



Prof. Taciano Balardin

[www.taciano.pro.br](http://www.taciano.pro.br)

[taciano@ulbra.edu.br](mailto:taciano@ulbra.edu.br)

2015-2



# Banco de Dados II

**E-MAIL DE CONTATO:**  
[taciano@ulbra.edu.br](mailto:taciano@ulbra.edu.br)

**SITE DA DISCIPLINA:**  
<http://www.taciano.pro.br/>



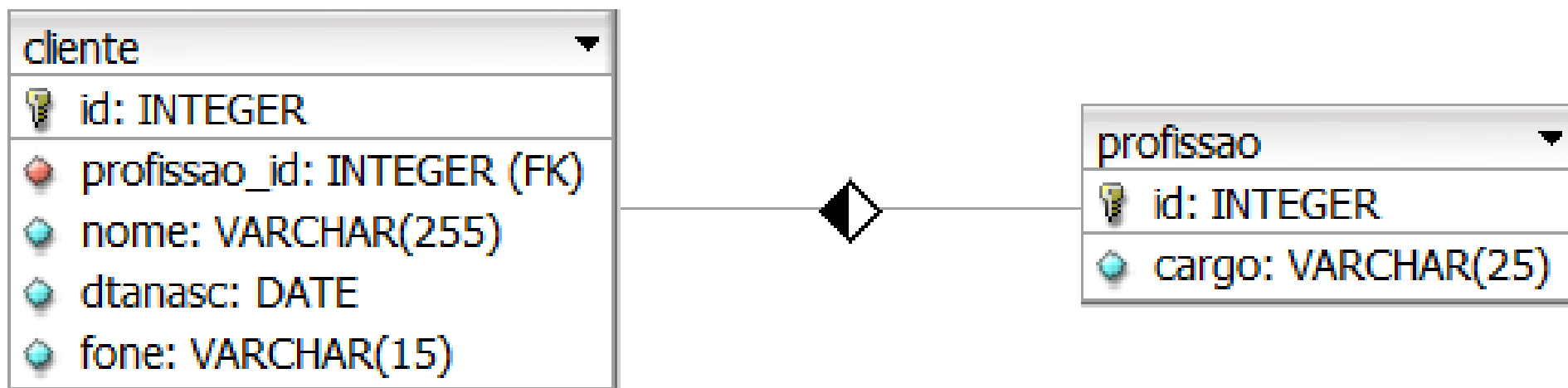
Junções com SQL:

- Left Join e Right Join
- Inner Join, Full Join e Cross Join

# **BANCO DE DADOS II**

## **AULA 02**

# Definição de tabelas



# Inserção de registros

id	profissao_id	nome	dtanasc	fone
1	1	Maria Antônia	1985-10-08	(51) 1234-5032
2	2	João Silva	1992-08-30	(51) 5432-1231
3	4	José Santos	1990-02-10	(51) 3451-2123
4	1	Ana Costa	1988-04-17	(51) 5431-2513
cliente				

id	cargo
1	Programadora
2	Analista de Sistemas
3	Administrador de BD
4	Suporte Técnico
profissao	

# INNER JOIN

- Uma **Junção Interna** é caracterizada por uma seleção que retorna apenas os dados que atendem às condições de junção, isto é, quais linhas de uma tabela se relacionam com as linhas de outras tabelas.
- Há **duas formas** diferentes de **expressar esta junção**:
  1. A **implícita** utiliza "," para separar as tabelas a combinar na cláusula **FROM** do **SELECT**. Então sempre é gerado o produto cruzado do qual são selecionadas as combinações que cumpram a cláusula **WHERE**.
  2. A **explícita** utiliza a palavra **JOIN** ou **INNER JOIN** combinada com a cláusula **ON**, que é semelhante à cláusula **WHERE**.

# INNER JOIN: Implícito

- Selecionar o nome do cliente e a sua profissão?

```
mysql > SELECT cliente.nome, profissao.cargo  
FROM cliente, profissao  
WHERE cliente.profissao_id = profissao.id;
```

**OU**

```
mysql > SELECT a.nome, b.cargo  
FROM cliente a, profissao b  
WHERE a.profissao_id = b.id;
```

# INNER JOIN: Explícito

- Selecionar o nome do cliente e a sua profissão?

```
mysql > SELECT c.nome, p.cargo  
FROM cliente c INNER JOIN profissao p  
ON c.profissao_id = p.id;
```

