



SÃO  
PAULO  
TECH  
SCHOOL



# **BD – Banco de Dados**

Tipos de atributos

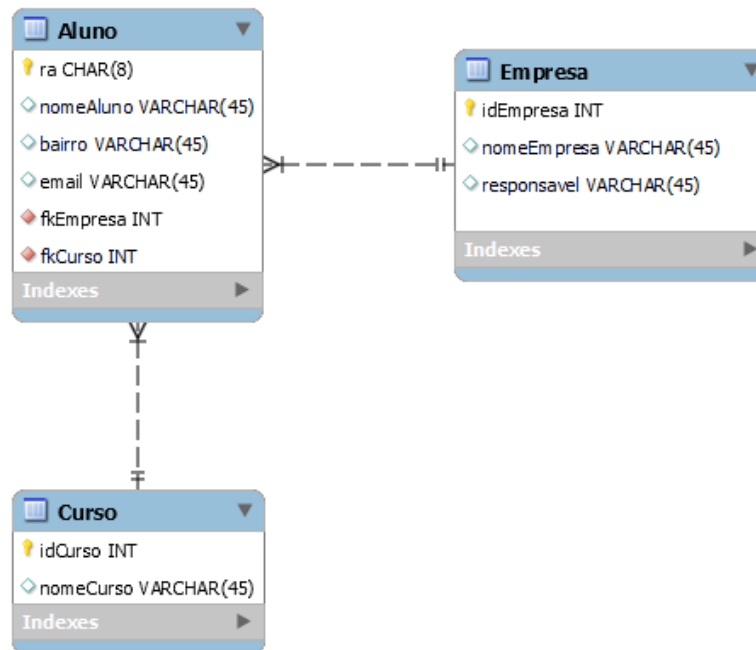
# Domínio de um atributo

- É o conjunto de valores que o atributo pode assumir
  - Exemplo: sexo – os valores válidos são conhecidos, permitindo a sua validação (masculino ou feminino)
- CHECK
  - Sintaxe:

```
CREATE TABLE curso (  
    idCurso int primary key auto_increment,  
    nome varchar(40),  
    sigla char(3),  
    semestre char(6),  
    CONSTRAINT chkSem CHECK (semestre in ('1º sem', '2º  
sem', '3º sem', '4º sem'))  
);  
OU  
ALTER TABLE curso ADD CONSTRAINT chkSem CHECK (semestre  
in ('1º sem', '2º sem', '3º sem', '4º sem'));
```

# Entidade primária ou Entidade forte

- A entidade é identificada como forte quando não tem dependência com nenhuma outra para formar seu conceito.
- A entidade forte tem sempre um atributo identificador (chave primária) próprio, que não depende do atributo identificador de outra entidade.
- No exemplo que fizemos das entidades ALUNO e CURSO, ambas são entidades fortes.



# Entidade dependente ou Entidade fraca

- A entidade é fraca quando ela não existe por si só e sua existência no MER está condicionada a outra entidade única, da qual ela depende.
- O atributo identificador da entidade fraca depende do atributo identificador da entidade forte da qual ela depende.
- Exemplo:
  - ITENS\_PEDIDO (complementa a entidade PEDIDO)
  - HISTORICO\_PACIENTE (complementa a entidade PACIENTE)
- Representada por um retângulo duplo, no DER do Peter Chen.

# Entidade dependente ou Entidade fraca

- No caso da entidade ALUNO, ela é uma entidade forte, quanto está no contexto da nossa Faculdade.
  - Tabela ALUNO, contendo alunos apenas da Faculdade SPTECH:

RA	NOME
01221001	João
01221002	Maria
01221003	Ana

- Nesse contexto, só existe um aluno da SPTECH com RA 01221001. Esse aluno é único na SPTECH.
- RA é a chave primária (PK) da tabela, que não depende de nenhuma outra tabela
- A entidade ALUNO, nesse contexto, é uma entidade FORTE.

