

FACEC

F a c u l d a d e

Cont. Linguagem SQL

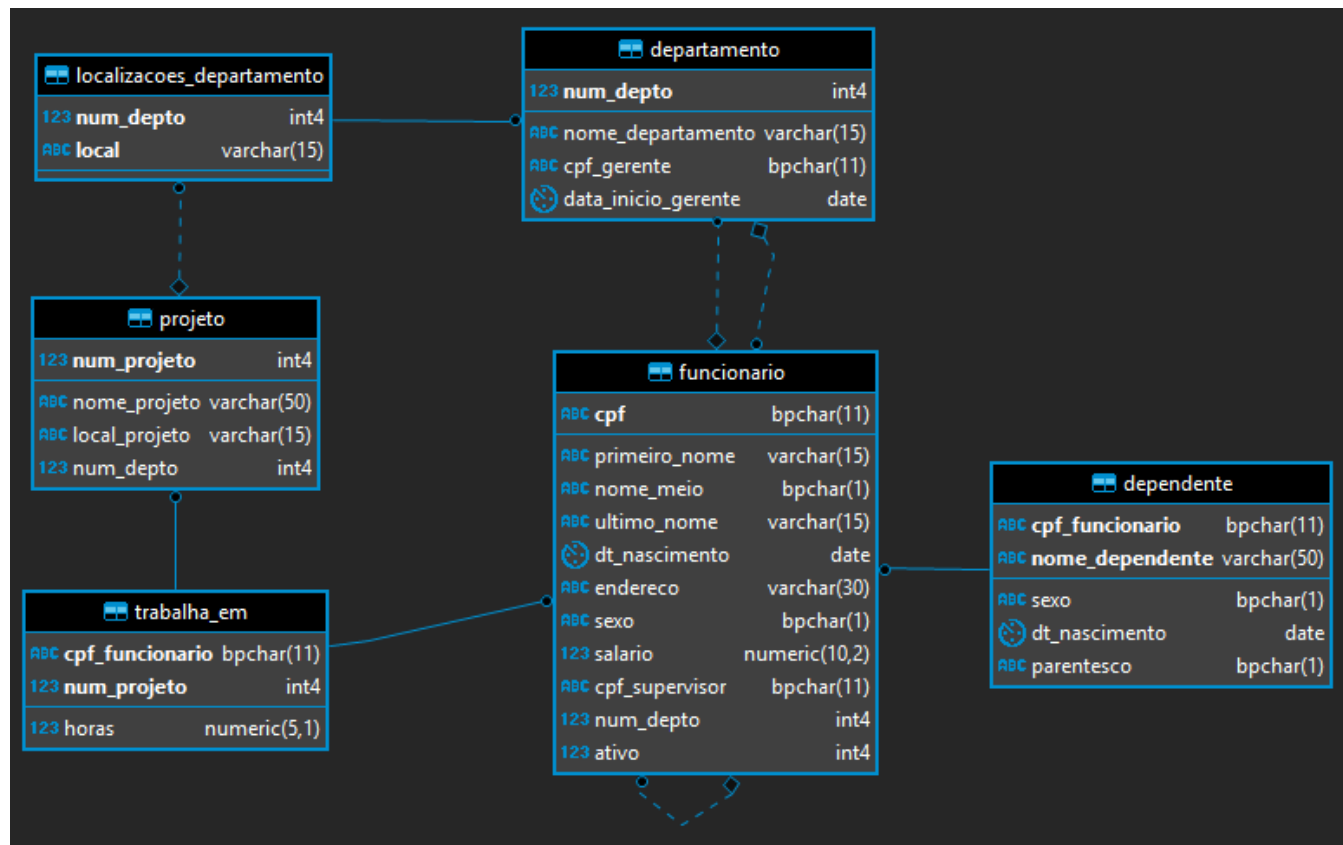
Professor: Yuri Ferreira

- Revisão aula anterior:
 - Diagrama de Esquema Empresa;
 - Operações de inserção de dados;
 - Introdução ao comando *Select*;
 - *Select*: operadores de comparação Lógicos;

➤ Conteúdo:

- Consultas simples e com produto cartesiano;
- Renomeação de Tabelas e Atributos;
- Cláusula **Where** condições:
 - seleção;
 - junção;
- Cláusula Where não especificada;
- Uso do Asterístico (*);
- Combinações de cadeia de caracteres;
- Operadores aritméticos;
- Operador de comparação auxiliar;
- Concatenação de atributos;

- Para o Diagrama de Esquema **Empresa**, criar as tabela e inserir as informações.



