



DISCIPLINA: Banco de Dados 1

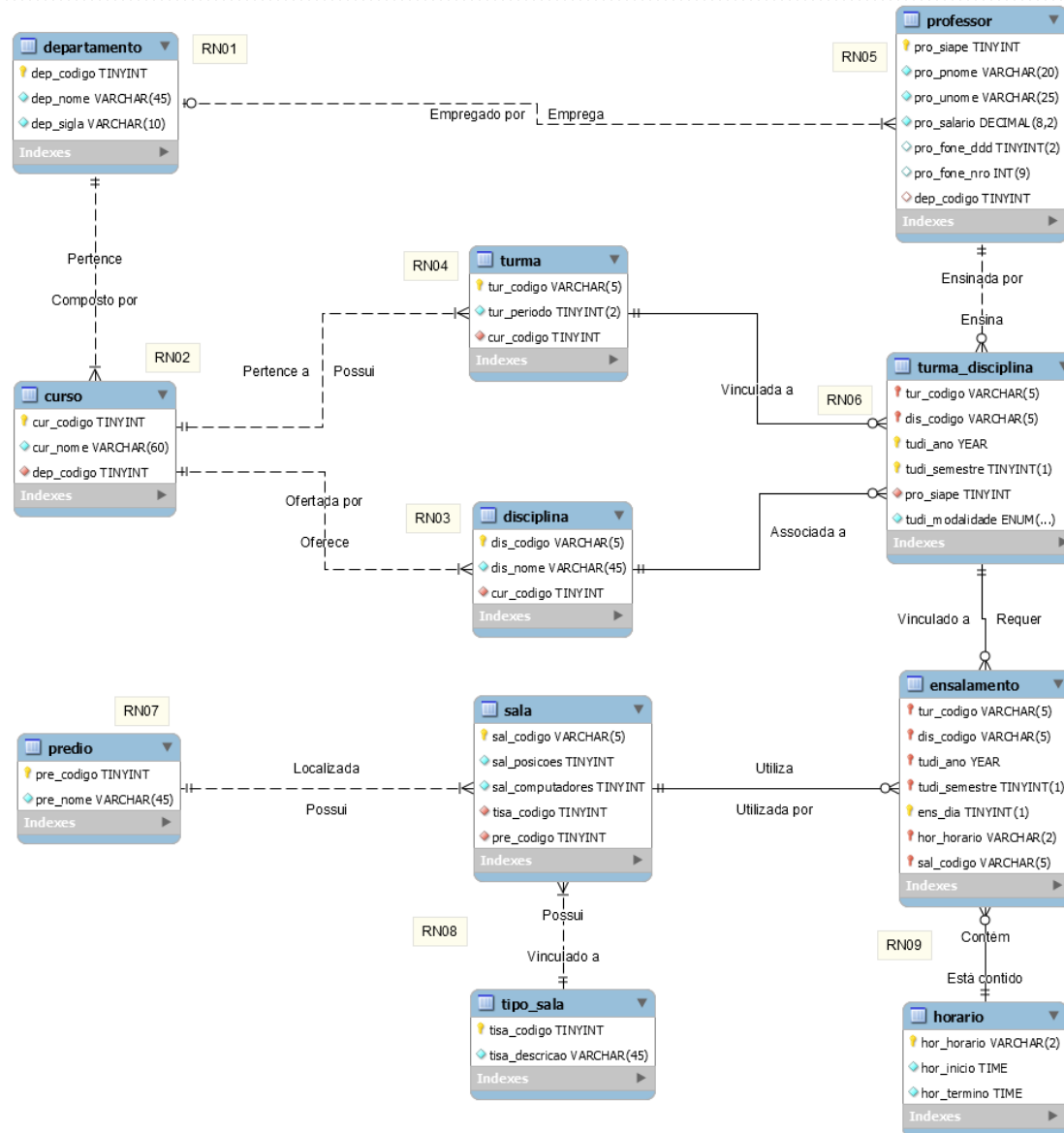
Prof. **GIOVANI** Volnei Meinerz

Aula 21 – SQL (cont.)

Objetivos da Aula

- Entender a operação de junção relacional interna e externa
- Como utilizar os operadores de junção em SQL
- Como utilizar algumas funções SQL que permitem manipular valores numéricos, de *string* e de data

Cenário



Junção Relacional

Operação que recupera e funde linhas de duas ou mais tabelas com base no tipo de junção empregada (interna ou externa) e na condição definida pela junção

Classificações das Junções

→ Junção Interna

- Retorna apenas linhas que atendam a determinados critérios
 - Condição de igualdade (junção natural ou comparação de igualdade)
 - Condição de desigualdade (utilizando operadores de comparação)

→ Junção Externa

- Retorna não apenas linhas correspondentes, mas também as linhas com valores de colunas sem correspondência em...
 - uma tabela
 - **left outer join** – junção externa à esquerda
 - **right outer join** – junção externa à esquerda
 - ou ambas as tabelas a serem unidas
 - **full outer join** – junção externa completa

Expressões de Junção em SQL

Classificação	Tipo	Sintaxe	Descrição
INNER	join	<code>select * from T1, T2 where T1.C1 = T2.C1</code>	Retorna apenas as linhas que atendam à condição de junção na cláusula where (manual, estilo “antigo”)
	natural join	<code>select * from T1 natural join T2</code>	Retorna apenas as linhas com valores coincidentes nos atributos comuns (que devem ter mesmo nome e tipo)
	join using	<code>select * from T1 join T2 using(C1)</code>	Retorna apenas as linhas com valores coincidentes no atributo indicado na cláusula using
	join on	<code>select * from T1 join T2 on T1.C1 = T2.C1</code>	Retorna apenas as linhas que atendam à condição de junção na cláusula on

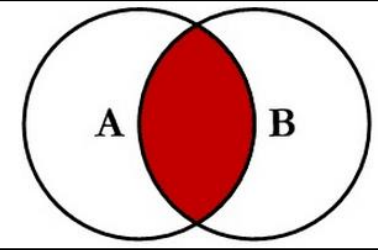
Expressões de Junção em SQL (cont.)

