

Modelo Conceitual

Banco de dados

Modelo Conceitual



Este é o modelo de mais alto nível, ou seja, que está mais próximo dos usuários.



Esse modelo pode ser elaborado de forma textual ou por meio de dois diagramas: Diagrama de Entidade e Relacionamento e/ou Diagrama de Classes (UML)

Finalidade



O modelo conceitual de banco de dados trata da modelagem, independentemente da plataforma ou implementação escolhida, representando as necessidades do mundo real.



É realizada a análise dos conjuntos de dados relevantes perante as necessidades do usuário final e então damos início às especificações do banco de dados através do modelo de entidade-relacionamento (MER)

MER



“

é um tipo de modelo descritivo conceitual, utilizado para descrever de forma abstrata os objetos que existirão em um banco de dados.



O MER é construído a partir dos objetos e conceitos de Entidade, Atributos e Relacionamentos.

Entidades

- São objetos do mundo real que têm importância para o sistema. Elas são qualquer coisa do mundo real que tenha um significado para o sistema e que tenha uma necessidade de armazenamento de dados.
 - Exemplos: cliente, pedido, produto, fornecedor, etc.
- Tipos:
- **Entidade Forte:** não depende de outra entidade.
- **Entidade Fraca:** depende de outra entidade.
- **Entidade Associativa:** uma entidade auxiliar, utilizada para associar duas outras entidades no caso de um relacionamento muitos para muitos.

DER
Símbologia
para
entidades

ENTIDADE

ENTIDADE
FRANCA

ASSOCIATIVA

O que é uma Entidade Associativa?

- No modelo conceitual, uma entidade associativa é uma entidade que **surge de um relacionamento Muitos para Muitos (N:M)** entre duas ou mais entidades. Ela é usada para resolver esse tipo de relacionamento, permitindo que você adicione atributos que pertencem *ao próprio relacionamento*, e não a uma das entidades envolvidas individualmente.
- Pense nela como uma "ponte" entre duas entidades que têm uma conexão complexa.

Exemplo:

- Imagine um relacionamento N:M clássico:
- ALUNO e CURSO
- Um aluno pode se matricular em vários cursos, e um curso pode ter vários alunos.
- Isso é um relacionamento N:M.
- Agora, pense em atributos específicos desse relacionamento:
- Qual é a data de matrícula do aluno naquele curso específico?
- Qual foi a nota final que o aluno obteve naquele curso?

Exemplo:

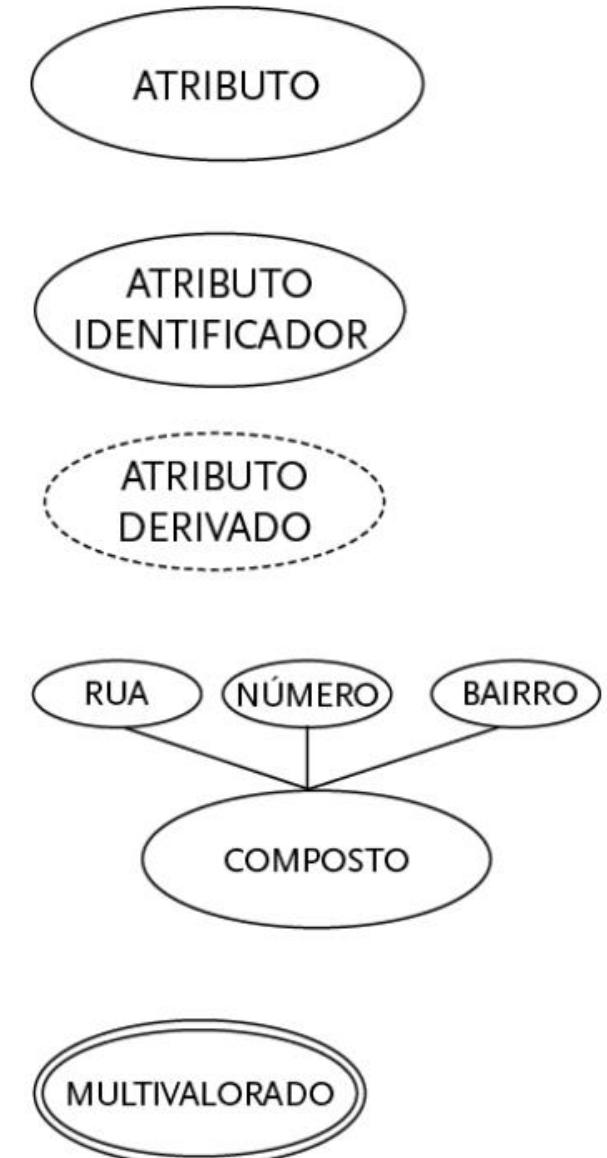
- Eses atributos (Data de Matrícula, Nota Final) não pertencem apenas ao ALUNO (um aluno pode ter várias datas de matrícula para cursos diferentes) e nem apenas ao CURSO (um curso tem várias notas, uma para cada aluno).
- Eles pertencem à combinação específica de um aluno em um curso.

Exemplo:

- Em vez de um relacionamento direto N:M entre ALUNO e CURSO, teríamos:
- Entidades: ALUNO, CURSO, MATRICULA (Entidade Associativa)
- Atributos:
 - ALUNO: ID_Aluno, Nome, DataNascimento
 - CURSO: ID_Curso, Titulo, CargaHoraria
 - MATRICULA: DataMatricula, NotaFinal
- Relacionamentos:
 - Um ALUNO realiza muitas MATRICULAS (1:N)
 - Um CURSO possui muitas MATRICULAS (1:N)

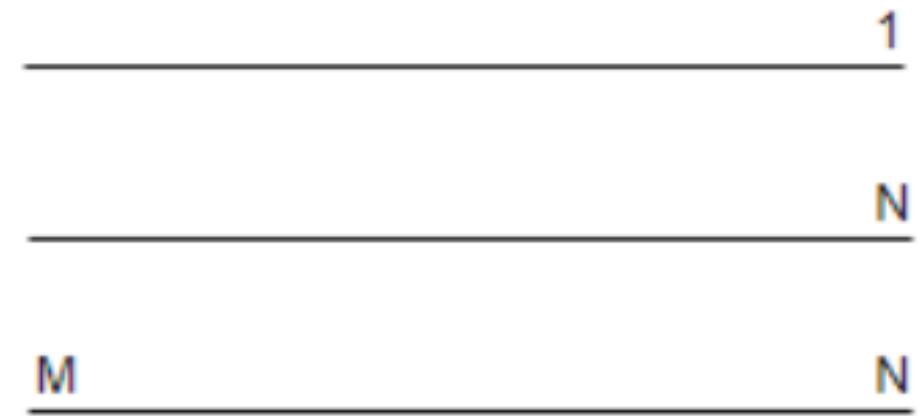
Atributos

Cada entidade possui uma ou mais características, chamadas de atributos.



