

# MAC0350: Introdução ao Desenvolvimento de Sistemas de Software

Aula: Mapeamento e dúvidas gerais

João Eduardo Ferreira;  
Décio Lauro Soares

Universidade de São Paulo (USP)

Instituto de Matemática e Estatística - USP - 2019



# Table of Contents

1 **Introdução**  
Objetivos

2 Modelagem  
Nosso modelo  
Modelagem  
Construção da base

3 Modelo conceitual final  
Nosso modelo para o EP

4 QA  
Contato

Os objetivos gerais dessa aula são:

- ▶ Revisar os conceitos envolvendo o ciclo completo de modelagem sobre um exemplo simples (MER-X  $\rightarrow$  Lógico  $\rightarrow$  Físico);
- ▶ Discutir adaptações desse modelo ao modelo que será trabalhado no EP;
- ▶ Tratar de dúvidas gerais.

Entretanto, por questões de pré-requisitos, no modelo lógico ainda não será abordado:

- ▶ Dependências Funcionais;
- ▶ Formas normais.

# Table of Contents

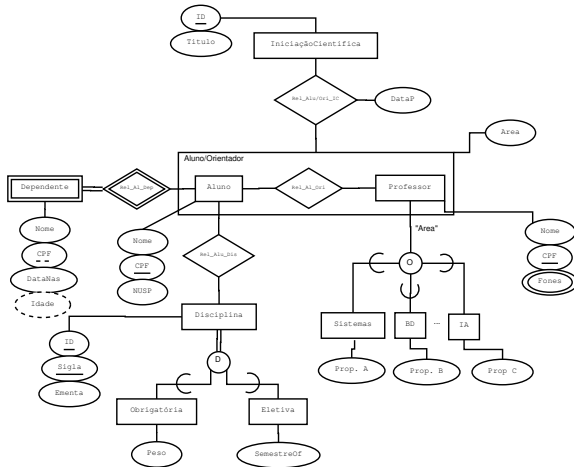
- 1 Introdução  
Objetivos
- 2 **Modelagem**  
Nosso modelo  
Modelagem  
Construção da base
- 3 Modelo conceitual final  
Nosso modelo para o EP
- 4 QA  
Contato

Design Conceitual **Simplificado** (Sem descrição detalhada, análise de requisitos ou restrições).

Sistema para gestão de alunos, professores e matriculas capaz de:

- ▶ Verificar as informações pessoais dos alunos (incluindo matriculas e dependentes),
- ▶ Verificar as informações pessoais e profissionais dos professores,
- ▶ Verificar as informações das disciplinas oferecidas,
- ▶ Quando necessário, listar informações sobre iniciações científicas realizadas.

Possível representação desse modelo construído para fins didáticos. (Ou seja, super-forçado)



Algoritmo de mapeamento MER-X  $\rightarrow$  Relacional (Repita até à convergência :-)): (1-9 extraído de Elmasri/Navati)

1. Mapeamento de entidades regulares.
2. Mapeamento de entidades fracas.
3. Mapeamento de relacionamento binário 1:1.
4. Mapeamento de relacionamento binário 1:N.
5. Mapeamento de relacionamento binário M:N.
6. Mapeamento de atributos multi valorados.
7. Mapeamento de relacionamento N-ário ( $N > 2$ ).
8. Mapeamento de especialização ou generalização.
9. Mapeamento de categorias (Union Types).
10. Mapeamento de agregações.



## Passo 1: Mapeamento de entidades regulares.

- ▶ Para cada entidade regular  $E$  crie a relação  $R$  que inclua todos os atributos **simples** de  $E$ .
- ▶ Escolha o conjunto de chaves primárias de  $E$ .
- ▶ Caso a entidade possua chaves múltiplas (secundária, terciária, ...) represente-as.

## Passo 2: Mapeamento de entidades fracas.

- ▶ Para cada entidade fraca  $W$  crie a relação  $R$  que inclua todos os atributos **simples** de  $W$ .
- ▶ Inclua também campos suficientes para mapear toda a chave estrangeira relacionada à entidade pai.
- ▶ Defina como chave primária de  $R$  o conjunto de chaves de  $W$  **mais** o conjunto de chaves estrangeiras do pai.
- ▶ Caso o pai de uma entidade  $E_2$  seja  $E_1$  e ambos sejam entidades fracas, mapeie  $E_1$  primeiro.

