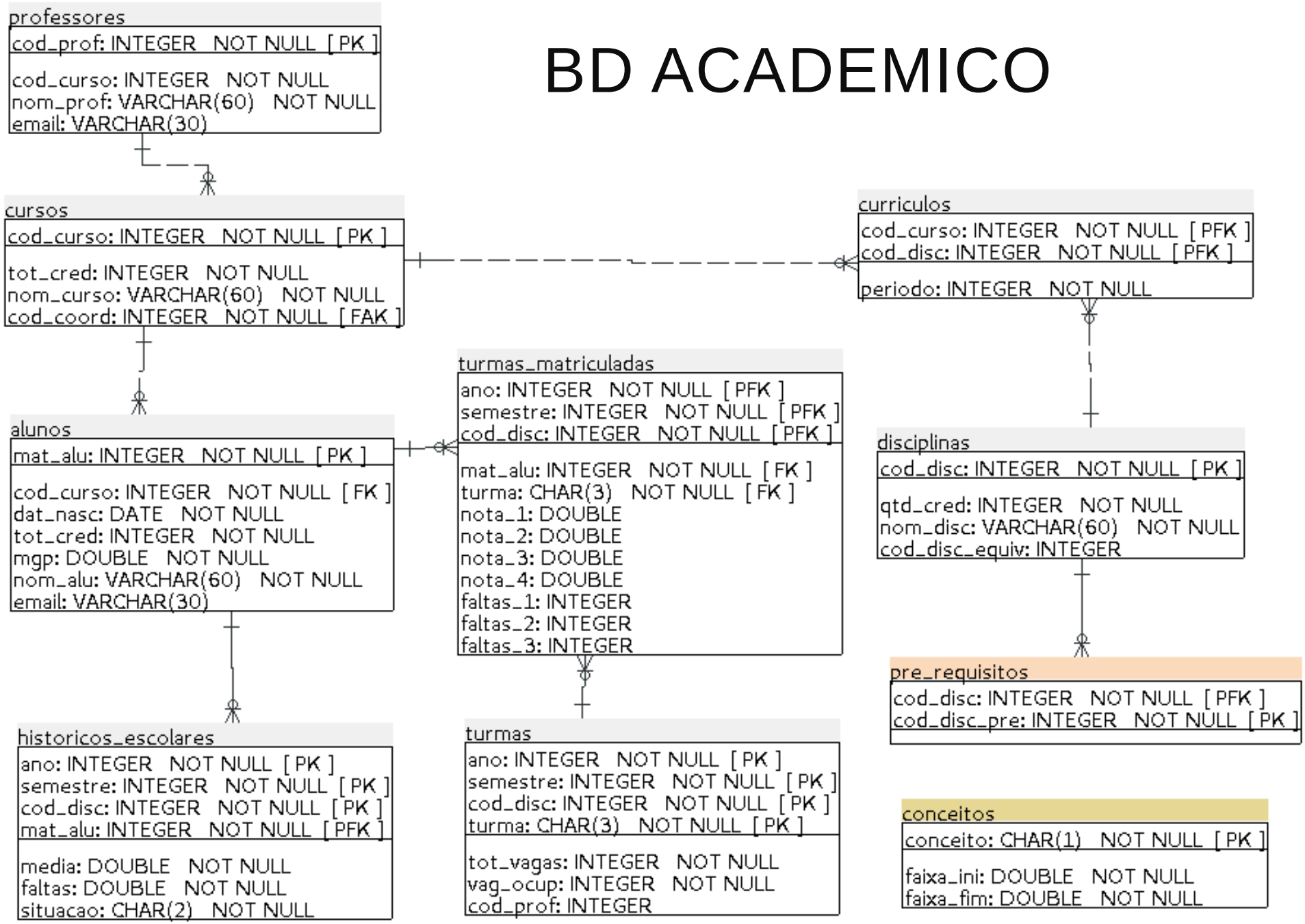
BANCO DE DADOS II

SQL JOINS

INSTITUTO FEDERAL DE
EDUCAÇÃO, CIÊNCIA E TECNOLOGIA
RIO GRANDE DO NORTE

Msc. Eliezio Soares
eliezio.soares@ifrn.edu.br | https://docente.ifrn.edu.br/elieziosoares

BD ACADEMICO



SQL JOINS

Queries podem acessar múltiplas tabelas de uma vez, ou acessar a mesma tabela de modo que múltiplas linhas da tabela sejam processadas ao mesmo tempo.

