

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

CARLA FERNANDES CURVELO
CARLAFCF@GMAIL.COM

TAD0010 – BANCO DE DADOS II

AULA 3

CONSULTAS AVANÇADAS

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)

COMPARAÇÕES ENVOLVENDO NULL

- ▶ Representa um valor que está faltando
- ▶ Situações
 - ▶ **Valor desconhecido:** a data de nascimento de uma pessoa não é conhecida
 - ▶ **Valor indisponível ou retido:** uma pessoa não deseja listar o telefone residencial
 - ▶ **Atributo não aplicável:** o atributo cônjuge não se aplica a uma pessoa que não é casada
- ▶ Utilizar **IS NULL** ou então **IS NOT NULL**

SQL – PESQUISA

PERSONAGEM (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo_chefe, Cnr)

Recuperar o primeiro nome (Pnome) de todos os personagens que não possuem chefe (Codigo_chefe)

SQL – PESQUISA

PERSONAGEM (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo_chefe, Cnr)

Recuperar o primeiro nome (Pnome) de todos os personagens que não possuem chefe (Codigo_chefe)

```
SELECT Pnome  
FROM PERSONAGEM  
WHERE Codigo_chefe IS NULL;
```

PERSONAGEM (Pnome, Unome, Codigo)

RESPONSAVEL (Codigo_responsavel, Casa_numero)

CONSULTAS ANINHADAS – IN / NOT IN

- ▶ Identifica se um elemento está presente em uma próxima lista
- ▶ Selecione as casas (Casa_numero) em que o **responsável tem o último nome (Unome) 'Stark'**

Codigo	Pnome	Unome
1	Eddard	Stark
2	Sansa	Stark
3	Arya	Stark
4	Roose	Bolton
5	Ramsey	Bolton
6	Balon	Greyjoy
7	Theon	Greyjoy
8	Cersei	Lannister

```
SELECT Codigo  
FROM PERSONAGEM  
WHERE Unome='Stark';
```

PERSONAGEM (Pnome, Unome, Codigo)

RESPONSAVEL (Codigo_responsavel, Casa_numero)

CONSULTAS ANINHADAS – IN / NOT IN

- Identifica se um elemento está presente em uma próxima lista
- Selecione as casas (Casa_numero) em que o responsável tem o último nome (Unome) 'Stark'

Codigo	Pnome	Unome
1	Eddard	Stark
2	Sansa	Stark
3	Arya	Stark
4	Roose	Bolton
5	Ramsey	Bolton
6	Balon	Greyjoy
7	Theon	Greyjoy
8	Cersei	Lannister

SELECT Codigo
FROM PERSONAGEM
WHERE Unome='Stark';

Codigo
1
2
3

PERSONAGEM (Pnome, Unome, Codigo)

RESPONSAVEL (Codigo_responsavel, Casa_numero)

CONSULTAS ANINHADAS – IN / NOT IN

- ▶ Identifica se um elemento está presente em uma próxima lista
- ▶ **Selecione as casas (Casa_numero) em que o responsável tem o último nome (Unome) 'Stark'**

```
SELECT Casa_numero  
FROM RESPONSAVEL  
WHERE Codigo_responsavel IN  
  (SELECT Codigo  
   FROM PERSONAGEM  
   WHERE Unome='Stark');
```

Codigo_Responsavel	Casa_numero
1	1
4	2
6	3
8	4
12	5
44	6

Codigo
1
2
3

PERSONAGEM (Pnome, Unome, Codigo)
RESPONSAVEL (Codigo_responsavel, Casa_numero)

CONSULTAS ANINHADAS – IN / NOT IN

- ▶ Identifica se um elemento está presente em uma próxima lista
- ▶ **Selecione as casas (Casa_numero) em que o responsável tem o último nome (Unome) 'Stark'**

```
SELECT Casa_numero
FROM RESPONSAVEL
WHERE Codigo_responsavel IN
  (SELECT Codigo
   FROM PERSONAGEM
   WHERE Unome='Stark');
```

Codigo_Responsavel		Casa_numero	
1		1	
4		2	
6		3	
8		4	
12		5	
44		6	
Codigo		Casa_numero	
1		1	
2			
3			

SQL – PESQUISA

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

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

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pcodigo	Nome_depen	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M

SQL – PESQUISA

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pcodigo	Nome_depen	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M
Pnome	Unome	
Eddard	Stark	

SQL – PESQUISA

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pcodigo	Nome_depen	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M
Pnome		Unome
Eddard		Stark

SQL – PESQUISA

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pcodigo	Nome_depen	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M
Pnome	Unome	
Eddard	Stark	
Cersei	Lannister	

PERSONAGEM (Pnome, Unome, Valor, Cnr)

SQL – PESQUISA

Retorne os nomes (Pnome, Unome) dos personagens cujo valor (Valor) é maior do que o valor de todos os personagens na casa 2 (Cnr).