### Passo 3: Mapeamento de relacionamentos binários 1:1. (entre $S$ e $T$ )

- ▶ Escolha um dos métodos de mapeamento (Foreign key, Merged relation ou Cross-reference).
- ▶ (FK) Escolha “de forma inteligente” uma relação (por exemplo  $S$ ) e inclua como chave estrangeira a chave primária de  $T$ .
- ▶ (MR) Quando **ambas as participações são totais**, faça um merge das duas entidades.
- ▶ (CR) Proceda de maneira similar ao passo 5.

## Passo 4: Mapeamento de relacionamentos binários 1:N. (entre $S$ e $T$ )

- ▶ Escolha um dos métodos de mapeamento (Regular ou Cross-reference).
- ▶ (R) Para cada relacionamento binário 1:N  $R$ , identifique a relação  $S$  que participa “pelo lado N da relação”.
- ▶ (R) Inclua em  $S$  um campo para uma chave estrangeira que aponta para a chave primária de  $T$ .
- ▶ (CR) Proceda de maneira similar ao passo 5.

Passo 5: Mapeamento de relacionamentos binários N:M. (entre  $S$  e  $T$ )

- ▶ Para cada relacionamento binário N:M  $R$ , crie uma nova relação  $U$  para representar  $R$ .
- ▶ Inclua como chave estrangeira de  $U$  as chaves primárias de  $S$  e  $T$ .
- ▶ Caso haja, inclua também todos os atributos simples da relação  $R$ .

## Passo 6: Mapeamento de atributos multi valorados.

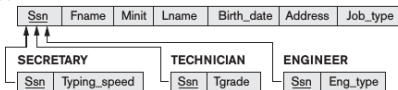
- ▶ Escolha um dos métodos (Regular ou Unroll).
- ▶ (R) Para cada atributo multi valorado  $A$ , crie uma nova relação  $R$  e inclua um atributo correspondente à  $A$  **mais** uma chave estrangeira  $K$  que aponta para a chave primária da relação sobre qual o atributo multi valorado  $A$  está definido.
- ▶ (R) Defina a chave primária de  $R$  como sendo a combinação do atributo criado com  $K$ .
- ▶ (U) Quando o número de termos do atributo multi valorado é **conhecido**, separe o atributo composto em  $n$  atributos simples.

## Passo 7: Mapeamento de relacionamento N-ário ( $N > 2$ ).

- ▶ Para cada relacionamento N-ário  $R$  onde  $N > 2$ , crie um novo relacionamento  $U$  para representar  $R$ .
- ▶ Inclua como chave estrangeira em  $U$  as chaves primárias das relações que compõem o relacionamento N-ário.
- ▶ Caso haja, inclua também todos os atributos simples do relacionamento N-ário.
- ▶ Cuidado nos casos onde a cardinalidade é 1!

## Passo 8: Mapeamento de especialização ou generalização.

(a) EMPLOYEE



(b) CAR

<u>Vehicle_id</u>	License_plate_no	Price	Max_speed	No_of_passengers
-------------------	------------------	-------	-----------	------------------

TRUCK

<u>Vehicle_id</u>	License_plate_no	Price	No_of_axles	Tonnage
-------------------	------------------	-------	-------------	---------

(c) EMPLOYEE

<u>Ssn</u>	Fname	Minit	Lname	Birth_date	Address	Job_type	Typing_speed	Tgrade	Eng_type
------------	-------	-------	-------	------------	---------	----------	--------------	--------	----------

(d) PART

<u>Part_no</u>	Description	Mflag	Drawing_no	Manufacture_date	Batch_no	Pflag	Supplier_name	List_price
----------------	-------------	-------	------------	------------------	----------	-------	---------------	------------

**Figure 9.5**

Options for mapping specialization or generalization. (a) Mapping the EER schema in Figure 8.4 using option 8A. (b) Mapping the EER schema in Figure 8.3(b) using option 8B. (c) Mapping the EER schema in Figure 8.4 using option 8C. (d) Mapping Figure 8.5 using option 8D with Boolean type fields Mflag and Pflag.