- Exemplo:
  - Tabela Tempo x Tabela Cidade
  - Listar todos os registros de tempo juntos com a localização da cidade associada.
  - Para fazer isso, precisamos comparar a coluna cidade com cada linha da tabela tempo, campo cidade, e selecionar os pares de linhas onde os valores são iguais.

```
     city      | temp_lo | temp_hi | prcp |    date    |     name      | location
---------------+---------+---------+------+------------+---------------+-----------
 San Francisco |      46 |      50 | 0.25 | 1994-11-27 | San Francisco | (-194,53)
 San Francisco |      43 |      57 |    0 | 1994-11-29 | San Francisco | (-194,53)
(2 rows)
```

SQL JOINS

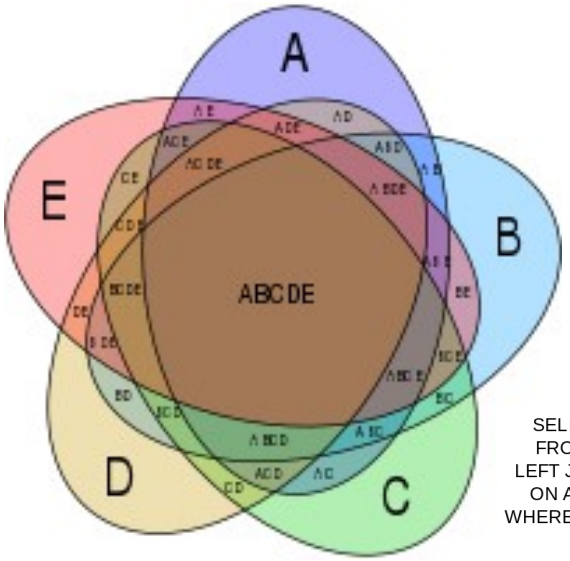
- Um SQL Join é usado para combinar linhas de duas ou mais tabelas, baseado em um campo comum entre elas.

- Relaciona-se os conjuntos de duas relações.

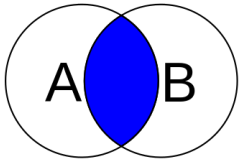
- Exemplo:
  - Para recuperar informações de “nome de um filme” e o “nome dos atores” que atuaram nele, é necessário juntar os dados da tabela Ator e da tabela Filme.

1	Benetti	Tecnologia em Processamento de Dados
2	Jorge	Tecnologia em Processamento de Dados
3	Marcelo	Tecnologia em Processamento de Dados
4	Wolney	Tecnologia em Processamento de Dados
5	Andre	Tecnologia em Processamento de Dados
6	Thadeu	Tecnologia em Processamento de Dados
7	Geniclecia	Tecnologia em Processamento de Dados
8	Denio	Tecnologia em Processamento de Dados

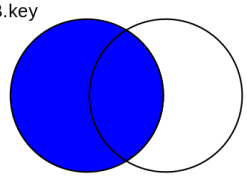
SQL JOINS



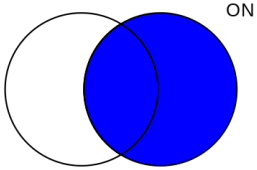
SELECT <fields>
FROM TableA A
INNER JOIN TableB B
ON A.key = B.key



SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key

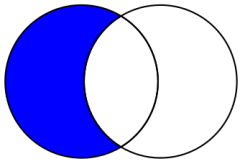


SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key

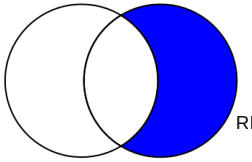


SQL JOINS

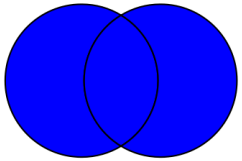
SELECT <fields>
FROM TableA A
LEFT JOIN TableB B
ON A.key = B.key
WHERE B.key IS NULL



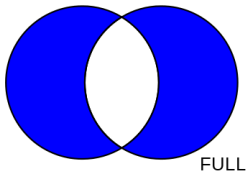
SELECT <fields>
FROM TableA A
RIGHT JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL



SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key

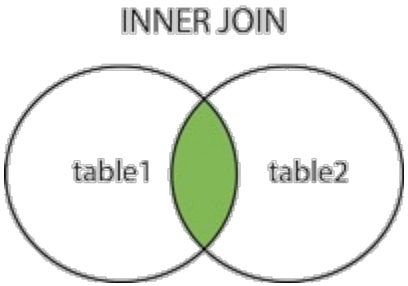


SELECT <fields>
FROM TableA A
FULL OUTER JOIN TableB B
ON A.key = B.key
WHERE A.key IS NULL
OR B.key IS NULL



INNER JOIN

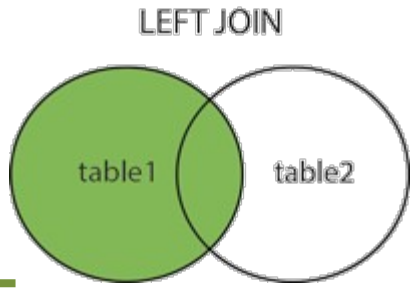
- O INNER JOIN seleciona todas as linhas de ambas as tabelas que hajam correspondentes nas duas tabelas.
  - Serão listadas apeas a intersessão.




```
SELECT column_name(s)
      FROM table1
      INNER JOIN table2
      ON table1.column_name=table2.column_name;
```