Classificação	Tipo	Sintaxe	Descrição
OUTER	left join	<code>select * from T1 left outer join T2 on T1.C1 = T2.C1</code>	Retorna todas as linhas da tabela a esquerda (T1), mesmo não havendo linhas com valores coincidentes na tabela da direita (T2)
	right join	<code>select * from T1 right outer join T2 on T1.C1 = T2.C1</code>	Retorna todas as linhas da tabela a direita (T2), mesmo não havendo linhas com valores coincidentes na tabela da esquerda (T1)
	full join	<code>select * from T1 full outer join T2 on T1.C1 = T2.C1</code>	Retorna todas as linhas de ambas as tabelas (T1 e T2), mesmo havendo linhas sem valores coincidentes

Expressões de Junção em SQL (cont.)

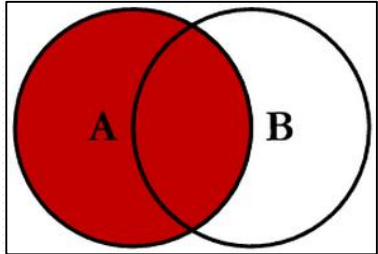
INNER JOIN

Retorna apenas as linhas com valores coincidentes nos atributos comuns nas tabelas A e B



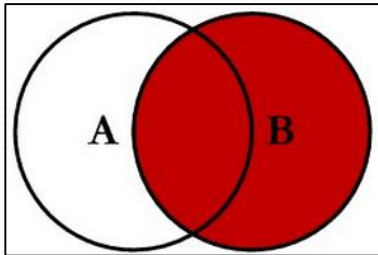
LEFT OUTER JOIN

Retorna todas as linhas da tabela A (a esquerda), mesmo não havendo linhas correspondentes, ou seja, não havendo valores coincidentes nos atributos comuns na tabela B (da direita)



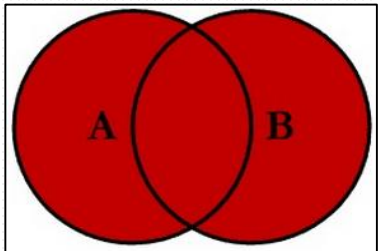
RIGHT OUTER JOIN

Retorna todas as linhas da tabela B (a direita), mesmo não havendo linhas correspondentes, ou seja, não havendo valores coincidentes nos atributos comuns na tabela A (da esquerda)



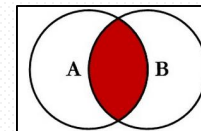
FULL OUTER JOIN

Retorna todas as linhas de ambas as tabelas (A e B), mesmo não havendo linhas correspondentes, ou seja, não havendo valores coincidentes nos atributos comuns de ambas as tabelas



INNER JOIN

Retorna apenas as linhas com valores correspondentes nas tabelas A e B

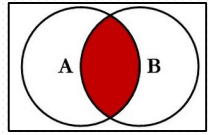


```
1  -- JUNÇÃO MANUAL (estilo antigo)
2  • SELECT *
3  FROM professor p, departamento d
4  WHERE p.dep_codigo = d.dep_codigo;
5  -- JUNÇÃO NATURAL
6  • SELECT *
7  FROM professor NATURAL JOIN departamento;
8  -- JOIN ... USING
9  • SELECT *
10 FROM professor JOIN departamento
11 USING (dep_codigo);
12 -- JOIN ... ON
13 • SELECT *
14 FROM professor p JOIN departamento d
15 ON p.dep_codigo = d.dep_codigo;
```

pro_siape	pro_pnome	pro_unome	pro_salario	pro_fone_ddd	pro_fone_nro	dep_codigo	dep_codigo	dep_nome	dep_sigla
1	Hanka	Ruebbert	60000.00	43	35239000	6	6	Departamento Acadêmico de Mecânica	DAMEC
2	Tilo	Gerhold	65000.00	NULL	NULL	2	2	Departamento Acadêmico de Elétrica	DAELE
3	Ekkehart	Schubbert	55000.00	11	754210000	3	3	Departamento Acadêmico de Matemática	DAMAT
4	Gerhard	Huettio	85000.00	14	33448888	2	2	Departamento Acadêmico de Elétrica	DAELE
5	Anoela	Lehmann	75000.00	43	35237777	2	2	Departamento Acadêmico de Elétrica	DAELE
6	Lisa	Reimann	95000.00	NULL	NULL	1	1	Departamento Acadêmico de Computação	DACOM
7	Corinna	Enoellmann	130000.00	55	995554500	1	1	Departamento Acadêmico de Computação	DACOM
8	Manfred	Schubbert	79000.00	43	998456587	2	2	Departamento Acadêmico de Elétrica	DAELE
9	Lena	Reimann	145000.00	14	997465544	3	3	Departamento Acadêmico de Matemática	DAMAT
10	Giovani	Meinerz	84000.00	55	999838457	1	1	Departamento Acadêmico de Computação	DACOM

INNER JOIN (cont.)

✈ NATURAL JOIN

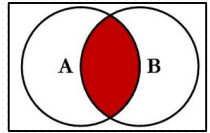


```
1  -- Encontre as turmas/disciplinas atribuídas aos professores do DACOM
2  • SELECT
3      tur_codigo, dis_codigo, pro_pnome, pro_unome, tudi_ano, tudi_semestre
4  FROM
5      departamento
6      NATURAL JOIN
7      professor
8      NATURAL JOIN
9      turma_disciplina
10 WHERE
11     dep_sigla LIKE 'DACOM';
```

tur_codigo	dis_codigo	pro_pnome	pro_unome	tudi_ano	tudi_semestre
C41	EC34D	Lisa	Reimann	2017	2
N12B	AN32C	Corinna	Enoellmann	2017	2
N12B	AN32C	Corinna	Enoellmann	2018	1

INNER JOIN (cont.)

→ NATURAL JOIN (cont.)

