

UNIVERSIDADE FEDERAL DO RIO GRANDE DO NORTE  
ESCOLA AGRÍCOLA DE JUNDIAÍ  
ANÁLISE E DESENVOLVIMENTO DE SISTEMAS

CARLA FERNANDES CURVELO  
[CARLAFCF@GMAIL.COM](mailto:CARLAFCF@GMAIL.COM)

TAD0010 – BANCO DE DADOS II

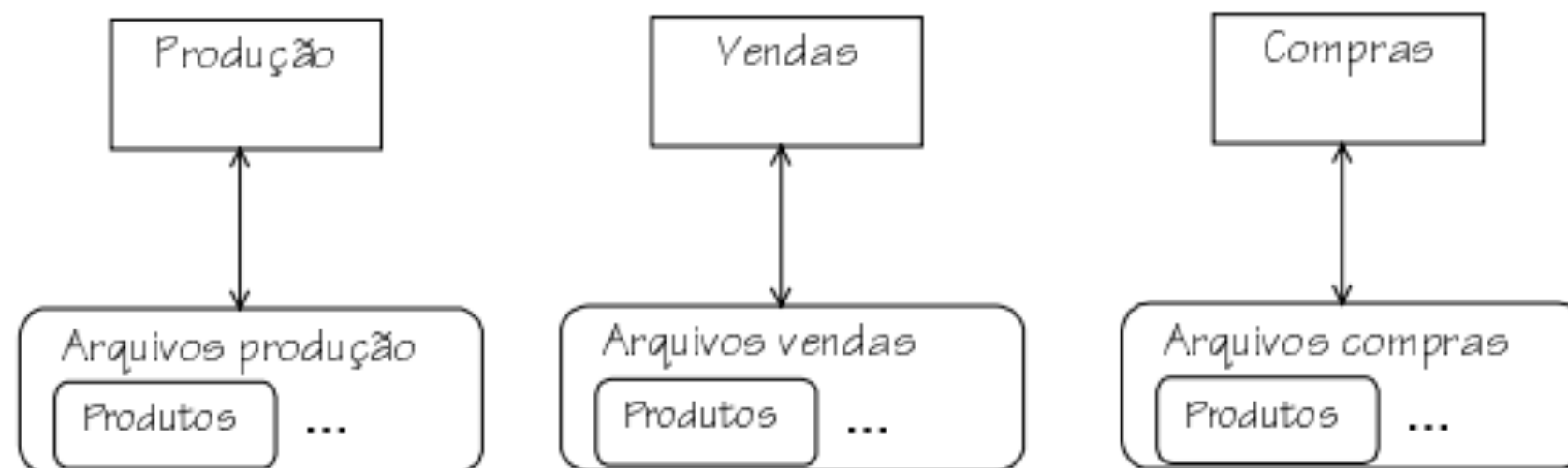
---

# AULA 2

# REVISÃO

# CONCEITOS BÁSICOS DE BANCO DE DADOS

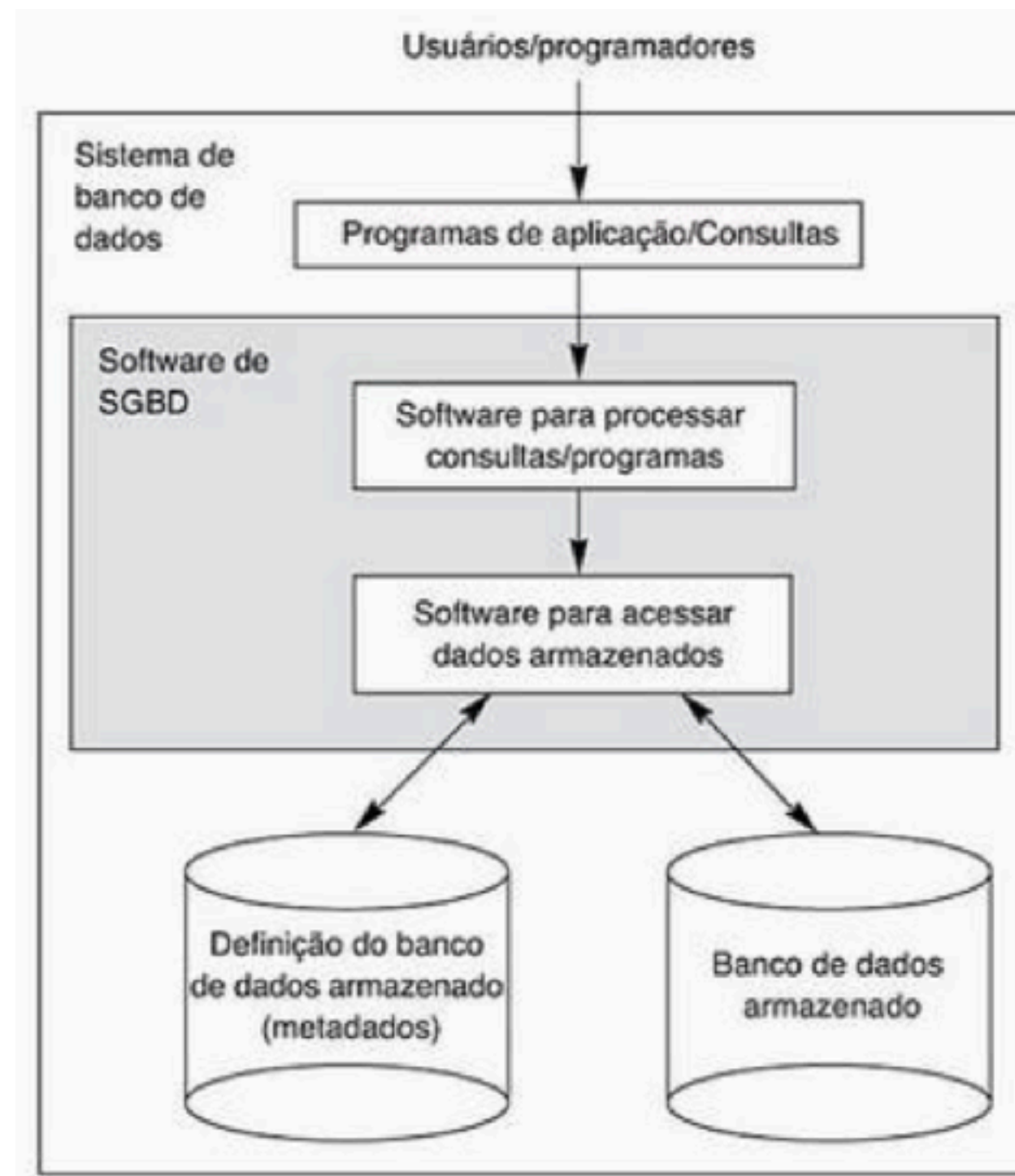
- ▶ Exemplos de sistemas que utilizam **banco de dados**?
- ▶ **Abordagem antiga:** programação com arquivos
  - ▶ Redundância na informação
  - ▶ Desperdício no espaço de armazenamento
  - ▶ Esforço redundante para manter os dados comuns atualizados (redigitação e inconsistência de dados)
  - ▶ Elementos de dados nomeados independentes



