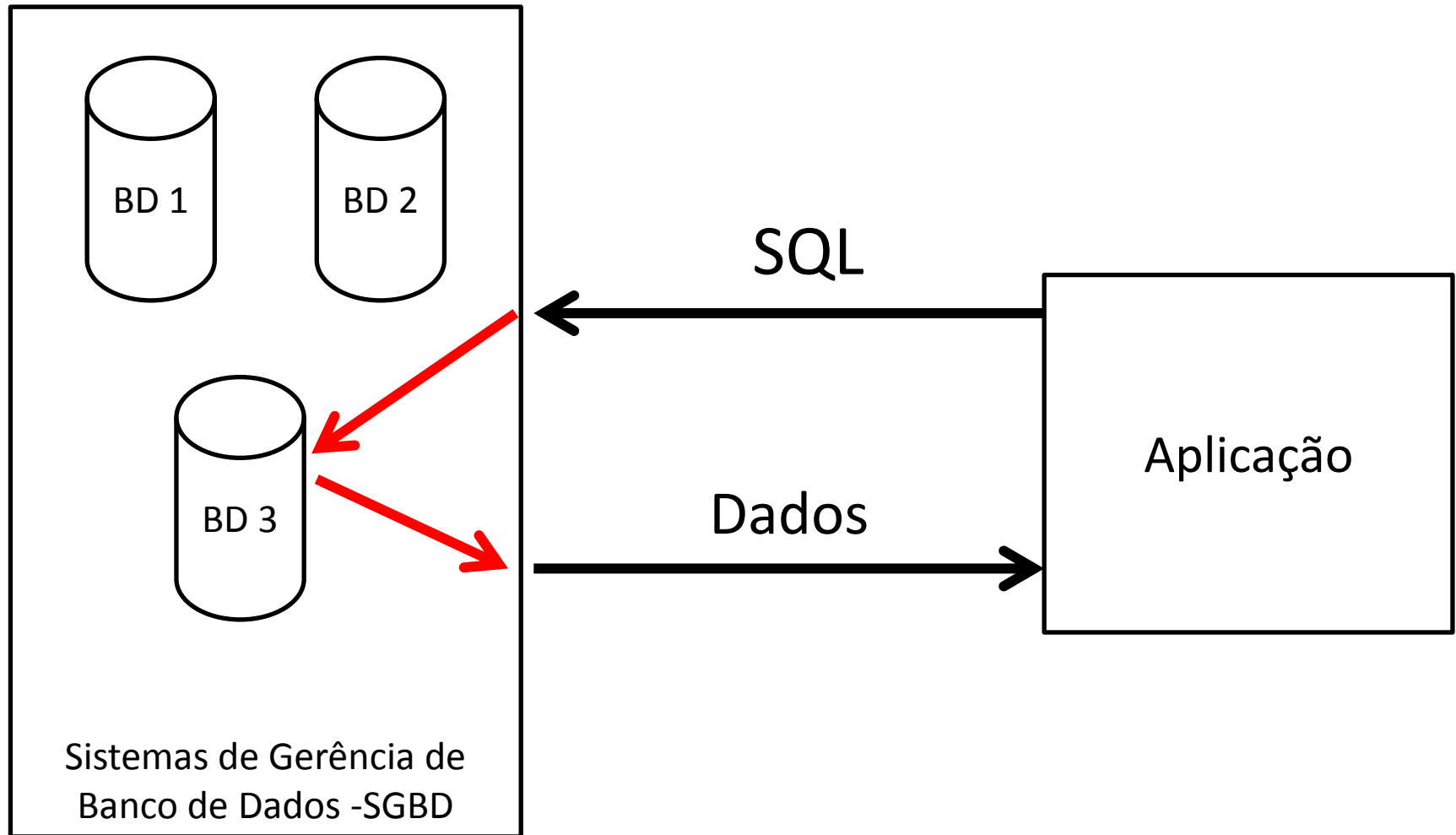


Curso de Banco de Dados

Prof. Rone Ilídio

Introdução



Introdução

Aplicação que controla os dados

Drives ODBC, BDE, ect

SQL Server, Oracle, Interbase

Arquivos contendo os dados



↑
Modelagem do Banco de Dados

Introdução

- Banco de Dados é um conjunto de dados integrados que tem com objetivo atender a um comunidade de usuários
- Sistema de Gerência de Banco de Dados (SGBD) é o software que incorpora as funções de definição, recuperação e alteração dos dados em um banco de dados
- Modelo dos dados é a definição formal da estrutura de um banco de dados