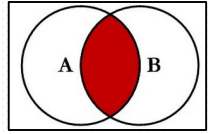


```
13 -- Encontre os dias e horários das turmas/disciplinas
14 -- atribuídas aos professores do DACOM, efetivamente ensaladas
15 • SELECT
16     tur_codigo,
17     dis_codigo,
18     pro_pnome,
19     pro_unome,
20     tudi_ano,
21     tudi_semestre,
22     ens_dia,
23     hor_horario
24 FROM
25     departamento
26     NATURAL JOIN
27     professor
28     NATURAL JOIN
29     turma_disciplina
30     NATURAL JOIN
31     ensalamento
32     NATURAL JOIN
33     horario
34 WHERE
35     dep_sigla LIKE 'DACOM';
```

ult Grid Filter Rows: <input type="text"/> Export: Wrap Cell Content:							
tur_codigo	dis_codigo	pro_pnome	pro_unome	tudi_ano	tudi_semestre	ens_dia	hor_horario
C41	EC34D	Lisa	Reimann	2017	2	3	T4
C41	EC34D	Lisa	Reimann	2017	2	3	T5

INNER JOIN (cont.)

➤ *NATURAL JOIN* com *JOIN..USING()*



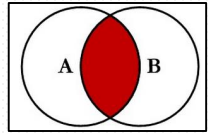
```
1  -- NATURAL JOIN com JOIN..USING()
2  -- Prédios, com suas respectivas salas que estes possuem, bem como tipo
3  -- da mesma, para todas as salas com quantidade de posições superior a 30
4  • SELECT
5      pre_nome, sal_codigo, tisa_descricao
6  FROM
7      predio
8      NATURAL JOIN
9      sala JOIN tipo_sala USING (tisa_codigo)
10 WHERE
11     sal_posicoes > 30;
```

Result Grid | | Filter Rows: | Export: | Wrap Cell Content:

pre_nome	sal_codigo	tisa_descricao
Bloco A	A040	Laboratório
Bloco A	A140	Teórica
Bloco A	A146	Teórica
Bloco I	I201	Laboratório
Bloco I	I202	Laboratório
Bloco I	I205	Laboratório
Bloco P	P005	Laboratório
Bloco P	P101	Teórica
Bloco P	P105	Laboratório
Bloco P	P205	Laboratório

INNER JOIN (cont.)

➤ *NATURAL JOIN / JOIN..USING() / JOIN..ON*

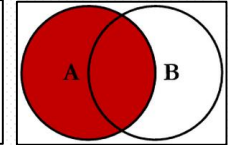


```
49 -- NATURAL JOIN / JOIN..USING() / JOIN..ON
50 -- Ensalamientos que utilizam salas, cuja quantidade de posições superior a 30
51 -- Mostrar turma, disciplina, ano, semestre, dia, horário, sala, qtde de posições,
52 -- tipo da sala e prédio
53 • SELECT
54     tur_codigo Turma, dis_codigo Disciplina, tudi_ano Ano,
55     tudi_semestre Semestre, ens_dia Dia, hor_horario Horário,
56     sal_codigo Sala, sal_posicoes Posições, tisa_descricao Tipo, pre_nome Prédio
57 FROM
58     ensalamento
59     NATURAL JOIN
60     sala s
61     JOIN
62     tipo_sala USING (tisa_codigo)
63     JOIN
64     predio p ON s.pre_codigo = p.pre_codigo
65 WHERE
66     sal_posicoes > 30;
```

Turma	Disciplina	Ano	Semestre	Dia	Horário	Sala	Posições	Tipo	Prédio
ES21	IF62H	2017	1	3	T4	A040	45	Laboratório	Bloco A
C41	EC34D	2017	2	3	T4	A040	45	Laboratório	Bloco A
C41	EC34D	2017	2	3	T5	A040	45	Laboratório	Bloco A
ES21	IF62H	2017	2	2	N1	I205	45	Laboratório	Bloco I
N12A	AN32C	2017	2	4	N2	I205	45	Laboratório	Bloco I
N12A	AN32C	2017	2	4	N3	I205	45	Laboratório	Bloco I
N12A	AN32C	2017	2	2	N4	I205	45	Laboratório	Bloco I
N12A	AN32C	2017	2	2	N5	I205	45	Laboratório	Bloco I
M31	EM33H	2017	2	5	M4	P105	45	Laboratório	Bloco P
M31	EM33H	2017	2	5	M5	P105	45	Laboratório	Bloco P

LEFT OUTER JOIN

Retorna todas as linhas da tabela A (a esquerda), mesmo não havendo linhas correspondentes na tabela B (da direita)



```
11 -- Selecione todos os professores e as turmas/disciplinas a eles atribuídas, mesmo que em relação
12 -- a alguns não tenha sido atribuída nenhuma turma/disciplina. No resultado, mostre o nome do
13 -- professor e os códigos das turmas/disciplinas a ele atribuídas, bem como ano e semestre.
14 -- Tudo deve estar ordenado ascendentemente pelo código da turma.
15 -- Utilize o conceito de junção relacional externa, empregando o comando de junção externa à esquerda.
16 • SELECT
17     pro_pnome, pro_unome, td.tur_codigo, td.dis_codigo, td.tudi_ano, td.tudi_semestre
18 FROM
19     professor p
20     LEFT OUTER JOIN
21     turma_disciplina td ON p.pro_siape = td.pro_siape
22 ORDER BY tur_codigo;
```

Table: professor

Columns:

<u>pro_siape</u>	tinyint(4) PK
pro_pnome	varchar(20)
pro_unome	varchar(25)
pro_salario	decimal(8,2)
pro_fone_ddd	tinyint(2)
pro_fone_nro	int(9)
dep_codigo	tinyint(4)

Table: turma_disciplina

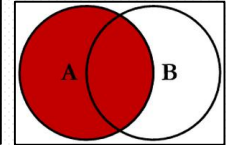
Columns:

<u>tur_codigo</u>	varchar(5) PK
<u>dis_codigo</u>	varchar(5) PK
<u>tudi_ano</u>	year(4) PK
<u>tudi_semestre</u>	tinyint(1) PK
pro_siape	tinyint(4)
tudi_modalidade	enum(Presenc

pro_pnome	pro_unome	tur_codigo	dis_codigo	tudi_ano	tudi_semestre
Giovani	Meinerz	NULL	NULL	NULL	NULL
Lena	Reimann	NULL	NULL	NULL	NULL
Ekkehart	Schubbert	NULL	NULL	NULL	NULL
Manfred	Schubbert	NULL	NULL	NULL	NULL
Tilo	Gerhold	NULL	NULL	NULL	NULL
Lisa	Reimann	C41	EC34D	2017	2
Cenair	Maicá	C51	EC35B	2017	2
Mano	Lima	C51	EC35B	2017	1
Gerhard	Huettio	E21	MA35B	2017	2
Luiz	Marengo	ES21	IF62H	2017	2
Luiz	Marengo	ES21	IF62H	2017	1
Anoela	Lehmann	ES31	IF63C	2016	1
Hanka	Ruebbert	M31	EM33H	2017	2
Luiz	Marengo	N12A	AN32C	2017	2
Corinna	Enoellmann	N12B	AN32C	2018	1
Corinna	Enoellmann	N12B	AN32C	2017	2
Luiz	Marengo	N12SP	AN32C	2017	2

LEFT OUTER JOIN (cont.)

