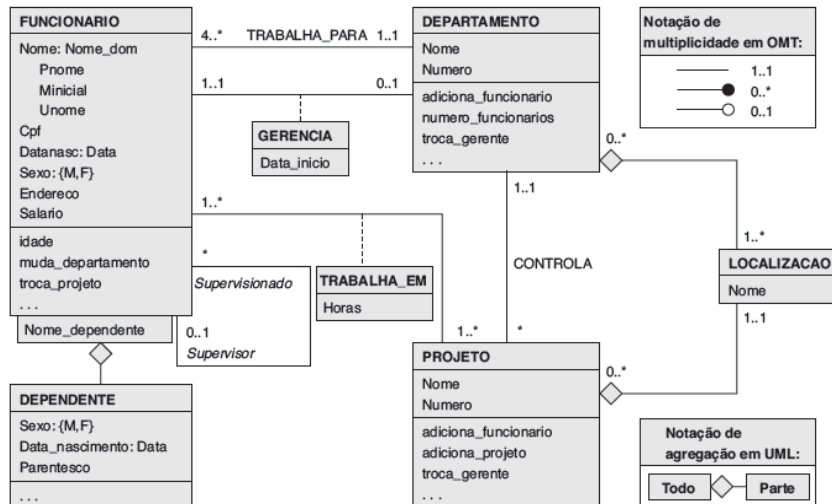


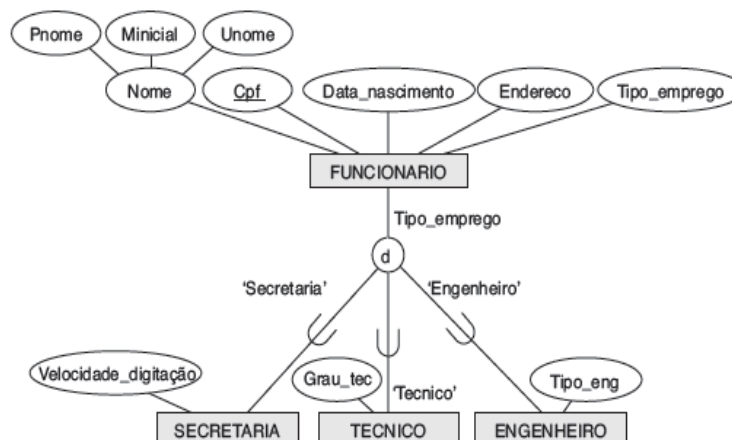
## Exercícios – Bancos de Dados Orientados a Objetos

1. Considere o diagrama de classes abaixo:



Mostre o código SQL para criação desse banco de dados em um SGBD objeto-relacional. Use ao máximo os conceitos de banco de dados de objeto.

- Especifique as consultas da Lista de Exercícios sobre SQL para o banco de dados da questão anterior.
- Mapeie o diagrama EER abaixo para um banco de dados de objetos e mostre o código SQL para sua criação em um SGBD objeto-relacional.



1. Resposta:

```
CREATE TYPE TipoFuncionario AS (  
    nome ROW (pNome VARCHAR(20),  
              mInicial CHAR(1),  
              uNome VARCHAR(20)),  
    cpf CHAR(11),  
    dataNasc DATE,  
    sexo ENUM('M','F'),  
    endereco VARCHAR(100),  
    salario NUMERIC(10,2),  
    supervisor REF (TipoFuncionario) SCOPE (Funcionario),  
    depto REF (TipoDepartamento) SCOPE (Departamento),  
    dependentes REF (TipoDependente) SET SCOPE (Dependente),  
    projetos REF (TipoTrabalha) SET SCOPE (Trabalha),  
    INSTANTIABLE  
    NOT FINAL  
    REF IS SYSTEM GENERATED  
    INSTANCE METHOD idade() RETURNS INTEGER;  
    INSTANCE METHOD mudaDepto (TipoDepartamento novoDepto) RETURNS BOOLEAN;  
    INSTANCE METHOD adicionaProjeto (TipoTrabalha proj, FLOAT numHoras) RETURNS  
        BOOLEAN;  
);  
  
CREATE TYPE TipoLocalizacao AS (  
    nomeLocal VARCHAR(50)  
);  
  
CREATE TYPE TipoDepartamento AS (  
    nomeDepto VARCHAR(50),  
    numDepto INTEGER,  
    gerente ROW (funcGerente REF (TipoFuncionario) SCOPE (Funcionario),  
                 dataInicio DATE),  
    localizacoes TipoLocalizacao SET,  
    funcionarios REF (TipoFuncionario) SET SCOPE (Funcionario),  
    projetos REF (TipoProjeto) SET SCOPE (Projeto),  
    INSTANTIABLE  
    NOT FINAL  
    REF IS SYSTEM GENERATED  
    INSTANCE METHOD adicionaFuncionario (TipoFuncionario func) RETURNS BOOLEAN;  
    INSTANCE METHOD qtdeFuncs () RETURNS INTEGER;  
    INSTANCE METHOD trocaGerente (TipoFuncionario novoGerente) RETURNS BOOLEAN;  
);
```

```
CREATE TYPE TipoProjeto AS (  
    nomeProj VARCHAR(50),  
    numProj INTEGER,  
    localizacao TipoLocalizacao,  
    depto REF (TipoDepartamento) SCOPE (Departamento),  
    funcionarios REF (TipoTrabalha) SET SCOPE (Trabalha),  
    INSTANTIABLE  
    NOT FINAL  
    REF IS SYSTEM GENERATED  
    INSTANCE METHOD adicionaFuncionario (TipoFuncionario func, FLOAT numHoras)  
        RETURNS BOOLEAN;  
);
```

```
CREATE TYPE TipoTrabalha AS (  
    numHoras NUMERIC(3,1),  
    func REF (TipoFuncionario) SCOPE (Funcionario),  
    proj REF (TipoProjeto) SCOPE (Projeto),  
    INSTANTIABLE  
    FINAL  
    REF IS SYSTEM GENERATED  
);
```

```
CREATE TYPE TipoDependente AS (  
    nomeDep VARCHAR(80),  
    sexo ENUM('M','F'),  
    dataNasc DATE,  
    parentesco VARCHAR(10),  
    func REF (TipoFuncionario) SCOPE (Funcionario),  
    INSTANTIABLE  
    NOT FINAL  
    REF IS SYSTEM GENERATED  
);
```

```
CREATE TABLE Funcionario OF TipoFuncionario  
    REF IS idFunc SYSTEM GENERATED;  
CREATE TABLE Departamento OF TipoDepartamento;  
CREATE TABLE Projeto OF TipoProjeto;  
CREATE TABLE Trabalha OF TipoTrabalha;  
CREATE TABLE Dependente OF TipoDependente;  
    REF IS idDep SYSTEM GENERATED;
```