PERSONAGEM (Pnome, Unome, Valor, Cnr)

SQL – PESQUISA

Retorne os nomes (Pnome, Unome) dos personagens cujo valor (Valor) é maior do que o valor de **todos os personagens na casa 2 (Cnr)**.

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

Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000
4	Cersei	Lannister	1000000

PERSONAGEM (Pnome, Unome, Valor, Cnr)

SQL – PESQUISA

Retorne os nomes (Pnome, Unome) dos personagens cujo valor (Valor) é maior do que o valor de **todos os personagens na casa 2 (Cnr)**.

```
SELECT Unome, Pnome
FROM PERSONAGEM
WHERE Valor > ALL
      (SELECT Valor
       FROM PERSONAGEM
       WHERE Cnr=2)
ORDER BY Unome, Pnome;
```

Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000
4	Cersei	Lannister	1000000

PERSONAGEM (Pnome, Unome, Valor, Cnr)

SQL – PESQUISA

Retorne os nomes (Pnome, Unome) dos **personagens cujo valor (Valor) é maior do que o valor de todos os personagens na casa 2 (Cnr).**

```
SELECT Unome, Pnome
FROM PERSONAGEM
WHERE Valor > ALL
      (SELECT Valor
       FROM PERSONAGEM
        WHERE Cnr=2)
ORDER BY Unome, Pnome;
```

Valor			
30000			
60000			
Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000
4	Cersei	Lannister	1000000

PERSONAGEM (Pnome, Unome, Valor, Cnr)

SQL – PESQUISA

Retorne os nomes (Pnome, Unome) dos **personagens cujo valor (Valor) é maior do que o valor de todos os personagens na casa 2 (Cnr).**

```
SELECT Unome, Pnome
FROM PERSONAGEM
WHERE Valor > ALL
      (SELECT Valor
       FROM PERSONAGEM
       WHERE Cnr=2)
ORDER BY Unome, Pnome;
```

Valor			
30000			
60000			
Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000
4	Cersei	Lannister	1000000

PERSONAGEM (Pnome, Unome, Valor, Cnr)
SQL – PESQUISA

Retorne os nomes (Pnome, Unome) dos personagens cujo valor (Valor) é maior do que o valor de todos os personagens na casa 2 (Cnr).

```
SELECT Unome, Pnome
FROM PERSONAGEM
WHERE Valor > ALL
      (SELECT Valor
       FROM PERSONAGEM
       WHERE Cnr=2)
ORDER BY Unome, Pnome;
```

Valor			
30000			
60000			
Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000
4	Cersei	Lannister	1000000
Pnome		Unome	
Arya		Stark	
Cersei		Lannister	

AS FUNÇÕES EXISTS E NOT EXISTS

- ▶ Verificar se o resultado de uma consulta aninhada é vazio (não contem tuplas) ou não

Resultado	EXISTS	NOT EXISTS
Pelo menos uma tupla	TRUE	FALSE
Nenhuma tupla	FALSE	TRUE

SQL – PESQUISA

PERSONAGEM (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo_chefe, Cnr)

DEPENDENTE (Pcodigo, Nome_dependente, Sexo, Datnasc, Parentesco)

Recuperar o nome (Pnome, Unome) de cada personagem que tem um dependente com o mesmo sexo do funcionário.

SQL – PESQUISA

PERSONAGEM (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo_chefe, Cnr)

DEPENDENTE (Pcodigo, Nome_dependente, Sexo, Datnasc, Parentesco)

Recuperar o nome (Pnome, Unome) 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

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pcodigo	Nome_dependente	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M

SQL – PESQUISA

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pcodigo	Nome_dependente	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M
Pnome		Unome
Eddard		Stark

SQL – PESQUISA

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pcodigo	Nome_dependente	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M
Pnome	Unome	
Eddard	Stark	

SQL – PESQUISA

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pcodigo	Nome_dependente	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M
Pnome	Unome	
Eddard	Stark	
Cersei	Lannister	

SQL – PESQUISA

PERSONAGEM (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo_chefe, Cnr)

DEPENDENTE (Pcodigo, Nome_dependente, Sexo, Datnasc, Parentesco)