# CONCEITOS BÁSICOS DE BANCO DE DADOS

- ▶ Exemplos de sistemas que utilizam **banco de dados**?
- ▶ **SGBD**: Sistema Gerenciador de Banco de Dados
  - ▶ Definição: especificar os tipos, estruturas e restrições dos dados
  - ▶ Construção: processo de armazenar os dados em algum meio controlado pelo SGBD
  - ▶ Manipulação: funções como consultas e atualizações dos dados
  - ▶ Compartilhamento: permite que diversos usuários e programas acessem o banco de dados simultaneamente

# CONCEITOS BÁSICOS DE BANCO DE DADOS



## EXEMPLO – UNIVERSIDADE

### ALUNO

Nome	Numero_aluno	Tipo_aluno	Curso
Silva	17	1	CC
Braga	8	2	CC

### DISCIPLINA

Nome_disciplina	Numero_disciplina	Creditos	Departamento
Introd. à ciência da computação	CC1310	4	CC
Estruturas de dados	CC3320	4	CC
Matemática discreta	MAT2410	3	MAT
Banco de dados	CC3380	3	CC

### PRE\_REQUISITO

Numero_disciplina	Numero_pre_requisito
CC3380	CC3320
CC3380	MAT2410
CC3320	CC1310

### TURMA

Identificacao_turma	Numero_disciplina	Semestre	Ano	Professor
85	MAT2410	Segundo	07	Kleber
92	CC1310	Segundo	07	Anderson
102	CC3320	Primeiro	08	Carlos
112	MAT2410	Segundo	08	Chang
119	CC1310	Segundo	08	Anderson
135	CC3380	Segundo	08	Santos

### HISTORICO\_ESCOLAR

Numero_aluno	Identificacao_turma	Nota
17	112	B
17	119	C
8	85	A
8	92	A
8	102	B
8	135	A

## EXEMPLO – UNIVERSIDADE

### ▶ Consultas:

- ▶ Recuperar uma lista de todas as disciplinas e notas de 'Silva'
- ▶ Listar os nomes dos alunos que realizaram a disciplina 'Banco de dados' oferecida no segundo semestre de 2008 e suas notas nessa turma
- ▶ Listar os pré-requisitos do curso de 'Banco de dados'

### ▶ Atualizações:

- ▶ Alterar o tipo de aluno de 'Silva' para segundo ano
- ▶ Criar outra turma para a disciplina 'Banco de dados' para este semestre
- ▶ Inserir uma nota 'A' para 'Silva' na turma 'Banco de dados' do último semestre

# PROJETO DE BANCO DE DADOS

1. Especificação e análise de requisitos
2. Projeto conceitual
  - ▶ Projeto que pode ser implementado em um banco de dados
  - ▶ Exemplo: DER
3. Projeto lógico
  - ▶ Modelo de dados implementado em um SGBD comercial
4. Projeto físico

# MODELO DE DADOS RELACIONAL

**ALUNO** (Nome, Cpf, Telefone\_residencial, Endereco, Telefone\_comercial, Idade, Media)

Nome da relação

**ALUNO**

Atributos

Nome	Cpf	Telefone_residencial	Endereco	Telefone_comercial	Idade	Media
Bruno Braga	305.610.243-51	(17)3783-1616	Rua das Paineiras, 2918	NULL	19	3,21
Carlos Kim	381.620.124-45	(17)3785-4409	Rua das Goiabeiras, 125	NULL	18	2,89
Daniel Davidson	422.111.232-70	NULL	Avenida da Paz, 3452	(17)4749-1253	25	3,53
Roberta Passos	489.220.110-08	(17)3476-9821	Rua da Consolação, 265	(17)3749-6492	28	3,93
Barbara Benson	533.690.123-80	(17)3239-8461	Rua Jardim, 7384	NULL	19	3,25

