

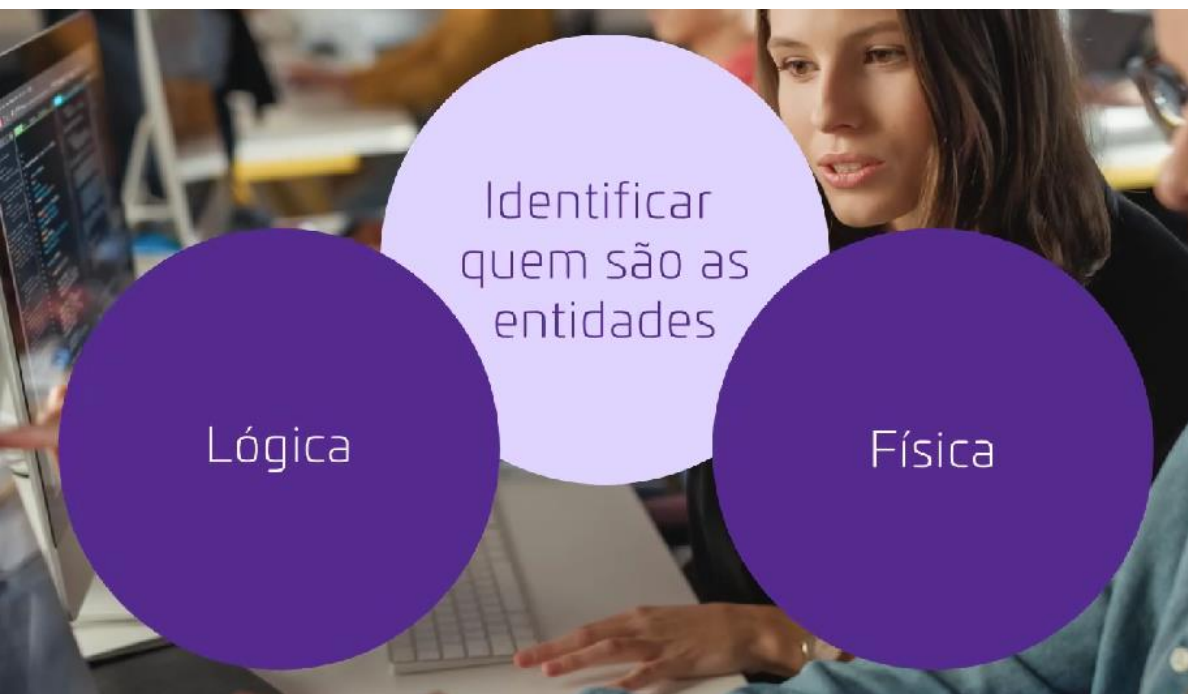
Entidades e atributos

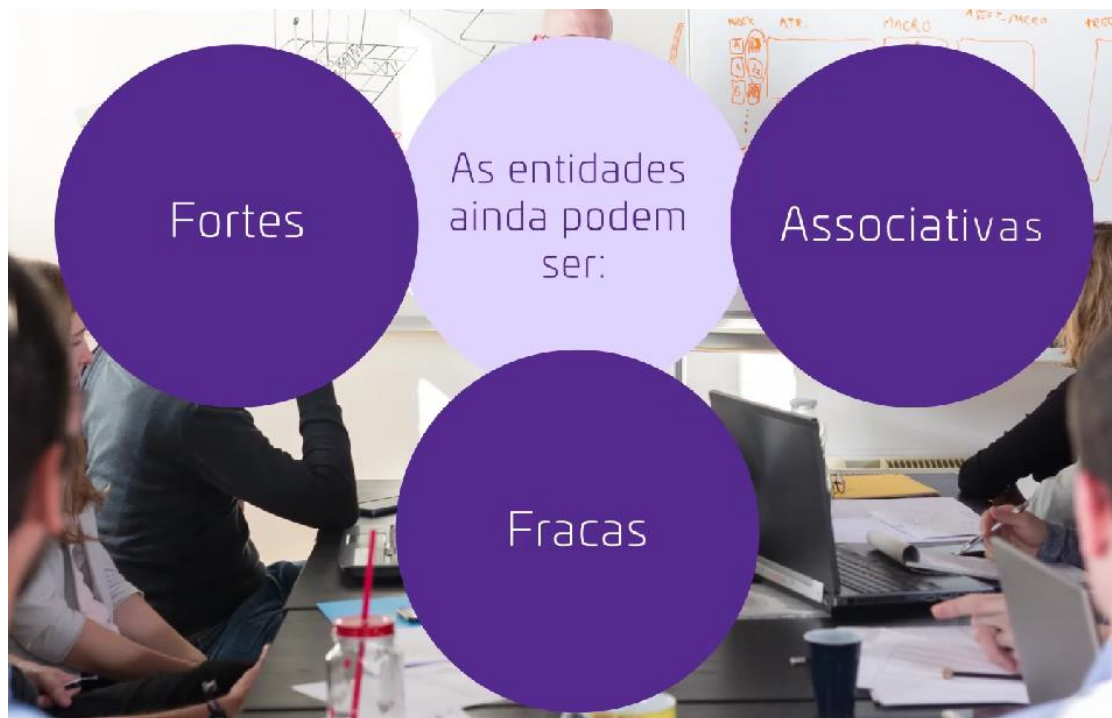
- A **entidade** pode ser definida como um objeto do mundo real, concreto ou abstrato, com existência independente.
- Os **atributos** compõem um conjunto particular de propriedades atreladas à entidade.