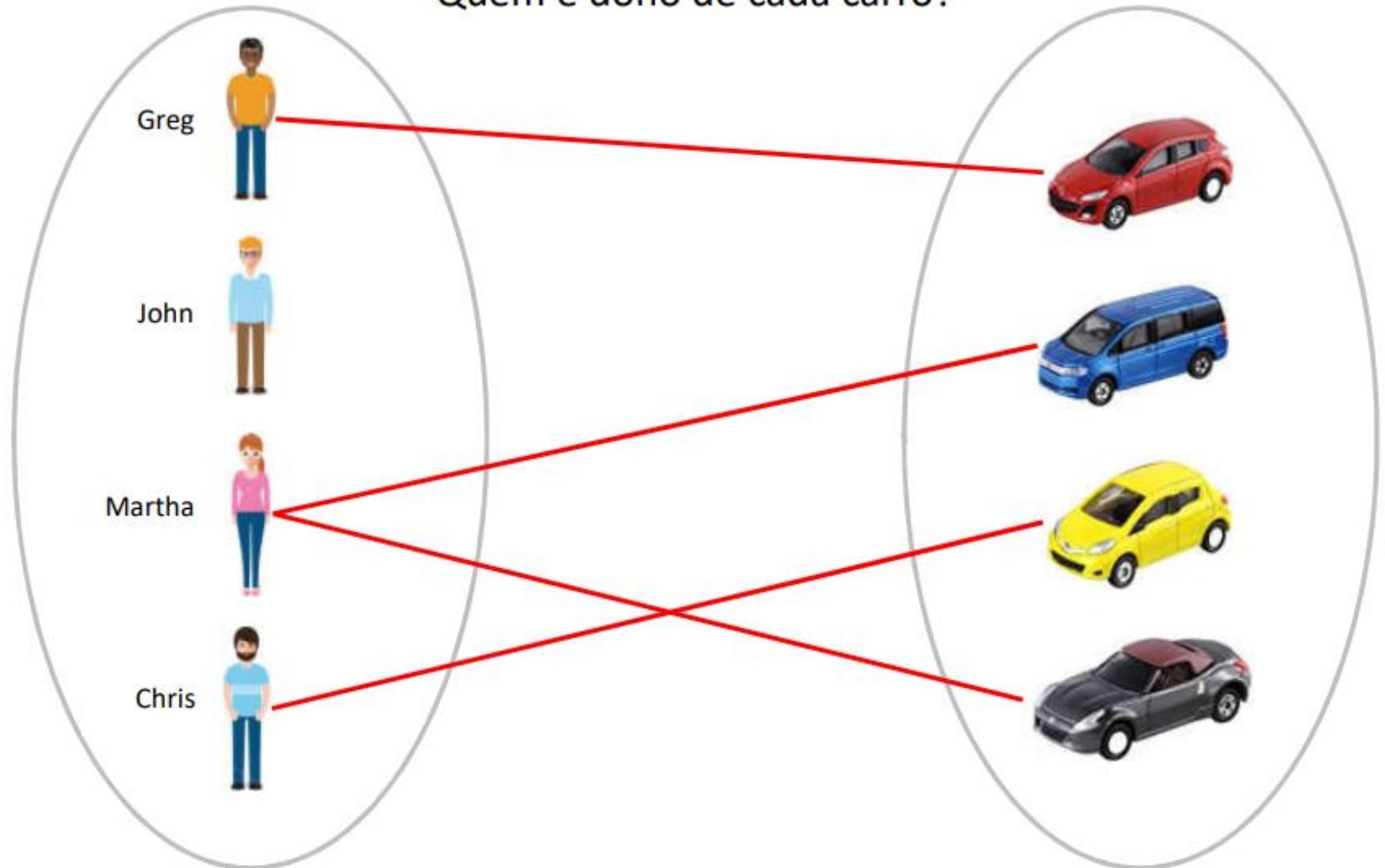
Relacionamento e cardinalidade

- Relacionamento é como essas entidades se associam.
- Tipos de cardinalidade:
 - **Um para um, ou unário (1:1)**: Esta regra é aplicada a entidades em que um registro pode ter ligação, ou seja, relacionamento com somente um registro de outra entidade.
 - Ex: Uma compra está relacionada a um cliente.
 - **Um para muitos (1:N)**: Esta regra segue a premissa que uma entidade pode ter uma ou mais associações com outra entidade.
 - EX: uma compra pode ter mais de um produto.
 - **Muitos para muitos, ou ternário (M:N)**: Neste relacionamento, uma entidade pode conter várias referências de uma outra entidade, e vice-versa.
 - Ex: um fornecedor pode se relacionar com vários produtos e vice-versa.



DER – Simbologia para relacionamento e cardinalidade

Quem é dono de cada carro?



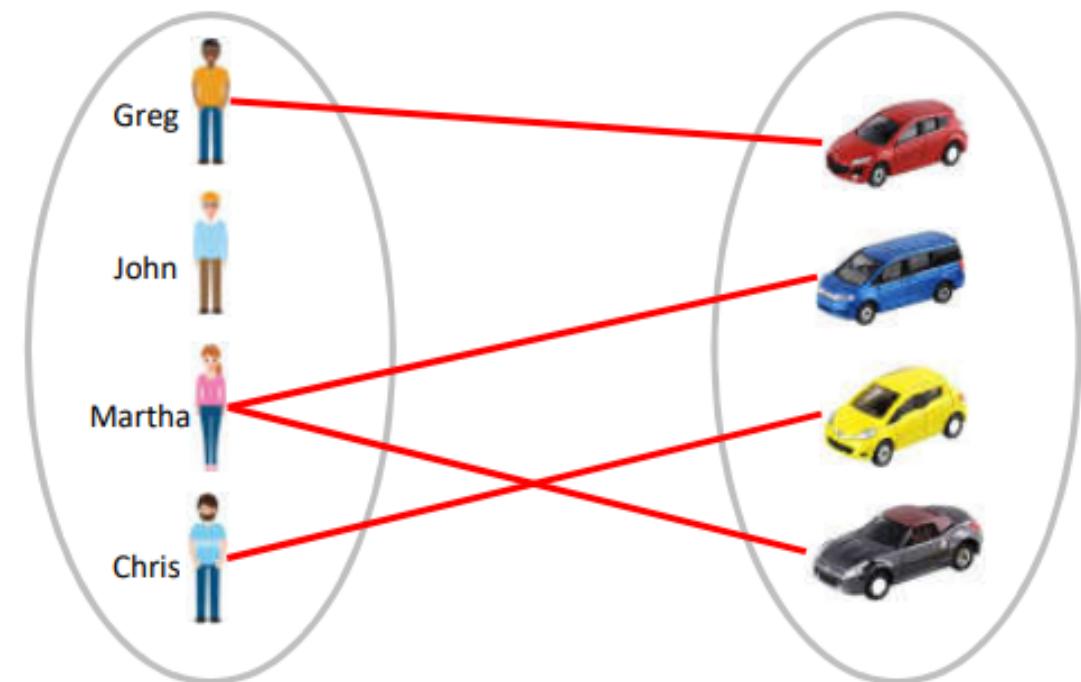
O que é multiplicidade?

