

53 45 52 45 49 20 46 49 45 4c 20
41 4f 53 20 50 52 45 43 45 49 54
4f 53 20 44 41 20 48 4f 4e 52 41
20 45 20 44 41 20 43 49 c3 8a 4e
43 49 41 2c 20 50 52 4f 4d 4f 56
45 4e 44 4f 20 4f 20 55 53 4f 20
45 20 4f 20 44 45 53 45 4e 56 4f
4c 56 49 4d 45 4e 54 4f 20 44 41
20 49 4e 46 4f 52 4d c3 81 54 49
43 41 20 45 4d 20 42 45 4e 45 46
c3 8d 43 49 4f 20 44 4f 20 43 49
44 41 44 c3 83 4f 20 45 20 44 41
20 53 4f 43 49 45 44 41 44 45 2e

RESIDÊNCIA DE SOFTWARE

**CAPACITAR
TREINAR
EMPREGAR**

TRANSFORMAR



Introdução a Banco de Dados
Aula 2
29/03/2022

Entidades

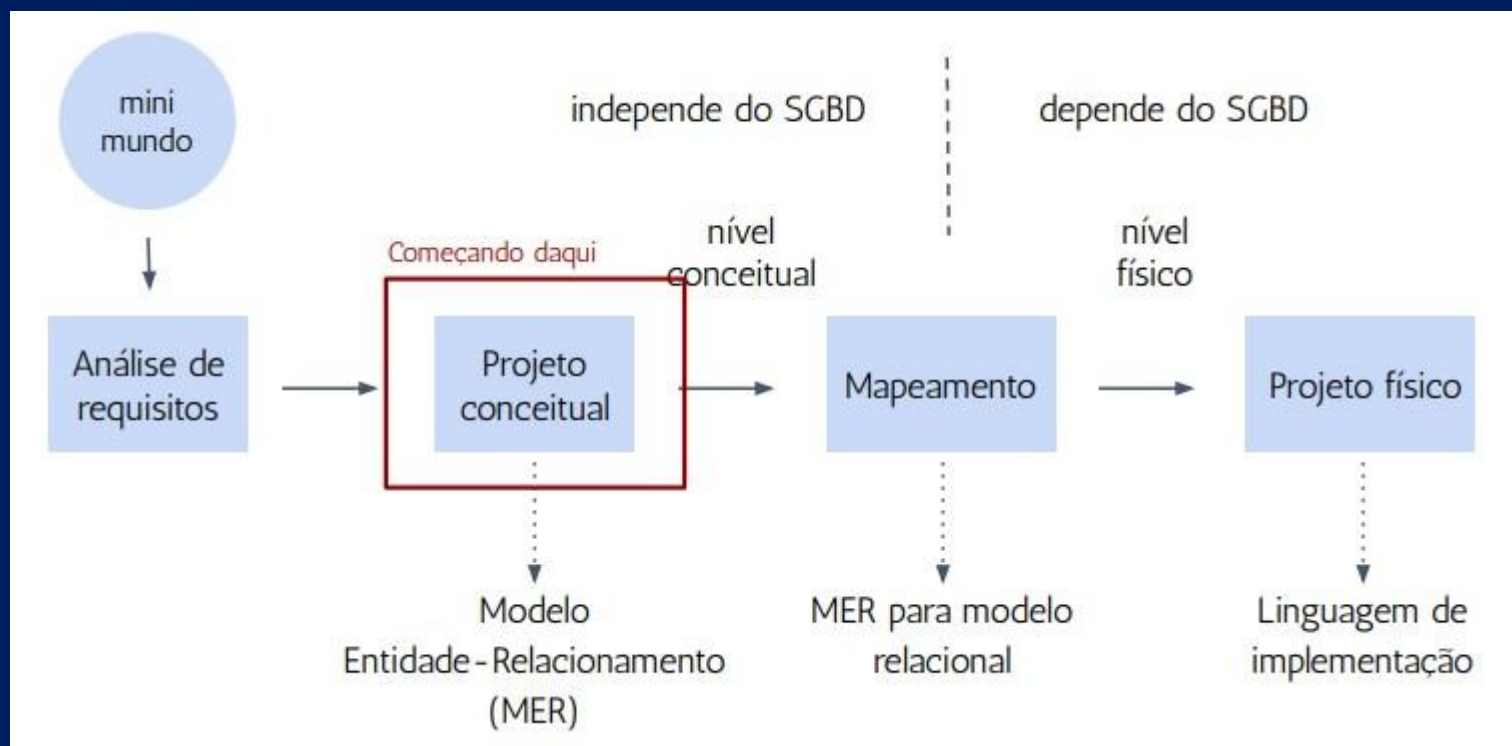
Item, coisa, do mini-mundo que precisa ser representado em um banco de dados

