

# Banco de dados

## 03 - Projeto Lógico de Banco de Dados

---

Marcos Roberto Ribeiro



Instituto Federal Minas Gerais - Campus Bambuí

2018

# Restrições de Chave

- Antes de abordarmos o Projeto Lógico de Banco de Dados é importante tomarmos conhecimento de alguns conceitos a respeito de restrições de chave sobre tabelas;
- Uma **chave** é um subconjunto mínimo de colunas de uma tabela que identifica unicamente cada tupla da tabela;
- Por exemplo na tabela *Alunos(matrícula, cpf, nome, média)*, podemos considerar uma chave sobre a coluna *cpf*. Desta maneira, cada aluno possui um CPF único e não é possível inserir um novo aluno com o mesmo CPF de outro já existente na tabela;
- Uma tabela pode ter mais de uma chave, no caso da tabela anterior podemos ter como chaves: *cpf* e *matrícula*. Entretanto uma das chaves deve ser eleita a **chave primária** (normalmente a chave primária é definida de acordo com a análise de requisitos);
- Cada tabela deve possuir uma única chave primária, mas podemos fazer uso de outras chaves se for necessário, por exemplo, para impor uma restrição de unicidade<sup>1</sup>;
- Considerando novamente a tabela *Alunos*, podemos considerar como chave primária o campo *matricula* e como chave o campo *cpf* para impedir erros de inserção de CPF duplicados;

---

<sup>1</sup>Veremos em mais detalhes as restrições de unicidade quando estudarmos SQL

# Chaves Estrangeiras

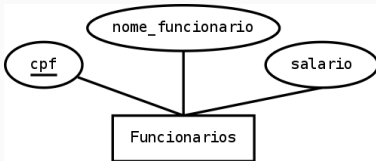
- Em determinadas situações os dados de uma tabela dependem dos dados de outra tabela. Neste caso, a alteração dos dados não pode ser feita sem verificar possíveis inconsistências;
- Por exemplo, considerando as tabelas:  
*Alunos*(*id\_aluno*: integer, *nome*: string, *idade*: integer, *media*: float)  
*Matriculados*(*\*id\_disciplina*: integer, *\*id\_aluno*: integer, *nota*: float)  
*\*Matriculados.id\_aluno* : *Alunos.id\_aluno*  
Uma tupla na tabela *Matriculados* não pode conter um valor em *id\_aluno* que não exista na tabela *Alunos*;
- Este tipo de restrição é chamado de **chave estrangeira**, ou seja, os valores das colunas que compõem a chave estrangeira devem corresponder a valores existentes na chave primária referenciada.
- No nosso exemplo a chave estrangeira é o campo *id\_aluno* da tabela *Matriculados* que referencia a chave primária da tabela *Alunos*;
- Note que as colunas que compõem alguma chave estrangeira são precedidos de \*;

# Conversão do DER para o Esquema Relacional

- O DER é conveniente para representar um projeto de banco de dados em alto nível;
- Porém para uma implementação prática de um banco de dados precisamos de um esquema lógico do mesmo;
- É possível obter um esquema lógico aproximado de um banco de dados a partir de seu DER.

# Entidades para Tabelas

- Uma entidade é mapeada em uma tabela de maneira direta;
- Cada atributo da entidade torna-se uma coluna na tabela seguida por seu respectivo tipo;
- Exemplo, considerando a seguinte entidade:



temos a seguinte tabela mapeada:

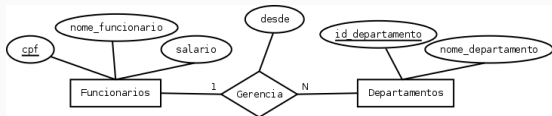
*Funcionários*(cpf: string, nome\_funcionario: string, salario: float)

# Mapeamento de Relacionamentos

- O mapeamento de relacionamentos deve considerar a cardinalidade. Como visto anteriormente as cardinalidades são as seguintes:
  - 1:N** Um para muitos;
  - 1:1** Um para um;
  - N:N** Muitos para muitos.

