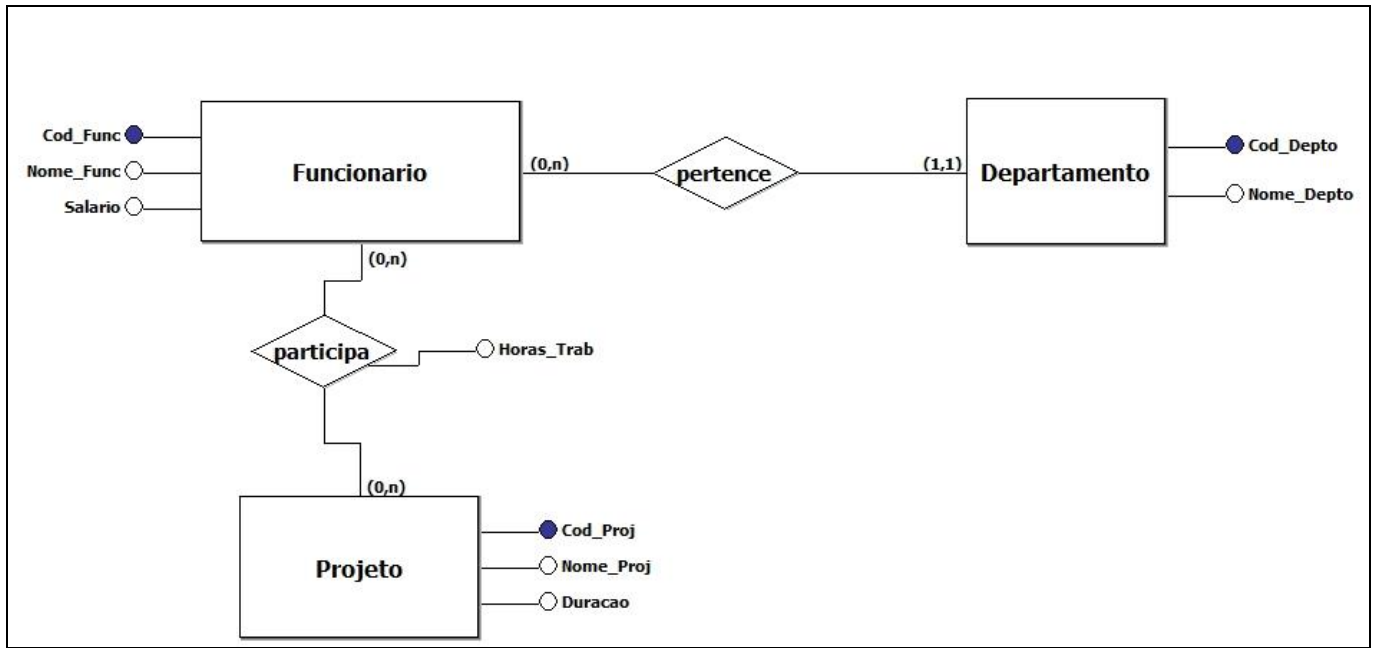


### Exercícios da Aula 19 – Linguagem PL/SQL

Considere o seguinte Banco de Dados “Projeto” para responder as questões deste exercício, utilizando a Linguagem SQL (o script de criação deste banco de dados se encontra no final deste arquivo).