Tuplas



## MODELO DE DADOS RELACIONAL

- ▶ O que aconteceria se fosse utilizada uma única tabela para representar mais de uma coisa?
  - ▶ Exemplo: ALUNO\_CURSO
  - ▶ Repetição de informação
  - ▶ Valores nulos

# RESTRICÇÕES EM MODELO RELACIONAL

- ▶ Restrição de tuplas duplicadas
- ▶ **Superchave**
  - ▶ Qualquer conjunto de atributos que garante que duas tuplas distintas não possuem valores idênticos para todos os atributos da chave
  - ▶ **Superchave padrão:** conjunto de todos os atributos da relação
  - ▶ **Problema:** atributos redundantes
    - ▶ **Chave:** superchave na qual a remoção de qualquer atributo resulta em um conjunto de atributos que não é mais uma superchave

# RESTRICÇÕES EM MODELO RELACIONAL

## ► Superchaves:

{Nome, Cpf, Tel1, Endereco, Tel2, Idade, Media}

{Cpf}

{Nome, Cpf}

{Cpf, Nome, Idade}

...

## ► Chaves:

► {Cpf}

ALUNO						
Nome	Cpf	Telefone_ residencial	Endereco	Telefone_ comercial	Idade	Media
Daniel Davidson	422.111.232-70	NULL	Avenida da Paz, 3452	(17)4749-1253	25	3,53
Barbara Benson	533.690.123-80	(17)3239-8461	Rua Jardim, 7384	NULL	19	3,25
Roberta Passos	489.220.110-08	(17)3476-9821	Rua da Consolação, 265	(17)3749-6492	28	3,93
Carlos Kim	381.620.124-45	(17)3785-4409	Rua das Goiabeiras, 125	NULL	18	2,89
Bruno Braga	305.610.243-51	(17)3783-1616	Rua das Paineiras, 2918	NULL	19	3,21

# RESTRICÇÕES EM MODELO RELACIONAL

## ► Superchaves:

{Placa, Chassi, Marca, Modelo, Ano}

{Placa, Chassi, Modelo}

{Placa, Chassi}

{Placa}

{Chassi}

...

## ► Chaves Candidatas:

{Placa}

{Chassi}

## ► Chave Primária:

{Placa} ou {Chassi}

CARRO				
<u>Placa</u>	Numero_chassi	Marca	Modelo	Ano
Itatiaia ABC-7039	A6935207586	Volkswagen	Gol	02
Itu TVP-3470	B4369668697	Chevrolet	Corsa	05
Santos MPO-2902	X8355447376	Fiat	Uno	01
Itanhaem TFY-6858	C4374268458	Chevrolet	Celta	99
Itatiba RSK-6279	Y8293586758	Renault	Clio	04
Atibaia RSK-6298	U0283657858	Volkswagen	Parati	04

# ESQUEMA DE BANCO DE DADOS RELACIONAL

**EMPRESA** = {FUNCIONARIO, DEPARTAMENTO, LOCALIZACAO\_DEP, PROJETO, TRABALHA\_EM, DEPENDENTE}

## **FUNCIONARIO**

Prnome	Minicial	Unome	<u>Cpf</u>	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
--------	----------	-------	------------	----------	----------	------	---------	----------------	-----

## **DEPARTAMENTO**

Dnome	<u>Dnumero</u>	Cpf_gerente	Data_inicio_gerente
-------	----------------	-------------	---------------------

## **LOCALIZACAO\_DEP**

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------

## **PROJETO**

Projnome	<u>Projnumero</u>	Projlocal	Dnum
----------	-------------------	-----------	------

## **TRABALHA\_EM**

<u>Fcpf</u>	<u>Pnr</u>	Horas
-------------	------------	-------

## **DEPENDENTE**

<u>Fcpf</u>	<u>Nome_dependente</u>	Sexo	Datanasc	Parentesco
-------------	------------------------	------	----------	------------

# ESTADO DE BANCO DE DADOS RELACIONAL

## FUNCIONARIO

Pnome	Minicial	Unome	Cpf	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
João	B	Silva	12345678966	09-01-1965	Rua das Flores, 751, São Paulo, SP	M	30.000	33344555587	5
Fernando	T	Wong	33344555587	08-12-1955	Rua da Lapa, 34, São Paulo, SP	M	40.000	88866555576	5
Alice	J	Zelaya	99988777767	19-01-1968	Rua Souza Lima, 35, Curitiba, PR	F	25.000	98765432168	4
Jennifer	S	Souza	98765432168	20-06-1941	Av. Arthur de Lima, 54, Santo André, SP	F	43.000	88866555576	4
Ronaldo	K	Lima	66688444476	15-09-1962	Rua Rebouças, 65, Piracicaba, SP	M	38.000	33344555587	5
Joice	A	Leite	45345345376	31-07-1972	Av. Lucas Obes, 74, São Paulo, SP	F	25.000	33344555587	5
André	V	Pereira	98798798733	29-03-1969	Rua Timbira, 35, São Paulo, SP	M	25.000	98765432168	4
Jorge	E	Brito	88866555576	10-11-1937	Rua do Horto, 35, São Paulo, SP	M	55.000	NULL	1