Recuperar os nomes (Pnome, Unome) de personagens que não possuem dependentes

* Na tabela DEPENDENTE não existe nenhuma entrada com este personagens (Pcodigo)

SQL - PESQUISA

PERSONAGEM (Pnome, Unome, Codigo, Datanasc, Residencia, Sexo, Valor, Codigo_chefe, Cnr)

DEPENDENTE (Pcodigo, Nome_dependente, Sexo, Datanaasc, Parentesco)

Recuperar os nomes (Pnome, Unome) 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

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pcodigo	Nome_dependente	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M

SQL - PESQUISA

SELECT P.Pnome, P.Unome

FROM PERSONAGEM P

```
WHERE NOT EXISTS (SELECT *  

FROM DEPENDENTE D  

WHERE P.Codigo = D.Pcodigo);
```

1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M

1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommom	M

SQL - PESQUISA

SELECT P.Pnome, P.Unome

FROM PERSONAGEM P

```
WHERE NOT EXISTS (SELECT *  

FROM DEPENDENTE D  

WHERE P.Codigo = D.Pcodigo);
```

1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M

Sansa	Stark

1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommom	M

SQL – PESQUISA

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pnome	Unome
Sansa	Stark
Arya	Stark

Pcodigo	Nome_dependente	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M

SQL – PESQUISA

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pnome	Unome
Sansa	Stark
Arya	Stark
Roose	Bolton
Ramsay	Bolton
Balon	Greyjoy
Theon	Greyjoy

Pcodigo	Nome_dependente	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M

SQL – PESQUISA

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

Codigo	Pnome	Unome	Sexo
1	Eddard	Stark	M
2	Sansa	Stark	F
3	Arya	Stark	F
4	Roose	Bolton	M
5	Ramsey	Bolton	M
6	Balon	Greyjoy	M
7	Theon	Greyjoy	M
8	Cersei	Lannister	F

Pnome	Unome
Sansa	Stark
Arya	Stark
Roose	Bolton
Ramsay	Bolton
Balon	Greyjoy
Theon	Greyjoy

Pcodigo	Nome_dependente	Sexo
1	Catelyn	F
1	Bran	M
1	Rickon	M
8	Myrcela	F
8	Tommem	M

SQL – PESQUISA

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 (Pnome, Unome) dos responsáveis das casas que possuem pelo menos um dependente.

SQL – PESQUISA

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 (Pnome, Unome) 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

```
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);
```

Codigo	Pnome	Unome
1	Eddard	Stark
2	Sansa	Stark
3	Arya	Stark
4	Roose	Bolton
5	Ramsey	Bolton
6	Balon	Greyjoy
7	Theon	Greyjoy
8	Cersei	Lannister

Casa_numero	Codigo_responsavel
1	1
2	4
3	6
4	8
5	12
6	44

Pcodigo	Nome_dependente
1	Catelyn
1	Bran
1	Rickon
8	Myrcela
8	Tommem

SQL – PESQUISA

```
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);
```

Codigo	Pnome	Unome
1	Eddard	Stark
2	Sansa	Stark
3	Arya	Stark
4	Roose	Bolton
5	Ramsey	Bolton
6	Balon	Greyjoy
7	Theon	Greyjoy
8	Cersei	Lannister

Casa_numero	Codigo_responsavel
1	1
2	4
3	6
4	8
5	12
6	44

Pcodigo	Nome_dependente
1	Catelyn
1	Bran
1	Rickon
8	Myrcela
8	Tommem

Pnome	Unome
Eddard	Stark

SQL – PESQUISA

```
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);
```

Codigo	Pnome	Unome
1	Eddard	Stark
2	Sansa	Stark
3	Arya	Stark
4	Roose	Bolton
5	Ramsey	Bolton
6	Balon	Greyjoy
7	Theon	Greyjoy
8	Cersei	Lannister

Casa_numero	Codigo_responsavel
1	1
2	4
3	6
4	8
5	12
6	44

Pcodigo	Nome_dependente
1	Catelyn
1	Bran
1	Rickon
8	Myrcela
8	Tommem

Pnome	Unome
Eddard	Stark

SQL – PESQUISA

```
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);
```

Codigo	Pnome	Unome
1	Eddard	Stark
2	Sansa	Stark
3	Arya	Stark
4	Roose	Bolton
5	Ramsey	Bolton
6	Balon	Greyjoy
7	Theon	Greyjoy
8	Cersei	Lannister

