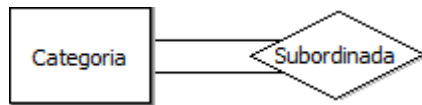


Resumo P1: Abordagem Entidade-Relacionamento

- Associação entre entidades
- Atributo de uma entidade que se refere a outra

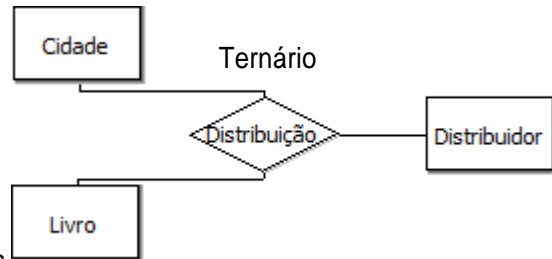


- Auto Relacionamento: Relacionamento entre ocorrências da mesma entidade



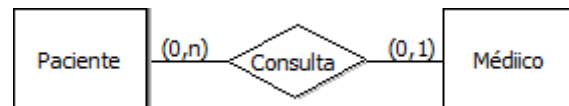
Ficção --- Subordinada --- Romance

- Grau de Relacionamento: Binário

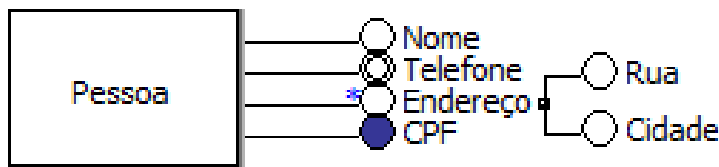


- Cardinalidade: Podendo ser (0,n) = Nenhuma ou várias

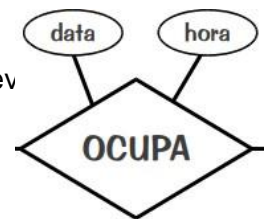
(0,1) = Nenhuma ou apenas uma (1,n) = Apenas uma ou várias (1,1) = Uma e apenas uma
Pode-se variar entre diferentes valores específicos com a mesma lógica, porém esses são os mais comuns



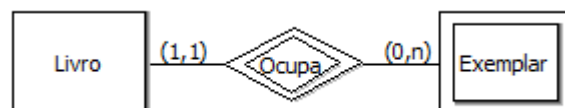
- Atributos: Cada instância de entidade ou relacionamento tem atributos que a descrevem



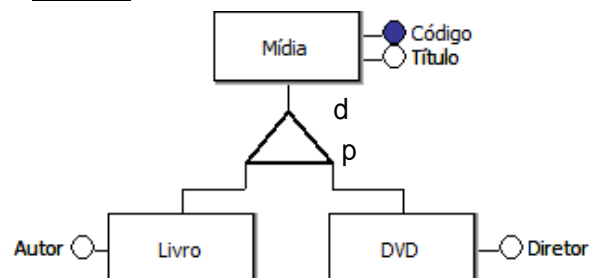
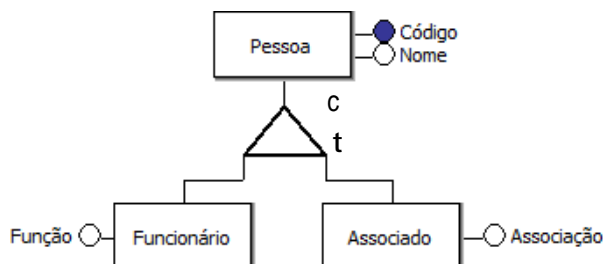
- Simples (Atômico)
- Multivalorado
- Composto
- Chave (únicos na relação), se houver mais de um, se torna Atributo-Chave Composto



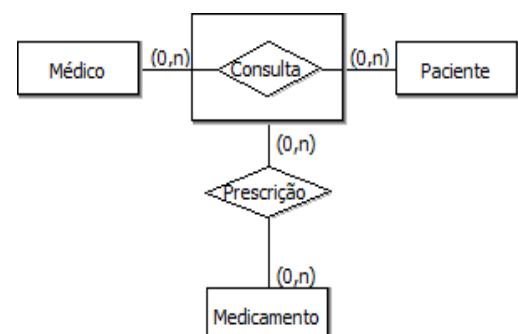
- Entidade Fraca: Quando uma entidade depende de outra para existir



