

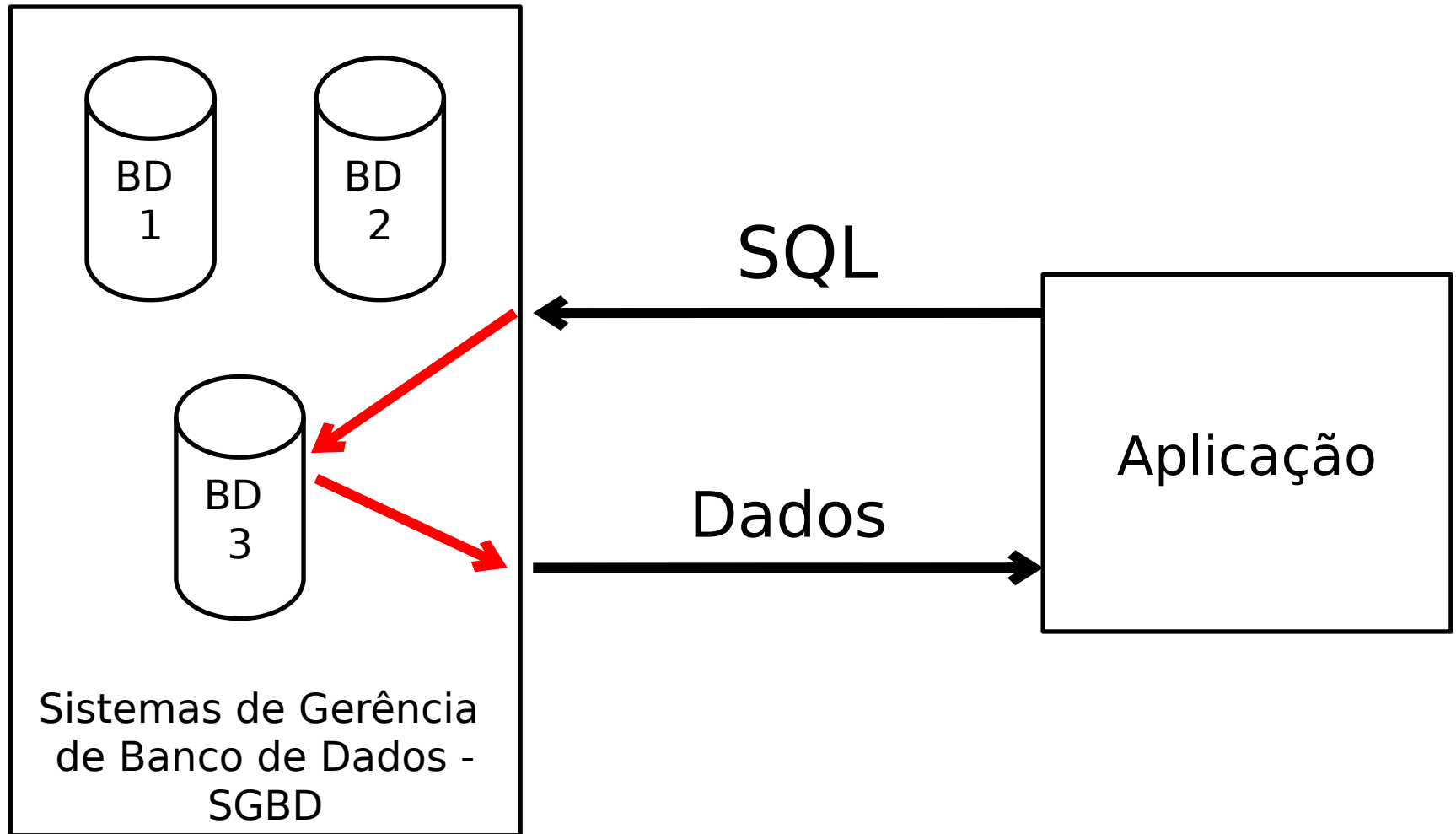
# Curso de Banco de Dados

Prof. Rone Ilídio

# Banco de Dados e Programação WEB

- Objetivos
  - Modelagem de Banco de Dados
  - Criação de Banco de Dados
  - Criação de Consultas: busca, atualização, deleção e inserção
  - Desenvolvimento de sites
  - Acesso e manipulação de dados via linguagem de programação