- Representação de temas, tópicos ou um conceito
- Pode existir de forma física ou conceitual

Exemplo: Empregado, aluno, livro, produto

- Precisam ser nomeados de forma clara e objetiva, representando sua função (substantivos)
- Instâncias são ocorrências da entidade, são valores que elas podem admitir

Projeto de Banco de Dados



Entidades - Tipos de entidades

- **Forte**: sua existência independe de outras entidades. São entidades que por si só possuem total sentido de existir.



- **Fraca** (ou dependente): precisa de outra entidade para garantir a sua existência. O identificador de uma entidade fraca possui em sua composição o(s) atributo(s) identificador(es) da entidade forte à qual está associada.



Atributos

Descrevem **características** de determinada entidade ou relação Cor, nome, modelo, telefone

- Possuem um nome, que identifica o que o atributo representa, um tipo de dados, ou domínio, e um valor

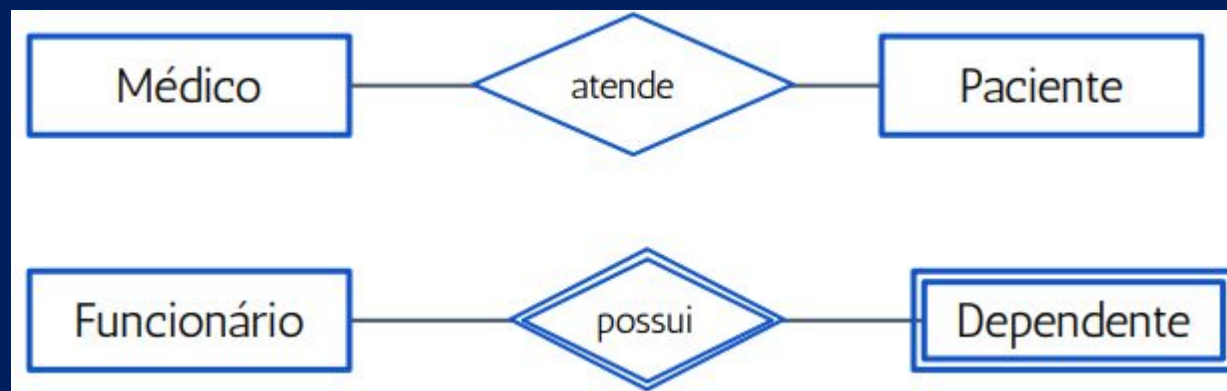
Exemplo: Nome, cadeia de caracteres, João

Atributos - Tipos de atributos

- **Simples**: atômico, indivisível. Ex.: nome
- **Composto**: pode ser subdividido em outros atributos. Ex.: endereço
- **Multivalorado**: vários valores para um mesmo registro. Ex.: idioma
- **Determinante**: define de forma única a instância de uma entidade. Ex.: matrícula
- **Derivado**: atributo que tem relação com outro. Ex. idade (nascimento)
- **Chaves**: identifica uma instância da entidade. Ex.: CPF