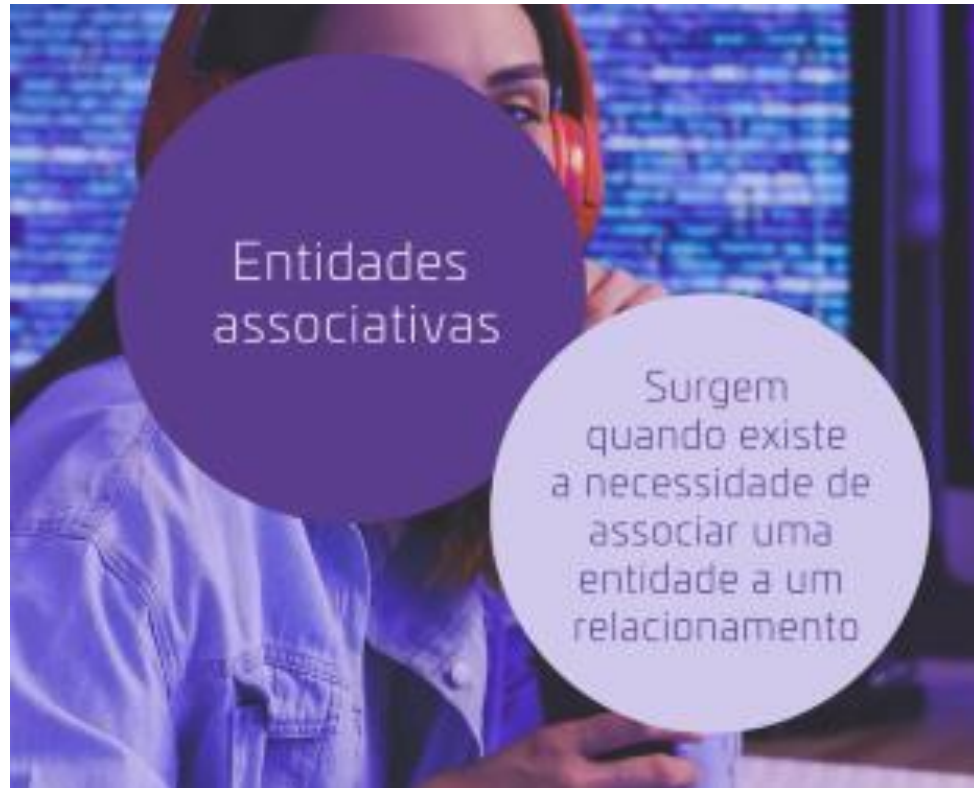


Exemplo de entidades e atributos









“Um para um”

As duas entidades envolvidas tem referência obrigatória a apenas uma unidade da outra



“Muitos para muitos”

