



Faculdade FUCAPI
Curso de Graduação em Sistemas de Informação

Linguagem SQL

Banco de Dados II
Professor: Eder Martins Franco

Agenda

- ▶ Linguagem SQL
 - ▶ Introdução;
 - ▶ Subconjuntos;
 - ▶ Principais componentes:
 - ▶ Operadores
 - ▶ Cláusulas
 - ▶ Instruções e exemplos
 - ▶ Funções de agregação (próxima aula);
 - ▶ Subconsultas;
- ▶ Exercícios
- ▶ Referências

Banco de Dados II

Linguagem SQL



Linguagem SQL

► Introdução:

- A sigla SQL é uma abreviação para *Structured Query Language* (Linguagem de Consulta Estruturada);
- Originalmente criada pela IBM no início dos anos 90;
- Padronizada pela ANSI em 1986;
- Linguagem responsável pela interação com os dados armazenados na maioria dos bancos de dados relacionais;
- Permite realizar consultas (recuperação de informações), atualizações de dados, filtros, ordenações, etc.
- Apesar de ser padronizadas pela ANSI e ISO, diferentes SGBD implementam extensões da linguagem para facilitar a interação com seus bancos de dados:
 - Exemplos: Oracle, PostgreSQL e MSSQL Server;

Linguagem SQL

- ▶ Subconjuntos:
 - ▶ DDL – Linguagem de Definição de Dados;
 - ▶ CREATE, ALTER e DROP;
 - ▶ DML – Linguagem de Manipulação de Dados:
 - ▶ INSERT, UPDATE e DELETE;
 - ▶ **DQL – Linguagem de Consulta de Dados:**
 - ▶ **SELECT**, SHOW e HELP;
 - ▶ DCL ou TCL – Linguagem de Controle de Transações ou Linguagem de Transação de Dados:
 - ▶ GRANT e REVOKE;
 - ▶ DCL – Linguagem de Controle de Dados:
 - ▶ TRANSACTION, SAVEPOINT, COMMIT e ROLLBACK;

Linguagem SQL

► Introdução:

- Além das DDL já estudamos algumas operações com DML:
 - INSERT – Inserção de dados em uma tabela;
 - INSERT INTO <tabela>(campos) VALUES (valores);
 - UPDATE – Atualização de dados em uma tabela;
 - UPDATE <tabela> SET campo = novo_valor, campo2 = novo_valor WHERE condicao;
 - DELETE – Remoção de dados em uma tabela;
 - DELETE FROM <tabela> WHERE condicao;

Linguagem SQL

▶ Principais componentes

▶ Operadores Lógicos

- ▶ AND – E. Utilizando para compor uma sentença lógica onde ambos os operandos são verdadeiros;
- ▶ OR – OU. Utilizando para compor uma sentença lógica onde pelo menos um dos valores da expressão é verdadeiro;
- ▶ NOT – Negação. Utilizado para negar o valor de uma expressão;

Linguagem SQL

▶ Principais componentes

▶ Operadores relacionais

- ▶ Maior >;
- ▶ Menor <;
- ▶ Maior ou igual >=
- ▶ Menor ou igual <=
- ▶ Igual =
- ▶ Diferente <>, !=

Linguagem SQL

▶ Principais componentes

▶ Operadores relacionais

- ▶ IN – Valores contidos em uma lista
- ▶ NOT IN – Valores não contidos em uma lista
- ▶ BETWEEN – Valores dentro de um intervalo fechado
- ▶ LIKE – Comparação textual
- ▶ IS NULL – Verifica se é nulo
- ▶ IS NOT NULL – Verifica se não é nulo

Linguagem SQL

▶ Principais componentes

▶ Cláusulas

- ▶ FROM – Seleciona a tabela que vai ser consultada
- ▶ WHERE – Estabelece as condições para a consulta
- ▶ GROUP BY – Agrupa os registros
- ▶ ORDER BY – Ordena os registros
- ▶ LIMIT – Limita a quantidade de resultados