# Introdução



# Introdução

Aplicação que  
controla os dados



Aplicação

*Drives* ODBC, BDE,  
ect  
SQL Server, Oracle,  
Interbase



Meio de acesso



SGBD

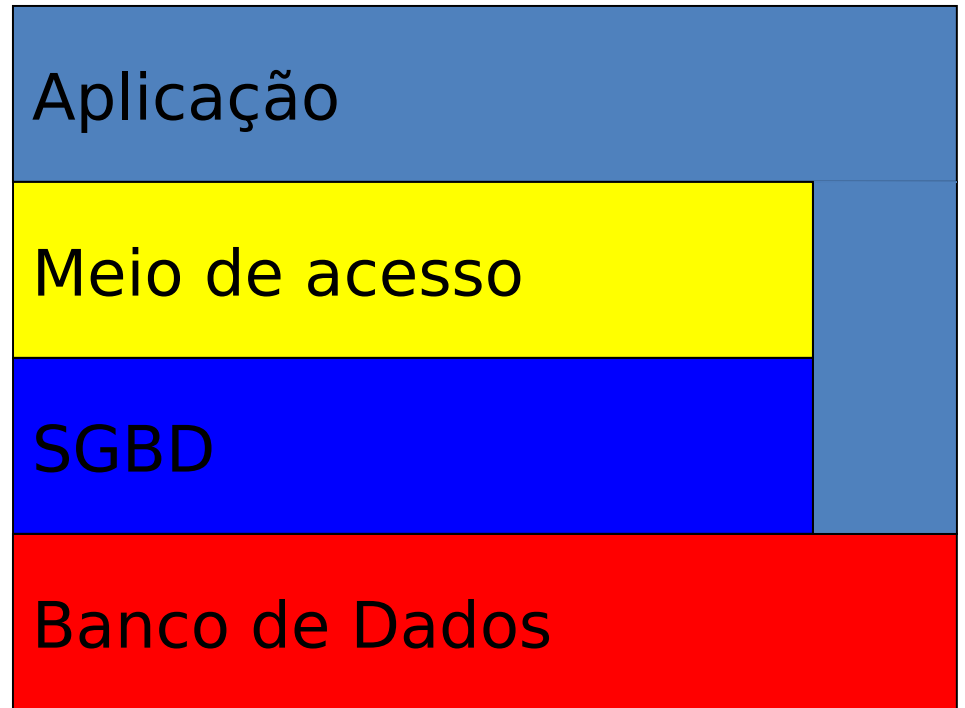
Arquivos contendo  
os dados



Banco de 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**