LEFT JOIN

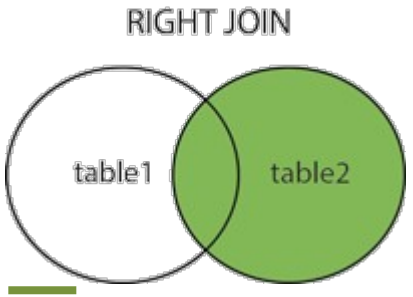
- O LEFT JOIN seleciona todas as linhas da tabela da esquerda (a primeira tabela) e inclui campos da tabela da direita (a segunda tabela).
  - Caso não haja correspondencia na segunda tabela, os campos serão null.




```
SELECT column_name(s)
	FROM table1
	LEFT JOIN table2
	ON table1.column_name=table2.column_name;
```

RIGHT JOIN

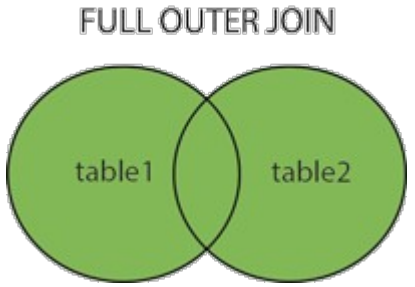
- O RIGHT JOIN seleciona todas as linhas da tabela da direita (a segunda tabela) e inclui campos da tabela da esquerda (a primeira tabela).
  - Caso não haja correspondencia na primeira tabela, os campos serão null.




```
SELECT column_name(s)
    FROM table1
    RIGHT JOIN table2
    ON table1.column_name=table2.column_name;
```

FULL OUTER JOIN

- O FULL OUTER JOIN seleciona todas as linhas da tabela da esquerda (a primeira tabela) e todas as linhas da tabela da direita (a segunda tabela).
  - É a junção do LEFT e RIGHT joins.




```
SELECT column_name(s)
	FROM table1
	FULL OUTER JOIN table2
	ON table1.column_name=table2.column_name;
```

SQL UNION

- O operador UNION realiza a combinação do result-set de dois ou mais SELECTS.
  - Os selects devem ter o mesmo número de colunas;
  - Os campos selecionados devem ter tipos de dados similares;

- Somente resultados distintos são selecionados.

```
SELECT column_name(s) FROM table1
      UNION
SELECT column_name(s) FROM table2;
```

SQL UNION ALL

- O operador UNION ALL realiza a combinação do result-set de dois ou mais SELECTS.
  - Esse comando permite a listagem de linhas repetidas;

```
SELECT column_name(s) FROM table1
      UNION ALL
SELECT column_name(s) FROM table2;
```

CROSS JOIN

- O CROSS JOIN seleciona todas as linhas de ambas as tabelas realizando o cruzamento entre as ocorrências.
- O CROSS JOIN não preza pelos relacionamentos entre os dados, ele apenas gera um cruzamento N-N entre as tuplas.

```
select alunos.nom_alu, conceitos.conceito from
alunos CROSS JOIN conceitos
```

172	alvaro	A
173	Andrezza	A
174	Luisa	A
175	Samanda	A
176	Benetti	B
177	Jorge	B
178	Marcelo	B
179	Wolney	B

DÚVIDAS?

EXERCÍCIO

Escreva para as tabelas Alunos (esquerda) e Cursos (direita), os seguintes joins:

1. Construa um SELECT que mostre todos os alunos e o respectivo nome do curso no qual ele está matriculado.
2. Construa um SELECT que mostre todos os alunos com matrícula.
3. Construa um SELECT que mostre todos os cursos com alunos matriculados.
4. Construa um SELECT que mostre todos os alunos sem curso.
5. Construa um SELECT que mostre todos os alunos e cursos. Caso o aluno não tenha vínculo com curso ainda assim ele deve aparecer no conjunto resultante.
6. Construa um SELECT que mostre todos as disciplinas que possuem pré-requisitos.
7. Construa um SELECT que mostre todos as disciplinas que NÃO possuem pré-requisitos.

REFERÊNCIAS BIBLIOGRÁFICAS

PostgreSQL 9.0.22 Documentation. Disponível em: <https://www.postgresql.org/files/documentation/pdf/9.0/postgresql-9.0-US.pdf>. Acesso em 27 Set. 2016.

SQL Joins. Disponível em: <https://www.w3schools.com/sql/sql_join.asp>. Acesso em 14 Fev. 2017.