Banco de Dados Relacional

Tabela Empregado

Chave
Primária

Coluna (atributo)

Chave
Estrangeira

Chave Alternativa

Codigo	Nome	CodDepart	CPF	Linha (tupla)
1	João	13	11111111111	
2	Maria	28	22222222222	
3	José	13	33333333333	

↑
Valor do campo (valor do atributo)

Tabela Departamento

Codigo	NomeDepart
13	Desenvolvimento
28	Pessoal

Banco de Dados Relacional

Tabela Empregado

Chave
Primária

Coluna (atributo)

Chave
Estrangeira

Chave Alternativa

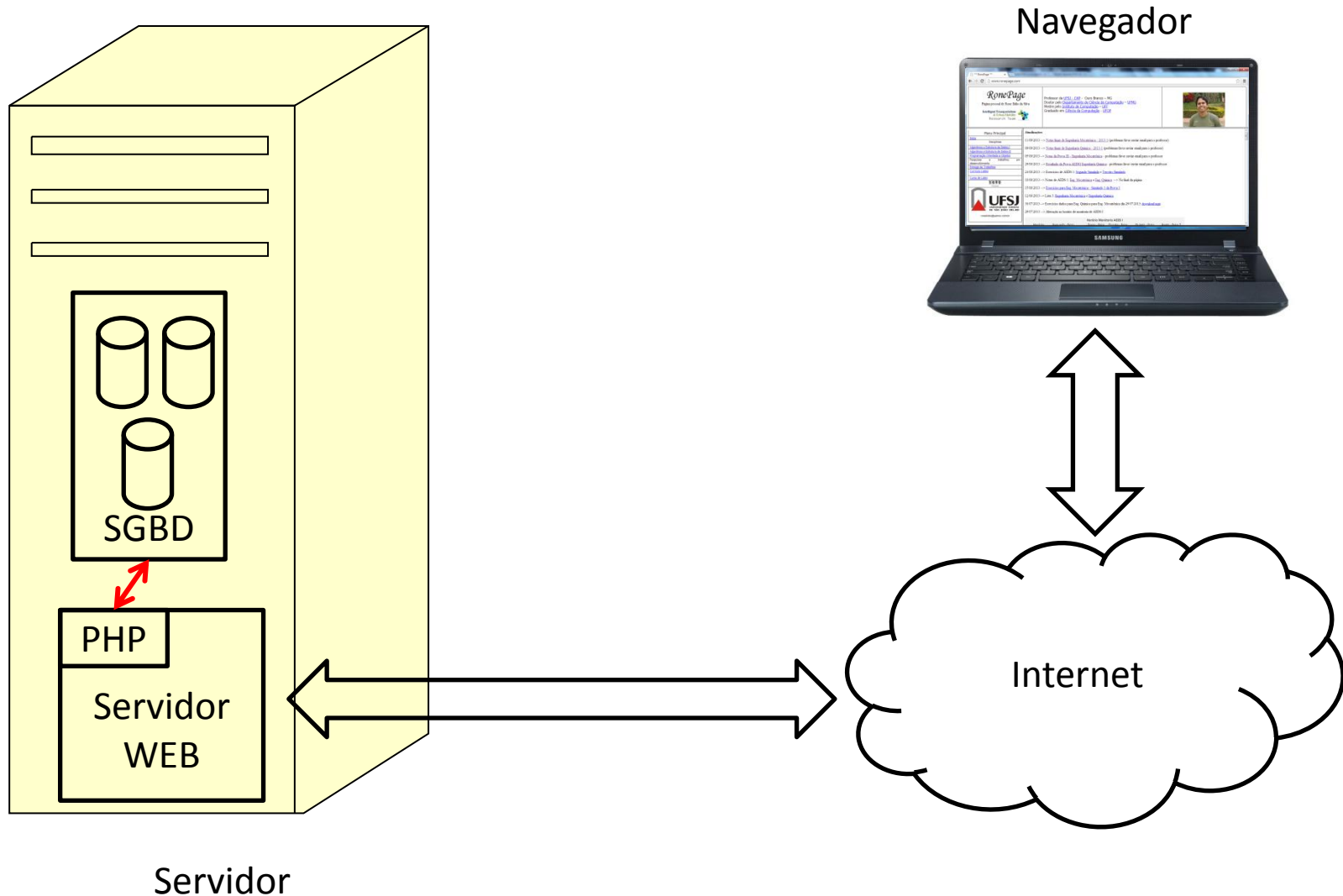
CodEmpregado	Nome	CodDepart	CPF	Linha (tupla)
E1	João	13	11111111111	
E2	Maria	28	22222222222	
E3	José	13	33333333333	

