

Exercícios da Aula 11 – Inserção, Atualização e Eliminação de dados

Considere o seguinte Banco de Dados “Projeto” para responder as questões deste exercício, utilizando a Linguagem SQL.

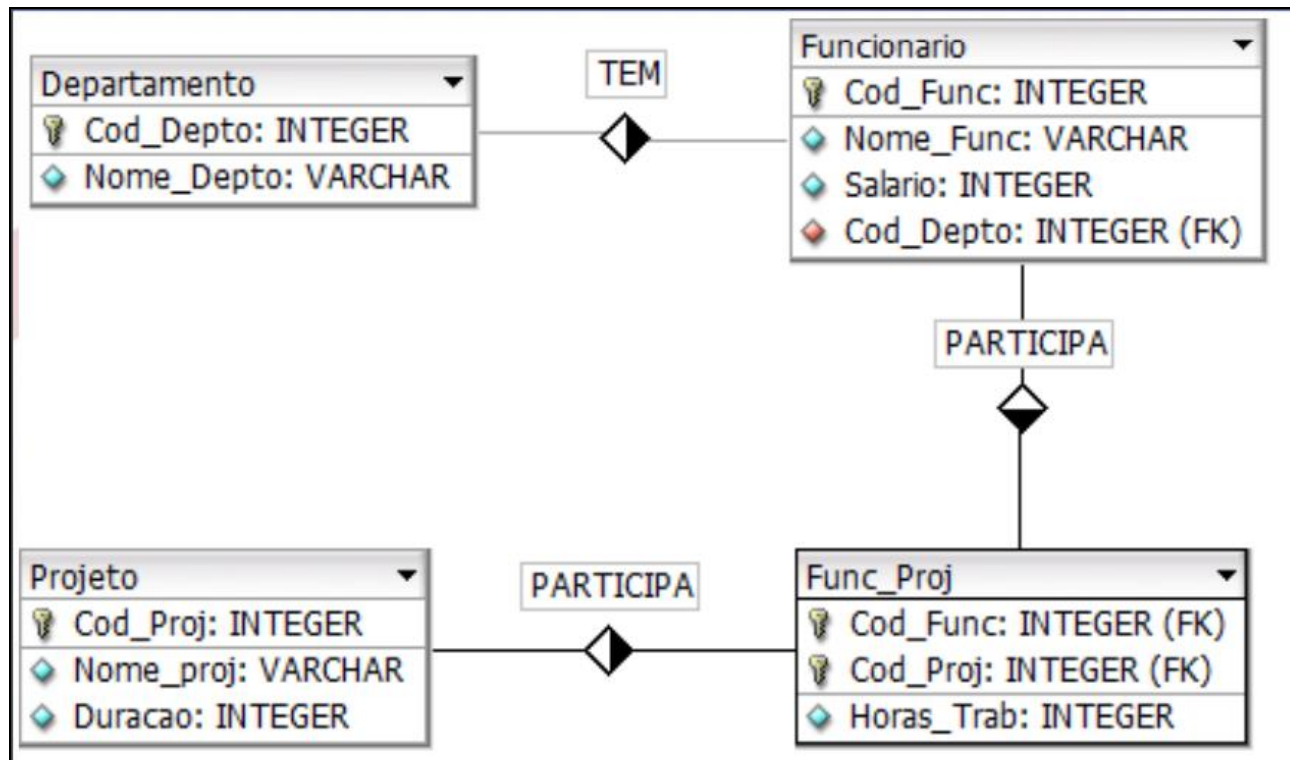
Modelo Relacional:

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.



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. (INSERT)** Insira todos os dados nas tabelas do banco de dados mostrado, a seguir (o script de criação das tabelas se encontra no final deste arquivo).
- 2. (INSERT)** Faça um teste de **PK (primary key)** em uma das tabelas que você já inseriu dados. Ou seja, viole um valor de PK e veja o que acontece (insira uma nova linha em qualquer tabela).
- 3. (INSERT ou UPDATE ou DELETE)** Faça um teste de **FK (foreign key)** em uma das tabelas que você já inseriu dados. Ou seja, viole um valor de FK e veja o que acontece (por exemplo, insira um novo funcionário em um departamento que não existe).
- 4. (UPDATE)** Altere o nome do departamento de 'Dados' para 'Banco de Dados'.
- 5. (UPDATE)** Aumente o número de horas trabalhadas em 20% daqueles funcionários que trabalharam mais de 100 horas no projeto de código 1002.
- 6. (UPDATE)** Dobre o salário daqueles funcionários que são do departamento de código 2 e ganham um salário diferente de 2000, ou então, daqueles funcionários que são do departamento de código 1 e ganham um salário menor que 1400.

7. (DELETE) A funcionária 'Maria Castro', que possui código de funcionário igual a 104, saiu da empresa, logo, seus dados deverão ser eliminados do banco de dados. Elimine todas as suas informações, respeitando as restrições de integridade.

8. (UPDATE e DELETE) O departamento de 'Pesquisa', cujo código é 4, deverá ser eliminado do banco de dados. Mas os funcionários ligados a ele não serão demitidos, e sim, realocados a quaisquer outros departamentos da empresa. Elimine este departamento e faça a realocação dos funcionários a outros departamentos que desejar.

```
-- Script de criação das tabelas

-- eliminação das tabelas
DROP TABLE Func_Proj CASCADE CONSTRAINT;
DROP TABLE Projeto CASCADE CONSTRAINT;
DROP TABLE Funcionario CASCADE CONSTRAINT;
DROP TABLE Departamento CASCADE CONSTRAINT;

-- 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)
);
```