# Entidade dependente ou Entidade fraca

- Mas se estivermos fazendo um sistema em que estamos cadastrando vários alunos de várias faculdades diferentes,
  - Nesse caso, a tabela ALUNO conterá alunos de várias faculdades diferentes:

Tabela Faculdade

idFaculdade	nomeFaculdade
1	Sptech
2	Fatec
3	USP

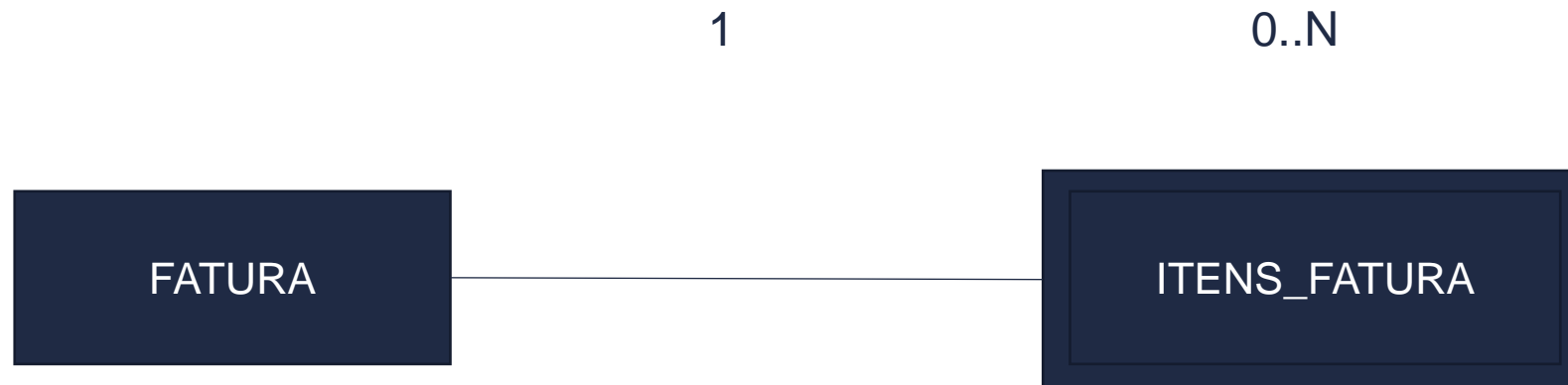
Tabela Aluno

fkFaculdade	RA	nomeAluno
1	01221001	João
1	01221002	Maria
2	01221001	Carlos
3	01221001	Beatriz

- Pode haver o aluno de RA 01221001 da Faculdade Sptech, da Fatec, da USP, etc
- Então, apenas o RA não é suficiente para identificar o aluno.
- É preciso juntar o fkFaculdade e o RA para identificar um aluno.
- ALUNO passa a ser uma entidade FRACA.
- Tabela Aluno: chave primária composta (fkFaculdade, RA)

# Relacionamento identificador ou tipo dependência

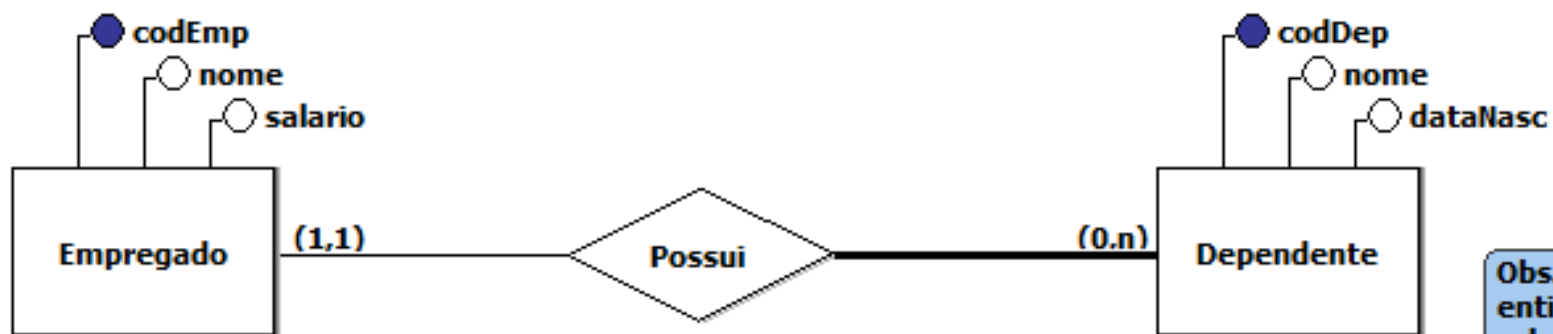
- Relacionamento entre uma entidade forte e uma entidade fraca, que dela é dependente.
- Exemplo:
  - Relacionamento entre entidade FATURA e ITENS\_FATURA





# Relacionamento identificador ou tipo dependência

- Outro exemplo:
  - Relacionamento entre entidade Empregado e Dependente
  - Modelagem no brModelo:



Obs.: Dependente é uma entidade fraca, em relação a Empregado. O brModelo marca isso através do negrito da linha do relacionamento.