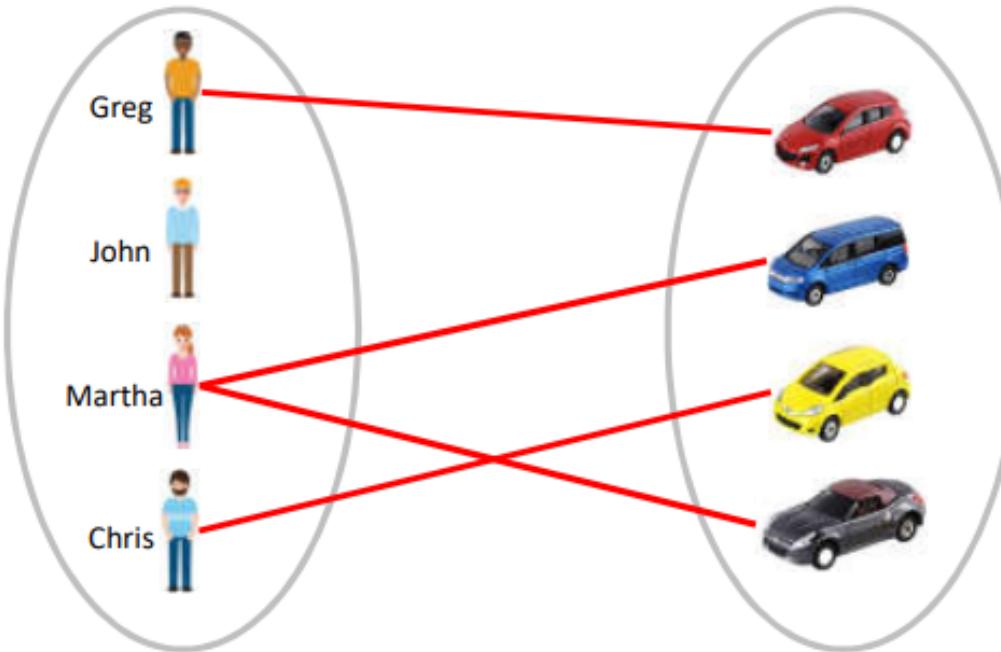
É a quantidade mínima e máxima de objetos que uma associação permite em cada um de seus papéis.

Exemplo: um carro pode ter quantos donos?

Mínimo: 1

Máximo: 1





Pergunte para os dois lados:

1 <conceito> **pode ter** quantos <papel>?

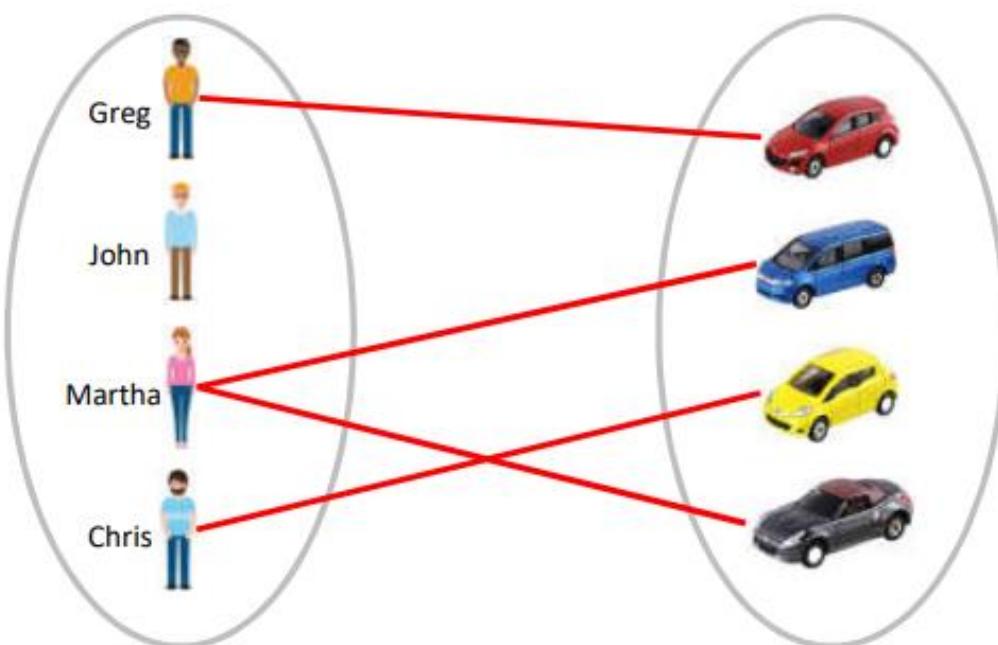
1 carro **pode ter** quantos **donos**?

1 pessoa **pode ter** quantos **carros**?

