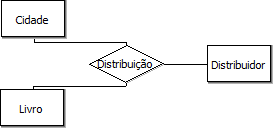
# Resumo P1: Abordagem Entidade-Relacionamento

### Associação entre entidades

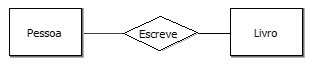
* **Atributo de uma entidade que se refere a outra**
* **Auto Relacionamento:** Relacionamento entre ocorrências da mesma entidade

Ficção --- Subordinada --- Romance

* **Grau de Relacionamento:** Binário

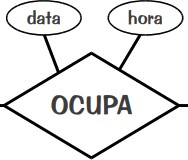


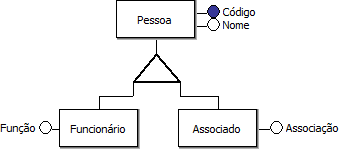
Ternário



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

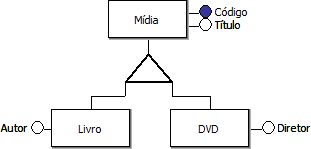
(0,1) = Nenhuma ou apenas uma (1,n) = Apenas uma ou várias (1,1) = Uma e apenas uma Pode-se variar entre diferentes valores específicos com a mesma lógica, porém esses são os mais comuns

* **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



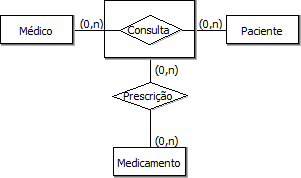
c

t



d

p

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

# Projeto Lógico - Modelo Relacional

### Mapeamento Entidades Fortes:

emprestimo(numero emprestimo, quantia)

* **Relacionamento:** É implementado com chave estrangeira



instituto (código, nome) funcionario (numero, nome) Onde incluir a chave estrangeira?

1. *Incluir a chave estrangeira em uma das tabelas*

instituto (código, nome, num\_gerente)

num\_ gerente referencia funcionario

funcionario (numero, nome)

1. *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)

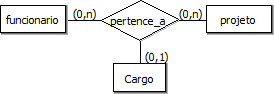
funcionario (numero, nome, codigo, descricao, bloco, sala, tipo\_restricao)

1. *N para N + com atributo*

conta (num, saldo,...) \*conta\_cliente(codigo, num, **tipo**) cliente (codigo, nome,...) codigo referencia cliente

(linha de baixo) \* num referencia conta

1. *Relacionamento N-ário: Relação U para o relacionamento*

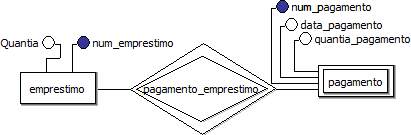
projeto (num, título,...) \*trabalho(num, codigoF, codigoC) funcionário (codigoF, nome,...) num referencia projeto

cargo (codigoC, nome,...)\* 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

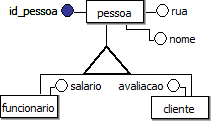
* **Mapeamento de Entidades Fracas:** É representado como uma relação

**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)

1. *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)

1. *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

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

**Linguagem de BD:** SQL “//” = Linha de baixo

**SGBD:** Postgres **DDL: Esquema** (Create, Drop, Alter)

## 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 digitos *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 tabel 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/restrição, além de alterar atributo

*alter table nomeTabela* ***add*** *columm* (opcional) *nomeAtributo tipoAtributo* ***=*** *alter table agencia* ***add*** *tel varchar(15) alter table nomeTabela* ***drop*** *columm nomeAtributo* ***=*** *alter table agencia* ***drop*** *cidade*

*alter table nomeTabela* ***alter*** *columm nomeAtributo novaDefinição* ***=*** *alter table agencia* ***alter*** *columm 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)*

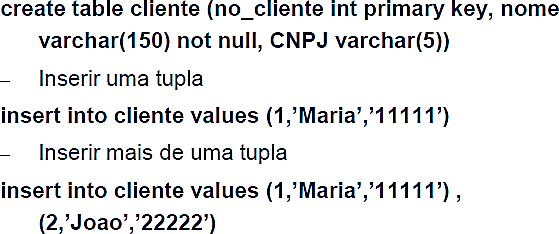
Chave candidata: *... add constraint nomeConstraint* ***unique*** *(nomeColuna)*

Check: *alter table* ***Conta*** *add constraint* ***conta\_saldo\_check*** *check (****saldo****>0)*

Apagar restrição: *alter table nomedaTabela* ***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**](https://sigaa.unifei.edu.br/sigaa/portais/discente/discente.jsf)**”**



## 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 alocacao drop column datai*

adicionar atributo

*alter table alocacao add column datai date*

adicionar atributo com check

*alter table alocacao 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 alocacao set qtdhora=30 where codf<>7; //atualizar onde codf for diferente de 7*

*delete from departamento where codigo=1;*