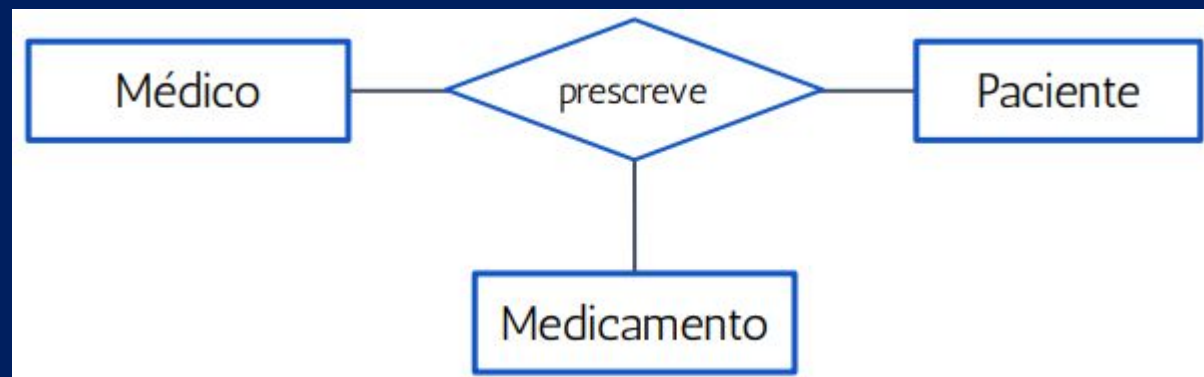
Relacionamentos

- São utilizados para indicar a associação entre as entidades
- Essa associação é feita entre **uma** ou mais entidades
- Entidades modelam itens de forma separada, e por isso os relacionamentos nos permitem combinar informações de diferentes



Relacionamentos - Grau de um relacionamento

Número de entidades que participam de um relacionamento Unário, binário, ternário...



Relacionamentos - Cardinalidade

Indica o número mínimo/máximo de valores que o atributo pode assumir em cada instância da entidade ou relacionamento.

Ou seja, ajuda a definir o relacionamento, pois define o número de ocorrências possíveis em um relacionamento.

Relacionamentos – Cardinalidade - Tipos de cardinalidade

- Mínima: número mínimo de instâncias de entidade que devem participar de um relacionamento 0 - opcional ou 1 - obrigatório
- Máxima: número máximo de instâncias de entidade que podem participar de um relacionamento 1 ou N - muitos

Relacionamentos – Tipos de relacionamento

- **Relacionamento 1:1 (um para um):** cada uma das duas entidades envolvidas referenciam obrigatoriamente

apenas uma instância da outra.