Valor do campo (valor do atributo)

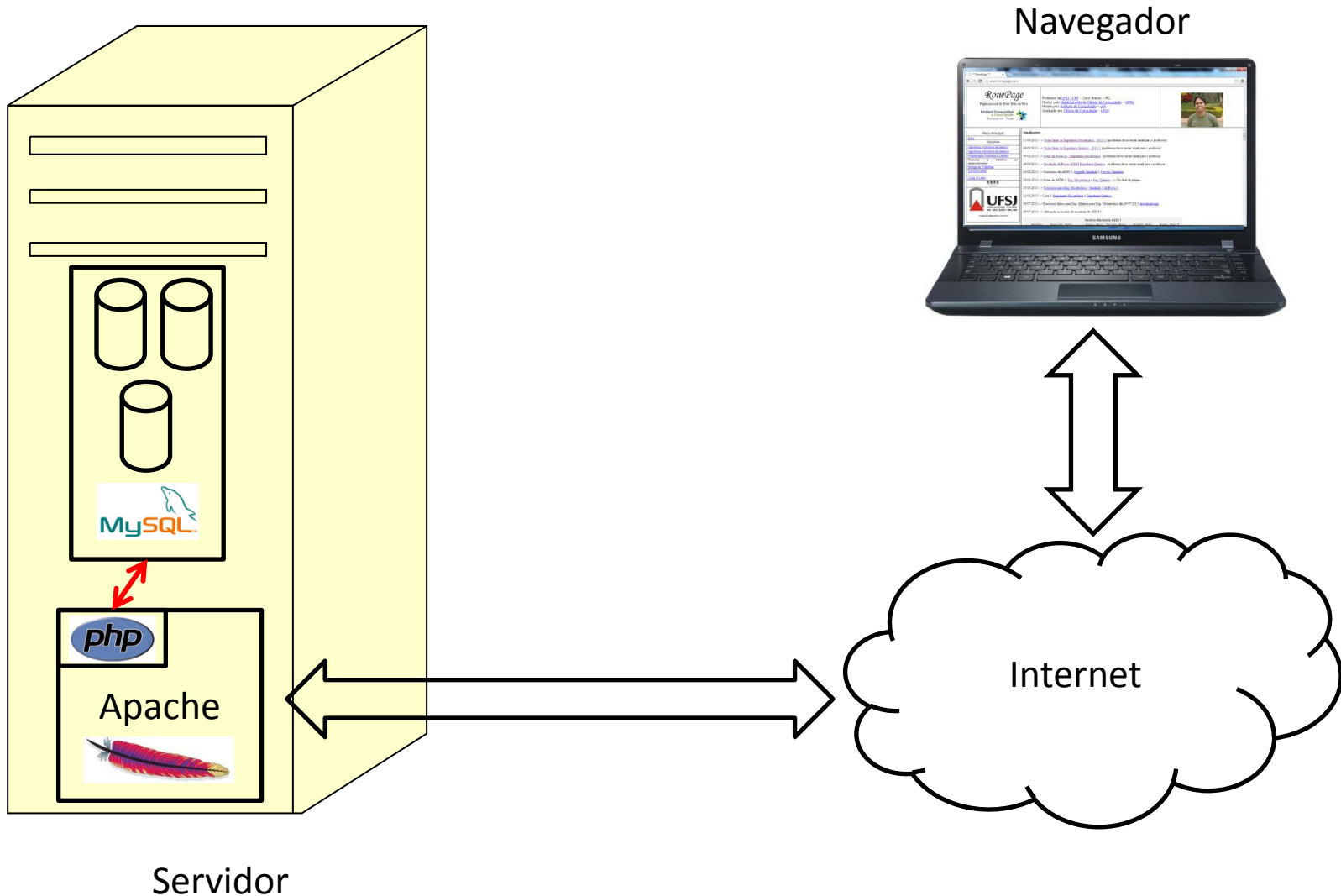
Tabela Departamento

CodDepartamento	NomeDepart
13	Desenvolvimento
28	Pessoal

Acesso a Banco de Dados na WEB



Acesso a Banco de Dados na WEB



SQL

- Structured Query Language
- Faz toda a manipulação de um banco de dados
- Operações que serão estudadas (manipulação de dados):
 - Inserção: insert
 - Exclusão: delete
 - Alteração: update
 - Busca: select