## 2. Resposta:

(1) Selecione o endereço e o salário do funcionário de nome 'Luciana S. Santos'.

```
SELECT F.endereco, F.salario  
FROM Funcionario F  
WHERE F.nome.pNome = 'Luciana' AND F.nome.mInicial = 'S' and F.nome.uNome = 'Santos';
```

(2) Selecione o nome e o salário dos funcionários que nasceram entre os anos de 1960 e 1969, inclusive, do sexo feminino e que ganham menos de 1000.

```
SELECT F.nome.pNome, F.nome.mInicial, F.nome.uNome, F.salario  
FROM Funcionario F  
WHERE F.dataNasc BETWEEN '01/01/1960' AND '31/12/1969' and F.sexo = 'F' and  
F.salario < 1000;
```

(3) Selecione o nome dos dependentes do funcionário de nome 'João B. Silva'.

```
SELECT D.nomeDep  
FROM Dependente D  
WHERE D.func->nome.pNome = 'João' AND D.func->nome.mInicial = 'B' and  
D.func->nome.uNome = 'Silva';
```

(4) Selecione o nome dos projetos que o funcionário de nome 'Frank T. Santos' trabalha.

```
SELECT T.proj->nomeProj  
FROM Trabalha T  
WHERE T.func->nome.pNome = 'Frank' AND T.func->nome.mInicial = 'T' and  
T.func->nome.uNome = 'Santos';
```

(5) Selecione o nome dos funcionários que trabalham em projetos controlados pelo departamento de nome 'Construção'.

```
SELECT T.func->nome.pNome, T.func->nome.mInicial, T.func->nome.uNome  
FROM Trabalha T  
WHERE T.proj->depto->nomeDepto = 'Construção';
```

(6) Selecione o nome dos funcionários supervisionados pelo funcionário de nome 'Frank T. Santos'.

```
SELECT F.nome.pNome, F.nome.mInicial, F.nome.uNome  
FROM Funcionario F  
WHERE F.supervisor->nome.pNome = 'Frank' and F.supervisor->nome.mInicial = 'T' and  
F.supervisor->nome.uNome = 'Santos';
```

(7) Selecione o nome e endereço dos funcionários que não tem nenhum dependente.

```
SELECT F.nome.pNome, F.nome.mInicial, F.nome.uNome, F.endereco  
FROM Funcionario F  
WHERE CARDINALITY (F.dependentes) = 0;
```

3. Resposta:

```
CREATE TYPE TipoFuncionario AS (  
    nome ROW (pNome VARCHAR(20),  
              mInicial CHAR(1),  
              uNome VARCHAR(20)),  
    cpf CHAR(11),  
    dataNasc DATE,  
    endereco VARCHAR(100),  
    tipoEmprego CHAR(1),  
    INSTANTIABLE  
    NOT FINAL  
    REF IS SYSTEM GENERATED  
);
```

```
CREATE TYPE TipoSecretaria UNDER TipoFuncionario AS (  
    velocidadeDigitacao INTEGER,  
    INSTANTIABLE  
    NOT FINAL  
);
```

```
CREATE TYPE TipoTecnico UNDER TipoFuncionario AS (  
    grauTecnico VARCHAR(30),  
    INSTANTIABLE  
    NOT FINAL  
);
```

```
CREATE TYPE TipoEngenheiro UNDER TipoFuncionario AS (  
    tipoEngenharia VARCHAR(30),  
    INSTANTIABLE  
    NOT FINAL  
);
```

```
CREATE TABLE Funcionario OF TipoFuncionario  
    REF IS idFunc SYSTEM GENERATED;  
CREATE TABLE Secretaria OF TipoSecretaria UNDER Funcionario;  
CREATE TABLE Tecnico OF TipoTecnico UNDER Funcionario;  
CREATE TABLE Engenheiro OF TipoEngenheiro UNDER Funcionario;
```