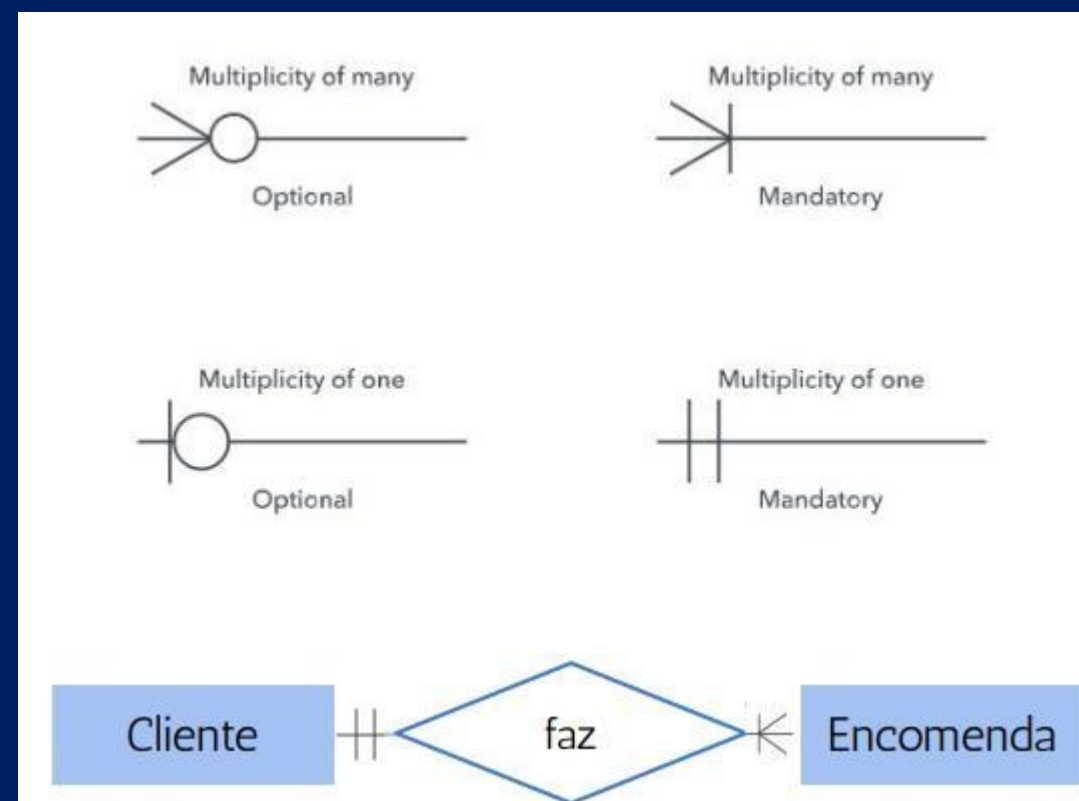
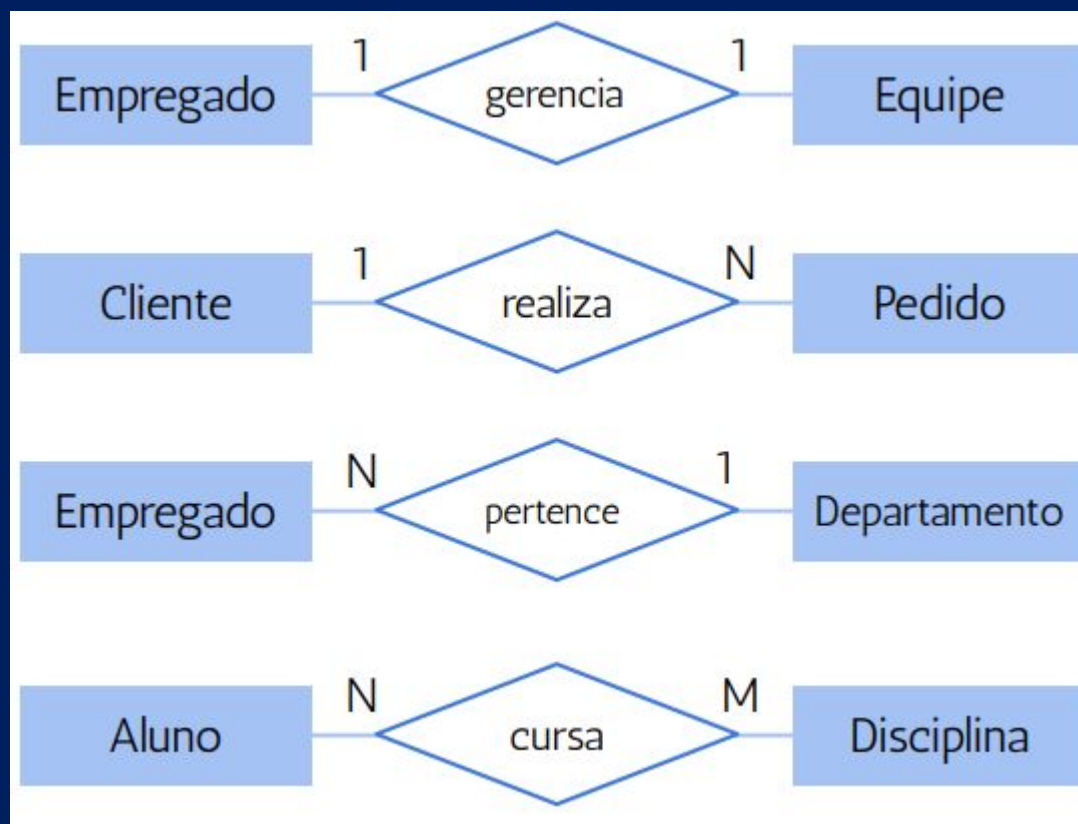
Usuário - Currículo

- **Relacionamento 1:N ou 1:* (um para muitos):** uma das entidades envolvidas pode referenciar **várias instâncias** da outra, porém, do outro lado cada uma das várias unidades referenciadas só pode estar ligada **uma instância** da outra entidade.

