados de *chaves primárias*, como também as chaves secundárias, que são geradas para associar uma entidade a outra.

As chaves primárias são colunas que não podem ser duplicadas, por exemplo, o CPF no cadastro de clientes e o CNPJ no cadastro de empresas. Os relacionamentos são importantes para representar a conexão entre as tabelas, representadas anteriormente no DER pelas entidades e também por manter a integridade do banco de dados, o qual irá utilizar essas definições de restrição para permitir ou não a duplicidade de um dado.

Segundo Puga, França e Goya (2013, p. 77), a modelagem de dados é uma técnica utilizada para:

- Conhecer melhor o contexto do negócio;
- Retratar os dados que suportam esse contexto de negócio;
- Projetar o banco de dados;
- Promover o compartilhamento dos dados e a integração dos sistemas por meio da reutilização de estruturas de dados comuns;
- Contribuir para que a perspectiva da organização a respeito dos seus dados seja unificada.

Dentro do modelo lógico, as entidades que eram representadas por retângulos (modelo conceitual) se tornam tabelas do banco de dados.



Figura 13 – Exemplo da representação gráfica do modelo lógico



Fonte: Puga; França; Goya, 2013, p. 163.

A Figura 13 é um exemplo da representação gráfica do modelo lógico, mas ele ainda pode ser descrito sem a representação gráfica.

Figura 14 – Exemplo de modelo lógico

```
Clientes (Codigo_Cliente, Nome_Cliente, CPF, Endereco)
Produtos (Codigo_Produto, Descricao_Produto, Unidade_Medida, Preco_Unitario)
```

Fonte: Alves, 2014, p. 30.

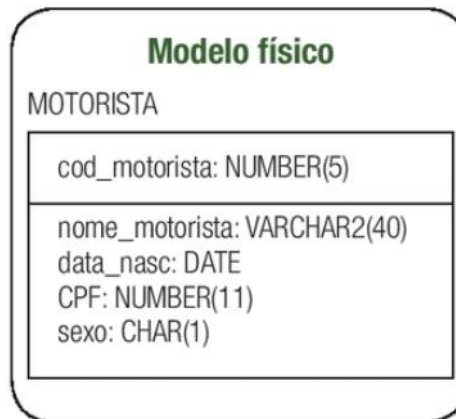
É importante destacar que, nesse ponto do projeto de banco de dados, ainda não se definem as características particulares de cada coluna, como tamanho ou tipo de dado, limitando-se apenas a relacioná-los com suas respectivas tabelas.

4.2 Modelo físico

O modelo físico é a etapa final do projeto de banco de dados, estando relacionado diretamente com o profissional do SGBD, podendo variar conforme o SGBD que está sendo utilizado. Já com a definição do SGBD, o modelo físico é criado e cada atributo é devidamente especificado, conforme os tipos de dados do SGBD escolhido.



Figura 15 – Exemplo da representação gráfica do modelo físico



Fonte: Puga; França; Goya (2013, p. 163).

O modelo físico representa a estrutura para armazenamento físico dos dados, expressando a forma como as informações serão armazenadas fisicamente, em termos computacionais. Para criar um banco de dados, geralmente se utiliza uma linguagem declarativa. Em SQL (Structured Query Language), essa linguagem é denominada *linguagem de definição de dados* (DDL).

Figura 16 – Exemplo de comando SQL para criação do modelo físico

```
CREATE TABLE Clientes(
Codigo_Cliente INTEGER,
Nome_Cliente VARCHAR(50),
CPF CHAR(14),
Endereco VARCHAR(50),
PRIMARY KEY (Codigo_Cliente));

CREATE TABLE Produtos(
Codigo_Produto CHAR(13),
Descricao_Produto VARCHAR(50),
Unidade_Medida CHAR(3),
Preco_Unitario DECIMAL(10,2),
PRIMARY KEY (Codigo_Produto));
```