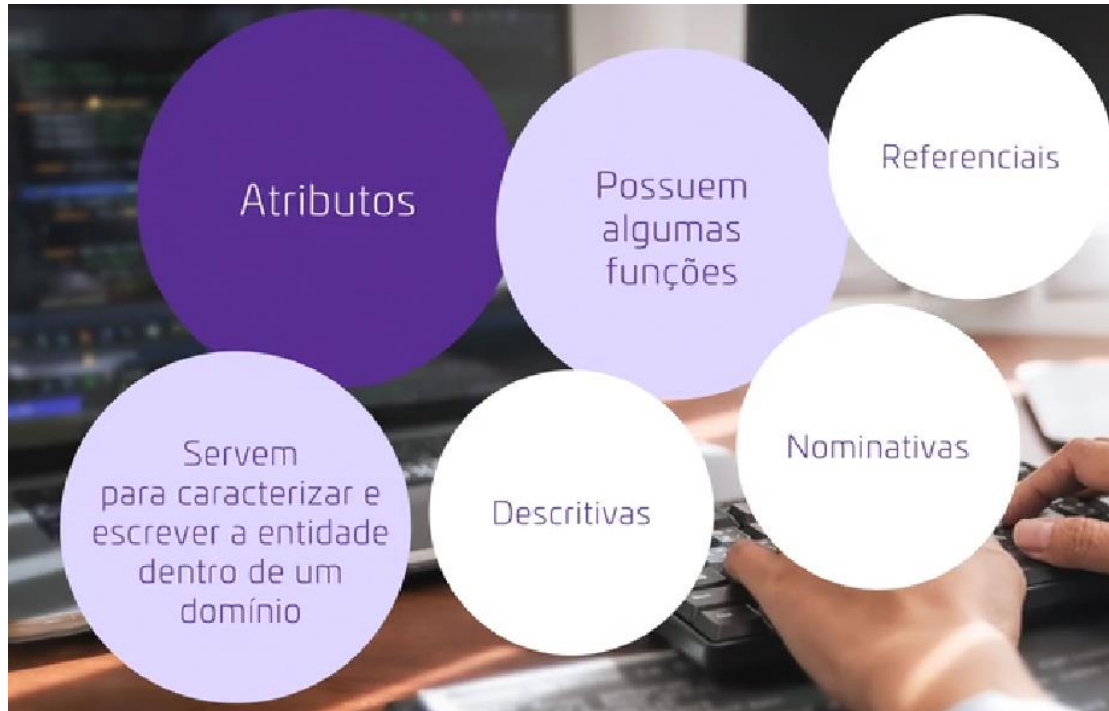
Uma entidade pode fazer referência a várias Unidades da outra e vice versa



“Um para muitos”

Uma entidade pode fazer referência a várias unidades da outra Enquanto essas unidades só podem fazer referência a uma





Descritivas

Um nome

Uma cor

Nominativas

Um nome

Um código

Um número

Referenciais

O CPF de um cliente após uma venda



Análise de requisitos

Análise de Requisitos

A **análise de requisitos** é a fase inicial do desenvolvimento do projeto. Nela, é realizada a identificação dos requisitos que constam em um banco de dados. Entrevistas são feitas e o **minimundo** é desenvolvido com a descrição textual geral do projeto.

As regras de negócios desse documento devem ser identificadas para que as informações sejam bem executadas, pois informações erradas afetam todo o projeto.

Projeto Conceitual

O **projeto conceitual** tem base nas informações coletadas pela **análise de requisitos**. Assim, um esquema é gerado, utilizando um modelo de dados conceitual de alto nível, o **Modelo Entidade-Relacionamento** (MER). Esse modelo é simplificado para aproximar a composição dos bancos de dados aos usuários, facilitando a compreensão das informações. Por isso, a representação dos dados armazenados não é tão relevante.

Fases de um projeto de banco de dados

Para entender as fases de um projeto de banco de dados, imagine um **ciclo com várias etapas**, em que cada uma delas é criada a partir da anterior.

A primeira etapa desse projeto consiste em reunir dados e regras relevantes para a concepção do **minimundo**, que se trata da apresentação abstrata de uma pequena parte do mundo real. Depois, inicia-se a **engenharia de requisitos**.