Tabelas Base

Campos:

Departamento: Codigo: Long Integer

Sala: Integer

Nome: VarChar(45)

Empregado: Matricula: Long Integer

CodDepartamento: Long Integer

Nome: VarChar(45)

Telefone: VarChar(45)

Salario: Float

Dependente: Codigo: Long Integer

MatEmpregado: Long Integer

Nome: VarChar(45)

Nascimento: Date

Tabelas Base

- Inserir alguns dados para visualização de consultas

Departamento

Codigo	Sala	Nome
1	101	Financeiro
2	201	Administrativo
3	102	Serviços Gerais

Empregado

Matricula	CodDepartamento	Nome	Telefone	Salario
2	1	ze	654564	R\$ 1.000,00
3	3	mane	11111	R\$ 2.000,00
4	1	maria	8978	R\$ 850,00

Dependente

Codigo	Nome	Nascimento	MatEmpregado
1	ana	20/06/1988	2
2	paulo	05/08/2000	2
3	joão vitor	23/12/1999	3

Insert

```
Insert into [tabela] (Campo1,...,CampoN)  
Values(Valor_Campo1, ...,Valor_CampoN)
```

Ex:

```
INSERT INTO Departamento(Nome, Sala)  
VALUES ("Departamento 1", 1)
```

Insert

Para campos do tipo Date os valores são passados no formato “#AAAA/MM/DD#”

Ex:

```
INSERT INTO Dependente(MatEmpregado,  
Nome, Nascimento)
```

```
VALUES (1,"Dependente 1", "#2003/01/01#")
```

Insert

Os comandos do tipo Datetime seguem o formato igual aos campos do tipo Date, mas inserindo um espaço e as horas separadas por :

“#AAAA/MM/DD# HH:MM:SS”

Ex:

```
INSERT INTO Dependente(MatEmpregado, Nome,  
Nascimento)
```

```
VALUES (1,"Dependente 1", "#2003/01/01 13:30:00#")
```

Insert

Como tratar outros tipos de dados

- Varchar (String): valores sempre entre aspas duplas (""")
- Float: utilizar ponto (3.1416)
- Integer: normalmente

Insert

O comando Now() retorna o dia e a hora atual.

Ex:

```
INSERT INTO Dependente(MatEmpregado,  
    Nome, Nascimento)
```

```
VALUES (1,"Dependente 1",Now())
```

Update

Atualiza os campos de uma tabela

Sintaxe:

Update Tabela Set Campo1=Valor, Campo2=Valor2

...

Where [Condição]

Update

- Exemplo

Update Dependente Set Nome="Maria José",
Nascimento = "#2003/10/01#"

WhereCodigo = 2

Delete

Apaga registros de uma tabela

Sintaxe:

Delete From Tabela

Where [Condição]

Delete

Exemplo:

Delete From Dependente
WhereCodigo = 5

Select

```
Select [Campo1], ..., [CampoN]  
From [Tabela1], ... [Tabela2]  
Where [Condição]
```

Obs: “*” = todos os campos das tabelas

Select

Exemplos

```
Select Nome,Telefone  
From Empregado  
Where Nome = "Joao"
```

```
Select Nome,Telefone  
From Empregado  
Where Nome = "Joao"
```

Obs: "*" = todos os **campos** das tabelas

Select

Ordenação – Comando Order By

Exemplo:

Select Nome,Telefone

From Empregado

Where Nome = “Joao”

Order By Salario

Select

Comando Distinct (só valores diferentes)

Exemplo:

```
Select Distinct Nome  
From Empregado
```

O resultado desta consulta é o nome de todos os empregados, sendo que cada nome aparece somente uma vez.