## DEPARTAMENTO

Dnome	Dnumero	Cpf_gerente	Data_inicio_gerente
Pesquisa	5	33344555587	22-05-1988
Administração	4	98765432168	01-01-1995
Matriz	1	88866555576	19-06-1981

## LOCALIZACAO\_DEP

Dnumero	Dlocal
1	São Paulo
4	Mauá
5	Santo André
5	Itu
5	São Paulo





# ESTADO DE BANCO DE DADOS RELACIONAL

TRABALHA_EM			PROJETO			
Fcpf	Pnr	Horas	Projnome	Projnumero	Projlocal	Dnum
12345678966	1	32,5	ProdutoX	1	Santo André	5
12345678966	2	7,5	ProdutoY	2	Itu	5
66688444476	3	40,0	ProdutoZ	3	São Paulo	5
45345345376	1	20,0	Informatização	10	Mauá	4
45345345376	2	20,0	Reorganização	20	São Paulo	1
33344555587	2	10,0	Novosbenefícios	30	Mauá	4
33344555587	3	10,0				
33344555587	10	10,0				
33344555587	20	10,0				
99988777767	30	30,0				
99988777767	10	10,0				
98798798733	10	35,0				
98798798733	30	5,0				
98765432168	30	20,0				
98765432168	20	15,0				
88866555576	20	NULL				

DEPENDENTE				
Fcpf	Nome_dependente	Sexo	Datanasc	Parentesco
33344555587	Alicia	F	05-04-1986	Filha
33344555587	Tiago	M	25-10-1983	Filho
33344555587	Janaina	F	03-05-1958	Esposa
98765432168	Antonio	M	28-02-1942	Marido
12345678966	Michael	M	04-01-1988	Filho
12345678966	Alicia	F	30-12-1988	Filha
12345678966	Elizabeth	F	05-05-1967	Esposa



# RESTRICÇÕES DE INTEGRIDADE

- ▶ **Integridade de entidade:** chaves primárias não podem ter valores NULL
- ▶ **Integridade referencial:** uma tupla em uma relação que referencia outra relação precisa se referir a uma tupla existente nessa relação
  - ▶ Chave estrangeira (*Foreign key*)

## FUNCIONARIO

Pnome	Minicial	Unome	<u>Cpf</u>	Datanasc	Endereco	Sexo	Salario	Cpf_supervisor	Dnr
-------	----------	-------	------------	----------	----------	------	---------	----------------	-----