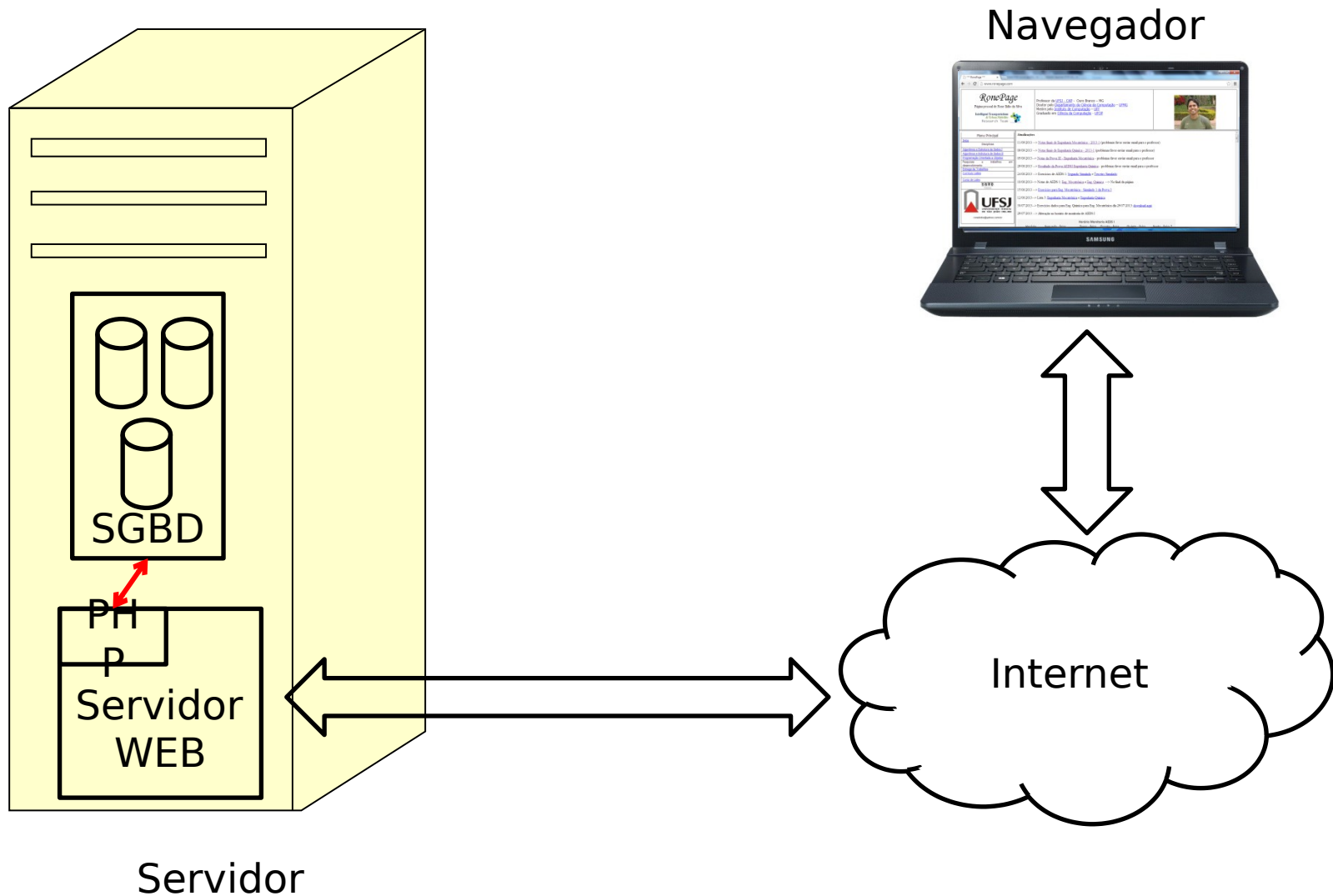
Chave Primária	Coluna (atributo)	Chave Estrangeira	Chave Alternativa	
	Codigo	CodDepart	CPF	Linha (tupla)
	1	13	11111111111	
	2	28	22222222222	
	3	13	33333333333	

Valor do campo (valor do atributo)