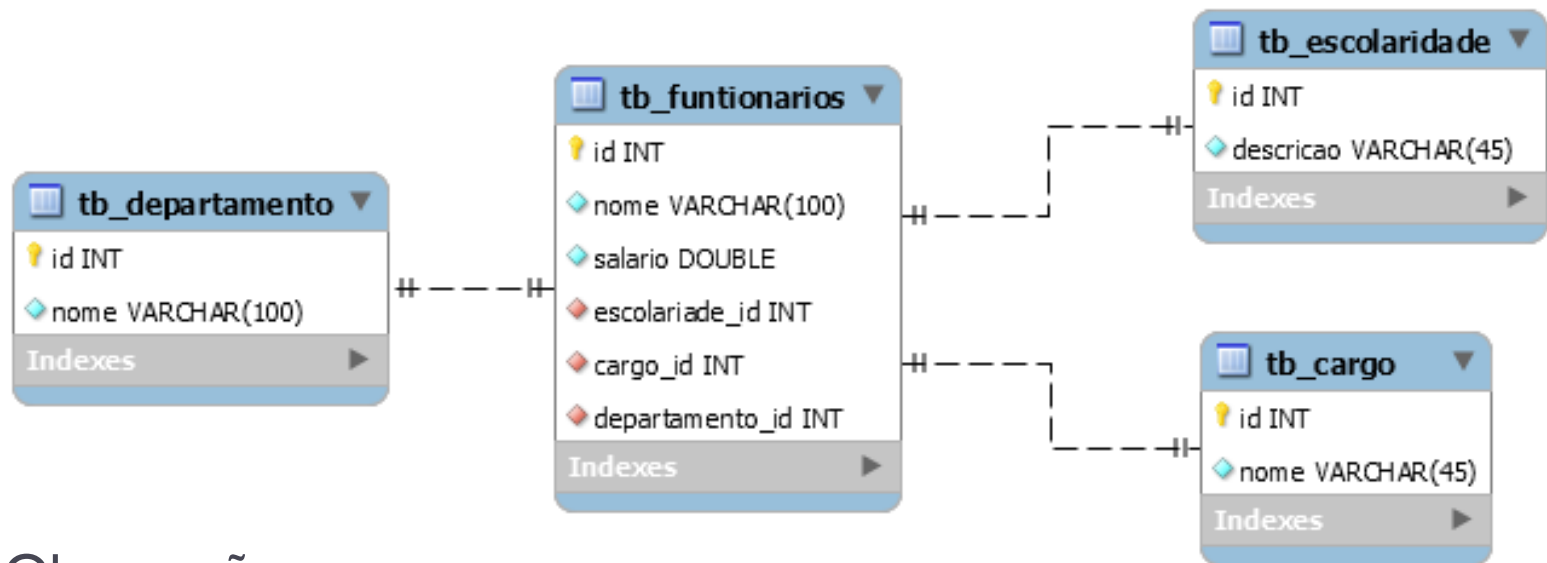
▶ Wildcards

- ▶ * - Seleciona todos as colunas de uma tabela
- ▶ % - Utilizado para definir o padrão de busca com LIKE

Linguagem SQL

► Instruções e exemplos

- Para esta aula serão criadas as seguintes tabelas de exemplo:



► Observação:

- Baixe e execute o arquivo .sql para criação desta base de dados no seu ambiente:
 - <http://moodle.franco.eti.br/mod/resource/view.php?id=347>

Linguagem SQL

► Instruções e exemplos

- SELECT - A instrução mais importante do SQL (DQL);
- Sintaxe básica:

SELECT <campo(s)> FROM <tabela>

► Exemplos:

- Para buscar todos os dados de todos os funcionários na tabela:
 - SELECT * FROM tb_funcionarios;

id	nome	salario	escolaridade_id	cargo_id	departamento_id
2	Rafael Maester	4.6	4	6	4
3	Leandro Divina da Costa	8.3	5	9	2
6	Luiz Lost Lustroza	6.6	4	4	2
7	Pee VeePeh Gaee	8.3	5	9	5
8	Keysuke Koisarara	1.2	2	11	1
10	Marcelo Maneiro	3.6	4	10	7
11	Toppercinda Amaral	2.2	7	4	6
12	José Bolso Mito	10.4	5	8	5
13	Gumercendo Jubarte	4.5	6	5	1
14	Pedro Monteiro Silveira	5	3	2	1
15	Pedro Capaldo	15	7	8	2