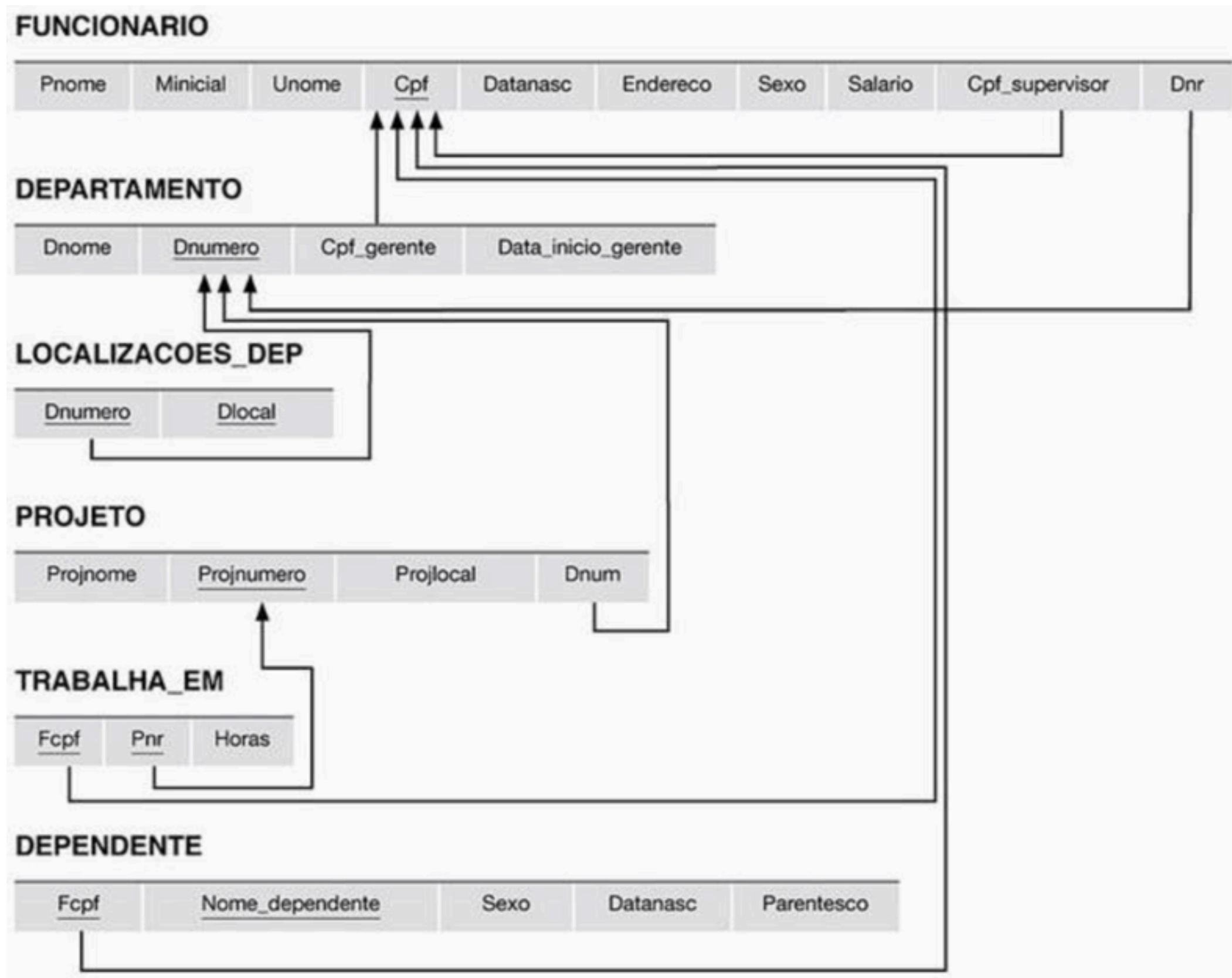
## DEPARTAMENTO

Dnome	<u>Dnumero</u>	Cpf_gerente	Data_inicio_gerente
-------	----------------	-------------	---------------------

## LOCALIZACAO\_DEP

<u>Dnumero</u>	<u>Dlocal</u>
----------------	---------------





# SQL

- ▶ *Structure Query Language* (Linguagem de consulta estruturada)
- ▶ Linguagem de banco de dados
  - ▶ Definição de dados, consultas e atualizações
  - ▶ DDL (*Data Definition Language*) e DML (*Data Manipulation Language*)

# SQL

**CREATE SCHEMA** GOT;

**CREATE TABLE** CASA (  
    Cnome VARCHAR(15) **NOT NULL**,  
    Cnumero CHAR(11),  
    **PRIMARY KEY** (Cnumero));

**CREATE TABLE** PERSONAGEM (  
    Pnome VARCHAR(15) **NOT NULL**,  
    Unome VARCHAR(15) **NOT NULL**,  
    Codigo CHAR(11),  
    Datanasc DATE,  
    Residencia VARCHAR(30),  
    Sexo CHAR,  
    Valor DECIMAL(10,2),  
    Codigo\_chefe CHAR(11),  
    Cnr INT **DEFAULT** 1,  
    **PRIMARY KEY** (Codigo),  
    **FOREIGN KEY** (Codigo\_chefe) **REFERENCES** PERSONAGEM(Codigo),  
    **FOREIGN KEY** (Cnr) **REFERENCES** CASA(Cnumero));

## SQL

```
INSERT INTO PERSONAGEM VALUES ('Eddard', 'Stark',  
'111111111111', '1955-11-09', 'Winterfell', 'M', 50000, NULL, 1);
```

```
INSERT INTO PERSONAGEM (Pnome, Unome, Cnr, Codigo)  
VALUES ('Eddard', 'Stark', 1, '111111111111');
```

```
DELETE FROM PERSONAGEM WHERE Unome='Stark';
```

```
DELETE FROM PERSONAGEM;
```

```
UPDATE PERSONAGEM  
SET Valor = Valor* 1.1  
WHERE Cnr=1;
```

## SQL – EXEMPLOS

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc,  
Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero,  
Data\_inicio)

**EPISODIO** (Episodio\_nome, Episodio\_numero,  
Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo,  
Datnasc, Parentesco)

## SQL – PESQUISA 1

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar a data de nascimento e o endereço do(s) personagens(s) cujo nome seja 'Eddard Stark'

## SQL – PESQUISA 1

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar a data de nascimento e o endereço do(s) personagens(s) cujo nome seja 'Eddard Stark'

```
SELECT Datnasc, Residencia  
FROM PERSONAGEM  
WHERE Pnome='Eddard' AND Unome='Stark';
```

PRODUTO CARTESIANO

CASA

Cnome	Cnumero
Stark	1
Bolton	2
Greyjoy	3

PERSONAGEM

Pnome	Unome	Codigo	Cnr
Eddard	Stark	1111	1
Sansa	Stark	2222	1
Roose	Bolton	4444	2

CASA, PERSONAGEM

Cnome	Cnumero	Pnome	Unome	Codigo	Cnr
Stark	1	Eddard	Stark	1111	1
Stark	1	Sansa	Stark	2222	1
Stark	1	Roose	Bolton	4444	2
Bolton	2	Eddard	Stark	1111	1
Bolton	2	Sansa	Stark	2222	1
Bolton	2	Roose	Bolton	4444	2
Greyjoy	3	Eddard	Stark	1111	1
Greyjoy	3	Sansa	Stark	2222	1
Greyjoy	3	Roose	Bolton	4444	2



PRODUTO CARTESIANO

CASA

Cnome	Cnumero
Stark	1
Bolton	2
Greyjoy	3

PERSONAGEM

Pnome	Unome	Codigo	Cnr
Eddard	Stark	1111	1
Sansa	Stark	2222	1
Roose	Bolton	4444	2

CASA, PERSONAGEM

Cnome	Cnumero	Pnome	Unome	Codigo	Cnr
Stark	1	Eddard	Stark	1111	1
Stark	1	Sansa	Stark	2222	1
Stark	1	Roose	Bolton	4444	2
Bolton	2	Eddard	Stark	1111	1
Bolton	2	Sansa	Stark	2222	1
Bolton	2	Roose	Bolton	4444	2
Greyjoy	3	Eddard	Stark	1111	1
Greyjoy	3	Sansa	Stark	2222	1
Greyjoy	3	Roose	Bolton	4444	2

# PRODUTO CARTESIANO

CASA

Cnome	Cnumero
Stark	1
Bolton	2
Greyjoy	3

PERSONAGEM

Pnome	Unome	Codigo	Cnr
Eddard	Stark	1111	1
Sansa	Stark	2222	1
Roose	Bolton	4444	2

CASA, PERSONAGEM

Cnome	Cnumero	Pnome	Unome	Codigo	Cnr
Stark	1	Eddard	Stark	1111	1
Stark	1	Sansa	Stark	2222	1
Bolton	2	Roose	Bolton	4444	2

## SQL – PESQUISA 2

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o nome e a residência de todos os personagens que pertencem à casa 'Stark'.