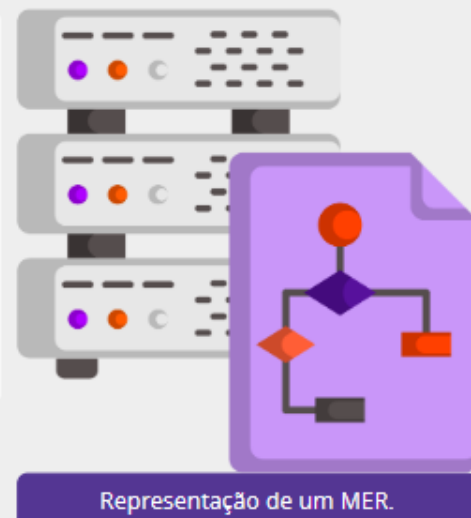
Já a **modelagem de dados**, etapa também usada no desenvolvimento de um banco de dados, consiste em **projeto conceitual**, **mapeamento dos conceitos**, **projeto lógico** e **projeto físico**, que é a fase estrutural do banco de dados.



Projeto conceitual

O MER é representado através de diagramas. As entidades são **retângulos**, os atributos são **elipses** e os relacionamentos são **losangos**. Nesta etapa, os três tipos de relacionamentos de entidades são fundamentais, sendo delas:

- um para um (1, 1);
- um para muitos (1, n) ou muitos para um ou (n, 1);
- muitos para muitos (n, n).



Atributos

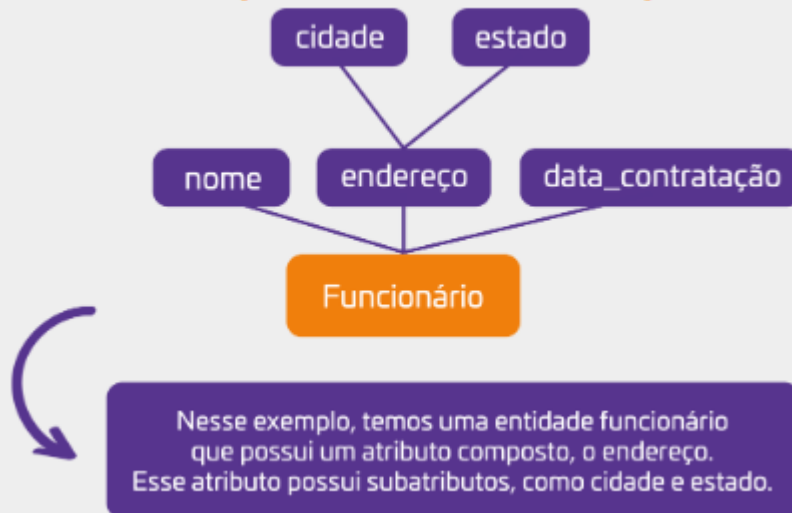
Eles podem ser classificados como:

- **Atributo simples** - não pode ser subdividido;
- **Atributo composto** - pode ser dividido em diversas partes, com significados independentes entre si;
- **Atributo monovalorado** - ou simplesmente valorado, só assume um valor em uma determinada instância;
- **Atributo multivalorado** - pode assumir diversos valores em uma mesma instância;
- **Atributo derivado** - é gerado a partir de outro atributo.