- Generalização/Especialização: Divide uma entidade em mais partes, com diferentes atributos. Além de termos:
 - Compartilhada ou Superposta: A Pessoa pode ser ambos
 - Exclusiva ou Disjunta: Se é um, não pode ser o outro
 - Total: Todas as pessoas são Funcionários ou Associados
 - Parcial: Nem todas as mídias são livros ou DVDs

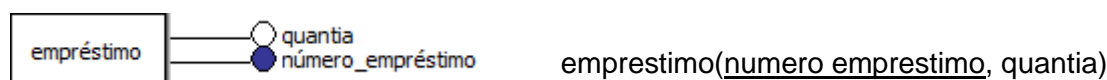


- Entidade Associativa: A relação pode se comportar como uma entidade

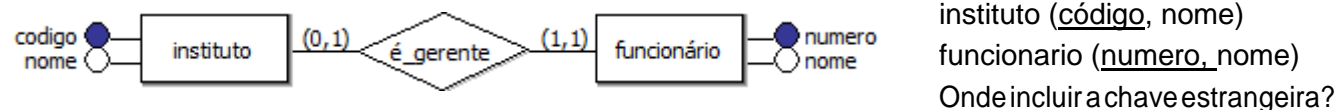


Projeto Lógico - Modelo Relacional

- Mapeamento Entidades Fortes:



- Relacionamento: É implementado com chave estrangeira

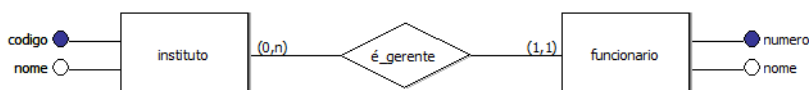


1) Incluir a chave estrangeira em uma das tabelas

instituto (codigo, nome, num_gerente)

num_gerente referencia funcionario

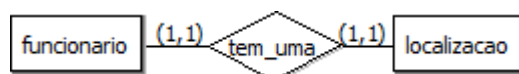
funcionario (numero, nome)



2) Incorporar os atributos de uma entidade em outra (útil quando o rel é total para as duas entidades envolvidas, e uma delas não possui outros relacionamentos)

localizacao (codigo, descricao, bloco, sala, tipo_restricao)

funcionario (numero, nome, codigo, descricao, bloco, sala, tipo_restricao)



3) N para N + com atributo

conta (num, saldo,...)

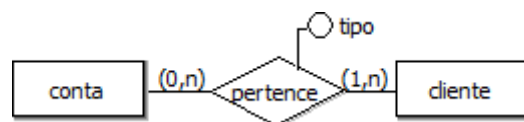
cliente (codigo, nome,...)

(linha de baixo) *

*conta_cliente(codigo, num, tipo)

codigo referencia cliente

num referencia conta



4) Relacionamento N-ário: Relação U para o relacionamento

projeto (num, título,...)

funcionário (codigoF, nome,...)

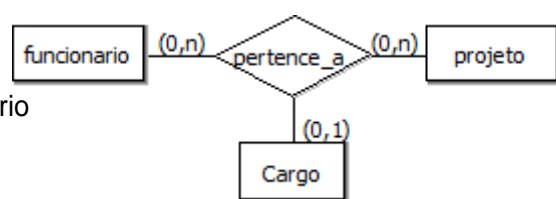
cargo (codigoC, nome,...)*

*trabalho(num, codigoF, codigoC)

num referencia projeto

codigoF referencia funcionario

codigoC referencia cargo

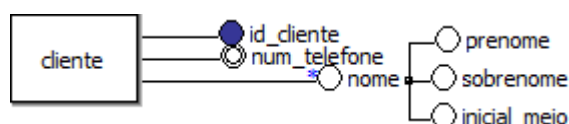


- Mapeamento de Atributos Compostos e Multivalorados:

cliente (id_cliente, prenome, inicial_meio, sobrenome)

telefone (id_cliente, num_telefone)

id_cliente referencia cliente



- Mapeamento de Entidades Fracas: É representado como uma relação

pagamento(num_emprestimo, num_pagamento, data_pagamento, quantia_pagamento)

num_emprestimo referencia empréstimo (num_emprestimo)



- Mapeamento Especialização e Generalização:

1) Criar relação para cada entidade (não disjuntas/parciais)

pessoa (id_pessoa, nome, rua)

funcionario (id_pessoa, salario) id_pessoa referencia pessoa(id_pessoa)

cliente (id_pessoa, avaliacao) id_pessoa referencia pessoa(id_pessoa)