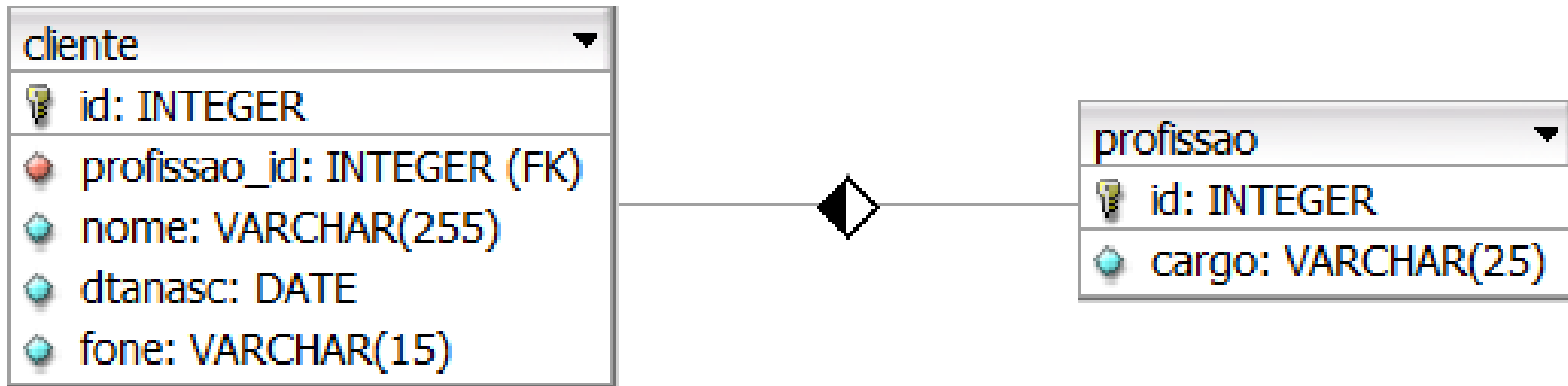
**OU**

```
mysql > SELECT c.nome, p.cargo  
FROM cliente c JOIN profissao p  
ON c.profissao_id = p.id;
```



# LEFT JOIN

- O resultado desta seleção sempre contém **todos os registros da tabela esquerda** (isto é, a primeira tabela mencionada na consulta), **mesmo quando não exista registros correspondentes na tabela direita**.
- Desta forma, esta seleção **retorna todos os valores da tabela esquerda** com os **valores da tabela direita correspondente**, ou quando não há correspondência retorna um valor NULL.



# LEFT JOIN

- Selecionar todos os clientes e a profissão (mesmo não tendo)?

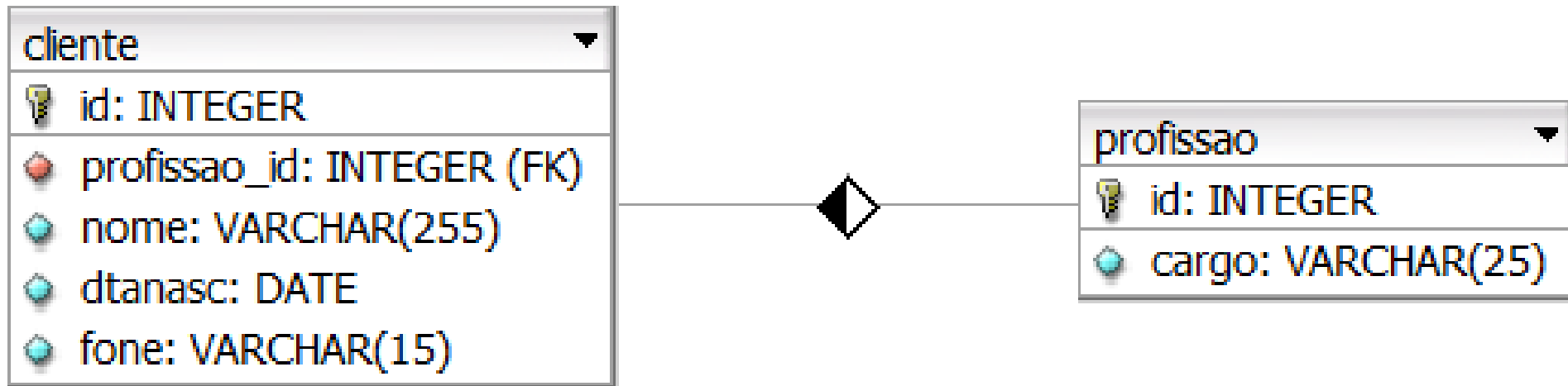
```
mysql > SELECT c.nome, p.cargo  
FROM cliente c LEFT JOIN profissao p  
ON c.profissao_id = p.id;
```

## SELECIONANDO TODAS AS COLUNAS DA TABELA CLIENTE?

```
mysql > SELECT c.*, p.cargo  
FROM cliente c LEFT JOIN profissao p  
ON c.profissao_id = p.id;
```

# RIGHT JOIN

- O resultado desta seleção sempre contém **todos os registros da tabela direita** (isto é, a segunda tabela mencionada na consulta), **mesmo quando não exista registros correspondentes na tabela direita**.
- Desta forma, esta seleção **retorna todos os valores da tabela da direita** com os **valores da tabela esquerda correspondente**, ou quando não há correspondência retorna um valor NULL.