➤ Estrutura **SELECT-FROM-WHERE**:

```
SELECT <lista de atributos>  
FROM <lista tabelas>  
WHERE <condicao>;
```

- Lista de atributos: nome dos atributos cujos valores devem ser recuperados da consulta;
- Lista de tabelas: tabelas exigidas para processar a consulta;
- Condição: expressão condicional (booleana) que identifica as tuplas a serem recuperadas pela consulta;

➤ **WHERE** - Operadores de Comparação Lógicos:

- São utilizados para comparar valores de atributos com constantes literais ou outros atributos;

= , < , <= , > , >= , <> , not;

- Ex: salario > 1500;
- nome = 'João';
- dt_nascimento = '2000-01-01';
- Salario >= 1500 and Salario <= 2000;
- Nome_meio not null;

➤ **SELECT** simples:

```
select  
dt_nascimento, endereco  
from funcionario  
where primeiro_nome='João'  
and nome_meio='B'  
and ultimo_nome='Silva'
```


➤ SELECT

- Utilizando apelidos “as” (*alias*) em tabelas;
- Renomeação de atributos;
- E produto cartesiano;

```
-- Lista de Funcionários e  
-- seus Departamentos  
select f.primeiro_nome,  
d.nome_departamento  
from funcionario as f  
      ,departamento as d  
where f.num_depto=d.num_depto;
```

```
-- Lista Departamentos e Localizações  
select d.num_depto,  
d.nome_departamento,  
l."local"  
from departamento as d  
      ,localizacoes_departamento as l  
where d.num_depto = l.num_depto
```

➤ Cláusula **WHERE**

- Em geral uma consulta SQL pode ter várias condições de **seleção / junção**;

➤ Cláusula **WHERE**

- Condição de **seleção** é utilizada para filtrar atributos de uma tabela com argumentos literais; Ex: primeiro_nome = 'Paulo';

```
-- Lista de Funcionarios com nome Paulo
-- contendo o funcionario, departamento e
-- o gerente
select
f.primeiro_nome,f.ultimo_nome,
d.nome_departamento,
s.primeiro_nome as gerente
from funcionario as f, departamento as d,
     funcionario as s
where f.primeiro_nome = 'Paulo'
     and f.num_depto = d.num_depto
     and d.cpf_gerente = s.cpf
```

➤ Cláusula **WHERE**

- Condição de **junção** (ou **join**) é utilizada para combinar duas tabelas através de suas chaves; Ex: f.num_depto = d.num_depto!

```
-- Lista de Funcionarios com nome Paulo
-- contendo o funcionario, departamento e
-- o gerente
select
f.primeiro_nome,f.ultimo_nome,
d.nome_departamento,
s.primeiro_nome as gerente
from funcionario as f, departamento as d,
     funcionario as s
where f.primeiro_nome = 'Paulo'
     and f.num_depto = d.num_depto
     and d.cpf_gerente = s.cpf
```

- Cláusula **WHERE** não especificada:
 - Como a cláusula Where não é obrigatória, ela pode ser ou não especificada;
 - Porém quando se trabalha com produto cartesiano entre tabelas (ou JOINS), é muito importante especificar, para não ter tabelas incorretas e/ou muito grandes;

```
-- Produto Cartesiano de todas  
-- notas fiscais para todos itens  
-- Ex: 1M * 1M = 1 Trilhão de linhas  
select *  
from NotaFiscal as N, ItensNF as I  
-- where N.cod_nf = I.cod_nf
```

➤ Utilização do asterisco (*)

- Para selecionarmos todos os atributos de uma tabela, não é necessário especifica-los explicitamente, pode ser utilizado o *;
- Pode ser utilizado o **nome_da_tabela.*** para referenciar os atributos de apenas uma tabela em específico;

```
select *  
from funcionario, departamento
```

➤ Combinação de padrões de subcadeias de caracteres

➤ **LIKE**

➤ (%): Substitui uma cadeia de caracteres;

➤ (_): Substitui um único caractere;

```
-- Liste os funcionarios que  
-- moram em uma Rua que inicia  
-- com a letra S  
select * from funcionario  
where endereco like 'Rua S%'
```

```
-- Liste os funcionarios que  
-- terminam o CPF com 1  
select * from funcionario as f  
where f.cpf like '_____1'
```

Obs: No **PostgreSQL** temos a opção de utilizar **ilike**, que é case-insensitive (não importa se o caractere é maiúsculo ou minúsculo);

➤ Operadores aritméticos

➤ (+) (-) (/) (*)

```
-- Listar os Funcionários  
-- com 10% de aumento de salário  
select primeiro_nome,  
salario * 1.1 as "Salario + 10%"  
from funcionario
```

```
-- Acrescimo na data de nascimento  
-- de um funcionário  
select dt_nascimento,  
dt_nascimento  
+ 1 as dt_nasc_1d,  
dt_nascimento  
+ interval '1 day 01:00:00' dt_nasc_1d_1h  
from funcionario  
where cpf='98765432112'
```


➤ Operador de comparação **BETWEEN**

```
select * from funcionario as f
where f.salario between 2000 and 3000
and dt_nascimento between '1990-01-01' and '2000-01-01'
```

➤ Concatenação de caracteres (||)

```
-- Concatena os nomes com  
-- espaços em branco  
select primeiro_nome || ' '  
|| nome_meio || ' '  
|| ultimo_nome  
from funcionario
```

➤ Referências:

- SILBERSCHATZ, A.; KORTH, F.; SUDARSHA, S. Database System Concepts. 6. ed. Nova York: MC Graw Hill, 2011.
- ELMASRI, R.; NAVATHE B. Sistemas de banco de dados. 6. Ed. São Paulo, SP: Pearson Addison-Wesley, 2011.