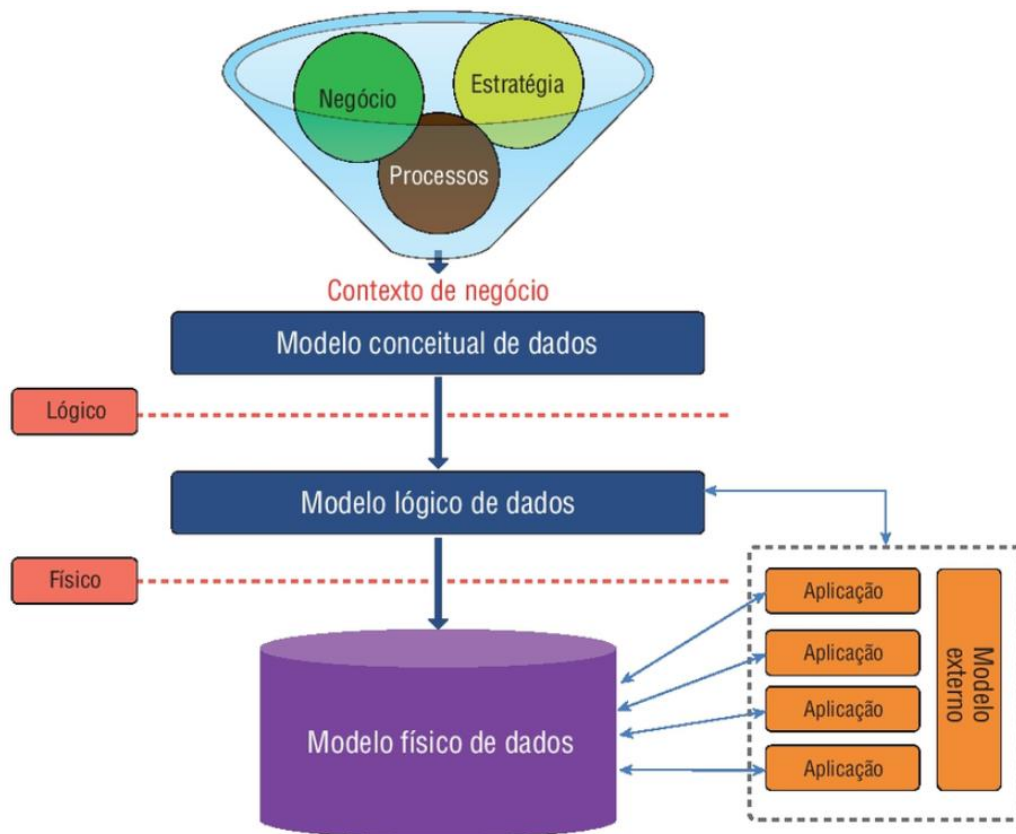
Fonte: Alves, 2014, p. 31.

A Figura 16 exemplifica um comando para a criação física das tabelas *Clientes* e *Produtos* no banco de dados. Observe que cada coluna possui um tipo de dado e um tamanho. A chave primária também foi definida nas duas tabelas. A linguagem utilizada na Figura 16 é a SQL e não se preocupe se você não entendeu os comandos, pois é assunto para a próxima aula.



Para finalizar os modelos de dados que foram estudados, a Figura 17 demonstra uma visão total dos estágios da modelagem de dados, podendo se ter a ideia do “posicionamento” dos três modelos vistos.

Figura 17 – Estágios da modelagem de dados



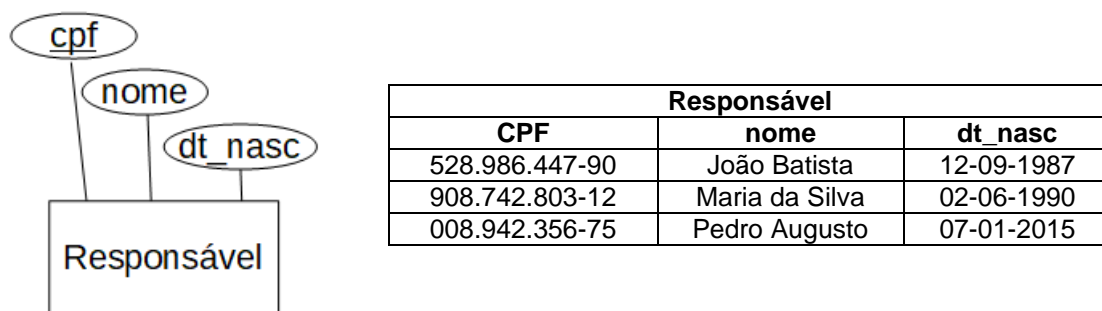
Fonte: Puga; França; Goya, 2013, p. 81.