Retorna todas as linhas da tabela A (a esquerda), mesmo não havendo linhas correspondentes na tabela B (da direita)



```

1 -- Selecione todas as turmas/disciplinas que foram criadas, mesmo que algumas não tenham sido ensaladas.
2 -- No resultado, mostre os códigos das turmas/disciplinas, ano e semestre, bem como dia, horário e sala.
3 -- Utilize o conceito de junção relacional externa, empregando o comando de junção externa à esquerda.
4 • SELECT
5     td.tur_codigo, td.dis_codigo, td.tudi_ano, td.tudi_semestre, e.ens_dia, e.hor_horario, e.sal_codigo
6 FROM
7     turma_disciplina td
8     LEFT OUTER JOIN
9     ensalamento e ON td.tur_codigo = e.tur_codigo;
    
```

Table: **turma_disciplina**

Columns:

<u>tur_codigo</u>	varchar(5) PK
<u>dis_codigo</u>	varchar(5) PK
<u>tudi_ano</u>	year(4) PK
<u>tudi_semestre</u>	tinyint(1) PK
<u>pro_siape</u>	tinyint(4)
tudi_modalidade	enum('Presenc

Table: **ensalamento**

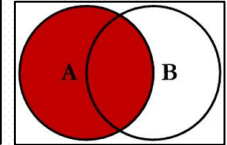
Columns:

<u>tur_codigo</u>	varchar(5) PK
<u>dis_codigo</u>	varchar(5) PK
<u>tudi_ano</u>	year(4) PK
<u>tudi_semestre</u>	tinyint(1) PK
<u>ens_dia</u>	tinyint(1) PK
<u>hor_horario</u>	varchar(2) PK
<u>sal_codigo</u>	varchar(5) PK

tur_codigo	dis_codigo	tudi_ano	tudi_semestre	ens_dia	hor_horario	sal_codigo
M31	EM33H	2017	2	5	M4	P105
M31	EM33H	2017	2	5	M5	P105
E21	MA35B	2017	2	6	T2	K009
E21	MA35B	2017	2	6	T3	K009
ES31	IF63C	2016	1	NULL	NULL	NULL
C41	EC34D	2017	2	3	T4	A040
C41	EC34D	2017	2	3	T5	A040
N12B	AN32C	2017	2	NULL	NULL	NULL
N12B	AN32C	2018	1	NULL	NULL	NULL
ES21	IF62H	2017	1	3	T4	A040
ES21	IF62H	2017	1	2	N1	I205
ES21	IF62H	2017	2	3	T4	A040
ES21	IF62H	2017	2	2	N1	I205
N12A	AN32C	2017	2	4	N2	I205
N12A	AN32C	2017	2	4	N3	I205
N12A	AN32C	2017	2	2	N4	I205
N12A	AN32C	2017	2	2	N5	I205
N12SP	AN32C	2017	2	NULL	NULL	NULL
C51	EC35B	2017	1	NULL	NULL	NULL
C51	EC35B	2017	2	NULL	NULL	NULL

LEFT OUTER JOIN (cont.)

Retorna todas as linhas da tabela A (a esquerda), mesmo não havendo linhas correspondentes na tabela B (da direita)



```
50 -- Selecione todos os departamentos, independente de ter, ou não, professores
51 -- vinculados a eles e, selecione também, as turmas/disciplinas atribuídas
52 -- a esses professores, mesmo que a alguns não tenham sido atribuídas turmas/disciplinas.
53 -- No resultado, mostre a sigla do departamento, o siape e nome do professor, os códigos
54 -- das turmas/disciplinas, ano e semestre. Utilize o conceito de junção relacional externa,
55 -- empregando a combinação de comandos de junção externa à esquerda.
56 • SELECT
57     dep_sigla, p.pro_siape, pro_pnome, pro_unome, tur_codigo, dis_codigo, tudi_ano, tudi_semestre
58 FROM
59     departamento d
60     LEFT OUTER JOIN
61     professor p ON d.dep_codigo = p.dep_codigo
62     LEFT OUTER JOIN
63     turma_disciplina td ON p.pro_siape = td.pro_siape;
```

Table: departamento

Columns:

<u>dep_codigo</u>	tinyint(4) PK
dep_nome	varchar(45)
dep_sigla	varchar(10)

Table: professor

Columns:

<u>pro_siape</u>	tinyint(4) PK
pro_pnome	varchar(20)
pro_unome	varchar(25)
pro_salario	decimal(8,2)
pro_fone_ddd	tinyint(2)
pro_fone_nro	int(9)
<u>dep_codigo</u>	tinyint(4)

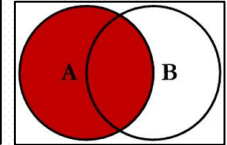
Table: turma_disciplina

Columns:

<u>tur_codigo</u>	varchar(5) PK
<u>dis_codigo</u>	varchar(5) PK
<u>tudi_ano</u>	year(4) PK
<u>tudi_semestre</u>	tinyint(1) PK
<u>pro_siape</u>	tinyint(4)
tudi_modalidade	enum('Presenc

LEFT OUTER JOIN (cont.)

