

# **BANCO DE DADOS**

**Linguagem SQL**

**Introdução à consulta - SELECT**

# SQL (STRUCTURED QUERY LANGUAGE)

Esquema para o modelo Relacional do contexto didático:  
**EMPRESA.**

FUNCIONARIO (ident, nome, sobrenome, endereco, dtnasc, salario, sexo, supident, dnumero)

DEPENDENTE (fident, nome, dt\_nasc, sexo, relacionamento)

DEPARTAMENTO (numero, nome, gident, dtinicio)

LOCALIZACOES (dnumero, localizacao)

PROJETO (numero, nome, localizacao, dnumero)

TRABALHA\_EM (pnumero, fident, horas)

Algumas simplificações nos nomes de variáveis foram realizadas para facilitar o seu uso nas consultas. A chave primária, em cada relação, está sublinhada.

# SQL - SELECT

Comando para recuperar dados em um ou mais objetos de um banco de dados. Estrutura básica do comando SELECT:

**SELECT** < lista de atributos >

**FROM** < lista de tabelas >

**WHERE** <condição>

**ORDER BY** <ordem>

Em que:

< **lista de atributos** >: é uma lista de nomes de atributos cujos valores serão recuperados pela consulta.

< **lista de tabelas** >: é uma lista de nomes de tabelas necessárias para encontrar a informação desejada na consulta.

< **condição** >: é uma expressão condicional (booleana) que identifica quais registros das tabelas mencionadas devem ser consideradas na consulta.

< **ordem** >: é o campo ou conjunto de campos que determina a ordem (classificação) em que os registros serão apresentados.

# **SELECT** < lista de atributos >

- Ao realizar uma seleção é possível recuperar dados de todos os campos disponíveis nas tabelas (\*) ou, então, escolher os campos a serem selecionados.
- Lembre-se sempre da questão relativa à performance e de onerar a rede para consultar dados.

ident	nome	sobrenome	endereco	dtnasc	salario	sexo	supident	dnumero
1163	Claudia	Morais	Rua A – 1 --SP	12/08/1974	5.558,00	F	NULL	4
1164	Jorge	Vila Verde	Rua M – 25 -- SP	29/01/1986	1.550,00	M	1163	1
1165	Moacir	Junqueira	Rua F – 57 -- SP	08/11/1981	1.550,00	M	1164	1
1166	Patrícia	Sorte	Rua F –22 -- SP	22/06/1979	2.200,00	F	1163	4
1167	Caio	Brotas	Rua I –100 -- SP	15/08/1977	3.400,00	M	1163	3

- **EXEMPLO:**

- select \* from FUNCIONARIO
- select NOME, DNUMERO from FUNCIONARIO

# FROM < lista de tabelas >

- As consultas simples utilizam dados de apenas uma tabela, entretanto, a grande maioria das consultas em bancos de dados relacionais exigirá consultas que juntem dados de mais de uma tabela.
- EXEMPLO:
  - select \* from FUNCIONARIO
  - select \* from FUNCIONARIO, DEPENDENTE

Nas aulas posteriores aprenderemos exatamente como trabalhar com mais de uma tabela usando os recursos das chaves estrangeiras.

# WHERE <condição>

- A cláusula WHERE permite criar um filtro, determinando o conjunto de registros que serão recuperados. A condição é sempre satisfeita por uma operação lógica (booleana).
- Há inúmeras possibilidades de condições que podem ser determinadas na cláusula WHERE.
- **Lembre-se sempre** que tem um papel muito representativo no seu comando SELECT.
- O campo a ser utilizado na cláusula Where não precisa ser listado na seleção.

ident	nome	sobrenome	endereco	dtnasc	salario	sexo	supident	dnumero
1163	Claudia	Morais	Rua A – 1 --SP	12/08/1974	5.558,00	F	NULL	4
1164	Jorge	Vila Verde	Rua M – 25 -- SP	29/01/1986	1.550,00	M	1163	1
1165	Moacir	Junqueira	Rua F – 57 -- SP	08/11/1981	1.550,00	M	1164	1
1166	Patrícia	Sorte	Rua F –22 -- SP	22/06/1979	2.200,00	F	1163	4
1167	Caio	Brotas	Rua I –100 -- SP	15/08/1977	3.400,00	M	1163	3

## EXEMPLO:

- select \* from FUNCIONARIO WHERE sexo='F'
- select \* from FUNCIONARIO WHERE salario>3000
- select \* from FUNCIONARIO WHERE dnumero<>1 AND salario<3000
- select \* from FUNCIONARIO WHERE dnumero>1 OR sexo='M'