4.3 Modelo relacional

O modelo ER é conveniente para representar um projeto de banco de dados inicial de alto nível. No modelo relacional, as entidades passam a ser tabelas e os atributos passam a ser colunas.

Heuser (2009, p. 120) define tabela como “um conjunto não ordenado de linhas (tuplas, na terminologia acadêmica)”. Tabela os dados (dispor em linhas e colunas) facilita a sua visualização e interpretação. A Figura 18 representa a transcrição da entidade *Responsável* e seus atributos para a tabela *Responsável*.

Figura 18 – Modelo ER X Modelo relacional



Analizando a Figura 18, a tabela *Responsável* possui três colunas: *cpf*, *nome*, *dt_nasc*. O atributo identificador (*cpf*) passa a ser a chave primária da tabela *Responsável*. Heuser (2009, p. 122) define chave primária (do inglês *primary key*) como “uma coluna ou uma combinação de colunas cujos valores distinguem uma linha das demais dentro de uma tabela”.

Outro conceito importante é o de *registro*. Os registros são os dados de uma mesma ocorrência, ou, os dados de uma mesma linha da tabela. A tabela da Figura 18 possui três registros, sendo que cada um deles possui dados nas três colunas, “*cpf*”, “*nome*” e “*dt_nasc*”. Podemos dizer que estes registros são informações cadastrais dos responsáveis nomeados por João, Maria e Pedro.

A nomeação dos elementos (tabelas, colunas, chaves, etc.) deve seguir o padrão do SGBD utilizado, e a grande maioria não aceita caracteres especiais (exceto o underline) nem espaço em branco. Exemplo:

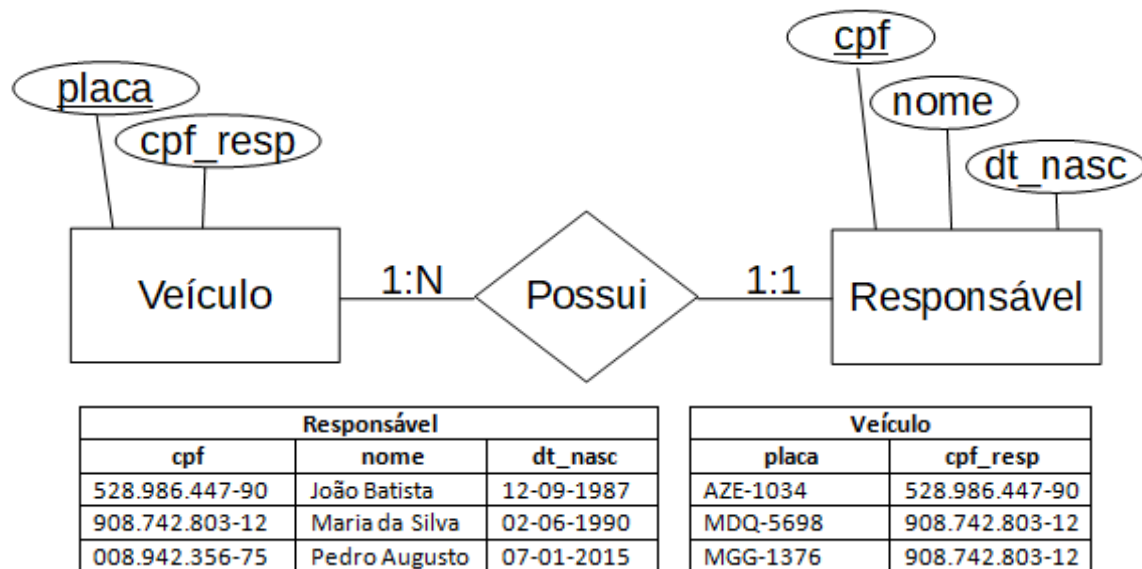
Tabela 2 – Nomeação dos elementos

Modelo ER	Modelo Relacional
Nome da mãe	nomeMae nomeDaMae nome_mae

A Figura 19 representa como é feito, fisicamente, um relacionamento entre duas tabelas.