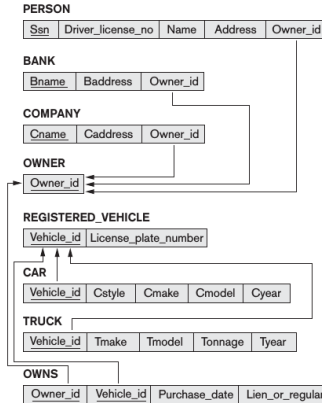
Figure: Fundamentals of Database Systems, 6th ed. Elmasri/Navati, pg. 295



## Passo 9: Mapeamento de categorias (Union Types).

**Figure 9.7**

Mapping the EER categories (union types) in Figure 8.8 to relations.

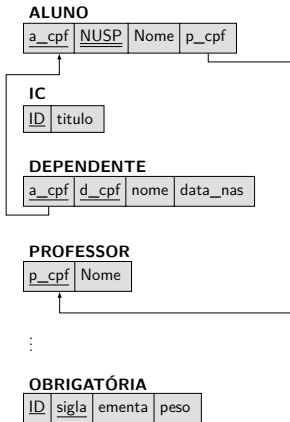


**Figure:** Fundamentals of Database Systems, 6th ed. Elmasri/Navati, pg. 298

## Passo 10: Mapeamento de agregações.

- ▶ Para cada agregação  $G$ , crie um novo relacionamento  $R$  para representar  $G$ .
- ▶ Inclua como chave estrangeira em  $R$  as chaves primárias das relações que compõem a agregação  $G$ .
- ▶ Caso haja, inclua também todos os atributos simples da agregação.

Por fim, seguindo os passos apresentados, uma parte do nosso mapeamento ficou:



Falta agora fazer o mapeamento do modelo lógico para o físico (SQL).

- ▶ Definir o SGBD à ser utilizado (postgresql).
- ▶ Definir o domínio e as restrições.
- ▶ Escrever o SQL.

## Apenas o início...

### example.sql

```
1 CREATE TABLE UD01_PROF (  
2     pr_nome    varchar(100) not NULL,  
3     pr_cpf     integer ,  
4     CONSTRAINT pk_pr PRIMARY KEY (pr_cpf)  
5 );  
6  
7 CREATE TABLE UD02_ALUNO (  
8     al_nome    varchar(100) not NULL,  
9     al_cpf     integer ,  
10    pr_cpf     integer ,  
11    CONSTRAINT pk_al PRIMARY KEY (al_cpf) ,  
12    CONSTRAINT fk_pr FOREIGN KEY (pr_cpf)  
13        REFERENCES UD01_PROF(pr_cpf)  
14        ON DELETE CASCADE  
15        ON UPDATE CASCADE  
16 );
```

Agora, de posse do modelo físico, vamos testar esse modelo no banco.

### Listing 1: Execução parte 1

```
1 $ psql --version
2 psql (PostgreSQL) 9.5.14
3
4 $ python --version
5 Python 2.7.12
6
7 $ python3 --version
8 Python 3.5.2
9
10 $ createdb exemplo1
11
12 $ psql exemplo1
13 psql (9.5.14)
14 Type "help" for help.
15
16 exemplo1=> \dt
17 No relations found.
18
19 exemplo1=> \q
```

## Listing 2: Execução parte 2

```
1 $ psql -d exemplo1 -a -f example.sql
2 CREATE TABLE UD01_PROF (
3     pr_nome    varchar(100) not NULL,
4     pr_cpf     integer,
5     CONSTRAINT pk_pr PRIMARY KEY (pr_cpf)
6 );
7 CREATE TABLE
8
9 CREATE TABLE UD02_ALUNO (
10     al_nome    varchar(100) not NULL,
11     al_cpf     integer,
12     pr_cpf     integer,
13     CONSTRAINT pk_al PRIMARY KEY (al_cpf),
14     CONSTRAINT fk_pr FOREIGN KEY (pr_cpf)
15         REFERENCES UD01_PROF(pr_cpf)
16         ON DELETE CASCADE
17         ON UPDATE CASCADE
18 );
19 CREATE TABLE
```

