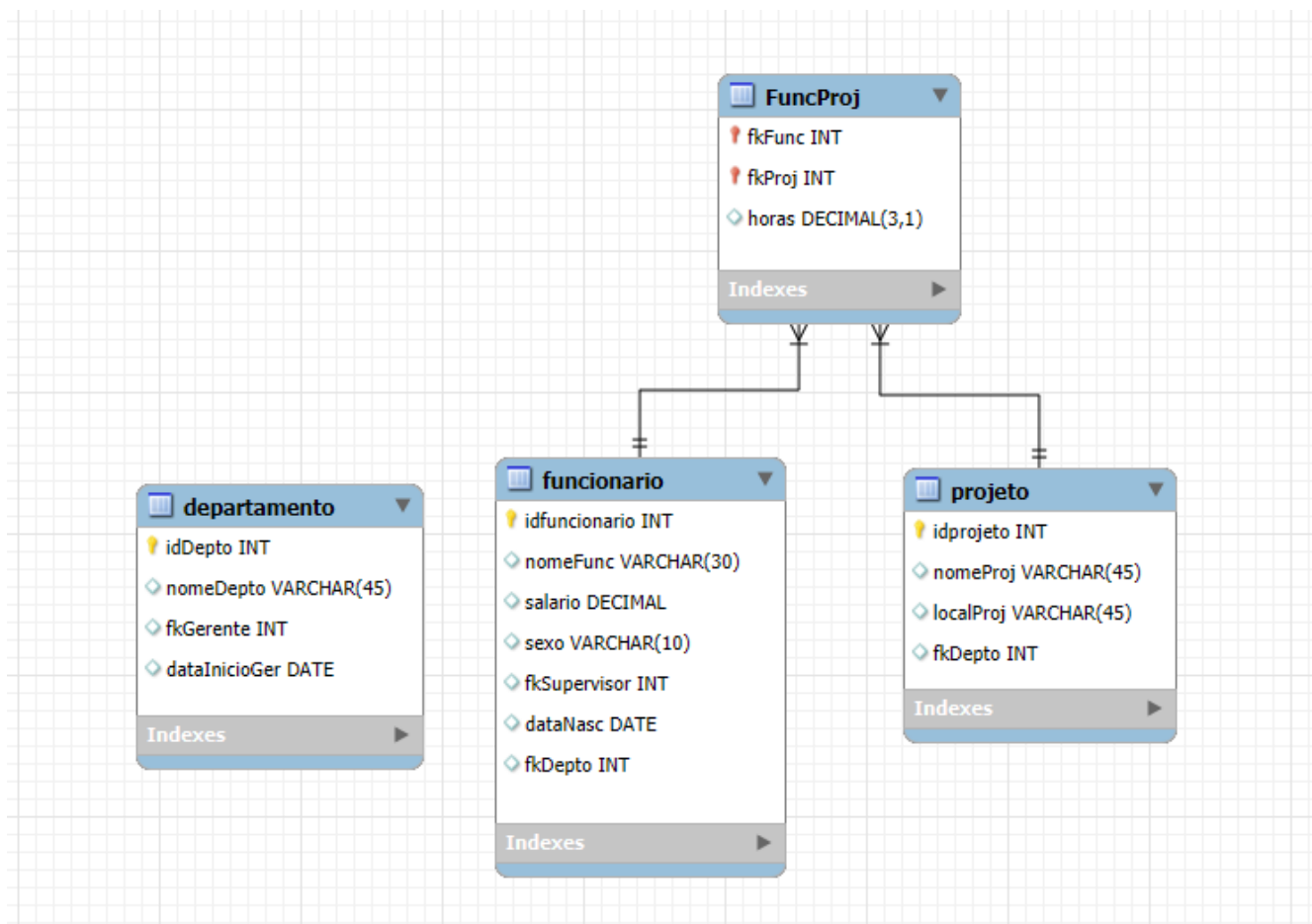


NOME	TURMA
Izael Reis de Oliveira Junior	1ADSB

Prática Desafio – BD



CREATE DATABASE Exercicio11;