Atenção:
sempre inicie
a pergunta
com "1"

Um para muitos

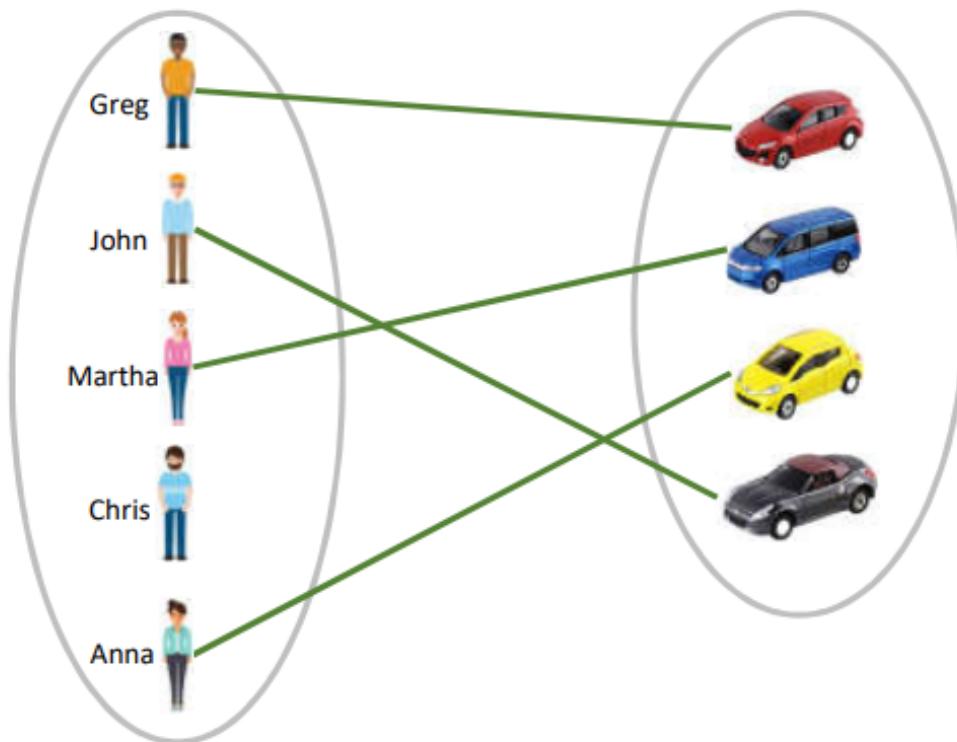
Exemplo: quem é dono de cada carro?



- Em um dos lados o máximo é 1
- No outro lado o máximo é "vários"

Um para um

Exemplo: quem é o responsável por cada carro?



1 carro pode ter quantos responsáveis?

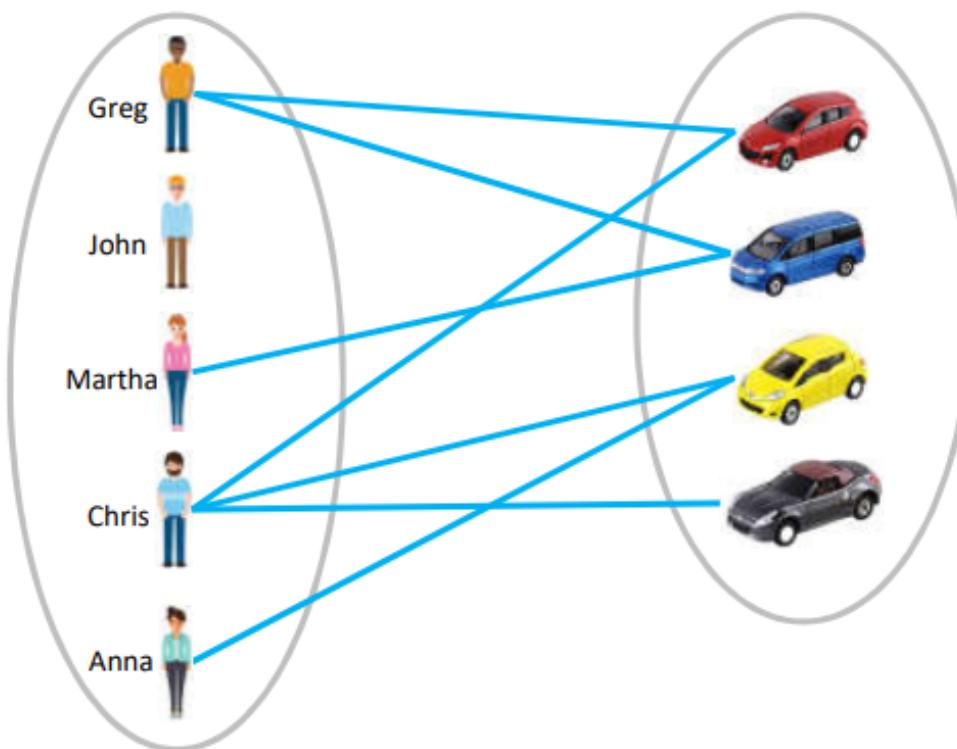
1 pessoa pode ser responsável por quantos carros?



- Em ambos os lados o máximo é 1

Muitos para muitos

Exemplo: quem dirige cada carro?



1 carro pode ter quantos motoristas?

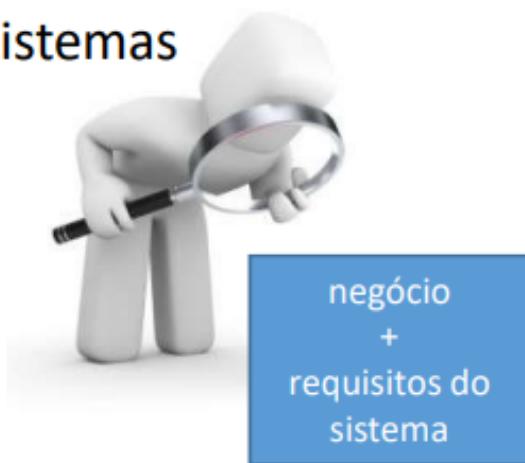
1 pessoa pode dirigir quantos carros?



- Em ambos os lados o máximo é "vários"

Orientações para identificação de conceitos

Analista de
sistemas



Com base nas informações, identificar conceitos relevantes para o negócio e com necessidade de armazenamento.

- Documentos (ordem de serviço, orçamento)
- Pessoas (cliente, fornecedor)
- Estruturas organizacionais (departamento)
- Eventos (venda, reserva, atendimento)

Orientações para identificação de conceitos

Analista de
sistemas



Atenção! Procure por:

Substantivos (pessoa, compra, produto, pagamento)

Expressões que denotem substantivos
(autorização de pagamento)

Verbos que indiquem um possível conceito
(comprar, pagar)

Exemplo 1 (especificação estilo "visão geral do sistema")

Deseja-se fazer um sistema para manter um cadastro dos funcionários de uma empresa. Deseja-se poder consultar o email e salário dos funcionários, bem como o telefone de seu departamento.