Retorna todas as linhas da tabela A (a esquerda), mesmo não havendo linhas correspondentes na tabela B (da direita)

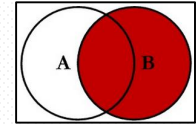


```
-- Selecione todos os departamentos, independente de ter, ou não, professores
-- vinculados a eles e, selecione também, as turmas/disciplinas atribuídas
-- a esses professores, mesmo que a alguns não tenham sido atribuídas turmas/disciplinas.
-- No resultado, mostre a sigla do departamento, o siape e nome do professor, os códigos
-- das turmas/disciplinas, ano e semestre. Utilize o conceito de junção relacional externa,
-- empregando a combinação de comandos de junção externa à esquerda.
56 • SELECT
57     dep_sigla, p.pro_siape, pro_pnome, pro_unome, tur_codigo, dis_codigo, tudi_ano, tudi_semestre
58 FROM
59     departamento d
60     LEFT OUTER JOIN
61     professor p ON d.dep_codigo = p.dep_codigo
62     LEFT OUTER JOIN
63     turma_disciplina td ON p.pro_siape = td.pro_siape;
```

dep_sigla	pro_siape	pro_pnome	pro_unome	tur_codigo	dis_codigo	tudi_ano	tudi_semestre
DACOM	6	Lisa	Reimann	C41	EC34D	2017	2
DACOM	7	Corinna	Enoellmann	N12B	AN32C	2017	2
DACOM	7	Corinna	Enoellmann	N12B	AN32C	2018	1
DACOM	10	Giovani	Meinerz	NULL	NULL	NULL	NULL
DAELE	2	Tilo	Gerhold	NULL	NULL	NULL	NULL
DAELE	4	Gerhard	Huettd	E21	MA35B	2017	2
DAELE	5	Anoela	Lehmann	ES31	IF63C	2016	1
DAELE	8	Manfred	Schubbert	NULL	NULL	NULL	NULL
DAMAT	3	Ekkehart	Schubbert	NULL	NULL	NULL	NULL
DAMAT	9	Lena	Reimann	NULL	NULL	NULL	NULL
DAELN	NULL	NULL	NULL	NULL	NULL	NULL	NULL
DAELT	NULL	NULL	NULL	NULL	NULL	NULL	NULL
DAMEC	1	Hanka	Ruebbert	M31	EM33H	2017	2
DAFIS	NULL	NULL	NULL	NULL	NULL	NULL	NULL
DAGEE	NULL	NULL	NULL	NULL	NULL	NULL	NULL

RIGHT OUTER JOIN

Retorna todas as linhas da tabela B (a direita), mesmo não havendo linhas correspondentes na tabela A (da esquerda)

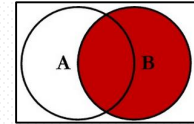


```
1 -- Selecione todas as salas, mesmo que em algumas salas não tenha sido ensalada nenhuma turma/disciplina.
2 -- No resultado mostre os códigos de sala, turma e disciplina.
3 -- Utilize o conceito de junção relacional externa, empregando o comando de junção externa à direita.
4 • SELECT s.sal_codigo, tur_codigo, dis_codigo
5 FROM ensalamento e RIGHT OUTER JOIN sala s ON e.sal_codigo = s.sal_codigo;
```

sal_codigo	tur_codigo	dis_codigo
I201	NULL	NULL
I202	NULL	NULL
I205	ES21	IF62H
I205	N12A	AN32C
I205	N12A	AN32C
I205	N12A	AN32C
I205	N12A	AN32C
K005	NULL	NULL
K008	NULL	NULL
K009	E21	MA35B
K009	E21	MA35B
P003	NULL	NULL
P005	NULL	NULL
P101	NULL	NULL
P105	M31	EM33H
P105	M31	EM33H
P205	NULL	NULL
A040	ES21	IF62H
A040	C41	EC34D
A040	C41	EC34D
A137	NULL	NULL
A140	NULL	NULL
A146	NULL	NULL

RIGHT OUTER JOIN (cont.)

Retorna todas as linhas da tabela B (a direita), mesmo não havendo linhas correspondentes na tabela A (da esquerda)

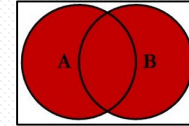


```
22 -- Selecione todos os tipos de sala, mesmo que em relação a
23 -- algum tipo ainda não exista nenhuma sala e, também, selecione
24 -- todas as salas, mesmo que em algumas salas não tenha sido
25 -- ensalada nenhuma turma/disciplina. No resultado mostre a
26 -- descrição do tipo da sala, os códigos de sala, turma e disciplina.
27 -- Utilize o conceito de junção relacional externa, empregando
28 -- a combinação de comandos de junção externa à direita.
29 • SELECT
30     tisa_descricao, s.sal_codigo, tur_codigo, dis_codigo
31 FROM
32     ensalamento e
33     RIGHT OUTER JOIN
34     sala s ON e.sal_codigo = s.sal_codigo
35     RIGHT OUTER JOIN
36     tipo_sala t ON s.tisa_codigo = t.tisa_codigo;
```

tisa_descricao	sal_codigo	tur_codigo	dis_codigo
Especial	NULL	NULL	NULL
Teórica	A140	NULL	NULL
Teórica	A146	NULL	NULL
Teórica	P101	NULL	NULL
Laboratório	A040	ES21	IF62H
Laboratório	A040	C41	EC34D
Laboratório	A040	C41	EC34D
Laboratório	I201	NULL	NULL
Laboratório	I202	NULL	NULL
Laboratório	I205	ES21	IF62H
Laboratório	I205	N12A	AN32C
Laboratório	I205	N12A	AN32C
Laboratório	I205	N12A	AN32C
Laboratório	I205	N12A	AN32C
Laboratório	K008	NULL	NULL
Laboratório	K009	E21	MA35B
Laboratório	K009	E21	MA35B
Laboratório	P003	NULL	NULL
Laboratório	P005	NULL	NULL
Laboratório	P105	M31	EM33H
Laboratório	P105	M31	EM33H
Laboratório	P205	NULL	NULL
Desenho	NULL	NULL	NULL
Pesquisa	K005	NULL	NULL
Monitoria	A137	NULL	NULL

FULL OUTER JOIN

Retorna todas as linhas de ambas as tabelas (A e B), mesmo havendo linhas sem valores correspondentes



➤ Problema

- Selecione todos os professores e departamentos, mesmo que haja professores que não estejam vinculados a um departamento, bem como departamentos aos quais porventura não tenha professor vinculado. No resultado, mostre o siape, o nome do professor e a sigla do departamento. Utilize o conceito de junção relacional externa, empregando o comando de junção externa completa.