Figura 19 – Relacionamento físico entre as tabelas



Ao analisar as duas tabelas (*Responsável* e *Veículo*), é possível observar que a coluna *cpf_resp* da tabela *Veículo* é a coluna que relaciona as duas tabelas. Na da tabela “*Veículo*”, essa coluna é uma chave estrangeira. Por meio desse relacionamento entre essas tabelas podemos observar que a responsável “Maria da Silva”, que tem o CPF 908.742.803-12, possui dois veículos (placas MDQ-5698 e MGG-1376), e que o responsável “Pedro Augusto” não possui veículos por não constar na tabela *Veículo*.

Para facilitar o entendimento desse processo de transformação de um modelo para um banco de dados, a Tabela 3 apresenta uma relação entre os elementos do modelo ER para o modelo relacional.

Tabela 3 – Modelo ER *versus* modelo relacional

Modelo ER	Modelo relacional
ENTIDADE	TABELA
ATRIBUTO SIMPLES	COLUNA
ATRIBUTO DERIVADO	COLUNA
ATRIBUTO IDENTIFICADOR	CHAVE PRIMÁRIA (OU SECUNDÁRIA)
ATRIBUTO MULTIVALORADO	NOVA TABELA E CHAVE ESTRANGEIRA
RELACIONAMENTO 1:1 OU 1:N	CHAVE ESTRANGEIRA
RELACIONAMENTO N:N	NOVA TABELA COM 2 CHAVES ESTRANGEIRAS
CONJUNTO DE VALORES	TIPO DE DADOS



4.4 Principais tipos de dados

Como explicado na aula 1, os tipos de dados são as últimas grandes evoluções nos bancos de dados, com bibliotecas digitais, vídeos interativos, etc. Na explicação a seguir, atente para os tipos de dados mais comuns e a sua utilização. Lembre-se de que, dependendo do SGBD escolhido, pode haver uma variação na nomenclatura dos tipos de dados. Neste material, utilizam-se os tipos de dados (do inglês *SQL Data types*) do banco de dados MySQL. No entanto, caso você tenha contato com outro banco de dados, deverá utilizar os tipos de dados desse outro sistema gerenciador.

Figura 20 – Números inteiros

TINYINT	1 Byte (0 a 255 ou -128 a 127)
SMALLINT	2 Bytes (0 a 65.535 ou -32.768 a 32.767)
MEDIUMINT	3 Bytes (0 a 16.777.215 ou -8,388,608 a 8,388,607)
INT	4 Bytes (0 a 4.294.967.295 ou -2,147,483,647 a 2,147,483,647)
BIGINT	8 Bytes
UNSIGNED	sem sinal (somente números positivos, dobra o limite dos números)

Fonte: Íparos, S.d.

Os tipos de dados da Figura 20 são utilizados para armazenar números inteiros, variando em tamanho utilizado para armazenar a coluna e a sua capacidade. Uma coluna do tipo TINYINT pode armazenar o número 255 como máximo.

Figura 21 – Armazenamento de arquivos

TINYBLOB	255 Bytes
BLOB	64KB
MEDIUMBLOB	16MB
LOB	4GB

Fonte: Íparos, S.d.

Os tipos de dados da Figura 21 são utilizados para armazenar arquivos como imagens e vídeos. O que vai diferenciar entre a escolha do tipo da coluna é, basicamente, o seu tamanho máximo de armazenamento.



Figura 22 – Tipo texto

CHAR	textos com um tamanho fixo de caracteres
VARCHAR	para textos de 2 a 255 caracteres
TINYTEXT	255 Bytes
TEXT	64KB
MEDIUMTEXT	16MB
LONGTEXT	4GB

Fonte: Íparos, S.d.