# Mapeamento de Relacionamentos com Cardinalidade 1:N

- Quando um relacionamento possui a cardinalidade 1:N (um-para-muitos), a chave primária da entidade com cardinalidade 1 e os atributos descritivos são mapeados para a entidade com cardinalidade N;
- A chave primária da tabela mapeada com cardinalidade 1 torna-se uma chave estrangeira na tabela mapeada com cardinalidade N;
- Exemplo, considerando o DER:



temos as seguintes tabelas mapeadas:

*Funcionários*(cpf: string, nome\_funcionario: string, salario: float)

*Departamentos*(id\_departamento: integer, nome\_departamento: string,  
\*cpf\_gerente: string, desde: date)

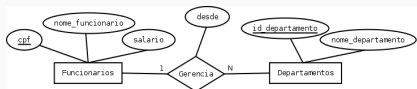
\**Departamentos.cpf\_gerente* : *Funcionários.cpf*

# Mapeamento de Relacionamentos com Cardinalidade 1:1

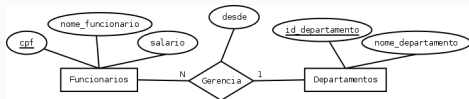
- Quando temos um relacionamento com cardinalidade 1:1, podemos fazer seu mapeamento considerando a cardinalidade 1:N;
- Como exemplo vamos considerar o relacionamento do DER a seguir com cardinalidade 1:1;



- Tal relacionamento pode ser mapeado de duas maneiras:



**Opção A**



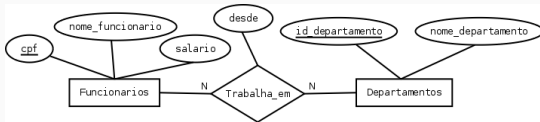
**Opção B**

- A cardinalidade 1:1 pode ser mantida através de uma restrição de unicidade que veremos nas próximas aulas.



# Mapeamento de Relacionamentos com Cardinalidade N:N

- O mapeamento de um relacionamento com cardinalidade N:N gera uma tabela. Tal mapeamento deve ser realizado considerando as chaves primárias das entidades participantes e os atributos descritivos;
- A chave primária da tabela resultante será formada pelas chaves primárias das entidades participantes;
- Além disto, a chave primária de cada entidade participante se tornará chave estrangeira na tabela;
- Exemplo, considerando o DER:



temos as seguintes tabelas mapeadas:

*Funcionários*(cpf: string, nome\_funcionario: string, salario: float)

*Departamentos*(id\_departamento: integer, nome\_departamento: strings)

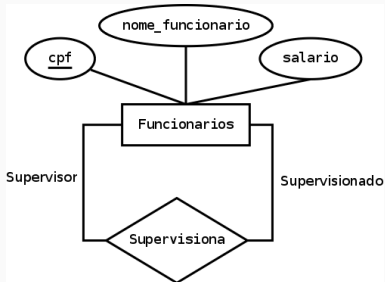
*Trabalha\_em*(\*id\_funcionario: string, \*id\_departamento: string, desde: date)

\*trabalha\_em.cpf : Funcionários.cpf

\*trabalha\_em.id\_departamento : Departamentos.id\_departamento




# Mapeamento de Auto-Relacionamentos

- No caso de auto-relacionamentos utilizamos os papéis associados ao relacionamento para compor os nomes das colunas da tabela;
- Exemplo, considerando o DER:



temos as seguintes tabelas mapeadas:

```
Funcionários(cpf: string, nome_funcionario: string, salario: float)
Supervisiona(*supervisor_cpf: string, *supervisionado_cpf: string)
*Supervisiona.supervisor_cpf : Funcionários.cpf
*Supervisiona.supervisionado_cpf : Funcionários.cpf
```

-  Elmasri, R. and Navathe, S. B. (2011).  
***Sistemas de banco de dados.***  
Pearson Addison Wesley, São Paulo, 6 edition.
-  Ramakrishnan, R. and Gehrke, J. (2008).  
***Sistemas de gerenciamento de banco de dados.***  
McGrawHill, São Paulo, 3 edition.
-  Takahashi, M. (2009).  
***Guia Mangá de Banco de Dados.***  
Novatec, São Paulo.