Casa_numero	Codigo_responsavel
1	1
2	4
3	6
4	8
5	12
6	44

Pcodigo	Nome_dependente
1	Catelyn
1	Bran
1	Rickon
8	Myrcela
8	Tommem

Pnome	Unome
Eddard	Stark

SQL – PESQUISA

```
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);
```

Codigo	Pnome	Unome
1	Eddard	Stark
2	Sansa	Stark
3	Arya	Stark
4	Roose	Bolton
5	Ramsey	Bolton
6	Balon	Greyjoy
7	Theon	Greyjoy
8	Cersei	Lannister

Casa_numero	Codigo_responsavel
1	1
2	4
3	6
4	8
5	12
6	44

Pcodigo	Nome_dependente
1	Catelyn
1	Bran
1	Rickon
8	Myrcela
8	Tommem

Pnome	Unome
Eddard	Stark
Cersei	Lannister

SQL – PESQUISA

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

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

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

Codigo	Pnome	Unome
1	Eddard	Stark
2	Sansa	Stark
3	Arya	Stark
4	Roose	Bolton
5	Ramsey	Bolton
6	Balon	Greyjoy
7	Theon	Greyjoy
8	Cersei	Lannister

Count
8

SQL – PESQUISA

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

Codigo	Pnome	Unome	Cnr
1	Eddard	Stark	1
2	Sansa	Stark	1
3	Arya	Stark	1
4	Roose	Bolton	2
5	Ramsey	Bolton	2
6	Balon	Greyjoy	3
7	Theon	Greyjoy	3
8	Cersei	Lannister	4

Cnumero	Cnome
1	Stark
2	Bolton
3	Greyjoy
4	Lannister
5	Targaryen
6	Baratheon

SQL – PESQUISA

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

Codigo	Pnome	Unome	Cnr	Cnumero	Cnome
1	Eddard	Stark	1	1	Stark
1	Eddard	Stark	1	2	Bolton
1	Eddard	Stark	1	3	Greyjoy
1	Eddard	Stark	1	4	Lannister
1	Eddard	Stark	1	5	Targaryen
1	Eddard	Stark	1	6	Baratheon
2	Sansa	Stark	1	1	Stark
2	Sansa	Stark	1	2	Bolton

SQL – PESQUISA

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

Codigo	Pnome	Unome	Cnr	Cnumero	Cnome
1	Eddard	Stark	1	1	Stark
1	Eddard	Stark	1	2	Bolton
1	Eddard	Stark	1	3	Greyjoy
1	Eddard	Stark	1	4	Lannister
1	Eddard	Stark	1	5	Targaryen
1	Eddard	Stark	1	6	Baratheon
2	Sansa	Stark	1	1	Stark
2	Sansa	Stark	1	2	Bolton

SQL – PESQUISA

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

Codigo	Pnome	Unome	Cnr	Cnumero	Cnome
1	Eddard	Stark	1	1	Stark
2	Sansa	Stark	1	1	Stark
3	Arya	Stark	1	1	Stark
4	Roose	Bolton	2	2	Bolton
5	Ramsey	Bolton	2	2	Bolton
6	Balon	Greyjoy	3	3	Greyjoy
7	Theon	Greyjoy	3	3	Greyjoy
8	Cersei	Lannister	4	4	Lannister

Count

3

SQL – PESQUISA

PERSONAGEM (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo_chefe, Cnr)

Para cada casa, recuperar o número da casa (Cnr), o número de personagens desta casa e o seu valor médio (Valor)

SQL – PESQUISA

PERSONAGEM (Pnome, Unome, Codigo, Datnasc, Residencia, Sexo, Valor, Codigo_chefe, Cnr)

Para cada casa, recuperar o número da casa (Cnr), o número de personagens desta casa e o seu valor médio (Valor)

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

SQL – PESQUISA

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

Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000
4	Cersei	Lannister	1000000

Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000

Cnr	Pnome	Unome	Valor
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000

Cnr	Pnome	Unome	Valor
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000

Cnr	Pnome	Unome	Valor
4	Cersei	Lannister	1000000

SQL – PESQUISA

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

Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000
4	Cersei	Lannister	1000000

Cnr	Count	AVG
1	3	53333
2	2	45000
3	2	4000
4	1	1000000

Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000

Cnr	Pnome	Unome	Valor
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000

Cnr	Pnome	Unome	Valor
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000

Cnr	Pnome	Unome	Valor
4	Cersei	Lannister	1000000

SQL – PESQUISA

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 (Cnumero ou Cnr) e o número de seus personagens que estão ganhando mais de R\$50.000 (Valor).