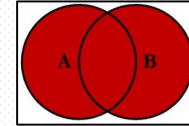
```
25 -- MYSQL não oferece suporte ao FULL OUTER JOIN
26 • SELECT
27     pro_siape, pro_pnome, pro_unome, dep_sigla
28 FROM
29     professor p
30 ✖     FULL OUTER JOIN
31     departamento d ON p.dep_codigo = d.dep_codigo;
```

Message

Error Code: 1064. You have an error in your SQL syntax; check the manual that corresponds to your MySQL server version

FULL OUTER JOIN (cont.)

Retorna todas as linhas de ambas as tabelas (A e B), mesmo havendo linhas sem valores correspondentes



```
1  -- Selecione todos os professores e departamentos, mesmo
2  -- que haja professores que não estejam vinculados a um
3  -- departamento, bem como departamentos aos quais porventura
4  -- não tenha professor vinculado. No resultado, mostre
5  -- o siape, nome do professor e a sigla do departamento.
6  -- Utilize o conceito de junção relacional externa,
7  -- empregando o comando de junção externa completa.
8  • SELECT
9      pro_siape, pro_pnome, pro_unome, dep_sigla
10 FROM
11     professor p
12     LEFT OUTER JOIN
13     departamento d ON p.dep_codigo = d.dep_codigo
14
15 UNION
16
17 SELECT
18     pro_siape, pro_pnome, pro_unome, dep_sigla
19 FROM
20     professor p
21     RIGHT OUTER JOIN
22     departamento d ON p.dep_codigo = d.dep_codigo;
```

pro_siape	pro_pnome	pro_unome	dep_sigla
1	Hanka	Ruebbert	DAMEC
2	Tilo	Gerhold	DAELE
3	Ekkehart	Schubbert	DAMAT
4	Gerhard	Huettia	DAELE
5	Angela	Lehmann	DAELE
6	Lisa	Reimann	DACOM
7	Corinna	Enoellmann	DACOM
8	Manfred	Schubbert	DAELE
9	Lena	Reimann	DAMAT
10	Giovani	Meinerz	DACOM
11	Luiz	Marengo	NULL
12	Mano	Lima	NULL
13	Cenair	Maicá	NULL
NULL	NULL	NULL	DAELN
NULL	NULL	NULL	DAELT
NULL	NULL	NULL	DAFIS
NULL	NULL	NULL	DAGEE

Função com Datas

→ EXTRACT()

→ Extrai unidade (ano, mês, dia) a partir de uma data

→ Sintaxe SQL

```
EXTRACT(Unidade FROM Data)
```

→ Para os exemplos com esta função, foi adicionado o atributo **pro_nascimento** à tabela **professor**

```
1 • SELECT * FROM professor;
```

pro_siape	pro_pnome	pro_unome	pro_salario	pro_fone_ddd	pro_fone_nro	pro_nascimento	dep_codigo
1	Hanka	Ruebbert	60000.00	43	35239000	1989-06-02	6
2	Tilo	Gerhold	65000.00	NULL	NULL	1989-09-20	2
3	Ekkehart	Schubbert	55000.00	11	754210000	1990-01-28	3
4	Gerhard	Huettig	85000.00	14	33448888	1990-04-01	2
5	Anoela	Lehmann	75000.00	43	35237777	1991-03-19	2
6	Lisa	Reimann	95000.00	NULL	NULL	1992-01-17	1
7	Corinna	Engelmann	130000.00	55	995554500	1992-02-28	1
8	Manfred	Schubbert	79000.00	43	998456587	1993-12-25	2
9	Lena	Reimann	145000.00	14	997465544	1994-06-21	3
10	Giovani	Meinerz	84000.00	55	999838457	1996-01-25	1
11	Luiz	Marengo	67000.00	55	996814596	1997-07-02	NULL
12	Mano	Lima	81000.00	55	975824684	1998-08-14	NULL
13	Cenair	Maicá	89000.00	55	997896341	1998-11-19	NULL

Função com Datas (cont.)

→ EXTRACT()

→ Extrai unidade (ano, mês, dia) a partir de uma data

→ Sintaxe SQL

EXTRACT(Unidade **FROM** Data)

```
1  -- Professores que nasceram em um determinado ano
2  • SELECT
3      pro_pnome Nome,
4      pro_unome Sobrenome,
5      pro_nascimento 'Data Nascimento'
6  FROM
7      professor
8  WHERE
9      EXTRACT(YEAR FROM pro_nascimento) = 1989
10 ORDER BY pro_nascimento;
```

Nome	Sobrenome	Data Nascimento
Hanka	Ruebbert	1989-06-02
Tilo	Gerhold	1989-09-20

Função com Datas (cont.)

→ EXTRACT()

→ Extrai unidade (ano, mês, dia) a partir de uma data

→ Sintaxe SQL

EXTRACT(Unidade **FROM** Data)

```
13 -- Professores, por determinado intervalo de ano de nascimento
14 • SELECT
15     pro_pnome Nome,
16     pro_unome Sobrenome,
17     pro_nascimento 'Data Nascimento'
18 FROM
19     professor
20 WHERE
21     EXTRACT(YEAR FROM pro_nascimento) BETWEEN 1986 AND 1995
22 ORDER BY pro_nascimento;
```

Nome	Sobrenome	Data Nascimento
Hanka	Ruebbert	1989-06-02
Tilo	Gerhold	1989-09-20
Ekkehart	Schubbert	1990-01-28
Gerhard	Huettig	1990-04-01
Anoela	Lehmann	1991-03-19
Lisa	Reimann	1992-01-17
Corinna	Enoellmann	1992-02-28
Manfred	Schubbert	1993-12-25
Lena	Reimann	1994-06-21

Função com Datas (cont.)