Linguagem SQL

► Instruções e exemplos

► SELECT - Exemplos:

- Para buscar somente o nome e salário de todos os funcionários na tabela:
 - `SELECT nome, salario FROM tb_funcionarios;`

nome	salario
Rafael Maester	4.6
Leandro Divina da Costa	8.3
Luiz Lost Lustroza	6.6
Pee VeePeh Gaae	8.3
Keysuke Koisarara	1.2
Marcelo Maneiro	3.6
Toppercinda Amaral	2.2
José Bolso Mito	10.4
Gumercendo Jubarte	4.5
Pedro Monteiro Silveira	5
Pedro Capaldo	15

Linguagem SQL

► Instruções e exemplos

► SELECT - WHERE:

- O parâmetro WHERE nos permite filtrar o conjunto de resultados a partir de uma condição que deve ser atendida;
- Exemplo: Buscar nome e salário de todos os funcionários que ganham mais de 5 mil reais.
 - `SELECT nome, salario FROM tb_funcionarios WHERE salario > 5.0;`

nome	salario
Leandro Divina da Costa	8.3
Luiz Lost Lustroza	6.6
Pee VeePeh Gaae	8.3
José Bolso Mito	10.4
Pedro Capaldo	15

Linguagem SQL

► Instruções e exemplos

► SELECT - WHERE:

- Exemplo: Buscar nome e salário de todos os funcionários com nível superior.
 - `SELECT nome, salario FROM tb_funcionarios WHERE escolaridade_id = 4;`

nome	salario
Rafael Maester	4.6
Luiz Lost Lustroza	6.6
Marcelo Maneiro	3.6

Linguagem SQL

▶ Instruções e exemplos

▶ SELECT – WHERE + AND:

- ▶ Pode incluir mais de uma restrição com o operador AND;
- ▶ Exemplo: Buscar nome e salário de todos os funcionários com nível superior e que ganham mais de 5 mil reais;
 - `SELECT nome, salario FROM tb_funcionarios WHERE escolaridade_id = 4 AND salario > 5.0;`

nome	salario
Luiz Lost Lustroza	6.6

Linguagem SQL

► Instruções e exemplos

► SELECT – WHERE + BETWEEN:

- Permite consultar dados em um intervalo;
- Exemplo: Buscar nome e salário de todos os funcionários com salário entre 4 e 8 mil reais;
 - `SELECT nome, salario FROM tb_funcionarios WHERE salario BETWEEN 4.0 AND 8.0;`

nome	salario
Rafael Maester	4.6
Luiz Lost Lustroza	6.6
Gumercendo Jubarte	4.5
Pedro Monteiro Silveira	5

Linguagem SQL

► Instruções e exemplos

► SELECT – AS:

- Permite atribuir alias a campos ou tabelas;
- Exemplo: Buscar nome_completo e salario de todos os funcionários;
 - SELECT nome AS nome_completo, salario FROM tb_funcionarios;

nome_completo	salario
Rafael Maester	4.6
Leandro Divina da Costa	8.3
Luiz Lost Lustroza	6.6
Pee VeePeh Gaee	8.3
Keysuke Koisarara	1.2
Marcelo Maneiro	3.6
Toppercinda Amaral	2.2
José Bolso Mito	10.4
Gumercendo Jubarte	4.5
Pedro Monteiro Silveira	5
Pedro Capaldo	15

Linguagem SQL

► Instruções e exemplos

► SELECT – ORDER BY:

- Permite ordenar os resultados de acordo com um ou mais campos, em ordem ASCendente ou DESCendente;
- Exemplo: Buscar os nomes dos funcionários ordenados pela coluna nome, de forma ascendente (do menor para o maior);
- A ordenação ascendente é o padrão quando nenhum tipo é definido;
 - `SELECT nome FROM tb_funcionarios ORDER BY nome;`

nome
Gumercindo Jubarte
José Bolso Mito
Keysuke Koisarara
Leandro Divina da Costa
Luiz Lost Lustroza
Marcelo Maneiro
Pedro Capaldo
Pedro Monteiro Silveira
Pee VeePeh Gaee
Rafael Maester
Toppercinda Amaral

Linguagem SQL

► Instruções e exemplos

► SELECT – ORDER BY:

- Permite ordenar os resultados de acordo com um ou mais campos, em ordem ASCendente ou DESCendente;
- Exemplo: Buscar os nomes dos funcionários ordenados pela coluna nome, de forma ascendente (do menor para o maior);
- A ordenação ascendente é o padrão quando nenhum tipo é definido;
 - `SELECT nome FROM tb_funcionarios ORDER BY nome;`

nome
Gumercindo Jubarte
José Bolso Mito
Keysuke Koisarara
Leandro Divina da Costa
Luiz Lost Lustroza
Marcelo Maneiro
Pedro Capaldo
Pedro Monteiro Silveira
Pee VeePeh Gaee
Rafael Maester
Toppercinda Amaral

Linguagem SQL

► Instruções e exemplos

► SELECT – ORDER BY:

- Exemplo: Buscar os nomes e salários dos funcionários ordenados pela coluna salario, de forma descendente (do maior para o menor);
- A ordenação ascendente é o padrão quando nenhum tipo é definido;
 - `SELECT nome, salario FROM tb_funcionarios ORDER BY salario DESC;`

nome	salario
Pedro Capaldo	15
José Bolso Mito	10.4
Leandro Divina da Costa	8.3
Pee VeePeh Gaae	8.3
Luiz Lost Lustroza	6.6
Pedro Monteiro Silveira	5
Rafael Maester	4.6
Gumercendo Jubarte	4.5
Marcelo Maneiro	3.6
Toppercinda Amaral	2.2
Keysuke Koisarara	1.2

Linguagem SQL

► Instruções e exemplos

► SELECT – GROUP BY:

- Permite agrupar os resultados de acordo com um parâmetro;
- Exemplo: Buscar os salários dos funcionários agrupados pela coluna salario;
 - `SELECT salario FROM tb_funcionarios GROUP BY salario;`

salario
1.2
2.2
3.6
4.5
4.6
5
6.6
8.3
10.4
15

Linguagem SQL

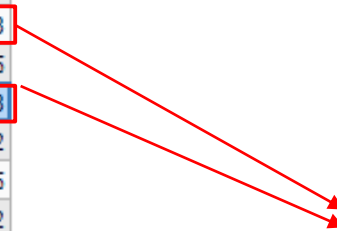
► Instruções e exemplos

► SELECT – GROUP BY:

- Exemplo: Buscar os salários dos funcionários agrupados pela coluna salario;
- Observe que dois funcionários possuem o salário 8.3, mas como os resultados foram agrupados, o valor é exibido somente uma vez;
 - SELECT salario FROM tb_funcionarios GROUP BY salario;

nome	salario
Rafael Maester	4.6
Leandro Divina da Costa	8.3
Luiz Lost Lustroza	6.6
Pee VeePeh Gaae	8.3
Keysuke Koisarara	1.2
Marcelo Maneiro	3.6
Toppercinda Amaral	2.2
José Bolso Mito	10.4
Gumercendo Jubarte	4.5
Pedro Monteiro Silveira	5
Pedro Capaldo	15

salario
1.2
2.2
3.6
4.5
4.6
5
6.6
8.3
10.4
15



Linguagem SQL

► Instruções e exemplos

► SELECT – LIMIT:

- Permite limitar a quantidade de resultados que serão retornados;
- Exemplo: Buscar os nomes e salários dos 3 primeiros funcionários na tabela;
 - `SELECT nome, salario FROM tb_funcionarios LIMIT 3;`

nome	salario
Rafael Maester	4.6
Leandro Divina da Costa	8.3
Luiz Lost Lustroza	6.6