# ORDER BY <ordem>

- A cláusula **ORDER BY** indica qual será o campo utilizado para ordenação dos resultados.
- Assim como na cláusula WHERE, o campo utilizado na ordenação não precisa estar contemplado na seleção.
- A ordenação é padronizada na maioria dos bancos de dados como ascendente. É possível usar a palavra DESC para que seja realizada de forma descendente. A palavra ASC garante/força que seja de forma ascendente. A ordem pode ser determinada por um ou mais campos.

ident	nome	sobrenome	endereco	dtnasc	salario	sexo	supident	dnumero
1163	Claudia	Morais	Rua A – 1 --SP	12/08/1974	5.558,00	F	NULL	4
1164	Jorge	Vila Verde	Rua M – 25 -- SP	29/01/1986	1.550,00	M	1163	1
1165	Moacir	Junqueira	Rua F – 57 -- SP	08/11/1981	1.550,00	M	1164	1
1166	Patrícia	Sorte	Rua F –22 -- SP	22/06/1979	2.200,00	F	1163	4
1167	Caio	Brotas	Rua I –100 -- SP	15/08/1977	3.400,00	M	1163	3

## EXEMPLO:

- select \* from FUNCIONARIO ORDER BY NOME
- select \* from FUNCIONARIO ORDER BY SOBRENOME DESC
- select \* from FUNCIONARIO ORDER BY DNUMERO DESC, SOBRENOME ASC
- select \* from FUNCIONARIO ORDER BY 2

# EXEMPLO

Liste o nome e salário dos funcionários do departamento 4, ordenados por data de nascimento.

```
SELECT NOME, SALARIO FROM FUNCIONARIO WHERE  
DNUMERO=4 ORDER BY DTNASC
```

nome	salario
Claudia	5.558,00
Patrícia	2.200,00

# SQL – CLÁUSULA DISTINCT

- A cláusula **DISTINCT** determina que o resultado da seleção não terá registros repetidos.
- Observe que ela é aplicada no resultado da operação select e não da tabela de origem.

```
SELECT DISTINCT dnumero  
FROM FUNCIONARIO;
```

```
SELECT DISTINCT dtnasc, nome  
FROM FUNCIONARIO;
```

# SQL – ALTERAR NOME DA COLUNA

- A cláusula **AS** possibilita a alteração do nome da coluna no resultado da seleção.
- Observe que ela não altera a estrutura da tabela de origem.

```
SELECT dnumero AS departamento  
FROM FUNCIONARIO;
```

```
SELECT nome, dtnasc AS nascimento  
FROM FUNCIONARIO;
```

# EXEMPLO

Liste o número dos departamentos que têm funcionários na empresa.

```
SELECT DISTINCT DNUMERO AS DEPARTAMENTO FROM  
FUNCIONARIO ORDER BY DNUMERO
```

departamento
1
3
4

# WHERE COM LIKE

- Um recurso muito importante e interessante que pode ser utilizado na cláusula where é o LIKE.
- O LIKE possibilita o filtro com recurso de busca de parte do texto.

```
SELECT * FROM funcionario  
WHERE nome LIKE 'José%'
```

```
SELECT * FROM funcionario  
WHERE nome LIKE '%José%'
```

# ALTERANDO VALORES

- É possível alterar os valores do resultados a partir dos dados originais. Os dados originais não são alterados, o resultado é uma projeção.

```
SELECT nome, salario * 1.3 FROM funcionario;
```

```
SELECT (1+2)*3 ;
```



# EXEMPLO

- Recuperar o nome, sobrenome, salário com a projeção de um aumento de salário de 15% para os funcionários, apresentando também o salário atual.
- Ordene por salário.

```
SELECT nome, sobrenome, salario, salario * 1.15 as novo_salario  
FROM funcionário ORDER BY salario;
```



ident	nome	sobrenome	endereco	dtnasc	salario	sexo	supident	dnumero
1163	Claudia	Morais	Rua A – 1 --SP	12/08/1974	5.558,00	F	NULL	4
1164	Jorge	Vila Verde	Rua M – 25 -- SP	29/01/1986	1.550,00	M	1163	1
1165	Moacir	Junqueira	Rua F – 57 -- SP	08/11/1981	1.550,00	M	1164	1
1166	Patrícia	Sorte	Rua F –22 -- SP	22/06/1979	2.200,00	F	1163	4
1167	Caio	Brotas	Rua I –100 -- SP	15/08/1977	3.400,00	M	1163	3

nome	sobrenome	salario	novo_salario
Jorge	Vila Verde	1.550,00	1.782,50
Moacir	Junqueira	1.550,00	1.782,50
Patrícia	Sorte	2.200,00	2.530,00
Caio	Brotas	3.400,00	3.910,00
Claudia	Morais	5.558,00	6.391,70

# **BANCO DE DADOS**

**Linguagem SQL**

**Introdução à consulta - SELECT**