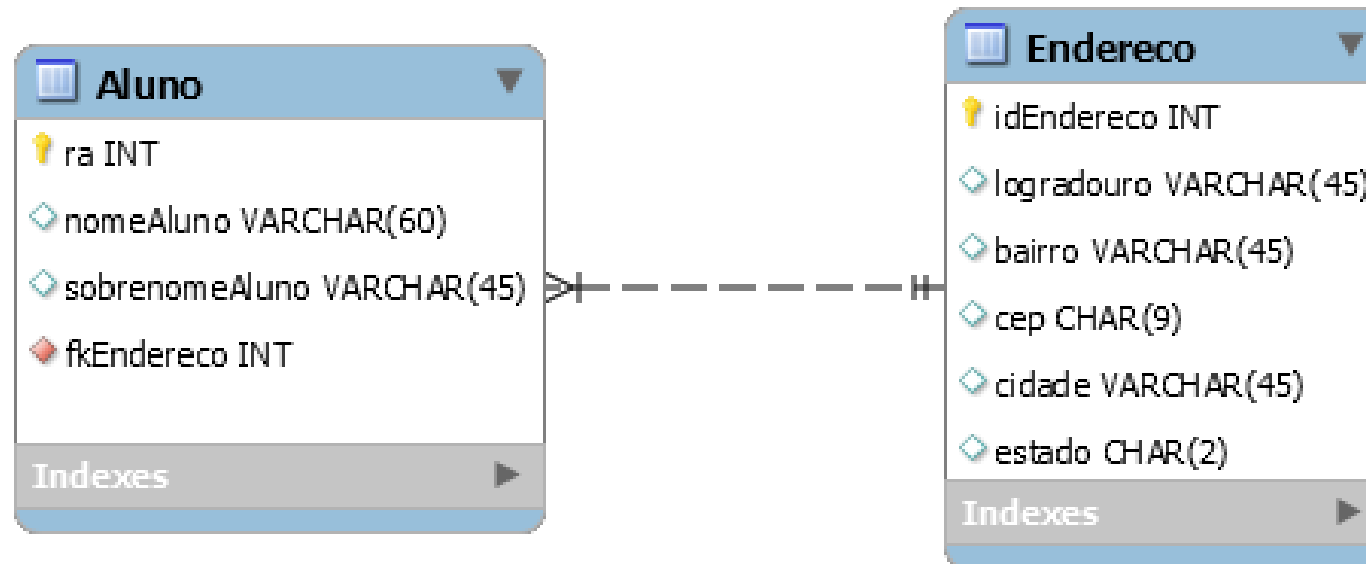
# Atributo atômico versus Atributo composto

- Atributo atômico ou primitivo ou simples:
  - Não pode ser dividido em subpartes menores
  - Exemplo: sexo
- Atributo composto:
  - Pode ser dividido em subpartes menores, com significado independente
  - Exemplo: Endereco – pode ser dividido em Endereço da Rua, Cidade, Estado, CEP



# Modelo lógico – MySQL Workbench

- **Atributo composto:** Ex: endereço - como vimos anteriormente, os campos do endereço podem estar na mesma tabela.
- Mas, por questão de organização, pode ser criada uma outra tabela Endereco, com um relacionamento 1:1
- Ou, como mostra a modelagem a seguir, podemos ter um relacionamento 1:N, em que 1 endereço pode ser de mais de 1 aluno (no caso de alunos irmãos):