SQL – PESQUISA

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 (Cnumero ou Cnr) e o número de seus personagens que estão ganhando mais de R\$50.000 (Valor).

```
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

```
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;			
Codigo	Pnome	Unome	Cnr
1	Eddard	Stark	1
2	Sansa	Stark	1
3	Arya	Stark	1
4	Roose	Bolton	2
5	Ramsey	Bolton	2
6	Balon	Greyjoy	3
7	Theon	Greyjoy	3
8	Cersei	Lannister	4

Cnr	Count
1	3
2	2
3	2
4	1

Cnr	Pnome	Unome	Valor
1	Eddard	Stark	50000
1	Sansa	Stark	10000
1	Arya	Stark	100000

Cnr	Pnome	Unome	Valor
2	Roose	Bolton	30000
2	Ramsey	Bolton	60000

Cnr	Pnome	Unome	Valor
3	Balon	Greyjoy	5000
3	Theon	Greyjoy	3000

Cnr	Pnome	Unome	Valor
4	Cersei	Lannister	1000000

SQL – PESQUISA

```
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;
```

Codigo	Pnome	Unome	Cnr	Cnumero	Cnome	Valor
1	Eddard	Stark	1	1	Stark	50000
2	Sansa	Stark	1	1	Stark	10000
3	Arya	Stark	1	1	Stark	100000
4	Roose	Bolton	2	2	Bolton	30000
5	Ramsey	Bolton	2	2	Bolton	60000
6	Balon	Greyjoy	3	3	Greyjoy	5000
7	Theon	Greyjoy	3	3	Greyjoy	3000
8	Cersei	Lannister	4	4	Lannister	1000000

SQL – PESQUISA

```
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;
```

Cnr	Count
1	3

Codigo	Pnome	Unome	Cnr	Cnumero	Cnome	Valor
1	Eddard	Stark	1	1	Stark	50000
2	Sansa	Stark	1	1	Stark	10000
3	Arya	Stark	1	1	Stark	100000
4	Roose	Bolton	2	2	Bolton	30000
5	Ramsey	Bolton	2	2	Bolton	60000
6	Balon	Greyjoy	3	3	Greyjoy	5000
7	Theon	Greyjoy	3	3	Greyjoy	3000
8	Cersei	Lannister	4	4	Lannister	1000000

SQL – PESQUISA

```
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;
```

Cnr	Count
1	3

Codigo	Pnome	Unome	Cnr	Cnumero	Cnome	Valor
1	Eddard	Stark	1	1	Stark	50000
2	Sansa	Stark	1	1	Stark	10000
3	Arya	Stark	1	1	Stark	100000
4	Roose	Bolton	2	2	Bolton	30000
5	Ramsey	Bolton	2	2	Bolton	60000
6	Balon	Greyjoy	3	3	Greyjoy	5000
7	Theon	Greyjoy	3	3	Greyjoy	3000
8	Cersei	Lannister	4	4	Lannister	1000000

Cnumero	Count
1	1

RESUMO DAS CONSULTAS EM SQL

SELECT <lista atributo e função>

FROM <lista tabela>

[**WHERE** <condição>]

[**GROUP BY** <atributo(s) de agrupamento>]

[**HAVING** <condição de grupo>]

[**ORDER BY** <lista atributos>];

RESTRIÇÕES GERAIS

- ▶ Foge das restrições embutidas no modelo relacional
 - ▶ Chaves primárias e únicas, integridade de entidade e integridade referencial
- ▶ Comando: **CREATE ASSERTION**
- ▶ É igual ao **CHECK**?

```
CREATE TABLE Persons (  
    ID int NOT NULL,  
    Nome Varchar(255) NOT NULL,  
    Idade int,  
    CHECK (Idade >= 18)  
);
```

RESTRIÇÕES GERAIS

Exemplo: Restrição de que o salário de um funcionário não pode ser maior do que o salário do gerente do departamento para o qual o funcionário trabalha

FUNCIONARIO (Nome, Cpf, Salario, Numero_depto)

DEPARTAMENTO (Nome, Numero, Cpf_gerente)

```
CREATE ASSERTION RESTRICAO_SALARIAL  
CHECK ( NOT EXISTS  
    ( SELECT *  
      FROM FUNCIONARIO F,  
            FUNCIONARIO G,  
            DEPARTAMENTO D  
      WHERE F.Salario > G.Salario AND  
            F.Numero_depto = D.Numero AND  
            D.Cpf_gerente = G.Cpf ) ) ;
```

BIBLIOGRAFIA

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