# RIGHT JOIN

- Selecionar todos os clientes e a profissão (mesmo não tendo)?

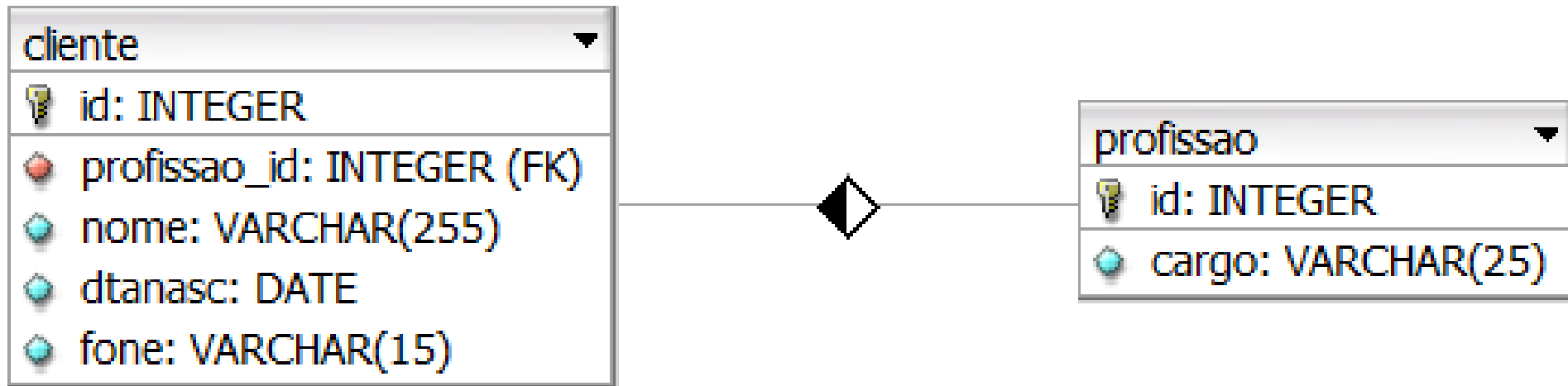
```
mysql > SELECT c.nome, p.cargo  
FROM cliente c LEFT JOIN profissao p  
ON c.profissao_id = p.id;
```

## SELECIONANDO TODAS AS COLUNAS DA TABELA CLIENTE?

```
mysql > SELECT c.*, p.cargo  
FROM cliente c LEFT JOIN profissao p  
ON c.profissao_id = p.id;
```

# FULL JOIN

- Esta operação apresenta **todos os dados das tabelas à esquerda e à direita**, mesmo que não possuam correspondência em outra tabela.
- A tabela combinada possuirá assim **todos os registros de ambas as tabelas** e apresentará **valores nulos** para os registros sem correspondência.



# FULL JOIN

- Selecionar todos os clientes e a profissões com FULL JOIN?

```
mysql > SELECT c.*, p.cargo  
FROM cliente c FULL JOIN profissao p  
ON c.profissao_id = p.id;
```



**POSSO AFIRMAR QUE O FULL JOIN É A  
UNIÃO DO LEFT JOIN E RIGHT JOIN?**

# FULL JOIN

- Como simular o FULL JOIN no MySQL?

```
mysql > SELECT c.nome, p.cargo  
FROM cliente c LEFT JOIN profissao p  
ON c.profissao_id = p.id
```

**UNION**

```
SELECT c.nome, p.cargo  
FROM cliente c RIGHT JOIN profissao p  
ON c.profissao_id = p.id ;
```



# CROSS JOIN

- Nesta junção **todos os dados da tabela à esquerda são cruzados com os dados da tabela à direita.**
- Exemplo:
  - Se a intenção é mostrar os dados de modo que **todos os funcionários tenham todos os cargos e vice-versa:**

```
mysql > SELECT * FROM cliente CROSS JOIN profissao;
```



[www.taciano.pro.br](http://www.taciano.pro.br)

# EXERCÍCIO

Defina como NULO o salário de todos os funcionários que possuem algum cargo e através de **um único UPDATE** resolva o seguinte desafio:

Insira valores aleatórios no campo salário da tabela funcionários, respeitando os limites impostos pelo cargo de cada funcionário.

## DESAFIO