Select

Comando Sum(campo) – Retorna a soma de todos valores contidos em campo

Ex:

Matricula	Nome	Idade
1	João	43
2	José	23
3	Maria	54
4	Paulo	32

Select Sum(Idade) From Empregado → 152

Select

Comando Count(campo) – Retorna a quantidade de registros

Ex:

Matricula	Nome	Idade
1	João	43
2	José	23
3	Maria	54
4	Paulo	32

Select Count(Idade)

From Empregado

Where idade < 30 → 1

Select

Comando Min(campo) – Retorna o menor valor de todos os campos

Ex:

Matricula	Nome	Idade
1	João	43
2	José	23
3	Maria	54
4	Paulo	32

Select Min(Idade) From Empregado → 23

Select

Comando Max(campo) – Retorna o maior valor de todos os campos

Ex:

Matricula	Nome	Idade
1	João	43
2	José	23
3	Maria	54
4	Paulo	32

Select Max(Idade) From Empregado → 54

Select

Comando Avg(campo) – Retorna a média aritmética do campo especificado

Ex:

Matricula	Nome	Idade
1	João	43
2	José	23
3	Maria	54
4	Paulo	32

Select Avg(Idade) From Empregado → 33

Select

Comando like – comparação

Sintaxe:

Select [Campo1], ..., [CampoN]

From [Tabela1], ... [Tabela2]

Where [Campo] like [texto]

Select

Exemplo:

Select *

From Empregado

Where Nome = "João" ou

Select *

From Empregaod

Where Nome like "João"

Select

No comando like podemos utilizar o “%” significando “qualquer letra”. Ex:

Select *

From Empregaod

Where Nome like “J%”

→ Retorna todos os empregados com nomes iniciados com “J”

Select

O “%” também pode ser utilizado no início.
Exemplo:

Select *

From Empregado

Where Nome like “%a”

→ Retorna todos os empregados com nomes terminados com “a”

Select

Comando as: nomeia um campo na exibição do resultado.

Exemplo:

Select Nome as NomeEmpregado
From Empregado

→ Retorno:

NomeEmpregado
João
José
Maria
Paulo

Join de Tabelas

- Junção → consulta que busca dados de mais de uma tabela
- Os dados deve ser relacionados, ou seja, deve haver uma chave estrangeira entre as tabelas
- Tipos
 - Join simples (nome criado pelo autor do curso)
 - Left outer join
 - Right outer join
 - Full outer join

Join de Tabelas

Tabela Empregado

Codigo	Nome	Telefone
1	Nelson	22222222
2	Joana	33332222
3	Manuel	66667777
4	Miguel	33336666

Tabela
Dependente

Codigo	Nome	CodEmpregado
1	Patricia	2
2	Juliana	2
4	Paulo	4
6	Francisco	3
43	Mecias	7

Join de Tabelas

Select Empregado.Nome, Dependente.Nome
From Empregado, Dependente
Where Empregado.Codigo = Dependente.
CodEmpregado

Joana	Patricia
Joana	Juliana
Miguel	Paulo
Manuel	Francisco

Left outer join

Para o empregado Nelson aparecer no resultado, mesmo não tendo dependentes use-se:

```
Select Empregado.Nome, Dependente.Nome  
From Empregado left outer join Dependente on  
(Empregado.Codigo = Dependente.  
Empregado_Codigo)
```

Left outer join

Resultado

Nelson	
Joana	Patricia
Joana	Juliana
Miguel	Paulo
Manuel	Francisco

Right outer join

O Dependente chamado Mecias não apareceu,
para ele aparecer usa-se:

```
Select Empregado.Nome, Dependente.Nome  
From Empregado right outer join Dependente  
on (Empregado.Codigo = Dependente.  
Empregado_Codigo)
```

Right outer join

Resultado

Joana	Patricia
Joana	Juliana
Miguel	Paulo
Manuel	Francisco
	Mecias

Full outer join

Para que todos os campos apareçam, deve-se utilizar a consulta:

```
Select Empregado.Nome, Dependente.Nome  
From Empregado full outer join Dependente on  
(Empregado.Codigo =  
Dependente.Empregado_Codigo)
```

Obs: Não funciona no MySQL

Full outer join

Resultado

Nelson	
Joana	Patricia
Joana	Juliana
Miguel	Paulo
Manuel	Francisco
	Mecias