## SQL – PESQUISA 2

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o nome e a residência de todos os personagens que pertencem à casa 'Stark'.

```
SELECT Pnome, Unome, Residencia  
FROM PERSONAGEM, CASA  
WHERE Cnome='Stark' AND Cnr=Cnumero;
```

```
SELECT P.Pnome, P.Unome, P.Residencia  
FROM PERSONAGEM P, CASA C  
WHERE C.Cnome='Stark' AND P.Cnr=C.Cnumero;
```

## SQL – PESQUISA 3

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

Para cada episódio localizado em 'Kings Landing', liste o número do episódio, o número da casa que o controla e o sobrenome, residência e data de nascimento do responsável pela casa.

## SQL – PESQUISA 3

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

Para cada episódio localizado em 'Kings Landing', liste o número do episódio, o número da casa que o controla e o sobrenome, residência e data de nascimento do responsável pela casa.

```
SELECT Episodio_numero, Cnum, Unome, Residencia, Datnasc  
FROM EPISODIO, PERSONAGEM, RESPONSAVEL  
WHERE Episodio_local='Kings Landing' AND Cnum=Casa_numero  
AND Codigo_responsavel=Codigo;
```

## SQL – PESQUISA 4

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o valor de cada personagem e todos os valores distintos.

## SQL – PESQUISA 4

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o valor de cada personagem e todos os valores distintos.

```
SELECT ALL Valor  
FROM PERSONAGEM;
```

```
SELECT DISTINCT Valor  
FROM PERSONAGEM;
```



## SQL – PESQUISA 5

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Fazer uma lista de todos os números de episódio para aqueles que envolvam um personagem cujo último nome é 'Stark', seja como participante do episódio ou como um chefe da casa que controla o episódio.

## SQL – PESQUISA 5

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Fazer uma lista de todos os números de episódio para aqueles que envolvam um personagem cujo último nome é 'Stark', seja como participante do episódio ou como um chefe da casa que controla o episódio.

```
(SELECT DISTINCT Episodio_numero
FROM PERSONAGEM, RESPONSAVEL, EPISODIO
WHERE Unome='Stark' AND Cnum=Casa_numero AND Codigo_responsavel=Codigo)
UNION
(SELECT DISTINCT Episodio_numero
FROM PERSONAGEM, EPISODIO, TRABALHA_EM
WHERE Unome='Stark' AND Pcodigo=Codigo AND Enr=Episodio_numero)
```

## SQL – PESQUISA 6

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o nome dos personagens cujo endereço esteja em um lugar que possua 'Winter' no nome.

Encontrar o nome dos personagens que nasceram durante a década de 50.

## SQL – PESQUISA 6

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o nome dos personagens cujo endereço esteja em um lugar que possua 'Winter' no nome.

```
SELECT Pnome, Unome, Residencia  
FROM PERSONAGEM  
WHERE Residencia LIKE '%Winter%';
```

Encontrar o nome dos personagens que nasceram durante a década de 50.

```
SELECT Pnome, Unome, Datanasc  
FROM PERSONAGEM  
WHERE Datanasc LIKE '__5_____';
```

## SQL – PESQUISA 7

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Mostrar os valores resultantes se cada personagem que trabalha no episódio 'Return of Dragons' aumentar seu valor em 10%.

## SQL – PESQUISA 7

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Mostrar os valores resultantes se cada personagem que trabalha no episódio 'Return of Dragons' aumentar seu valor em 10%.

```
SELECT P.Pnome, P.Unome, 1.1 * P.Valor AS Aumento_valor  
FROM PERSONAGEM P, EPISODIO E, TRABALHA_EM T  
WHERE E.Episodio_nome = 'Return of Dragons' AND  
T.Pcodigo=P.Codigo AND T.Enr=E.Episodio_numero;
```

## SQL – PESQUISA 8

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc,  
Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar todos os personagens da casa 'Stark' cujo valor esteja entre R\$20.000 e R\$70.000.

## SQL – PESQUISA 8

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar todos os personagens da casa 'Stark' cujo valor esteja entre R\$20.000 e R\$70.000.

**SELECT \***

**FROM** PERSONAGEM, CASA

**WHERE** Cnr=Cnumero **AND** Cnome='Stark' **AND** (Valor **BETWEEN** 20000 **AND** 70000);



## SQL – PESQUISA 9

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Recuperar uma lista dos personagens e dos episódios em que estão trabalhando, ordenada por casa e, dentro de cada casa, ordenada alfabeticamente pelo sobrenome, depois pelo nome.