#### MER



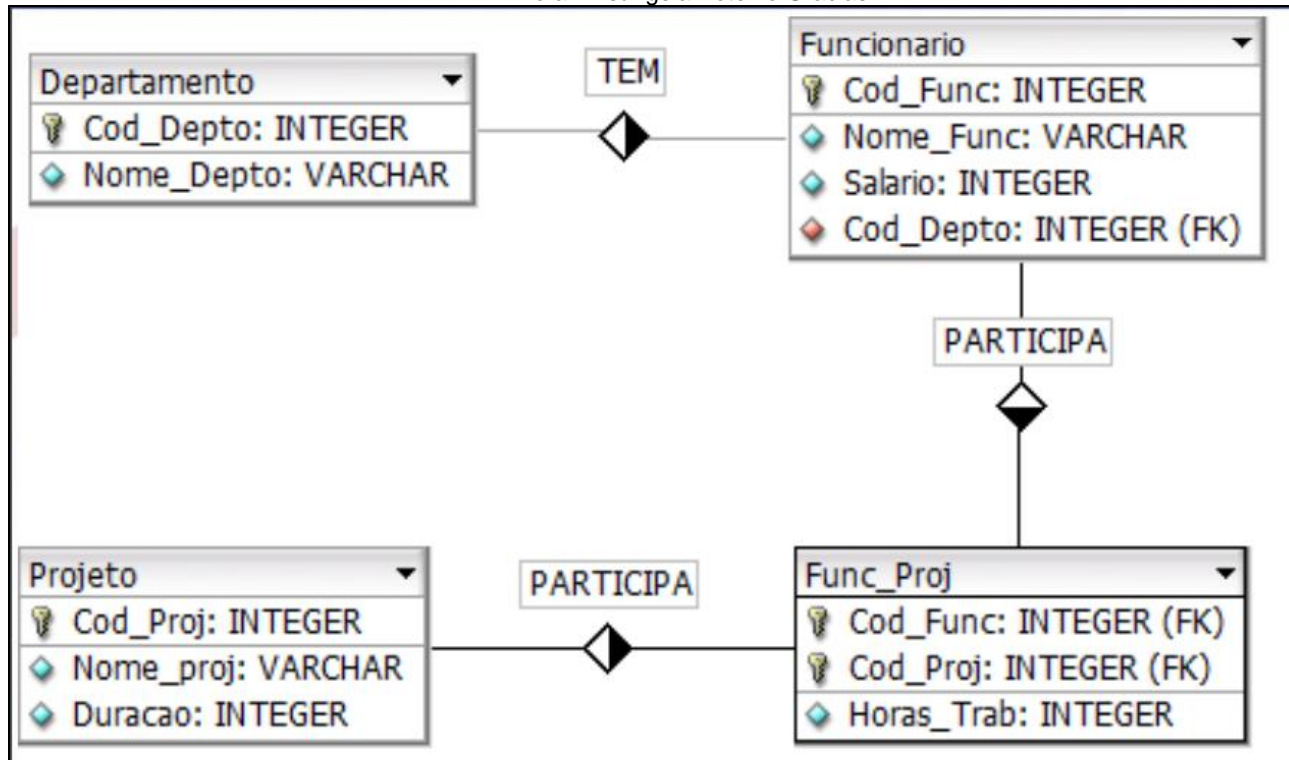
#### Modelo Relacional:

Departamento = {Cod\_Depto, Nome\_Depto}

Funcionário = {Cod\_Func, Nome\_Func, Salario, Cod\_Depto} – Cod\_depto é chave estrangeira que referencia o atributo Cod\_depto da tabela Departamento.

Projeto = {Cod\_Proj, Nome\_Proj, Duracao}

Func\_Proj = {Cod\_Func, Cod\_Proj, Horas\_Trab} - Cod\_Func é chave estrangeira que referencia o atributo Cod\_Func da tabela Funcionário e Cod\_Proj é chave estrangeira que referencia o atributo Cod\_Proj da tabela Projeto.



### Tabelas com Dados

**Departamento**

Cod_Depto	Nome_Depto
1	Marketing
2	Vendas
3	Dados
4	Pesquisa

**Funcionario**

Cod_Func	Nome_Func	Salario	Cod_Depto
101	Joao da Silva Santos	2000	2
102	Mario Souza	1500	1
103	Sergio Silva Santos	2400	2
104	Maria Castro	1200	1
105	Marcio Silva Santana	1400	4

**Projeto**

Cod_Proj	Nome_Proj	Duracao
1001	SistemaA	2
1002	SistemaB	6
1003	SistemaX	4

**Func\_Proj**

Cod_Func	Cod_Proj	Horas_Trab
101	1001	24
101	1002	160
102	1001	56
102	1003	45
103	1001	86
103	1003	64
104	1001	46
105	1001	84
105	1002	86

1) Desenvolva um bloco PL/SQL que faça a **atualização do salário do funcionário de código 103, observando os seguintes critérios:**

- Se o salário deste funcionário de código 103 for menor que 1500, então aumente o salário dele em 30%;
- Senão, se o salário dele for menor ou igual a 2000, aumente em 20%;
- Senão, aumente somente em 10%;
- No final, mostre na tela o valor do novo salário do funcionário 103.

2) **(utilize a estrutura WHILE)** Desenvolva um bloco PL/SQL que **aumente o salário do funcionário de código 101, de cada vez, em 10%, enquanto o salário dele for menor ou igual a 10000.**

A cada vez que o salário do funcionário de código 101 for aumentado, insira uma nova linha na tabela Historico\_Salario (cujo script de criação se encontra logo abaixo).

No final, **mostre na tela o valor final do novo salário do funcionário 101**, após todas as atualizações feitas.

```
CREATE TABLE Historico_Salario
(Cod_Func INTEGER,
 Salario_Novo INTEGER,
 Dt_Atualizacao DATE,
 PRIMARY KEY(Cod_Func, Salario_Novo)
);
```

3) Desenvolva um bloco PL/SQL que **mostre na tela o quanto o funcionário de nome 'Mario Souza' irá receber pelo número de horas que ele trabalhou nos projetos**, considerando que **cada hora trabalhada dele é 100.**

**Mostre na tela**, também, a **classificação deste funcionário**, de acordo com o número total de horas que ele trabalhou nos projetos, obedecendo aos seguintes critérios:

- se o número total de horas for maior ou igual a 200, então a classificação será "ótima participação";
- senão, se for maior ou igual a 100, então sua classificação será "boa participação";
- senão, sua classificação será "participação normal".