→ EXTRACT()

→ Extrai unidade (ano, mês, dia) a partir de uma data

→ Sintaxe SQL

EXTRACT(Unidade **FROM** Data)

```
24 -- Professores, por determinado intervalo de dias de um determinado mês
25 • SELECT
26     pro_pnome Nome,
27     pro_unome Sobrenome,
28     pro_nascimento 'Data Nascimento'
29 FROM
30     professor
31 WHERE
32     EXTRACT(DAY FROM pro_nascimento) BETWEEN 16 AND 30
33     AND EXTRACT(MONTH FROM pro_nascimento) = 01
34 ORDER BY pro_nascimento;
```

Nome	Sobrenome	Data Nascimento
Ekkehart	Schubbert	1990-01-28
Lisa	Reimann	1992-01-17
Giovani	Meinerz	1996-01-25

Função de Concatenação

→ CONCAT()

→ Concatena duas ou mais expressões

→ Sintaxe SQL

```
CONCAT(expressão1, expressão2, expressão3, ...)
```

→ Problema

→ Dado o ensalamento de turmas/disciplina a seguir, concatene as expressões Ano e Semestre, e Dia e Horário

Turma	Disciplina	Ano	Semestre	Dia	Horário	Sala
M31	EM33H	2017	2	5	M4	P105
M31	EM33H	2017	2	5	M5	P105
ES21	IF62H	2017	2	2	N1	I205
N12A	AN32C	2017	2	4	N2	I205
N12A	AN32C	2017	2	4	N3	I205
N12A	AN32C	2017	2	2	N4	I205
N12A	AN32C	2017	2	2	N5	I205
E21	MA35B	2017	2	6	T2	K009
E21	MA35B	2017	2	6	T3	K009
ES21	IF62H	2017	1	3	T4	A040
C41	EC34D	2017	2	3	T4	A040
C41	EC34D	2017	2	3	T5	A040

Função de Concatenação (cont.)

→ Sintaxe SQL

CONCAT(expressão1, expressão2, expressão3,...)

```
14 -- CONCAT()
15 -- Concatene Ano e Semestre, e Dia e Horário
16 • SELECT
17     tur_codigo Turma,
18     dis_codigo Disciplina,
19     CONCAT(tudi_ano,'/',tudi_semestre) 'Ano/Semestre',
20     CONCAT(ens_dia, hor_horario) Horário,
21     sal_codigo Sala
22 FROM
23     ensalamento
24     NATURAL JOIN
25     horario;
```

Turma	Disciplina	Ano/Semestre	Horário	Sala
M31	EM33H	2017/2	5M4	P105
M31	EM33H	2017/2	5M5	P105
ES21	IF62H	2017/2	2N1	I205
N12A	AN32C	2017/2	4N2	I205
N12A	AN32C	2017/2	4N3	I205
N12A	AN32C	2017/2	2N4	I205
N12A	AN32C	2017/2	2N5	I205
E21	MA35B	2017/2	6T2	K009
E21	MA35B	2017/2	6T3	K009
ES21	IF62H	2017/1	3T4	A040
C41	EC34D	2017/2	3T4	A040
C41	EC34D	2017/2	3T5	A040

Função de Concatenação (cont.)

- O SGBD adota o padrão americano para armazenar os valores de data (AAAA-MM-DD)

pro_pnome	pro_unome	pro_nascimento
Hanka	Ruebbert	1989-06-02
Tilo	Gerhold	1989-09-20
Ekkehart	Schubbert	1990-01-28
Gerhard	Huettig	1990-04-01
Anaela	Lehmann	1991-03-19
Lisa	Reimann	1992-01-17
Corinna	Engellmann	1992-02-28
Manfred	Schubbert	1993-12-25
Lena	Reimann	1994-06-21
Giovani	Meinerz	1996-01-25
Luiz	Marenco	1997-07-02
Mano	Lima	1998-08-14
Cenair	Maicá	1998-11-19

Função de Concatenação (cont.)

- ➔ Utilize a função **CONCAT()** para apresentar as datas de nascimento no formato dia, mês e ano. Concatene, também, o nome do professor

```
1 SELECT
2     CONCAT(pro_pnome, ' ', pro_unome) Professor,
3     CONCAT(EXTRACT(DAY FROM pro_nascimento),
4             '-',
5             EXTRACT(MONTH FROM pro_nascimento),
6             '-',
7             EXTRACT(YEAR FROM pro_nascimento)) Nascimento
8 FROM
9     professor;
```

Professor	Nascimento
Hanka Ruebbert	2-6-1989
Tilo Gerhold	20-9-1989
Ekkehart Schubbert	28-1-1990
Gerhard Huettig	1-4-1990
Angela Lehmann	19-3-1991
Lisa Reimann	17-1-1992
Corinna Endellmann	28-2-1992
Manfred Schubbert	25-12-1993
Lena Reimann	21-6-1994
Giovani Meinerz	25-1-1996
Luiz Marengo	2-7-1997
Mano Lima	14-8-1998
Cenair Maicá	19-11-1998

Função de Arredondamento

→ ROUND()

→ Arredonda um valor por determinada precisão

→ Sintaxe SQL

ROUND(Valor Numérico, precisão)

Operador de comparação

Subconsulta no **SELECT**

```
68 -- Professores cujo salário seja superior a média salarial, mostrando, também, a média salarial
69 • SELECT pro_pnome, pro_unome, pro_salario, (SELECT AVG(pro_salario) FROM professor) AS 'Média Salarial'
70 FROM professor
71 WHERE pro_salario > (SELECT AVG(pro_salario) FROM professor)
72 ORDER BY pro_salario;
```

Subconsulta no **WHERE**

pro_pnome	pro_unome	pro_salario	Média Salarial
Cenair	Maicá	89000.00	85384.615385
Lisa	Reimann	95000.00	85384.615385
Corinna	Engellmann	130000.00	85384.615385
Lena	Reimann	145000.00	85384.615385

Função de Arredondamento (cont.)