Produzindo o MER

- **Necessidade:** Para o crescimento da empresa, o diretor decidiu investir em um sistema, e para isso necessita que sejam mapeadas as necessidades operacionais, a fim de que a modelagem do banco de dados do sistema seja entregue e o banco de dados implementado. O sistema deve ser capaz de armazenar informações de **produtos**, **clientes**, **vendas realizadas**, **funcionários** e seus respectivos **departamentos**.
- Observações: Na loja, um cliente pode fazer compras, e cada compra pode conter diversos produtos. Os produtos vendidos são fornecidos por diversos fornecedores. As vendas são realizadas por um funcionário que pertence a somente um departamento. Os produtos vendidos são fornecidos por diversos fornecedores.

Produzindo o MER

- **Cenário proposto**
- **Situação atual:** Uma loja de produtos diversos atualmente utiliza planilhas para a organização das vendas realizadas. Porém, algumas vezes os funcionários anotam as transações em papel, ou não preenchem todas as informações solicitadas nas planilhas. Com isso, a empresa vem sofrendo prejuízos, perda de informação e retrabalho. A empresa possui **funcionários** e **fornecedores**.
- A cada transação de venda, pode haver **produtos** de mais de um fornecedor, porém a venda não pode ser compartilhada entre dois ou mais **clientes**.

Produzindo o MER

- Produto
- Cliente
- Compra (venda)
- Funcionário
- Departamento
- Produto
- Fornecedor

Produzindo o MER

CLIENTE (codigo, nome, endereco, cpf, telefone)
Entidade forte: CLIENTE

Atributo identificador: codigo

Atributo composto: endereco

Atributos simples: nome, cpf

Atributo multivalorado: telefone

COMPRA (codigo, quantidade_produtos)

Entidade fraca: COMPRA

Atributo identificador: codigo

Atributo derivado: quantidade_produtos

FUNCIONARIO (codigo, nome, cpf, telefone)

Entidade forte: FUNCIONARIO

Atributo identificador: codigo

Atributos simples: nome, cpf

Atributo multivalorado: telefone

DEPARTAMENTO (codigo, descricao)

Entidade forte: DEPARTAMENTO

Atributo identificador: codigo

Atributo simples: descricao

Produzindo o MER

FORNECEDOR (codigo, nome, telefone)

Entidade forte: FORNECEDOR

Atributo identificador: codigo

Atributo simples: nome

Atributo multivalorado: telefone

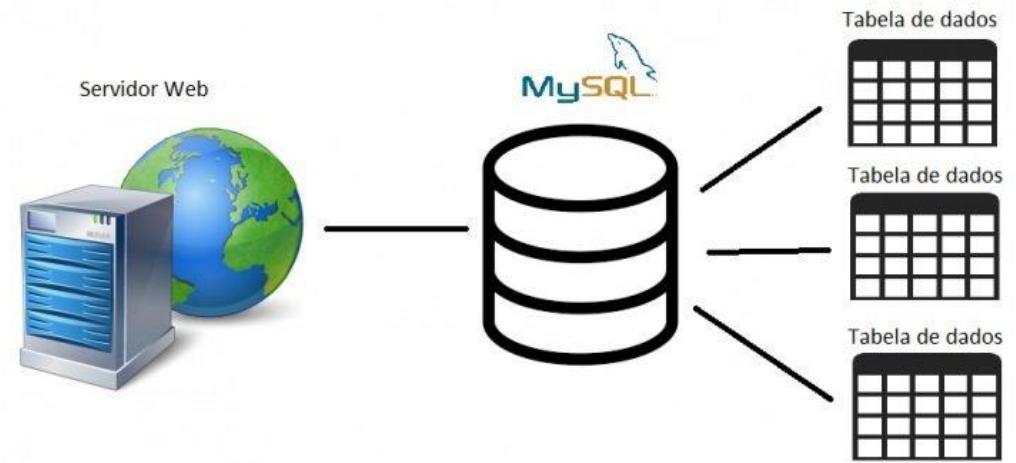
PRODUTO (codigo, descrição, valor)

Entidade forte: PRODUTO

Atributo identificador: codigo

Atributos simples: descricao, valor

Acessar o site: Draw.io



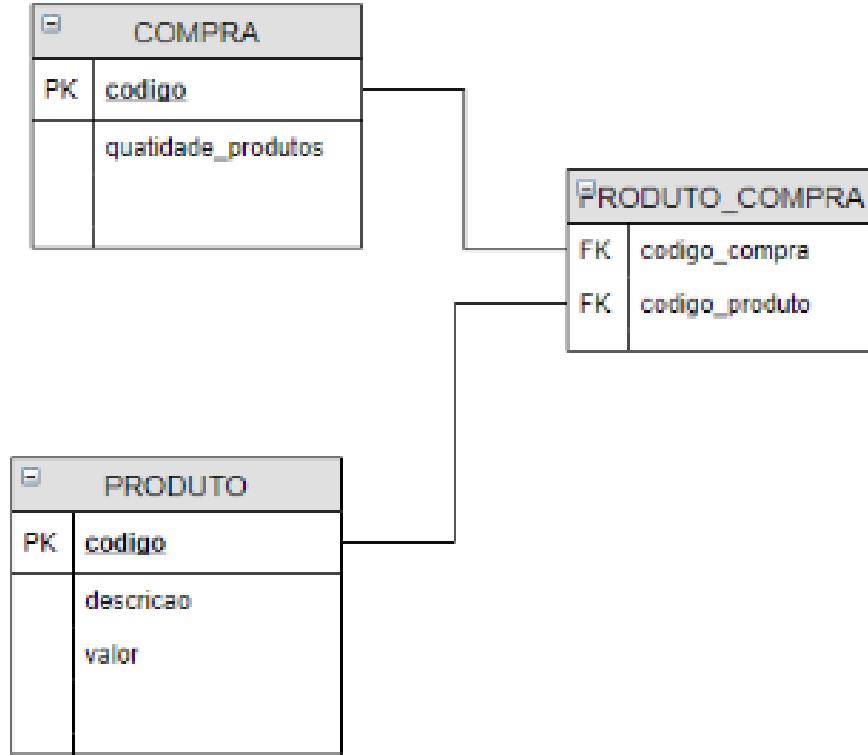
Modelo Lógico

Banco de dados

Modelo Lógico

- A modelagem lógica trata da definição da estrutura do banco de dados onde serão armazenados os dados, descrevendo detalhadamente suas tabelas, campos e relacionamentos.
- Essa modelagem define como será a implementação no SGBD.