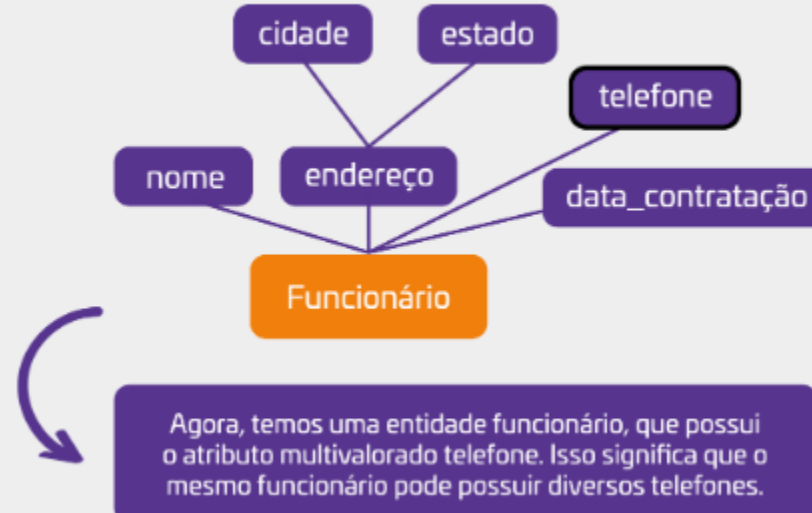
Exemplo de atributo monovalorado



Exemplo de atributo compostos



Exemplo de atributo multivalorado

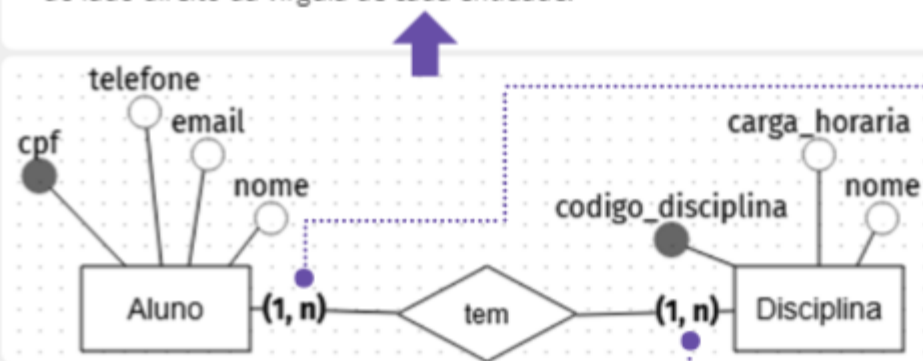


Exemplo de atributo derivado



Exemplo de projeto conceitual (n, n)

Representação de um banco de dados com duas tabelas e um relacionamento. Neste exemplo, está sendo utilizado o relacionamento de entidades **muitos para muitos**, pois temos (n, n) do lado direito da vírgula de cada entidade.

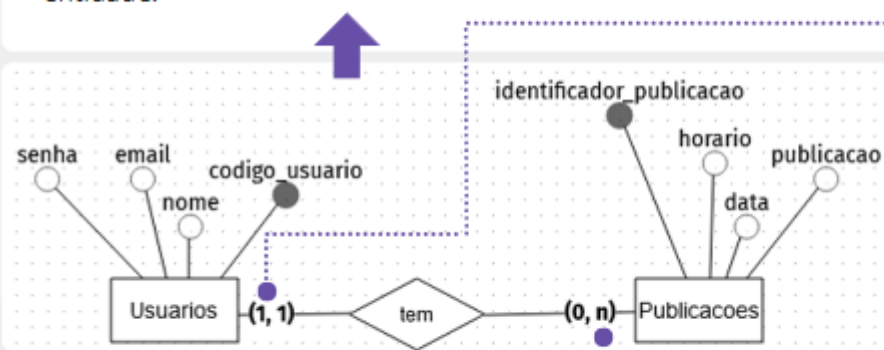


Neste relacionamento, um aluno pode ter uma ou várias disciplinas.

Neste caso, uma disciplina pode pertencer a, no mínimo, um aluno ou a vários alunos.

Exemplo de projeto conceitual (1, n)

Neste exemplo, está sendo utilizado o relacionamento de entidades **um para muitos**, pois temos (1, n) do lado direito da vírgula de cada entidade.



Neste relacionamento, uma publicação pode pertencer a apenas uma pessoa.

Neste caso, um usuário não é obrigado a fazer publicações. Por isso, ele pode ter nenhuma publicação ou várias.

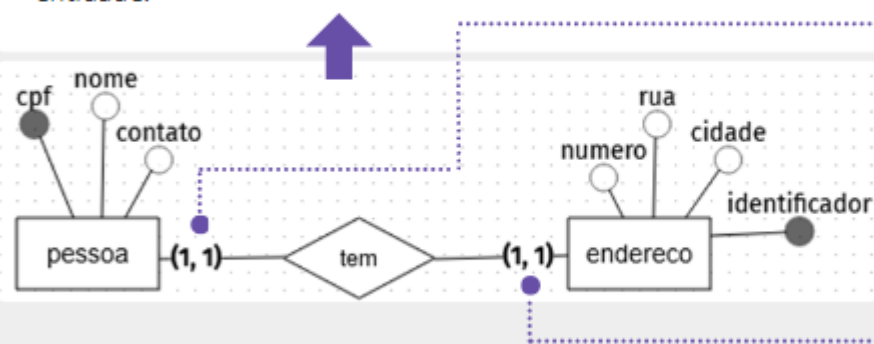
Exemplo de projeto conceitual (n, n)

Toda tabela possui uma chave primária, que se trata de um identificador único e não pode ter mais de um valor.