USE Exercicio11;

CREATE table departamento (
idDeppto int primary key,
nomeDeppto varchar(45),
fkGerente int,
dataInicioGer date
);

CREATE TABLE funcionario (
idFunc int primary key,
nomeFunc varchar(30),
salario decimal,
sexo varchar(15),

```
fkSupervisor int,  
dataNasc date,  
fkDepto int  
);
```

```
CREATE TABLE projeto (  
idProj int primary key,  
nomeProj varchar(45),  
localProj varchar(45),  
fkDepto int  
);
```

```
CREATE TABLE FuncProj (  
fkFunc int,  
fkProj int,  
constraint pkFuncProj primary key (fkFunc, fkProj),  
horas decimal(3,1)  
);
```

```
INSERT INTO departamento VALUES  
(105, 'Pesquisa', 2, '2008-05-22'),  
(104, 'Administração', 7, '2015-01-01'),  
(101, 'Matriz', 8, '2001-06-19');
```

```
INSERT INTO funcionario VALUES  
(1, 'João Silva', 3500, 'm', 2, '1985-01-09', 105),  
(2, 'Fernando Wong', 4500, 'm', 8, '1975-12-08', 105),  
(3, 'Alice Sousa', 2500, 'f', 7, '1988-01-19', 104),  
(4, 'Janice Moraes', 4300, 'f', 8, '1970-06-20', 104),  
(5, 'Ronaldo Lima', 3800, 'm', 1, '1982-09-15', 105),  
(6, 'Joice Leite', 2500, 'f', 1, '1992-07-31', 105),  
(7, 'Antonio Pereira', 2500, 'm', 4, '1989-03-29', 104),  
(8, 'Juliano Brito', 5500, 'm', null, '1957-11-10', 101);
```

```
INSERT INTO projeto VALUES  
(1, 'Produto X', 'Santo André', 105),  
(2, 'Produto Y', 'Itu', 105),  
(3, 'Produto Z', 'São Paulo', 105),  
(10, 'Informatização', 'Mauá', 104),  
(20, 'Reorganização', 'São Paulo', 101),  
(30, 'Benefícios', 'Mauá', 104);
```

```
INSERT INTO FuncProj VALUES  
(1, 1, 32.5),
```

```
(1, 2, 7.5),  
(5, 3, 40.0),  
(6, 1, 20.0),  
(6, 2, 20.0),  
(2, 2, 10.0),  
(2, 3, 10.0),  
(2, 10, 10.0),  
(2, 20, 10.0),  
(3, 30, 30.0),  
(3, 10, 10.0),  
(7, 10, 35.0),  
(7, 30, 5.0),  
(4, 30, 20.0),  
(4, 20, 15.0),  
(8, 20, null);
```

```
SELECT * FROM departamento;  
SELECT * FROM funcionario;  
SELECT * FROM projeto;  
SELECT * FROM FuncProj;
```

```
ALTER TABLE departamento  
ADD CONSTRAINT fkGerenteDep FOREIGN KEY (fkGerente) REFERENCES funcionario (idFunc);
```

```
ALTER TABLE funcionario  
ADD CONSTRAINT fkSuperFunc FOREIGN KEY (fkSupervisor) REFERENCES funcionario (idFunc);
```

```
ALTER TABLE funcionario  
ADD CONSTRAINT fkDepFunc FOREIGN KEY (fkDepto) REFERENCES departamento (idDepto);
```

```
ALTER TABLE projeto  
ADD CONSTRAINT fkDepProj FOREIGN KEY (fkDepto) REFERENCES departamento (idDepto);
```

```
ALTER TABLE FuncProj  
ADD CONSTRAINT fkFuncProj FOREIGN KEY (fkFunc) REFERENCES funcionario (idFunc);
```

```
ALTER TABLE FuncProj  
ADD CONSTRAINT fkProjProj FOREIGN KEY (fkProj) REFERENCES projeto (idProj);
```

```
INSERT INTO funcionario VALUES  
(null, 'Cecília Ribeiro', 2800, 'f', null, '1980-04-05', 104);  
-- Conseguiu inserir? Por que?  
-- Não foi possível inserir porque a chave primária não pode ser nula
```

INSERT INTO funcionario VALUES

(3, 'Alice Sousa', 2800, 'f', 4, '1980-04-05', 104);

-- Conseguiu inserir? Por que?