Linguagem SQL

► Instruções e exemplos

► SELECT – IN:

- Realiza a busca levando em consideração uma lista de valores;
- Exemplo: Buscar os ids e nomes dos funcionários com ID 6, 7, e 8;
 - `SELECT id, nome FROM tb_funcionarios WHERE id IN (6,7,8);`

id	nome
6	Luiz Lost Lustroza
7	Pee VeePeh Gaee
8	Keysuke Koisarara

Linguagem SQL

► Instruções e exemplos

► SELECT – LIKE:

- LIKE permite realizar uma busca textual não exata. Para isso, utilizamos o wildcard %, da seguinte maneira:
 - ☐ % no início + texto:
 - ☐ Retornará resultados onde o texto da coluna termina com o texto procurado;
 - ☐ % no fim + texto:
 - ☐ Retornará resultados onde o texto da coluna começa com o texto procurado;
 - ☐ % + texto + %:
 - ☐ Retornará resultados onde o texto desejado esteja presente em qualquer parte do valor da coluna;
 - ☐ texto1 + % + texto2:
 - ☐ Retornará resultados que comecem com o texto1 e terminem com texto2;

Linguagem SQL

▶ Instruções e exemplos

▶ SELECT – LIKE:

▶ Exemplos:

- SELECT nome FROM tb_funcionarios WHERE nome LIKE '%silveira';

nome
Pedro Monteiro Silveira

- SELECT nome FROM tb_funcionarios WHERE nome LIKE 'leandro%';

nome
Leandro Divina da Costa

Linguagem SQL

► Instruções e exemplos

► SELECT – LIKE:

► Exemplos:

- SELECT nome FROM tb_funcionarios WHERE nome LIKE '%an%';

nome
Leandro Divina da Costa
Marcelo Maneiro

- SELECT nome FROM tb_funcionarios WHERE nome LIKE 'josé%mito';

nome
José Bolso Mito

Linguagem SQL

► Instruções e exemplos

► JOIN:

- O comando JOIN permite buscar dados em mais de uma tabela com uma só consulta;
- Para isso, é necessário identificar as tabelas com um alias ou com seu nome completo, permitindo identificar de quais tabelas devem ser coletados os campos desejados;
- As tabelas são unidas por meio de um campo em comum (as chaves estrangeiras, neste caso);
- Exemplo: Buscar os nomes e salários e cargos dos 5 primeiros funcionários:

```
SELECT A.nome,A.salario,B.nome AS cargo  
FROM tb_funcionarios A  
JOIN tb_cargo B ON (A.cargo_id = B.id)  
LIMIT 5;
```

Linguagem SQL

► Instruções e exemplos

► JOIN:

- Exemplo: Buscar os nomes e salários e cargos dos 5 primeiros funcionários:

```
SELECT A.nome,A.salario,B.nome AS cargo
FROM tb_funcionarios A
JOIN tb_cargo B ON (A.cargo_id = B.id)
LIMIT 5;
```

nome	salario	cargo
Rafael Maester	4.6	Analista Contábil
Leandro Divina da Costa	8.3	Coordenador
Luiz Lost Lustroza	6.6	Desenvolvedor
Pee VeePeh Gaee	8.3	Coordenador
Keysuke Koisarara	1.2	Estagiário

Linguagem SQL

► Instruções e exemplos

► JOIN:

- O comando JOIN nos permite decidir em relação a qual tabela queremos relacionar a busca: se à esquerda (LEFT) ou direta (JOIN);
- No exemplo abaixo, para cada registro existente na tabela da direita (funcionário) buscaremos o registro equivalente na tabela cargo;
- Exemplo: Buscar os nomes e cargos dos cinco primeiros funcionários;

```
SELECT A.nome, B.nome AS cargo
FROM tb_funcionarios A
LEFT JOIN tb_cargo B ON (A.cargo_id = B.id)
LIMIT 5;
```

nome	cargo
Rafael Maester	Analista Contábil
Leandro Divina da Costa	Coordenador
Luiz Lost Lustroza	Desenvolvedor
Pee VeePeh Gaee	Coordenador
Keysuke Koisarara	Estagiário