## SQL – PESQUISA 9

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Recuperar uma lista dos personagens e dos episódios em que estão trabalhando, ordenada por casa e, dentro de cada casa, ordenada alfabeticamente pelo sobrenome, depois pelo nome.

```
SELECT C.Cnome, P.Unome, P.Pnome, E.Episodio_nome  
FROM PERSONAGEM P, CASA C, EPISODIO E, TRABALHA_EM T  
WHERE C.Cnumero=P.Cnr AND P.Codigo=T.Pcodigo AND  
T.Enr=E.Episodio_numero  
ORDER BY C.Cnome, P.Unome, P.Pnome;
```

## SQL – PESQUISA 10

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Fazer uma lista de todos os números de episódio para aqueles que envolvam um personagem cujo último nome é 'Stark', seja como participante do episódio ou como um chefe da casa que controla o episódio.

## SQL – PESQUISA 10

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Fazer uma lista de todos os números de episódio para aqueles que envolvam um personagem cujo último nome é 'Stark', seja como participante do episódio ou como um chefe da casa que controla o episódio.

**SELECT DISTINCT** Episodio\_numero

**FROM** EPISODIO

**WHERE** Episodio\_numero **IN**

(**SELECT** Episodio\_numero

**FROM** EPISODIO, RESPONSAVEL, PERSONAGEM

**WHERE** Cnum=Casa\_numero **AND** Codigo\_responsavel=Codigo **AND** Unome='Stark')

**OR** Episodio\_numero **IN**

(**SELECT** Enr

**FROM** TRABALHA\_EM, PERSONAGEM

**WHERE** Pcodigo=Codigo **AND** Unome='Stark');

## SQL – PESQUISA 11

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Retorne os nomes dos personagens cujo valor é maior do que o valor de todos os personagens na casa 3.

## SQL – PESQUISA 11

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Retorne os nomes dos personagens cujo valor é maior do que o valor de todos os personagens na casa 3.

```
SELECT Unome, Pnome
FROM PERSONAGEM
WHERE Valor > ALL
      (SELECT Valor
       FROM PERSONAGEM
       WHERE Cnr=3)
ORDER BY Unome, Pnome;
```

## SQL – PESQUISA 12

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datnasc, Parentesco)

Recuperar o nome de cada personagem que tem um dependente com o mesmo sexo do personagem.

## SQL – PESQUISA 12

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datnasc, Parentesco)

Recuperar o nome de cada personagem que tem um dependente com o mesmo sexo do personagem.

```
SELECT P.Pnome, P.Unome  
FROM PERSONAGEM P  
WHERE P.Codigo IN (SELECT D.Pcodigo  
                     FROM Dependente D  
                     WHERE P.Sexo=D.Sexo);
```



## SQL – PESQUISA 13

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datnasc, Parentesco)

Recuperar o nome de cada personagem que tem um dependente com o mesmo sexo do funcionário.

## SQL – PESQUISA 13

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datnasc, Parentesco)

Recuperar o nome de cada personagem que tem um dependente com o mesmo sexo do funcionário.

```
SELECT P.Pnome, P.Unome  
FROM PERSONAGEM P  
WHERE EXISTS (SELECT *  
                FROM DEPENDENTE D  
                WHERE P.Codigo = D.Pcodigo AND P.Sexo=D.Sexo);
```

## SQL – PESQUISA 14

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datnasc, Parentesco)

Recuperar os nomes de personagens que não possuem dependentes

# SQL – PESQUISA 14

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datanasc, Parentesco)

## Recuperar os nomes de personagens que não possuem dependentes

```
SELECT P.Pnome, P.Unome
FROM PERSONAGEM P
WHERE NOT EXISTS (SELECT *
                   FROM DEPENDENTE D
                   WHERE P.Codigo = D.Pcodigo);
```

## SQL – PESQUISA 15

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datanasc, Parentesco)

Listar os nomes dos responsáveis das casas que possuem pelo menos um dependente.

## SQL – PESQUISA 15

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datanasc, Parentesco)

Listar os nomes dos responsáveis das casas que possuem pelo menos um dependente.

**SELECT** P.Pnome, P.Unome

**FROM** PERSONAGEM P

**WHERE EXISTS** (**SELECT** \*

**FROM** DEPENDENTE D

**WHERE** P.Codigo = D.Pcodigo)

**AND EXISTS** (**SELECT** \*

**FROM** RESPONSAVEL R

**WHERE** P.Codigo = R.Codigo\_responsavel);

## SQL – PESQUISA 16

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o nome e a residência de todos os personagens da casa 'Stark'

## SQL – PESQUISA 16

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o nome e a residência de todos os personagens da casa 'Stark'

```
SELECT Pnome, Unome, Residencia  
FROM PERSONAGEM JOIN CASA  
      ON Cnr=Cnumero  
WHERE Cnome = 'Stark';
```

```
SELECT Pnome, Unome, Residencia  
FROM (PERSONAGEM NATURAL JOIN (  
      SELECT Cnome, Cnumero AS Cnr  
      FROM CASA) AS NOVA_CASA)  
WHERE Cnome = 'Stark';
```



## SQL – PESQUISA 17

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSAVEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

Para cada episódio que acontece em 'Kings Landing', liste o número do episódio, o número da casa que o controla e o nome completo, residência e data de nascimento do responsável pela casa.