Neste caso, as chaves primárias são os atributos referenciais, que são o CPF do aluno e o código da disciplina. Isso porque cada aluno tem seu próprio CPF e cada disciplina tem um código para representá-la.

Exemplo de projeto conceitual (1, 1)

Neste exemplo, está sendo utilizado o relacionamento de entidades **um para um**, pois temos (1, 1) do lado direito da vírgula de cada entidade.



Neste relacionamento, uma pessoa pode ter apenas um endereço no banco de dados.

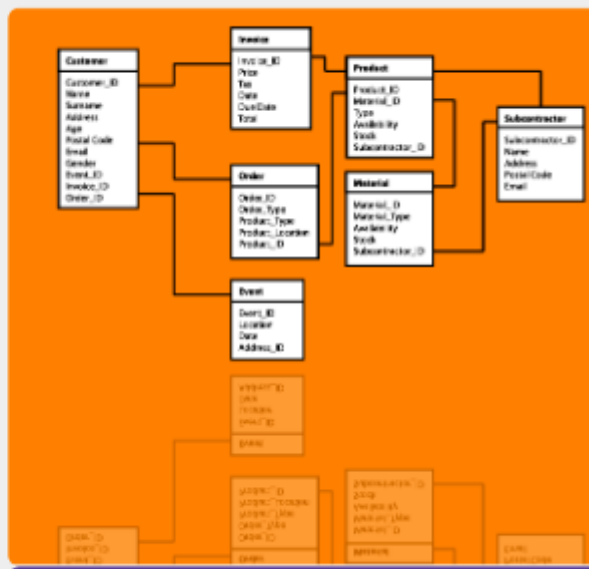
Neste caso, no banco de dados, um endereço é vinculado a um CPF, que pode ter apenas um endereço.

Projeto lógico

O **projeto lógico** é uma etapa do processo do MER. Nele, o mapeamento do projeto conceitual é elaborado.

A construção dos modelos internos é realizada detalhando **tabelas**, regras, relacionamentos, dados das colunas (tamanho e tipo), entre outros.

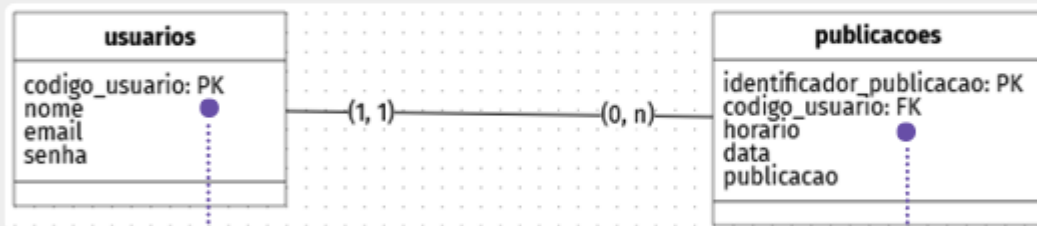
O projeto lógico tem como resultado o esquema detalhado do banco de dados.



Representação de um projeto lógico.

Exemplo de projeto lógico

Aqui, temos a conversão do modelo conceitual **um para muitos** (1, n) para o modelo lógico. Neste modelo, cada entidade se torna uma tabela e seus atributos, os dados de cada tabela.

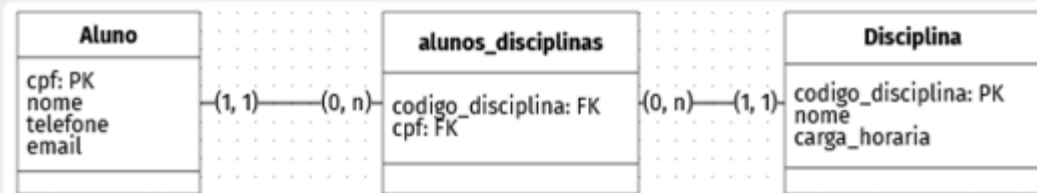


Tudo que tiver "**PK**" representa uma chave primária.