Usuário - Dependente

- **Relacionamento N..N ou *.* (muitos para muitos):** cada entidade, de ambos os lados, podem referenciar **múltiplas instâncias** da outra. Livro - Autor

Relacionamentos – Tipos de notação comuns



Instalando Um SGBD



Structured Query language

- É uma linguagem de pesquisa declarativa padrão para banco de dados relacionais
 - Tipo de paradigma onde o foco não está no “como” e sim no “que” Mais detalhes sobre as diferenças de paradigmas aqui.
- Criada no início dos anos 70 com o objetivo de demonstrar que era viável implementar o modelo relacional
 - Proposto por E. F. Codd, membro do laboratório de pesquisa da IBM em San Jose, CA
- Padrão utilizado pelos sistemas de banco de dados relacionais
- Pode ser dividida nas seguintes categorias:
 - DQL - Data Query Language
 - DDL - Data Definition Language
 - DML - Data Manipulation Language
 - DCL - Data Control Language
 - DTL - Data Transaction Language

Modelo Relacional

- Modelo de dados para um SGBD, que se baseia no princípio que todos os dados estão guardados em tabelas (entidades do MER).
- Exemplo : Tabela Aluno

Num_Matrícula	Nome_Aluno	Data_Nascimento
1	Raul	1993-11-01
2	Clara	2000-09-17
4	Denis	1995-07-25
5	Fred	1996-06-13

- Regra : Nome de tabelas devem ser únicos no BD;

Modelo Relacional - Colunas || Atributos

- Um nome de atributo deve ser único em uma tabela e dizer exatamente o tipo de informação que ele representa
- Nome de uma coluna deve expressar exatamente o que ela armazena
- Sempre que possível utilizar prefixos padronizados

Num_Matricula	Nome do Aluno	data de nasc.
---------------	---------------	---------------



Num_Matricula	Nome_Aluno	Data_Nascimento
---------------	------------	-----------------



Modelo Relacional - Linha || Registros

- A tabela Aluno possui cinco registros
- Cada linha da tabela é única e possui um atributo identificador (Num_Matrícula)
 - Este atributo identificador é chamado de **chave_primária**, **lembra**?
- Regras :
 - Em uma tabela não devem existir linhas duplicadas;
 - As linhas de um tabela não seguem uma ordem específica.

1	Raul	01/11/1993
2	Clara	17/09/2000
1	Raul	01/11/1993
1	Clara	17/09/2000



Modelo Relacional - Domínio

- A tabela Aluno possui três atributos
- Para cada atributo existe um conjunto de valores permitidos chamado domínio daquele atributo
 - Para o atributo Num_Matricula o domínio é o conjunto de números naturais
 - Para o atributo Nome_Aluno o domínio é qualquer nome válido
 - Enquanto que para Data_Nascimento o domínio são as Datas no formato YYYY- MM-DD

Modelo Relacional - Tabelas e Entidades

- Para a criação de banco de dados, tabelas e atributos em um SGBD, utilizaremos a linguagem SQL que é composta de comandos de manipulação, definição e controle de dados
- Esses conjuntos de comandos de definição de dados são denominados pela sigla **DDL** (Data Definition Language), que disponibiliza um conjunto de comandos para criação (CREATE), alteração (ALTER) e remoção (DROP) de tabelas e outras estruturas

Comando CREATE DATABASE

- A maioria dos SGBDs disponibiliza ferramentas que permitem a criação de Banco de Dados, mas é possível criar o próprio Banco de Dados a partir de um comando SQL.
- A sintaxe do comando é:

```
CREATE DATABASE nome_do_banco_de_dados
```

```
CREATE DATABASE RESIDENCIA_DE_SOFTWARE
```

Comando DROP DATABASE

- O comando DROP DATABASE permite remover um determinado Banco de Dados, apagando todas as tabelas e estruturas associadas e, consequentemente, todos os dados existentes nelas
- A sintaxe do comando é:

DROP DATABASE nome_do_banco_de_dados

DROP DATABASE RESIDENCIA_DE_SOFTWARE

Comando CREATE TABLE

- O comando CREATE TABLE é o principal comando DDL da linguagem SQL. A criação de tabelas é realizada em SQL utilizando este comando.
- A sintaxe básica do comando é:

```
CREATE TABLE nome_da_table (Coluna1 Tipo,  
                               Coluna2 Tipo, ColunaN Tipo)  
CREATE TABLE Aluno (Num_Matricula INTEGER, Nome_Aluno  
                      CHAR(50), Data_Nasc DATE )
```