2) Criar tabelas somente para entidade de nível inferior

cliente (id_pessoa, nome, rua, avaliacao) id_pessoa referencia pessoa(id_pessoa)

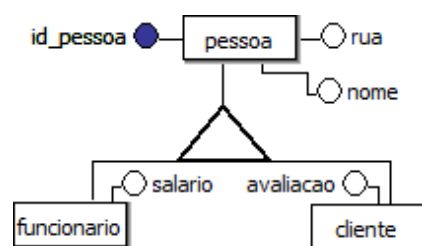
funcionario (id_pessoa, nome, rua, salario) id_pessoa referencia pessoa(id_pessoa)

3) Única tabela com todos os atributos das entidades envolvidas + atributos tipo
pessoa (id_pessoa, nome, rua, avaliacao, salario, **tipo**) id_pessoa referencia pessoa(id_pessoa)

3.5) Semelhante ao anterior, porém com atributo booleano, indicando a entidade específica ISA

pessoa (id_pessoa, nome, rua, **f_tipo**, avaliacao, **c_tipo**, salario)

id_pessoa referencia pessoa(id_pessoa)



Projeto Físico – SQL

Linguagem de BD: SQL
SGBD: Postgres

DML: Tupla (Insert, Delete, Update, Select)

DDL: Esquema (Create, Drop, Alter)

“//” = Linha de baixo

DDL:

- **Comando Create:** Criar elementos do esquema de banco de dados (Banco de Dados e/ou Tabela)

create database nomeDoBanco / create database empresa (nomeDoBanco: identificador de base de dados)

Tabela: *create table nomeTabela (// nomeColuna1 domínioDaColuna1, // nomeColuna2 domínioDaColuna2, // ...)*

- **Comando Drop:** Remover elementos do esquema de banco de dados (Banco de Dados e/ou Tabela)

drop database nomeDoBanco / drop database empresa

- **Tipos (padrão SQL):** *char(n)*: string tamanho fixo (*varchar(n)* é tamanho variado)

int: inteiro (*smallint* é inteiro pequeno)

numeric(p,d): Ponto físico, com precisão de p dígitos

date: Tipo para data, formato YYYY – MM - DD

float (n): Ponto flutuante (*real*, *double precision* além de *float*, também é ponto flutuante de precisão dupla)

- **Restrições:** *create table r (A1D1 restrição-de-integridade1, // ..., // restrição-de-integridadek)*

- **Primary Key:** Chave primária - *create table cliente (// matricula char(5) **primary key**)*

- **Restrições de Vazio:** *create table cliente (// nome varchar(50) **not null**)*

- **Chave Candidata:** *create table cliente (// identidade char(7) **not null unique**)*

- **Chave primária e Candidata compostas:** *(...// **primary key**(codigoLetra, codigoNum), // **unique**(nome, sobrenome))*

- **Restrições de Domínio:** *create table cliente (// email varchar(100) **default** ‘desconhecido’)*

- **Check:** *create table cliente (// idade int **not null**, // **check**(idade>0))* ou *(idade int **not null check**(idade>0))*

*create table cliente (// **tipo enum**(‘Especial’, ‘Ouro’, ‘Premium’))* (enum pode ser nula, mas se não for **not null** e não preenchida, é atribuído o primeiro valor)

*... (// tipo varchar(10) **not null**, **check**(tipo = ‘Especial’ OR tipo = ‘ouro’ OR tipo = ‘Premium’))*

*...(// dataContratacao date, // dataInicioGer date, // **check**(dataContratacao > dataInicioGer))*

- **Chave Estrangeira:** *create table conta (// ...// nome_agencia char(15) **not null**,*

***foreign key**(nome_agencia) **references** agencia (nome))*

//

*create table agencia (// nome char(15) **primary key**) // create table conta (// ...// **foreign key**(nome_agencia) **references** agencia (nome))*

*... **foreign key**(nome_agencia) **references** agencia (nome)) **on delete** (ou Update) **cascade***

- **Modificar Tabela (Alter table):** Adicionar ou remover atributo/restrrição, além de alterar atributo

*alter table nomeTabela **add** columnm (opcional) nomeAtributo tipoAtributo = alter table agencia **add** tel varchar(15)*

*alter table nomeTabela **drop** columnm nomeAtributo = alter table agencia **drop** cidade*