Universidade Presbiteriana Mackenzie  
Banco de Dados  
Profa. Elisângela Botelho Gracias

-- SCRIPT COMPLETO DO BANCO

-- eliminação das tabelas

```
DROP TABLE Func_Proj;
DROP TABLE Projeto;
DROP TABLE Funcionario;
DROP TABLE Departamento;
```

-- criação das tabelas

```
CREATE TABLE Departamento
(Cod_Depto INTEGER,
 Nome_Depto VARCHAR(20) NOT NULL,
 PRIMARY KEY (Cod_Depto)
);
```

```
CREATE TABLE Funcionario
(Cod_Func INTEGER,
 Nome_Func VARCHAR(20) NOT NULL,
 Salario INTEGER,
 Cod_Depto INTEGER,
 PRIMARY KEY (Cod_Func),
 FOREIGN KEY (Cod_Depto) REFERENCES Departamento (Cod_Depto)
);
```

```
CREATE TABLE Projeto
(Cod_Proj INTEGER,
 Nome_Proj VARCHAR(20) NOT NULL,
 Duracao INTEGER,
 PRIMARY KEY (Cod_Proj)
);
```

```
CREATE TABLE Func_Proj
(Cod_Func INTEGER,
 Cod_Proj INTEGER,
 Horas_Trab INTEGER,
 PRIMARY KEY (Cod_Func, Cod_Proj),
 FOREIGN KEY (Cod_Func) REFERENCES Funcionario (Cod_Func),
 FOREIGN KEY (Cod_Proj) REFERENCES Projeto (Cod_Proj)
);
```

-- Inserção de dados na tabela Departamento

```
INSERT
INTO Departamento (Cod_Depto, Nome_Depto)
VALUES (1, 'Marketing');
```

```
INSERT
INTO Departamento (Cod_Depto, Nome_Depto)
VALUES (2, 'Vendas');
```

```
INSERT
INTO Departamento (Cod_Depto, Nome_Depto)
VALUES (3, 'Dados');
```

```
INSERT
INTO Departamento (Cod_Depto, Nome_Depto)
VALUES (4, 'Pesquisa');
```

Universidade Presbiteriana Mackenzie  
Banco de Dados  
Profa. Elisângela Botelho Gracias

-- Inserção de dados na tabela Funcionario

```
INSERT
INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto)
VALUES (101, 'Joao da Silva Santos', 2000, 2);
```

```
INSERT
INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto)
VALUES (102, 'Mario Souza', 1500, 1);
```

```
INSERT
INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto)
VALUES (103, 'Sergio Silva Santos', 2400, 2);
```

```
INSERT
INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto)
VALUES (104, 'Maria Castro', 1200, 1);
```

```
INSERT
INTO Funcionario (Cod_Func, Nome_Func, Salario, Cod_Depto)
VALUES (105, 'Marcio Silva Santana', 1400, 4);
```

-- Inserção de dados na tabela Projeto

```
INSERT
INTO Projeto (Cod_Proj, Nome_Proj, Duracao)
VALUES (1001, 'SistemaA', 2);
```

```
INSERT
INTO Projeto (Cod_Proj, Nome_Proj, Duracao)
VALUES (1002, 'SistemaB', 6);
```

```
INSERT
INTO Projeto (Cod_Proj, Nome_Proj, Duracao)
VALUES (1003, 'SistemaX', 4);
```

-- Inserção de dados na tabela Func\_Proj

```
INSERT
INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab)
VALUES (101, 1001, 24);
```

```
INSERT
INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab)
VALUES (101, 1002, 160);
```

```
INSERT
INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab)
VALUES (102, 1001, 56);
```

```
INSERT
INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab)
VALUES (102, 1003, 45);
```

```
INSERT
INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab)
VALUES (103, 1001, 86);
```

```
INSERT
INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab)
VALUES (103, 1003, 64);
```

```
INSERT
INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab)
VALUES (104, 1001, 46);
```

Universidade Presbiteriana Mackenzie  
Banco de Dados  
Profa. Elisângela Botelho Gracias

```
INSERT  
INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab)  
VALUES (105, 1001, 84);
```

```
INSERT  
INTO Func_Proj (Cod_Func, Cod_Proj, Horas_Trab)  
VALUES (105, 1002, 86);
```