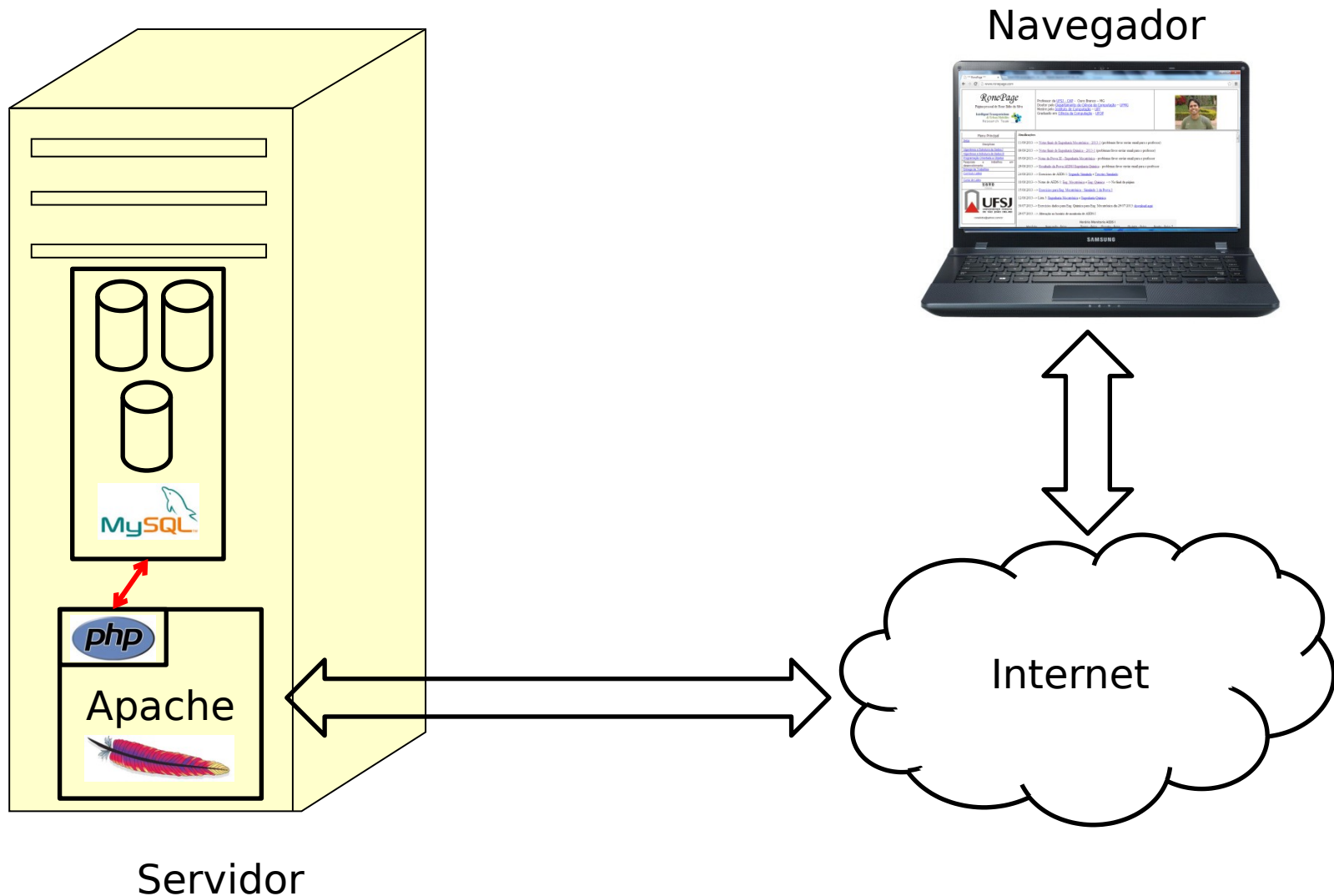
**Tabela Departamento**

Codigo	NomeDepart
13	Desenvolvimento
28	Pessoal

# Acesso a Banco de Dados na WEB



# Acesso a Banco de Dados na WEB





# Modelagem de Dados

- Modelo Conceitual: define as estruturas de dados que armazenarão os dados
- Modelo Lógico: define as estruturas de dados com características do tipo banco de dados que será utilizado. Ex: Modelo Lógico Relacional para banco de dados relacional, Modelo lógico orientado a objetos para banco de dados orientado a objeto
- Modelo Físico

# 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(255)

**Empregado:** Codigo: Long Integer  
CodDepartamento: Long Integer  
Nome: VarChar(255)  
Telefone: VarChar(255)  
Salario: Float

**Dependente:** Codigo: Long Integer  
CodEmpregado: Long Integer  
Nome: VarChar(255)  
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_Ca  
mpoN)
```

Ex:

```
INSERT INTO Departamento(Nome,  
    Sala)
```

# 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 ':'

“#DD/MM/AAAA HH:MM:SS”

Ex:

```
INSERT INTO Dependente(CodEmpregado,  
Nome, Nascimento)  
VALUES (2,"Dependente 1","04/05/2016")
```

# 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(CodEmpregado, Nome,  
    Nascimento)
```

```
VALUES (2,"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 = "01/10/2003"

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	Select *
From Empregado	From Empregado
Where Nome = "Joao"	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 Empregado

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 usa-se:

```
Select Empregado.Nome,  
Dependente.Nome
```

```
From Empregado left outer join  
Dependente on (Empregado.Codigo =  
Dependente.CodEmpregado)
```

# 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 =  
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