Modelo Lógico



- **Atributo Chave:** Também conhecido como chave primária, primary key ou pk, é um campo que deve possuir valor único (sem repetição) em todo o conjunto de dados da entidade. Este atributo é usado para identificar unicamente um registro da tabela.
- **Chave Estrangeira:** Também conhecido como foreign key ou fk, é um atributo presente em uma entidade que indica um relacionamento e representa a chave primária de uma outra entidade.

Modelo Lógico - normalização

- Em 1970, Edgar F. Codd apresentou o conceito de normalização, que visa analisar e melhorar a estrutura de um banco de dados por meio de um processo lógico para realizar a divisão de uma tabela complexa em uma ou mais tabelas simples, construído em torno do conceito de formas normais.
- O conceito apresentado por Codd possui três formas normais que são adotadas pela maioria dos bancos de dados, chamadas de primeira, segunda e terceira forma normal (abreviadas como 1NF, 2NF e 3NF, respectivamente).

Regras de normalização

- **Primeira forma normal (1NF):** As informações são armazenadas em uma tabela na qual cada coluna contém valores atômicos. cada célula da tabela contenha um único valor; cada registro seja exclusivo. Ou seja, não pode ser multivalorado. Nem composto.
- **Segunda forma normal (2NF):** A tabela tem que atender à regra da primeira forma normal e todas as colunas dependem da chave primária da tabela.
- **Terceira forma normal (3NF):** A tabela está na segunda forma normal e todas as colunas dependam somente da chave primária da tabela, sem que haja dependência com outras colunas além da chave primária.

Primeira Forma Normal (1FN)

Uma entidade estará na primeira forma normal (1FN) se todos os campos forem atômicos (simples) e não multivalorados (com múltiplos valores).

Exemplo: Tabela Cliente

Código_cliente	Nome	Telefone	Endereço
C001	José	9563-6352 9847-2501	Rua Seis, 85 Morumbi 12536-965
C002	Maria	3265-8596	Rua Onze, 64 Moema 65985-963
C003	Janio	8545-8956 9598-6301	Praça ramos Liberdade 68858-633

Note que temos dois problemas aqui que estão em desacordo com a 1FN:

- 1 - O campo telefone é multivalorado;
- 2 - O campo endereço é multivalorado;

Primeira Forma Normal (1FN)

Uma entidade estará na primeira forma normal (1FN) se todos os campos forem atômicos (simples) e não multivalorados (com múltiplos valores).

Exemplo: Tabela Cliente

Código_cliente	Nome	Telefone	Endereço
C001	José	9563-6352 9847-2501	Rua Seis, 85 Morumbi 12536-965
C002	Maria	3265-8596	Rua Onze, 64 Moema 65985-963
C003	<u>Janio</u>	8545-8956 9598-6301	Praça ramos Liberdade 68858-633

Como resolver?

Note que o campo endereço possui sempre 3 partes: rua, bairro e cep. Podemos então dividir isso em 3 campos nesta mesma tabela.

Primeira Forma Normal (1FN)

Uma entidade estará na primeira forma normal (1FN) se todos os campos forem atômicos (simples) e não multivalorados (com múltiplos valores).

Exemplo: Tabela Cliente

Código_cliente	Nome	Telefone	Rua	Bairro	Cep
C001	José	9563-6352 9847-2501	Rua Seis, 85	Morumbi	12536-965
C002	Maria	3265-8596	Rua Onze, 64	Moema	65985-963
C003	Janio	8545-8956 9598-6301	Praça ramos	Liberdade	68858-633

Como resolver?

Desta forma resolvemos um dos problemas. Mas e quanto ao telefone? Em algumas entradas temos somente um número e em outras temos dois números. Podemos fazer o mesmo? Ou seja, criar dois novos campos nesta mesma tabela: telefone1 e telefone2?

Primeira Forma Normal (1FN)

Uma entidade estará na primeira forma normal (1FN) se todos os campos forem atômicos (simples) e não multivalorados (com múltiplos valores).

Exemplo: Tabela Cliente e Tabela Telefones Cliente

Código_cliente	Nome	Rua	Bairro	Cep
C001	José	Rua Seis, 85	Morumbi	12536-965
C002	Maria	Rua Onze, 64	Moema	65985-963
C003	Janio	Praça ramos	Liberdade	68858-633

Código_cliente	Telefone
C001	9563-6352
C001	9847-2501
C002	3265-8596
C003	8545-8956
C003	9598-6301

Pronto! Nossa tabela inicial, Cliente, está normalizada na 1FN e pode passar para a próxima Forma Normal.

OBS: O processo de normalização é sequencial. Isso significa que você só pode aplicar as regras da segunda forma normal se as tabelas já se encontrarem na primeira forma normal.

Segunda Forma Normal - 2FN

Uma entidade estará na 2FN se ela já se encontrar na 1FN e todos os atributos não chave forem totalmente dependentes da chave primária.

Exemplo: Tabela Pedidos

N_pedido	Codigo_produto	Produto	Quant	Valor_unit	Subtotal
1005	1-934	Impressora laser	5	1.500,00	7.500,00
1006	1-956	Impressora desjet	3	350,00	1.050,00
1007	1-923	Impressora matricial	1	190,00	190,00
1008	1-908	Impressora mobile	6	980,00	5.880,00

Qual o problema aqui?

Segunda Forma Normal - 2FN

Uma entidade estará na 2FN se ela já se encontrar na 1FN e todos os atributos não chave forem totalmente dependentes da chave primária.

Exemplo: Tabela Pedidos e Tabela Produtos

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Codigo_produto	Produto
1-934	Impressora laser
1-956	Impressora desjet
1-923	Impressora matricial
1-908	Impressora mobile

Pronto! Nossa tabela está dentro da 2FN e pronta para ser validada com a 3FN.

Veja que a partir desse momento, o código do produto da tabela de Pedidos faz referência à chave primária da tabela de Produtos, portanto há um relacionamento entre as tabelas.

OBS: O valor unitário poderia/deveria estar na tabela de Produtos e não na tabela Pedidos, mas está aqui para ajudarmos a entender a 3FN que estudaremos na próxima aula.

Terceira Forma Normal (3FN)

Cada uma das formas normais tende a ir refinando a modelagem e deixando a estrutura de dados mais íntegra e exclusiva, evitando repetições desnecessárias e possíveis sobrecarga no gerenciador de banco de dados.

Uma tabela estará na 3FN se ela estiver na 2FN e se nenhuma coluna não-chave depender de outra coluna não-chave.

Ou seja, na 3FN temos simplesmente que eliminar os campos que podem ser obtidos pela equação de outros campos da mesma tabela.