-- Não foi possível inserir porque a chave primária que foi indicada (3) já existe

INSERT INTO funcionario VALUES

(9, 'Cecília Ribeiro', 2800, 'f', 4, '1980-04-05', 107);

-- Conseguiu inserir? Por que?

-- Não foi possível inserir porque o departamento 107 não foi criado na sua tabela

INSERT INTO funcionario VALUES

(9, 'Cecília Ribeiro', 2800, 'f', 4, '1980-04-05', 104);

-- Conseguiu inserir? Por que?

-- Sim, foi possível inserir porque os dados estão de acordo com as regras que foram definidas

-- Excluir a tupla (registro) da tabela FuncProj, em que fkFunc=3 e fkProj = 10.

DELETE FROM FuncProj

WHERE fkFunc = 3 and fkProj = 10;

-- Conseguiu excluir? Por que?

-- Sim, a fkFunc e a fkProj existia na tabela

-- Excluir a tupla da tabela Funcionario, em que idFunc=4.

DELETE FROM funcionario

WHERE idFunc = 4;

-- Conseguiu excluir? Por que?

-- Não foi possível porque a id especificada está atrelada a uma fk

-- Excluir a tupla da tabela Funcionário, em que idFunc=2.

DELETE FROM funcionario

WHERE idFunc = 2;

-- Conseguiu excluir? Por que?

-- Não foi possível porque a id especificada está atrelada a uma fk

-- Alterar o salário do Funcionário com idFunc=3 para 2800.

UPDATE funcionario

SET salario = 2800

WHERE idFunc = 3;

-- Conseguiu alterar? Por que?

-- Sim, as regras esperadas para atualizar um campo foram atendidas

-- Alterar o fkDepto do Funcionário com idFunc=3 para 101.

```
UPDATE funcionario
```

```
SET fkDepto = 101
```

```
WHERE idFunc = 3;
```

-- Conseguiu alterar? Por que?

-- Sim, as regras esperadas para atualizar um campo foram atendidas

-- Alterar o fkDepto do Funcionário com idFunc=3 para 107.

```
UPDATE funcionario
```

```
SET fkDepto = 107
```

```
WHERE idFunc = 3;
```

```
SELECT * FROM departamento;
```

-- Conseguiu alterar? Por que?

-- Não foi possível atualizar a fkDepto porque não existe o departamento especificado na tabela departamento

-- Exibir a data de nascimento e o salário do funcionário João Silva.

```
SELECT dataNasc, salario FROM funcionario
```

```
WHERE nomeFunc = 'João Silva';
```

-- Exibir uma lista apenas dos salários de todos os funcionários.

```
SELECT salario FROM funcionario;
```

-- Exibir uma lista apenas dos salários distintos de todos os funcionários.

```
SELECT distinct salario FROM funcionario;
```

-- Exibir os dados dos funcionários ordenados pelo nome.

```
SELECT * FROM funcionario ORDER BY nomeFunc;
```

-- Exibir os dados dos funcionários ordenados pelo salário, de forma decrescente.

```
SELECT * FROM funcionario ORDER BY salario DESC;
```

-- Exibir os dados dos funcionários que tenham salário entre 2000 e 4000.

```
SELECT * FROM funcionario
```

```
WHERE salario >= 2000 and salario <= 4000;
```

-- Exibir os nomes e os salários dos funcionários cujos nomes começam com a letra 'J'.

```
SELECT nomeFunc, salario FROM funcionario  
WHERE nomeFunc LIKE 'J%';
```

-- Exibir os nomes e os salários dos funcionários cujos nomes terminam com a letra 'a'.

```
SELECT nomeFunc, salario FROM funcionario  
WHERE nomeFunc LIKE '%a %';
```