Nesta representação, temos a chave estrangeira, que é uma chave primária utilizada em uma tabela que não é a sua de origem. Tudo que tiver "**FK**" representa uma chave estrangeira.

Exemplo de projeto lógico

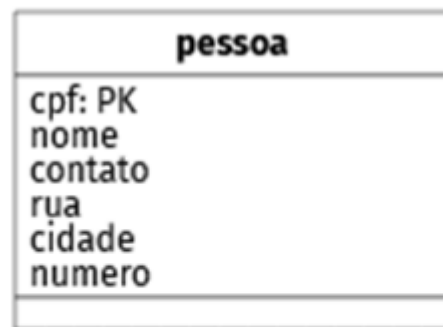
Aqui, temos a conversão do modelo conceitual **muitos para muitos** (n, n) para o modelo lógico. Neste modelo, cada entidade se torna uma tabela, seus atributos se tornam os dados de cada tabela e o relacionamento se torna uma nova tabela.



Uma nova tabela é criada para que ocorra o relacionamento entre as outras tabelas. Por isso, ela possui apenas as chaves estrangeiras.

Exemplo de projeto lógico

Aqui, temos a conversão do modelo conceitual **um para um** (1, 1) para o modelo lógico. Neste modelo, todas as entidades se tornam uma única tabela, ou seja, ocorre uma união dos dados.



Não existe chave estrangeira nesse tipo de tabela. Por isso, ela é considerada a mais simples.

Projeto físico

O **projeto físico** está no nível mais baixo em relação ao usuário final. Nessa etapa, são definidas as estruturas de armazenamento, os índices e a organização de arquivos do banco de dados.

Ela é considerada a parte final do projeto, pois apresenta a forma de armazenamento, as permissões de acesso ao usuário e os *scripts* para criação dos objetos, tabelas, colunas, **visões**, funções e entre outros

Exemplo de projeto físico

A seguir veremos a conversão do modelo lógico **um para muitos** (1, n) para o modelo físico. Neste modelo, os dados são representados em linhas de código SQL, que são utilizadas para a criação das tabelas em um banco de dados.

Exemplo de projeto físico

```
CREATE TABLE usuarios  
(  
  codigo_usuario INT PRIMARY KEY,  
  nome VARCHAR(n) DEFAULT '30',  
  email VARCHAR(n) DEFAULT '30',  
  senha VARCHAR(n) DEFAULT '30',  
  CHECK (codigo_usuario = undefined)  
);
```

```
CREATE TABLE publicacoes  
(  
  identificador_publicacao INT PRIMARY KEY,  
  codigo_usuario INT,  
  horario VARCHAR(n) DEFAULT '5',  
  data DATE,  
  publicacao VARCHAR(n) DEFAULT '1000',  
);
```

```
ALTER TABLE publicacoes ADD FOREIGN KEY(codigo_usuario) REFERENCES usuarios  
(codigo_usuario)
```

Cada tabela é criada através do comando *create table* e os seus atributos são descritos juntamente com a descrição do seu respectivo tipo. Neste exemplo, temos dados do tipo *varchar*, que guardam textos. Também há o número de caracteres que esse dado pode guardar, neste caso, *varchar default 30*.

Exemplo de projeto físico

```
CREATE TABLE usuarios  
(  
  codigo_usuario INT PRIMARY KEY,  
  nome VARCHAR(n) DEFAULT '30',  
  email VARCHAR(n) DEFAULT '30',  
  senha VARCHAR(n) DEFAULT '30',  
  CHECK (codigo_usuario = undefined)  
);
```

```
CREATE TABLE publicacoes  
(  
  identificador_publicacao INT PRIMARY KEY,  
  codigo_usuario INT,  
  horario VARCHAR(n) DEFAULT '5',  
  data DATE,  
  publicacao VARCHAR(n) DEFAULT '1000',  
);
```

```
ALTER TABLE publicacoes ADD FOREIGN KEY(codigo_usuario) REFERENCES usuarios  
(codigo_usuario)
```

Quando temos uma chave estrangeira, é necessário descrever qual é a sua origem.

Banco de dados

Existem ferramentas *on-line* destinadas à criação de cada um desses modelos vistos. Uma dessas é o **brModelo**, que é simples e de fácil uso.

Continue praticando!