Linguagem SQL

► Instruções e exemplos

► JOIN:

- Se mudarmos a busca para utilizar o RIGHT JOIN, a consulta mudará: para cada cargo (tabela à esquerda) serão consultados os registros equivalentes na tabela de funcionários;
- Observe que dois registros retornaram sem resultado porque não encontraram equivalência.
- Exemplo: Buscar os nomes e cargos dos cinco primeiros funcionários;

```
SELECT A.nome, B.nome AS cargo
FROM tb_funcionarios A
RIGHT JOIN tb_cargo B ON (A.cargo_id = B.id)
LIMIT 5;
```

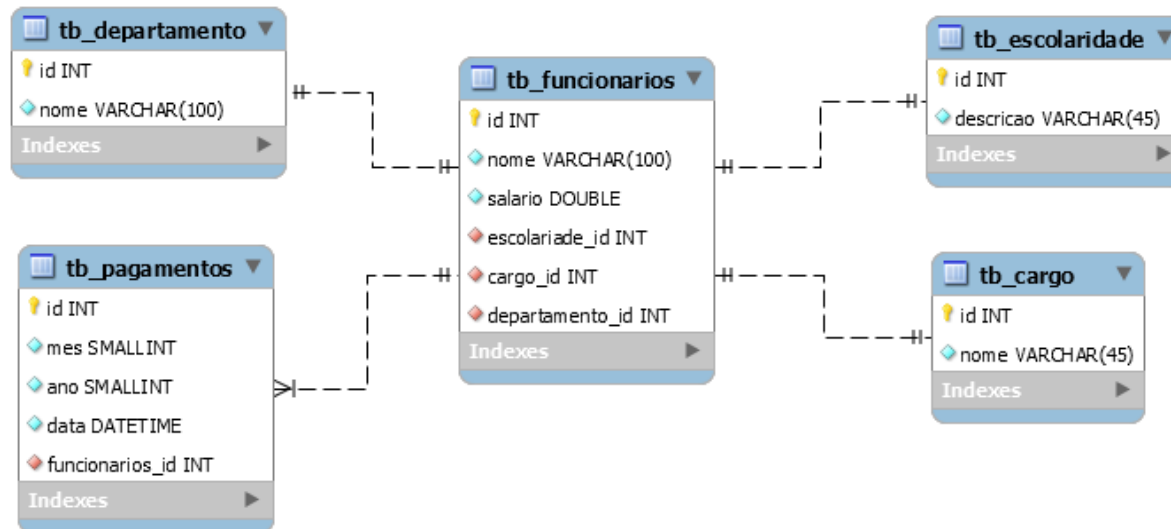
nome	cargo
(NULL)	Analista de Negócios
Pedro Monteiro Silveira	Analista de TI
(NULL)	Analista de Sistemas
Luiz Lost Lustroza	Desenvolvedor
Toppercinda Amaral	Desenvolvedor

Linguagem SQL

► Instruções e exemplos

► JOIN:

- Existem ainda outros dois modificadores para o JOIN: INNER e OUTER;
- Para simular esta operação, vamos criar uma nova tabela em nosso modelo, a tabela de pagamentos:



Linguagem SQL

► Instruções e exemplos

► JOIN:

- Exercício: criar a tb_pagamentos e inserir alguns registros;

```
CREATE TABLE tb_pagamentos (  
  id INT(11) NOT NULL AUTO_INCREMENT,  
  mes SMALLINT(6) NOT NULL,  
  ano SMALLINT(6) NOT NULL,  
  DATA DATETIME NOT NULL,  
  funcionario_id INT(11) NOT NULL,  
  PRIMARY KEY (id),  
  KEY fk_funcionario_id (funcionario_id),  
  CONSTRAINT fk_funcionario_id FOREIGN KEY (funcionario_id) REFERENCES tb_funcionarios (id)  
) ENGINE=INNODB;
```

```
ALTER TABLE tb_pagamentos ADD CONSTRAINT UQ_Mes_Ano_Funcionario UNIQUE(funcionario_id, mes, ano);
```

```
INSERT INTO tb_pagamentos (mes, ano, DATA, funcionario_id) VALUES (1,2017, '2017-01-01 00:00:00', 3);  
INSERT INTO tb_pagamentos (mes, ano, DATA, funcionario_id) VALUES (2,2017, '2017-01-01 00:00:00', 3);  
INSERT INTO tb_pagamentos (mes, ano, DATA, funcionario_id) VALUES (3,2017, '2017-01-01 00:00:00', 3);  
INSERT INTO tb_pagamentos (mes, ano, DATA, funcionario_id) VALUES (1,2017, '2017-01-01 00:00:00', 6);
```