### Listing 3: Execução parte 3

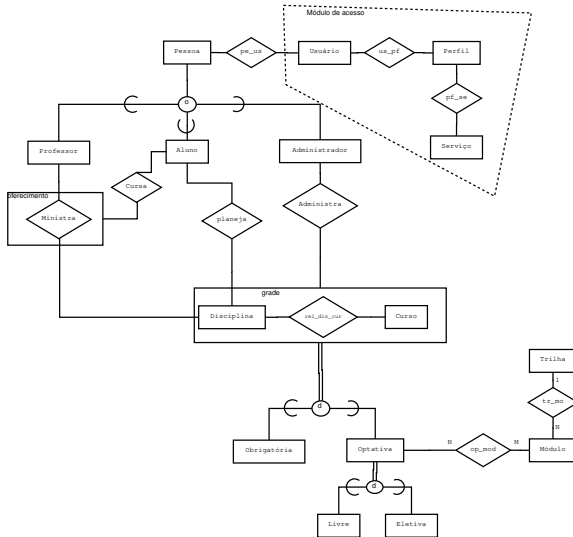
```
1 exemplo1=> INSERT INTO UD01_PROF(pr_nome, pr_cpf) VALUES ('Jef', 111222);
2 INSERT 0 1
3 exemplo1=> INSERT INTO UD01_PROF(pr_nome, pr_cpf) VALUES ('Decio', 333444);
4 INSERT 0 1
5 exemplo1=> INSERT INTO UD02_ALUNO(al_nome, al_cpf, pr_cpf) VALUES ('Alberto', 444555, 111222);
6 INSERT 0 1
7 exemplo1=> INSERT INTO UD02_ALUNO(al_nome, al_cpf, pr_cpf) VALUES ('Bernardo', 666777, 111222);
8 INSERT 0 1
9 exemplo1=> INSERT INTO UD02_ALUNO(al_nome, al_cpf, pr_cpf) VALUES ('Carlos', 888999, 333444);
10 INSERT 0 1
11 exemplo1=> SELECT A.al_nome
12 FROM UD01_PROF AS P
13 INNER JOIN UD02_ALUNO AS A
14 ON P.pr_cpf = A.pr_cpf
15 WHERE P.pr_nome = 'Decio';
16      al_nome
17 -----
18      Carlos
19 (1 row)
```



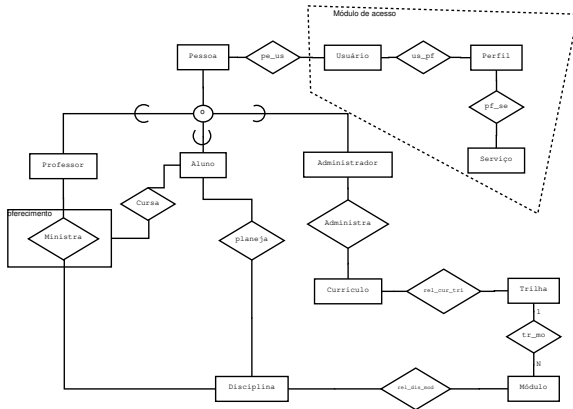
## Listing 4: Execução parte 4

```
1 exemplo1=> delete from UD01_PROF AS P
2 where P.pr_nome = 'Decio';
3 DELETE 1
4 exemplo1=> select * from UD02_ALUNO;
5   al_nome | al_cpf | pr_cpf
6   -----|-----|-----
7   Alberto | 444555 | 111222
8   Bernardo | 666777 | 111222
9   (2 rows)
10 exemplo1=>\q
11
12 $ dropdb exemplo1
13 $ psql exemplo1
14 psql: FATAL:  database "exemplo1" does not exist
```

Modelo conceitual simplificado para o EP. (Sem atributos e passível de pequenas mudanças)



Modelo conceitual simplificado para o EP. (Sem atributos e passível de pequenas mudanças)



Dúvidas? Sugestões? Revisões?

**Contato:**

- ▶ João Eduardo Ferreira (jef@ime.usp.br)
- ▶ Décio Lauro Soares (deciolauro@gmail.com)