Tipos de Dados

- Em SQL os tipos de dados são agrupados em 3 categorias:
 - Caracteres (Strings)
 - CHAR
 - CHAR(N)
 - VARCHAR(N)
 - Numéricos
 - NUMERIC
 - INTEGER ou INT
 - SMALLINT
 - FLOAT
 - BIT
 - Tempo e Data
 - DATE
 - TIME
 - DATETIME

Atributos NOT NULL

- No exemplo apresentado anteriormente

```
CREATE TABLE Aluno(Num_Matricula INTEGER, Nome_Aluno CHAR(50), Data_Nasc DATE )
```

- Acima os valores podem ser nulos se nada for especificado. Para tornar um campo obrigatório podemos utilizar o atributo NOT NULL

```
CREATE TABLE Aluno(Num_Matricula INTEGER NOT NULL, Nome_Aluno CHAR(50),  
Data_Nasc DATE )
```


Outras restrições (constraints)

- Existem alguns outros tipos distintos de restrições que se podem aplicar a colunas:
 - NOT NULL
 - CHECK
 - UNIQUE
 - PRIMARY KEY
 - REFERENCES



Normalização

- Normalização é o processo que permite a simplificação da estrutura de um banco de dados, de modo que esta se apresente em um ótimo estado.
- A ideia é minimizar ou eliminar a redundância de informações. Apesar de existirem 6 formas normais, é considerado em um bom nível quando um esquema de banco de dados se encontra na terceira forma normal

1a Forma Normal

- Diz-se que uma relação está na primeira forma normal quando:
 - Não contém atributos multivalorados (grupo de atributos)
 - Não contém grupos repetitivos

**Fatura (NumeroFatura (PK) ,
CodigoCliente, NomeCliente, Endereco,
CodigoProduto, DescricaoProduto, Preco,
Quantidade)**

1a Forma Normal

- Para fazer a passagem para a 1a FN será necessário separar a tabela em duas : **Fatura e Itens.**

**Fatura (NumeroFatura (PK),CodigoCliente,
NomeCliente, Endereco, CodigoProduto,
DescricaoProduto, Preco, Quantidade)**

**Fatura (NumeroFatura,CodigoCliente, NomeCliente,
Logradouro, Bairro,Cidade, Cep)**

**Itens (NumeroFatura (PK), CodigoProduto(PK) ,
DescricaoProduto,Preco, Quantidade)**

2a Forma Normal

- Diz-se que uma relação está na segunda forma normal quando:
 - Está na 1FN
 - Todos os atributos não chave dependem da totalidade da chave.

**Itens (NumeroFatura (PK) ,
CodigoProduto (PK) ,
DescricaoProduto, Preço, Quantidade)**

2a Forma Normal

- A tabela Itens não se encontra na 2FN porque os atributos **DescricaoProduto** e **Preco** não dependem de **CodigoFatura**, mas só de **CodigoProduto**.

**Itens (NumeroFatura (PK), CodigoProduto (PK),
DescricaoProduto, Preco, Quantidade)**

**Itens (NumeroFatura (PK), CodigoProduto (PK),
Quantidade)**

Produto (CodigoProduto (PK), DescricaoProduto, Preco)

3a Forma Normal

- Uma relação está na 3a FN se:
 - Está na 2a FN
 - Todos os seus atributos não chave NAO são identificados por outro também não chave

Fatura (NumeroFatura,CodigoCliente,
NomeCliente, Logradouro,
Bairro,Cidade, Cep)

3a Forma Normal

- A tabela Fatura não se encontra na 3FN porque os atributos **NomeCliente, Logradouro, Bairro, Cidade** e **Cep** são identificados por **CodigoCliente** e não por **CodigoFatura**. Para colocar na 3FN deve-se separar a tabela Fatura em duas tabelas:

Fatura (NumeroFatura (PK), CodigoCliente, NomeCliente, Logradouro, Bairro, Cidade, Cep)

Fatura (NumeroFatura (PK), CodigoCliente)

Cliente (CodigoCliente (PK), NomeCliente, Logradouro, Bairro, Cidade, Cep)

Exercícios e Dinâmicas