Exemplo: **Tabela Pedidos**

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Qual o problema aqui?

Veja que se multiplicarmos a quantidade do produto pelo seu preço unitário teremos o subtotal.

Portanto o campo subtotal é desnecessário e deve ser removido.

Terceira Forma Normal (3FN)

Cada uma das formas normais tende a ir refinando a modelagem e deixando a estrutura de dados mais íntegra e exclusiva, evitando repetições desnecessárias e possíveis sobrecarga no gerenciador de banco de dados.

Uma tabela estará na 3FN se ela estiver na 2FN e se nenhuma coluna não-chave depender de outra coluna não-chave.

Ou seja, na 3FN temos simplesmente que eliminar os campos que podem ser obtidos pela equação de outros campos da mesma tabela.

Exemplo: **Tabela Pedidos**

N_pedido	Codigo_produto	Quant	Valor_unit	Subtotal
1005	1-934	5	1.500,00	7.500,00
1006	1-956	3	350,00	1.050,00
1007	1-923	1	190,00	190,00
1008	1-908	6	980,00	5.880,00

Terceira Forma Normal (3FN)

Cada uma das formas normais tende a ir refinando a modelagem e deixando a estrutura de dados mais íntegra e exclusiva, evitando repetições desnecessárias e possíveis sobrecarga no gerenciador de banco de dados.

Uma tabela estará na 3FN se ela estiver na 2FN e se nenhuma coluna não-chave depender de outra coluna não-chave.

Ou seja, na 3FN temos simplesmente que eliminar os campos que podem ser obtidos pela equação de outros campos da mesma tabela.

Exemplo: **Tabela Pedidos**

N_pedido	Codigo_produto	Quant	Valor_unit
1005	1-934	5	1.500,00
1006	1-956	3	350,00
1007	1-923	1	190,00
1008	1-908	6	980,00

Pronto! Temos uma tabela normalizada, dentro das 3 formas normais.

Terceira Forma Normal (3FN)

Cada uma das formas normais tende a ir refinando a modelagem e deixando a estrutura de dados mais íntegra e exclusiva, evitando repetições desnecessárias e possíveis sobrecarga no gerenciador de banco de dados.

Uma tabela estará na 3FN se ela estiver na 2FN e se nenhuma coluna não-chave depender de outra coluna não-chave.

Ou seja, na 3FN temos simplesmente que eliminar os campos que podem ser obtidos pela equação de outros campos da mesma tabela.

Exemplo: **Tabela Pedidos**

N_pedido	Codigo_produto	Quant	Valor_unit
1005	1-934	5	1.500,00
1006	1-956	3	350,00
1007	1-923	1	190,00
1008	1-908	6	980,00

OBS: Lembra que eu falei sobre o campo 'Valor unitário' durante a 2FN?

O campo valor unitário pertence à tabela Produtos, e não deveria estar aqui.

Pronto! Temos uma tabela normalizada, dentro das 3 formas normais.

Terceira Forma Normal (3FN)

Cada uma das formas normais tende a ir refinando a modelagem e deixando a estrutura de dados mais íntegra e exclusiva, evitando repetições desnecessárias e possíveis sobrecarga no gerenciador de banco de dados.

Uma tabela estará na 3FN se ela estiver na 2FN e se nenhuma coluna não-chave depender de outra coluna não-chave.

Ou seja, na 3FN temos simplesmente que eliminar os campos que podem ser obtidos pela equação de outros campos da mesma tabela.

Exemplo: **Tabela Pedidos e Tabela Produtos**

N_pedido	Codigo_produto	Quant
1005	1-934	5
1006	1-956	3
1007	1-923	1
1008	1-908	6

Codigo_produto	Produto	Valor_unit
1-934	Impressora laser	1.500,00
1-956	Impressora desjet	350,00
1-923	Impressora matricial	190,00
1-908	Impressora mobile	980,00

Pronto! Temos uma tabela normalizada, dentro das 3 formas normais.



Descrição do Negócio: Loja de Venda de Tênis

- Uma loja de tênis precisa armazenar informações sobre seus **tênis**, seus **clientes** e as **vendas** que são realizadas.
- Cada **tênis** possui um nome, marca, tamanho, cor, preço unitário e a quantidade em estoque.
- Os **clientes** são identificados por um nome e um e-mail.
- Uma **venda** registra qual cliente fez a compra, a data da venda e pode incluir um ou mais tênis. Para cada tênis em uma venda, é importante saber qual a quantidade comprada.



Panorama Geral do Modelo Lógico Simplificado

- Para essa situação, precisaremos de tabelas para:
- **Tênis:** Para os dados dos produtos que a loja vende.
- **Cliente:** Para os dados dos compradores.
- **Venda:** Para registrar cada transação de compra.
- **ItemVenda:** Uma tabela associativa para detalhar quais tênis foram comprados em cada venda e em que quantidade.

Modelo Lógico Detalhado (Tabela: Tênis)



- **id_tênis (PK)**: Inteiro, Chave Primária, Auto-incremento. É um identificador único para cada tipo de tênis.
- **nome (VARCHAR)**: Texto (ex: "Air Max 270"), obrigatório.
- **marca (VARCHAR)**: Texto (ex: "Nike", "Adidas"), obrigatório.
- **tamanho (DECIMAL(4,1))**: Número decimal (ex: 42.5), obrigatório.
- **cor (VARCHAR)**: Texto (ex: "Preto", "Branco"), obrigatório.
- **preço_unitário (DECIMAL(10,2))**: Valor monetário do tênis, obrigatório.
- **quantidade_estoque (INT)**: Inteiro, para controlar quantos tênis desse modelo estão disponíveis, obrigatório.

Modelo Lógico Detalhado (Tabela: Cliente)



- **id_cliente (PK)**: Inteiro, Chave Primária, Auto-incremento.
Identificador único para cada cliente.
- **nome_cliente (VARCHAR)**: Texto completo do nome do cliente,
obrigatório.
- **email (VARCHAR)**: Endereço de e-mail do cliente, obrigatório e **único**
(não pode haver dois clientes com o mesmo e-mail).

Modelo Lógico Detalhado (Tabela: Venda)



- **id_venda (PK)**: Inteiro, Chave Primária, Auto-incremento. Identificador único para cada venda.
- **data_venda (DATE)**: A data em que a venda foi realizada, obrigatório.
- **id_cliente (FK)**: Inteiro, Chave Estrangeira referenciando Cliente.id_cliente. Isso conecta cada venda a um cliente específico (Um cliente pode fazer várias vendas, mas uma venda é feita por apenas um cliente - relacionamento 1:N).