O tipo mais comum, para os armazenamentos de texto, é o *varchar*, pois é utilizado para armazenar colunas como nome, endereço, bairro, cidade etc.

Figura 23 – Tipo data

DATE	armazena a data no formato YYYY-MM-DD
TIME	armazena a hora no formato HH:MM:SS
DATETIME	armazena a data e a hora no formato YYYY-MM-DD HH:MM:SS

Fonte: Íparos, S.d.

Um fato curioso no armazenamento das datas é que o padrão é o americano, onde o ano é invertido com o dia. Geralmente a coluna do tipo DATETIME é utilizado na gravação automática de logs e acessos.

Figura 24 – Tipo decimal

FLOAT	números flutuantes
DOUBLE	números flutuantes com precisão dupla

Fonte: Íparos, S.d.

Os tipos de dados da Figura 24 também são muito utilizados, pois serve para armazenar os números não inteiros (com vírgula). Como exemplos têm salário, preço, quantidade etc.

Saiba mais

As 12 Regras de Codd

Ao definir o modelo relacional, Codd estabeleceu 12 regras para determinação de um banco de dados relacional.



Estas regras são usadas, portanto para se verificar a fidelidade de um banco de dados ao modelo relacional.

Na prática são poucos os gerenciadores de banco de dados que atendem a todas as 12 regras.

Na maior parte dos casos são atendidas no máximo 10 regras.

1. Toda informação num banco de dados relacional é apresentada a nível lógico na forma de tabelas;
2. Todo dado em um banco de dados relacional tem a garantia de ser logicamente acessível, recorrendo-se a uma combinação do nome da tabela, um valor de chave e o nome da coluna;
3. Tratamento sistemático de valores nulos; (ausência de informação)
4. O dicionário de dados, catálogo, do banco de dados é baseado no modelo relacional;
5. Há uma linguagem não procedural para a definição, manipulação e controle dos dados;
6. Tratamento das atualizações de visões dos dados;
7. Tratamento de alto nível para inserção, atualização e eliminação de dados;
8. Independência física dos dados; (mudança na memória e no método de acesso, criação de um novo índice, criação de uma nova coluna)
9. Independência lógica dos dados; (mudança no tamanho de uma coluna)
10. Restrição de Integridade; (Identidade, Referencial e Domínio)
11. Independência de Distribuição dos dados;
12. Não subversão das regras de integridade ou restrições quando se usa uma linguagem hospedeira (Siqueira, S.d.).

FINALIZANDO

Nesta aula foram aprofundados os conceitos sobre a modelagem conceitual e do modelo ER. Foram explanados os conceitos de entidades (fortes e fracas), relacionamentos (fortes e fracos), atributos (identificador, não identificado, multivalorados e compostos) e apresentados os tipos de relacionamentos com as suas cardinalidades. O estudo de caso é a parte da aula que merece uma atenção ímpar, pois nele transforma-se o conhecimento teórico para o conhecimento prático. Na continuação da aula, apresentaram-se os modelos lógicos e físicos, com suas características finalizando com a transformação de um modelo ER para um modelo relacional.



REFERÊNCIAS

ALVES, W. P. **Banco de dados**. 1. ed. São Paulo: Érica, 2014.

HEUSER, C. A. (Org.). **Projeto de banco de dados**. 6. ed. Porto Alegre: Bookman, 2009.

PUGA, S.; FRANÇA, E.; GOYA, M. **Banco de dados**: implementação em SQL, PL/SQL, Oracle 11g. São Paulo: Pearson Education do Brasil, 2013.

SILBERSCHATZ, A.; KORTH, H. F.; SUDARSHAN, S. **Sistemas de banco de dados**. São Paulo: Makron Books, 1999.

SIQUEIRA, F. de. Banco de Dados I. **Google Sites**. Disponível em <<https://sites.google.com/site/uniplibancodedados1/aulas/modelo-relacional>>. Acesso em: 19 jun. 2018.

TIPOS de dados básicos do Mysql. **Íparos**. Disponível em: <<http://www.iparos.com.br/cursos/php/mysql/tipos-de-dados-basicos-do-mysql>>. Acesso em: 19 jun. 2018.