→ Sintaxe SQL

ROUND(Valor Numérico, precisão)

```
81 -- Professores cujo salário seja superior a média salarial, mostrando,
82 -- também, a média salarial (empregando função ROUND()).
83 • SELECT
84     pro_pnome Nome,
85     pro_unome Sobrenome,
86     pro_salario Salário,
87     ROUND((SELECT AVG(pro_salario) FROM professor),2) AS 'Média Salarial'
88 FROM professor
89 WHERE pro_salario > (SELECT AVG(pro_salario) FROM professor)
90 ORDER BY pro_salario;
```

Nome	Sobrenome	Salário	Média Salarial
Cenair	Maicá	89000.00	85384.62
Lisa	Reimann	95000.00	85384.62
Corinna	Endellmann	130000.00	85384.62
Lena	Reimann	145000.00	85384.62

Função de *String*

→ UCASE()

→ Converte o valor de um campo *string* para maiúsculo

→ Sintaxe SQL

```
UCASE(Atributo)
```

→ LCASE()

→ Converte o valor de um campo *string* para minúsculo

→ Sintaxe SQL

```
LCASE(Atributo)
```


Função de *String* (cont.)

```
9 • SELECT cur_nome, dep_sigla
10 FROM curso NATURAL JOIN departamento;
```

cur_nome	dep_sigla
Engenharia de Computação	DACOM
Engenharia Elétrica	DAELE
Engenharia Mecânica	DAMEC
Engenharia de Software	DACOM
Tecnologia em Análise e Desenvolvimento de Sistemas	DACOM

- Converta o nome do curso para maiúsculo e a sigla do departamento para minúsculo

```
1 • SELECT
2     UCASE(cur_nome) Curso,
3     LCASE(dep_sigla) Departamento
4 FROM curso NATURAL JOIN departamento;
```

Curso	Departamento
ENGENHARIA DE COMPUTAÇÃO	dacom
ENGENHARIA ELÉTRICA	daele
ENGENHARIA MECÂNICA	damec
ENGENHARIA DE SOFTWARE	dacom
TECNOLOGIA EM ANÁLISE E DESENVOLVIMENTO DE SISTEMAS	dacom

Função de *String* (cont.)

→ LENGTH()

→ Retorna a quantidade de caracteres de um valor de *string*

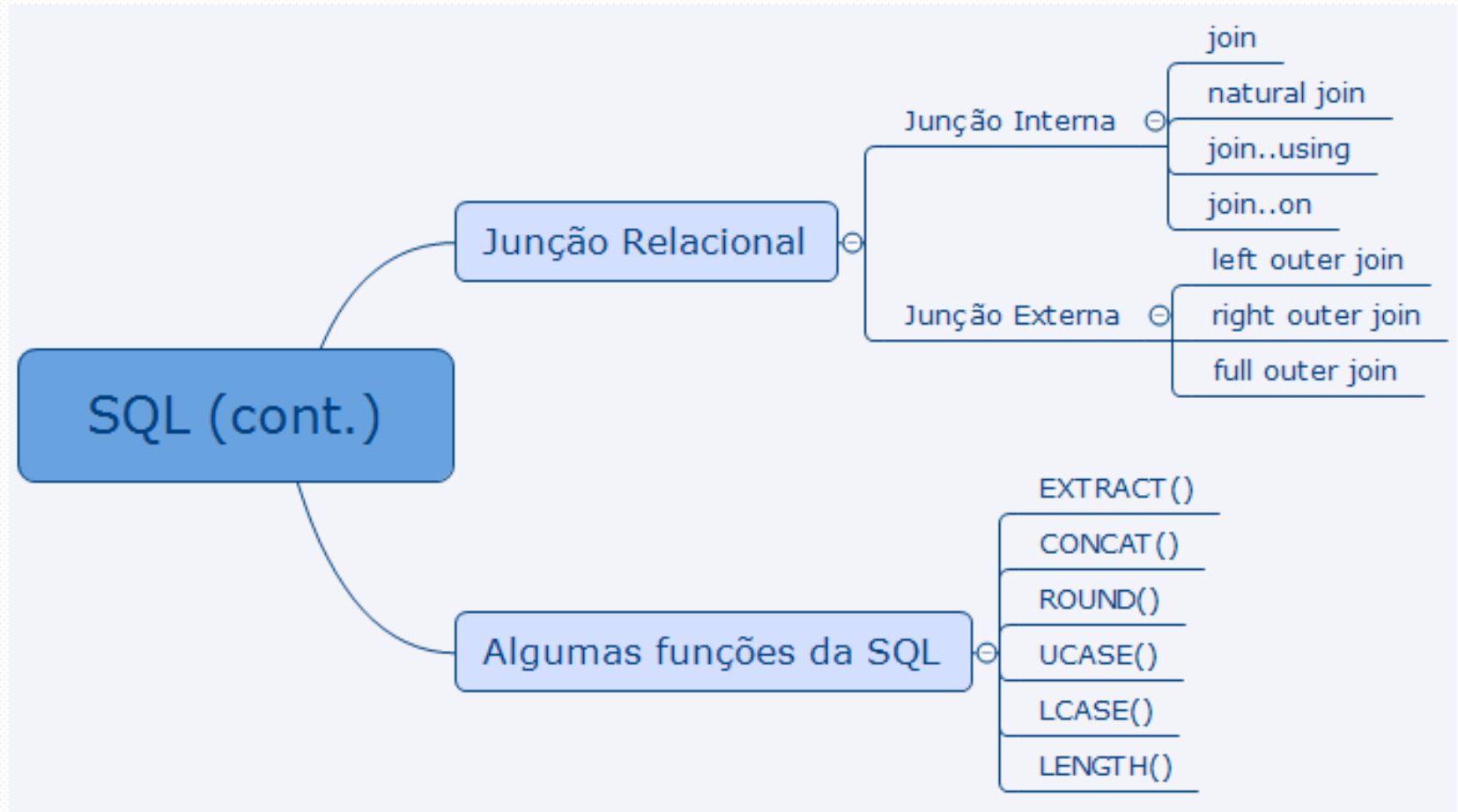
→ Sintaxe SQL

```
LENGTH(Atributo)
```

```
1 • SELECT
2     cur_nome Curso,
3     LENGTH(cur_nome) 'Qtde Caracteres'
4 FROM
5     curso;
```

Curso	Qtde Caracteres
Engenharia de Computação	26
Engenharia Elétrica	20
Engenharia Mecânica	20
Engenharia de Software	22
Tecnologia em Análise e Desenvolvimento de Sistemas	52

Resumo da Aula





DISCIPLINA: Banco de Dados 1

Prof. **GIOVANI** Volnei Meinerz

Aula 21 – SQL (cont.)