Modelo Lógico Detalhado (ItensVenda)



- **id_venda (PK, FK):** Inteiro, Chave Estrangeira referenciando Venda.id_venda. Faz parte da chave primária composta.
- **id_tênis (PK, FK):** Inteiro, Chave Estrangeira referenciando Tênis.id_tênis. Faz parte da chave primária composta.
- **quantidade_comprada (INT):** Inteiro, a quantidade de um tênis específico que foi comprado nesta venda, obrigatório.

Representação dos Relacionamentos



- **Cliente (1) --- (N) Venda:** Um cliente pode realizar várias vendas, mas uma venda é associada a um único cliente.
- Implementação: `id_cliente` como FK na tabela `Venda`.
- **Venda (N) --- (M) Tênis:** Uma venda pode conter vários tênis, e um tênis pode aparecer em várias vendas.
- Implementação: Tabela associativa `ItemVenda` com `id_venda` e `id_tênis` como Chave Primária Composta (PK) e Chaves Estrangeiras (FK).

Cenário - Pizzaria

- Uma pizzaria de tele-entrega apresenta um cardápio composto por diversos tipos de pizza, cujos dados são: número do item, nome da pizza, lista de ingredientes e preços das pizzas pequena, média e grande, como por exemplo: (12, francesa, (queijo, presunto, champignon, espargos), 10.00, 15.00, 18.00). Na pizzaria trabalham funcionários que emitem pedidos de pizzas. Cada pedido possui um número e uma data de emissão, além do nome, telefone e endereço do cliente que solicitou o pedido. Um pedido, emitido por apenas um funcionário, é composto por vários itens: cada item possui um número e se refere a uma determinada pizza do cardápio, em um determinado tamanho (pequena, média ou grande) e em uma certa quantidade. Dos funcionários entregadores (ou seja, responsáveis pela entrega de um ou mais pedidos), deve-se saber o número do telefone celular para um eventual contato durante uma entrega. Uma entrega destina-se a um bairro, e para cada bairro existe um tempo máximo de espera para a entrega de um pedido. Defina outras entidades/atributos que julgar relevante.

Cenário - Fábrica de picolé

Exercício Modelagem de Dados

Uma empresa fabricante de picolés deseja armazenar informações acerca de seus negócios. Os picolés fabricados são divididos em normal (com água) e ao leite. As informações comuns armazenadas dos picolés são: sabor, ingredientes, preço e tipo da embalagem. Especificamente, picolés normais possuem um conjunto de aditivos nutritivos (vitaminas ou sais minerais) cada um com nome e fórmula química; e picolés ao leite contêm um conjunto de conservantes, cada um com nome e descrição. Os dois tipos de picolés são vendidos em lotes exclusivos (ou normais, ou ao leite) para os revendedores e cada venda gera uma nota fiscal que pode conter um ou vários lotes. As notas fiscais possuem data, valor, número de série e descrição. Todo revendedor possui uma pessoa de contato para eventuais resoluções de problemas, além disso, armazena-se do revendedor o CNPJ e a razão social. Deseja-se obter relatórios sobre as vendas mensais dos picolés de cada tipo e quais revendedores compraram mais picolés nos últimos meses.

Cenário - Registro

Exercício 1 (RESOLVIDO): Deseja-se construir um sistema para manter um registro de artistas musicais e seus álbuns. Cada álbum possui várias músicas, as quais poderão ser consultadas pelo sistema. O sistema também deve permitir a busca de artistas por nome ou nacionalidade. O sistema também deve ser capaz de exibir um relatório dos álbuns de um artista, o qual pode ser ordenado por nome, ano, ou duração total do álbum. Um álbum pode ter a participação de vários artistas, sem distinção. Já a música pode possuir um ou mais autores e intérpretes (todos considerados artistas).

Cenário - Sistema Escolar

Visão geral do sistema: documento de formato livre que especifica, em linhas gerais, os requisitos do sistema.

Sistema de Controle Escolar

Deseja-se construir um sistema acadêmico. Para isso, são registrados os cursos disponíveis, onde cada um possui um nome, carga horária e valor. Quando um curso vai ser oferecido, é registrada uma turma, informando os seguintes dados: número da turma, data de início e número de vagas. Uma matrícula de um aluno em uma turma consiste na data de matrícula e no número de prestações em que o aluno vai pagar o curso. Para cada aluno, é necessário cadastrar seu nome, cpf, e data de nascimento.

Cada aluno passa por várias avaliações durante o desenrolar do curso que está cursando. Uma avaliação possui nota e data. Depois que a avaliação ocorre, é registrado resultado de cada aluno da turma (a nota que ele tirou). Um aluno é aprovado em um curso se sua nota total for pelo menos 70% da nota prevista do curso.

É importante saber a porcentagem de aprovação por turma e por curso (considerando somente as turmas que já finalizadas). Deseja-se saber também a nota final de um aluno em um curso que ele cursou, e se ele foi aprovado ou não no curso. Além disso, o sistema deve ser capaz de saber os alunos aprovados e reprovados em uma turma, bem como o aluno com melhor desempenho da turma (pode haver empates).

Cenário - Esporte

Exercício 2: Deseja-se construir um sistema para gerenciar as informações de campeonatos de handebol, que ocorrem todo ano. Deseja-se saber nome, data de nascimento, gênero e altura dos jogadores de cada time, bem qual deles é o capitão de cada time. Cada partida do campeonato ocorre em um estádio, que possui nome e endereço. Cada time possui seu estádio-sede e, assim, cada partida possui um time mandante (anfitrião) e o time visitante. O sistema deve ser capaz de listar as partidas já ocorridas e não ocorridas de um campeonato. O sistema deve também ser capaz de listar a tabela do campeonato, ordenando os times por classificação, que é calculada em primeiro lugar por saldo de vitórias e em segundo lugar por saldo de gols.

Cenário - Rede Social

Exercício 3: Deseja-se fazer um sistema de rede social. Nesta rede social, os usuários podem seguir e ser seguidos por outros usuários. O perfil do usuário deve permitir cadastrar nome, email, data de nascimento, website, gênero, telefone e foto do perfil. Os usuários podem fazer postagens de texto em sua própria "linha do tempo" (*timeline*) da rede social, sendo que podem anexar também fotos às postagens. Uma foto é referenciada pela URI de seu local de armazenamento. As fotos podem ser organizadas em álbuns, sendo que cada álbum possui um título.