*alter table nomeTabela **alter** columnm nomeAtributo novaDefinição = alter table agencia **alter** columnm tel varchar(30) **not null***

Esse é o padrão SQL, em MySQL usa-se **modify** ao invés de **alter**, em Postgresql coloca **set data type** antes de **varchar**

- **Modificar Restrições:** *alter table nomeTabela **add constraint** nome da restrição RESTRIÇÃO = ... constraint contaNum **Primary key**(num)*

Chave candidata: *... **add constraint** nomeConstraint **unique**(nomeColuna)*

Check: *alter table Conta **add constraint** conta_saldo_check **check**(saldo>0)*

Apagar restrição: *alter table nome da Tabela **drop** constraint nomeRestrição = alter table Conta **drop** constraint pkNum*

Apagar tabela: ***drop** table nomeTable = **drop** table agencia*

- **Comentários:** *-- ou /* ... */*

Para mais informações, acessar o PDF “[SQL-DDL](#)”

DML:

- **Inclusão Tupla:** Insert – **insert into** 'nome da relação' (lista de atributos) values (valores) – Os valores devem seguir a mesma ordem com que os atributos foram criados no comando create
- **Exclusão de Tuplas:** Delete – **delete from cliente**
- **Cláusula Where:** Define uma expressão da qual identifica as tuplas que devem ser consideradas (>,<,>=,<=,<>=) **where id = 3 / where id > 4 / where nome <> 'Joao'**
- **Exclusão de Tuplas com condição:** **delete from tabela//where**(condição)=**delete from cliente//where** nome='Maria'
- **AND e OR:** **delete from cliente // where** nome = 'Maria' **and** (ou então **or**) CNPJ = '11111'
- **Between:** **delete from cliente // where** no_cliente **between** 1 and 10 (seria "entre 1 e 10")
- **Update:** Atualização de dados – **update nome_tabela**
set coluna1 = valornovo, coluna2 = valornovo, ..., colunan = novovalor
update cliente // set cnpj= '00000'
- **Atualização com Condição:** **update cliente // set** cnpj= '00000' **// where** no_cliente = 1 (**where** condição)

Exemplos:

create table conta

```
(
numero char(5) primary key,
saldo numeric(2,1) not null,
nome_agencia char(15) not null,
foreign key (nome_agencia) references agencia
(nome)
)
```

add restrição

```
alter table departamento alter column nome
set not null
```

setar campo como unique

```
alter table funcionario add unique(identidade)
```

atributo multivalorado chave composta

```
create table telefone(
codf int not null,
numtel varchar(15) not null,
primary key(codf,numtel),
foreign key(codf) references funcionario(codigo)
)
```

datafim > datainicial

```
alter table projeto add constraint
projeto_datafim_datainicial check(datafim > datainicial)
```

apagar atributo

```
alter table alocao drop column
datai
```

adicionar atributo

```
alter table alocao add column
datai date
```

adicionar atributo com check

```
alter table alocao add qtdHora int not null check(qtdHora > 0);
```

adicionar atributo

```
alter table projeto add custo float not null;
```

adicionando chave estrangeira na tabela

```
alter table empregado add constraint cargo_fk foreign key(codCargo)
references cargo(codigo) on delete set null;
```

Inserção

```
insert into departamento values (1,'TI'),(2,'Qualidade');
insert into funcionario values
(1,'Bruno','222222','1134644646','bruno@gmail.com','1'),
(2,'Felipe','3333333','5534644646','felipe@gmail.com','1'),
(3,'Carlos','4444444','6634644646','carlos@gmail.com','1'),
(4,'Pedro','5555555','7734644646','pedro@gmail.com','2'),
(5,'Ana','6666666','8834644646','ana@gmail.com','2'),
(6,'Bia','7777777','9934644646','bia@gmail.com','2');
```

atualizar

```
update funcionario set nome='Bruno Moraes' where cpf='1134644646';
```

```
update alocao set qtdhora=30 where codf<>7; //atualizar onde codf for
diferente de 7
```

```
delete from departamento where codigo=1;
```

```
create table cliente (no_cliente int primary key, nome
varchar(150) not null, CNPJ varchar(5))
```

– Inserir uma tupla

```
insert into cliente values (1,'Maria','11111')
```

– Inserir mais de uma tupla

```
insert into cliente values (1,'Maria','11111') ,
(2,'Joao','22222')
```