Linguagem SQL

► Instruções e exemplos

► JOIN:

- Retomando os exemplos: O INNER JOIN cria uma relação exclusiva. Na consulta abaixo, serão retornados somente os funcionários que possuem registro de pagamento;

- Exemplo:

```
SELECT A.nome, B.mes, B.ano  
FROM tb_funcionarios A  
INNER JOIN tb_pagamentos B ON (A.id = B.funcionario_id);
```

nome	mes	ano
Leandro Divina da Costa	1	2017
Leandro Divina da Costa	2	2017
Leandro Divina da Costa	3	2017
Luiz Lost Lustroza	1	2017

Linguagem SQL

► Instruções e exemplos

► JOIN:

- O OUTER JOIN cria uma relação não exclusiva. Na consulta abaixo, serão retornados todos os funcionários, independente de possuírem um registro de pagamento;

► Exemplo:

```
SELECT A.nome, B.mes, B.ano
FROM tb_funcionarios A
LEFT OUTER JOIN tb_pagamentos B
ON (A.id = B.funcionario_id);
```

nome	mes	ano
Rafael Maester	(NULL)	(NULL)
Leandro Divina da Costa	1	2017
Leandro Divina da Costa	2	2017
Leandro Divina da Costa	3	2017
Luiz Lost Lustroza	1	2017
Pee VeePeh Gaae	(NULL)	(NULL)
Keysuke Koisarara	(NULL)	(NULL)
Marcelo Maneiro	(NULL)	(NULL)
Toppercinda Amaral	(NULL)	(NULL)
José Bolso Mito	(NULL)	(NULL)
Gumercendo Jubarte	(NULL)	(NULL)
Pedro Monteiro Silveira	(NULL)	(NULL)
Pedro Capaldo	(NULL)	(NULL)

Linguagem SQL

▶ Instruções e exemplos

▶ JOIN:

▶ Observações:

- ❑ Existem outros tipos de JOIN que não serão abrangidos neste a aula;
- ❑ Dependendo do SGBD o uso do JOIN pode mudar: no MySQL, por exemplo, não é possível utilizar somente o OUTER JOIN sem especificar a ordem das tabelas (LEFT ou RIGHT);
- ❑ Para compreender melhor os joins, devemos relembrar as operações com conjuntos;

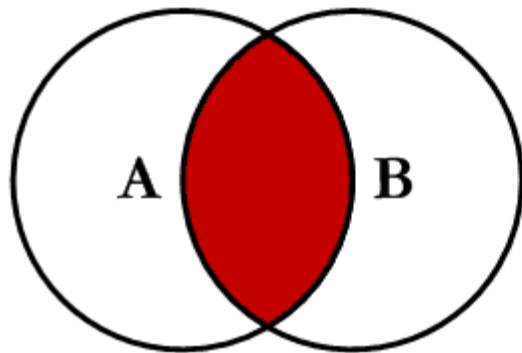
Linguagem SQL

► Instruções e exemplos

► JOIN:

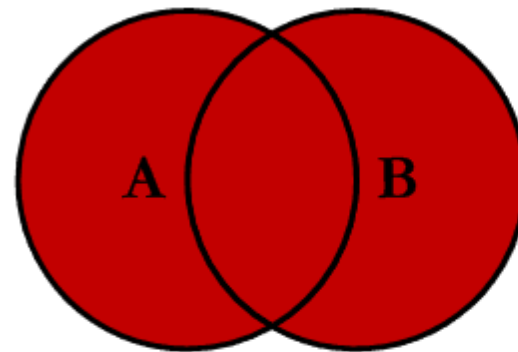
► Observações:

Inner JOIN



```
SELECT <select_list>
FROM Table_A A
INNER JOIN Table_B B
ON A.Key = B.Key
```

Outer JOIN



```
SELECT <select_list>
FROM Table_A A
FULL OUTER JOIN Table_B B
ON A.Key = B.Key
```

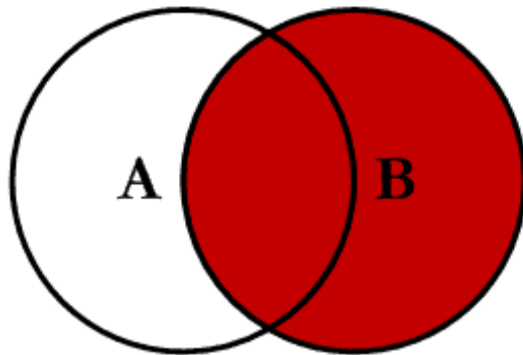
Linguagem SQL

► Instruções e exemplos

► JOIN:

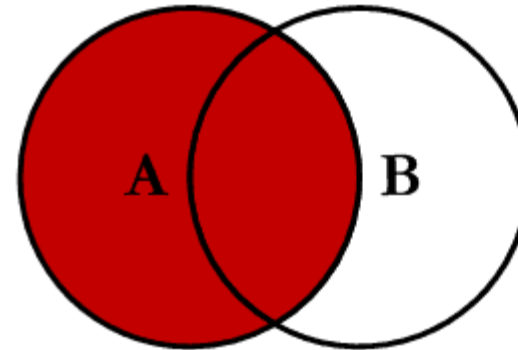
► Observações:

Right JOIN



```
SELECT <select_list>
FROM Table_A A
RIGHT JOIN Table_B B
ON A.Key = B.Key
```

Left JOIN



```
SELECT <select_list>
FROM Table_A A
LEFT JOIN Table_B B
ON A.Key = B.Key
```

Funções de Agregação

Funções de agregação

- ▶ Em 03/05/2017;



Subconsultas

Subconsultas

- ▶ Em 03/05/2017;

Banco de Dados II

Exercícios

Exercícios

- ▶ I. Lista de Exercícios 2:
 - ▶ Em 03/05/2017;

Referências

1. MILANI, André. **MySQL - Guia do Programador**. Ed. Novatec. 2006. Apêndice B. Páginas 339 a 355.
2. Microsoft Technet. **Noções básicas sobre subconsultas**. Disponível em <<http://bit.ly/2q3zvMv>> Acesso em 26/04/2017;
3. Database Administration – **ANSI SQL Standard Guidelines**. Disponível em <<http://bit.ly/2p2X2hR>> Acesso em 26/04/2017;
4. SQL Zoo. **SQL Tutorial – SELECTS**. Disponível em <<http://sqlzoo.net/>> Acesso em 26/04/2017;
5. Microsoft Technet. **Como usar caracteres curinga nas instruções SQL SELECT**. Disponível em <<http://bit.ly/2ph3ZdX>> Acesso em 26/04/2017;
6. Tech on the net. **SQL Comparsion Operators**. Disponível em <<http://bit.ly/2q3r5EJ>> Acesso em 26/04/2017;
7. Moffat, C. L. **Visual Representations of SQL Joins**. Disponível em <<http://bit.ly/2ph0aFi>> Acesso em 26/04/2017;

Banco de Dados II

Eder Martins Franco

efranco23@gmail.com

<http://moodle.franco.eti.br>

<http://fb.me/edermartinsfranco>



Este trabalho está licenciado sob uma Licença Creative Commons
Atribuição-NãoComercial-Compartilhalgual 4.0 Internacional.

Para ver uma cópia desta licença, visite <http://creativecommons.org/licenses/by-nc-sa/4.0/>.