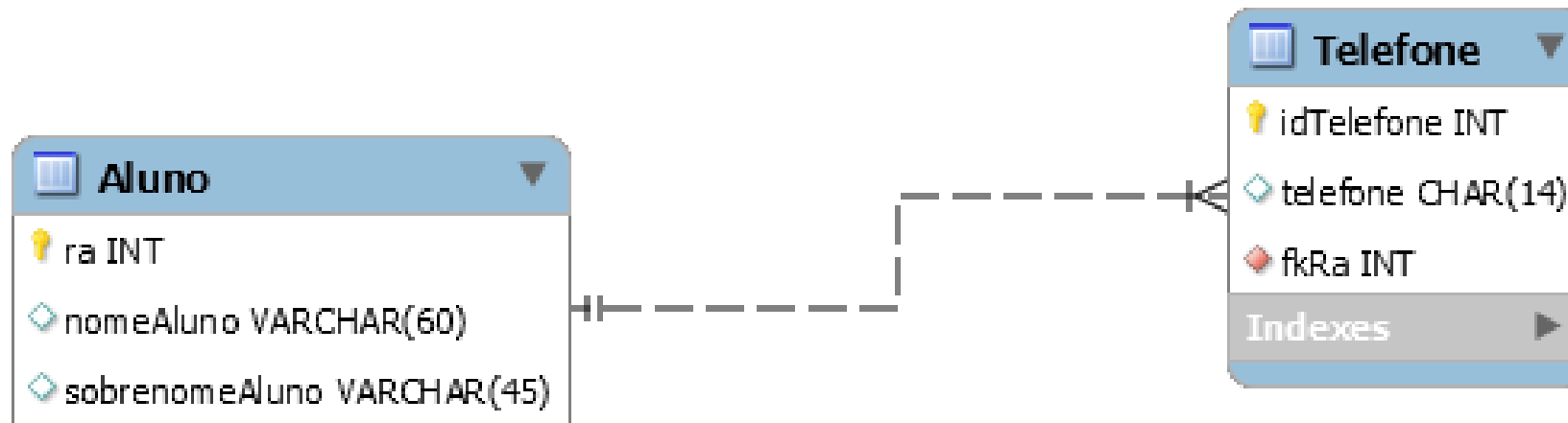
# Atributo monovalorado vs Atributo multivalorado

- Atributo monovalorado:
  - Tem apenas um valor na mesma entidade
  - Exemplo: bairro
- Atributo multivalorado:
  - Pode ter mais de um valor
  - Exemplo: Telefone – pode haver o telefone celular e o telefone residencial
  - Representado por um traço duplo ligando a entidade ao atributo:

ra	primeiroNome	sobrenome	Telefone_Fixo	Telefone_Movel	Telefone_Responsa
01201200	Lucas	Silva	(11) 5678-2233	(11) 99123-7676	null
01201201	Pedro	Sousa	(11) 3365-5567	(11) 98765-8109	(11) 99000-6523

# Modelo lógico – MySQL Workbench



- **Atributo multivalorado:**
- Ex: telefone – segue a mesma regra da implementação no SQL
- Se forem mais do que 3 telefones, é melhor criar uma tabela Telefone separada:
- Nesse caso, há um relacionamento 1:N entre Aluno e Telefone
  - 1 aluno pode ter muitos telefones
  - 1 telefone é somente de 1 aluno



# Atributo derivado

- Atributo cujo valor deve ser obtido após algum processamento de informações do próprio banco de dados:
  - Exemplo: valor\_fatura (no cenário do Rádio Táxi On-Line) – calculado por meio da soma dos valores das corridas realizadas num período, para uma conveniada
  - Outro exemplo: idade

```
58 • Select TIMESTAMPDIFF(YEAR, '1983-10-13',NOW()) idade;
```

Result Grid    Filter Rows: <input type="text"/>   Export:  Wrap Cell Content	
	idade
	38

# Atributo identificador ou determinante

- Atributo cujo valor identifica de forma única uma ocorrência da entidade
- Esse atributo corresponde à chave primária da tabela.
  - Exemplo: RA do aluno, código do produto
- Boas práticas: recomenda-se escolher como atributo identificador alguma informação que o sistema tenha controle, como um código gerado internamente.
- Recomenda-se não utilizar RG, pois o RG é um número gerado pelo estado, podendo haver RGs de mesmo número para pessoas de estados diferentes

**Agradeço**  
**a sua atenção!**

**Vivian Silva**

[vivian.silva@sptech.school](mailto:vivian.silva@sptech.school)

SÃO  
PAULO  
TECH  
SCHOOL