## SQL – PESQUISA 17

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**RESPONSABEL** (Codigo\_responsavel, Casa\_numero, Data\_inicio)

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

Para cada episódio que acontece em 'Kings Landing', liste o número do episódio, o número da casa que o controla e o nome completo, residência e data de nascimento do responsável pela casa.

```
SELECT Episodio_numero, Cnum, Pnome, Unome, Residencia, Datnasc
FROM ((EPISODIO JOIN RESPONSABEL ON Cnum=Casa_numero)
        JOIN PERSONAGEM ON Codigo_responsavel=Codigo)
WHERE Episodio_local = 'Kings Landing';
```

## SQL – PESQUISA 18

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Achar a soma dos valores de todos os personagens da casa 'Lannister', bem como o valor máximo, o valor mínimo e a média dos valores nesta casa.

## SQL – PESQUISA 18

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Achar a soma dos valores de todos os personagens da casa 'Lannister', bem como o valor máximo, o valor mínimo e a média dos valores nesta casa.

```
SELECT SUM(Valor), MAX(Valor), MIN(Valor), AVG(Valor)  
FROM (PERSONAGEM JOIN CASA ON Cnr=Cnumero)  
WHERE Cnome = 'Lannister';
```

## SQL – PESQUISA 19

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o número total de personagens da série e o número de personagens na casa 'Stark'.

## SQL – PESQUISA 19

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Recuperar o número total de personagens da série e o número de personagens na casa 'Stark'.

```
SELECT COUNT(*)  
FROM PERSONAGEM;
```

```
SELECT COUNT(*)  
FROM PERSONAGEM, CASA  
WHERE Cnr=Cnumero AND Cnome = 'Stark';
```

## SQL – PESQUISA 20

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datnasc, Parentesco)

Recuperar os nomes de todos os personagens que tem três ou mais dependentes.

## SQL – PESQUISA 20

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

**DEPENDENTE** (Pcodigo, Nome\_dependente, Sexo, Datnasc, Parentesco)

Recuperar os nomes de todos os personagens que tem três ou mais dependentes.

```
SELECT Unome, Pnome  
FROM PERSONAGEM  
WHERE (SELECT COUNT(*)  
        FROM DEPENDENTE  
        WHERE Codigo=Pcodigo) >= 3;
```



## SQL – PESQUISA 21

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Para cada casa, recuperar o número da casa, o número de personagens desta casa e o seu valor médio

## SQL – PESQUISA 21

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Para cada casa, recuperar o número da casa, o número de personagens desta casa e o seu valor médio

```
SELECT Cnr, COUNT(*), AVG(Valor)  
FROM PERSONAGEM  
GROUP BY Cnr;
```

## SQL – PESQUISA 22

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Para cada episódio, recuperar o número do episódio, o nome do episódio e o número de personagens que trabalham nesse episódio.

## SQL – PESQUISA 22

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Para cada episódio, recuperar o número do episódio, o nome do episódio e o número de personagens que trabalham nesse episódio.

```
SELECT Episodio_numero, Episodio_nome, COUNT(*)  
FROM EPISODIO, TRABALHA_EM  
WHERE Episodio_numero=Enr  
GROUP BY Episodio_numero, Episodio_nome;
```

## SQL – PESQUISA 23

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Para cada episódio em que mais de dois personagens trabalham, recuperar o número do episódio, o nome do episódio e o número de personagens que trabalham nesse projeto.

## SQL – PESQUISA 23

**EPISODIO** (Episodio\_nome, Episodio\_numero, Episodio\_local, Cnum)

**TRABALHA\_EM** (Pcodigo, Enr)

Para cada episódio em que mais de dois personagens trabalham, recuperar o número do episódio, o nome do episódio e o número de personagens que trabalham nesse projeto.

```
SELECT Episodio_numero, Episodio_nome, COUNT(*)  
FROM EPISODIO, TRABALHA_EM  
WHERE Episodio_numero=Enr  
GROUP BY Episodio_numero, Episodio_nome  
HAVING COUNT(*) > 2;
```

## SQL – PESQUISA 24

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Para cada casa que tem mais de dois personagens, recuperar o número da casa e o número de seus personagens que estão ganhando mais de R\$50.000.

## SQL – PESQUISA 24

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Para cada casa que tem mais de dois personagens, recuperar o número da casa e o número de seus personagens que estão ganhando mais de R\$50.000.

```
SELECT Cnumero, COUNT(*)  
FROM CASA, PERSONAGEM  
WHERE Cnumero=Cnr AND Valor>50000 AND  
      Cnr IN ( SELECT Cnr  
              FROM PERSONAGEM  
              GROUP BY Cnr  
              HAVING COUNT(*) > 2)  
GROUP BY Cnumero;
```



## SQL – PESQUISA 24 (CONSULTA ERRADA)

**CASA** (Cnome, Cnumero)

**PERSONAGEM** (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo\_chefe, Cnr)

Para cada casa que tem mais de dois personagens, recuperar o número da casa e o número de seus personagens que estão ganhando mais de R\$50.000.

```
SELECT Cnumero, COUNT(*)  
FROM CASA, PERSONAGEM  
WHERE Cnumero=Cnr AND Valor>50000  
GROUP BY Cnr  
HAVING COUNT(*) > 2
```

## BIBLIOGRAFIA

- ▶ ELMASRI, Ramez; NAVATHE, Sham. **Sistemas de banco de dados**. 6. ed. São Paulo: Pearson, 2011. 788 p. [cap. 1, 2, 4 e 5]