-- Exibir os nomes dos funcionários que tem a letra 'n' como terceira letra do nome.

```
SELECT nomeFunc FROM funcionario  
WHERE nomeFunc LIKE '__n%';
```

-- Exibir os nomes e as datas de nascimento dos funcionários cujos nomes tenham a letra 'S' como 5ª letra de trás para frente.

```
SELECT nomeFunc, dataNasc FROM funcionario  
WHERE nomeFunc LIKE '%S_____';
```

-- Exibir os dados dos funcionários que trabalham no departamento Pesquisa.

```
SELECT * FROM funcionario  
WHERE fkDepto = 105;
```

-- Exibir os dados dos funcionários que trabalham no departamento Pesquisa e que tenham salário acima de 3500.

```
SELECT * FROM funcionario  
WHERE fkDepto = 105 and salario > 3500;
```

-- Exibir os dados dos funcionários que trabalham no departamento Pesquisa e que tenham o nome com inicial 'J'.

```
SELECT * FROM funcionario  
WHERE fkDepto = 105 and nomeFunc LIKE 'J%';
```

-- Exibir o idFunc e o nome de cada funcionário, juntamente com o idFunc e o nome do seu supervisor. Faça com que o título da coluna seja "idFuncionario" para

-- funcionário e "idSupervisor" para o id do supervisor

```
SELECT f.idFunc as 'Funcionário', f.nomeFunc as 'Funcionário', s.idFunc as 'Supervisor', s.nomeFunc as  
'Supervisor' FROM funcionario as f JOIN funcionario as s ON f.fkSupervisor = s.idFunc;
```

-- Exibir para cada projeto localizado em São Paulo, o idProj do projeto, o número do departamento que o controla, o nome e a data de nascimento do gerente do departamento.

```
SELECT p.idProj, p.fkDepto, f.nomeFunc, f.dataNasc FROM projeto as p JOIN funcionario as f ON  
f.fkDepto = p.fkDepto  
WHERE p.localProj = 'São Paulo';
```

-- Exibir o idFunc e o nome do funcionário, o projeto e o nome do projeto em que trabalha, e a quantidade de horas que trabalha nesse projeto.

```
SELECT f.idFunc, f.nomeFunc, p.idProj, p.nomeProj, fp.horas  
FROM funcionario as f  
JOIN FuncProj as fp ON f.idFunc = fp.fkFunc  
JOIN projeto as p ON fp.fkProj = p.idProj;
```

-- Exibir os nomes dos funcionários que nasceram antes de 1980.

```
SELECT nomeFunc FROM funcionario  
WHERE dataNasc < '1980-01-01';
```

-- Exibir a quantidade de salários diferentes que existem na empresa.

```
SELECT COUNT(distinct salario) as SalariosDiferentes FROM funcionario;
```

-- Exibir a quantidade de locais diferentes de projeto.

```
SELECT COUNT(distinct localProj) as LocaisDiferentes FROM projeto;
```

-- Exibir o salário médio da empresa e a soma dos salários.

```
SELECT truncate(avg(salario),2) as MediaSalarial, sum(salario) as SomaSalarios FROM funcionario;
```

-- Exibir o menor e o maior salário da empresa.

```
SELECT min(salario) as MenorSalario, max(salario) as MaiorSalario FROM funcionario;
```

-- Exibir o idDepto, o salário médio e a soma do salário de cada departamento (agrupado por departamento)

```
SELECT d.idDepto, truncate(avg(f.salario),2) as MediaSalarial, sum(f.salario) as SomaSalarial  
FROM funcionario as f JOIN departamento as d ON f.fkDepto = d.idDepto  
GROUP BY d.idDepto;
```

-- Exibir o idDepto, o menor e o maior salário de cada departamento (agrupado por departamento).

```
SELECT d.idDepto, min(f.salario) as MenorSalario, max(f.salario) as MaiorSalario
```

```
FROM funcionario as f JOIN departamento as d ON f.fkDepto = d.idDepto  
GROUP BY d.idDepto;
```

```
INSERT INTO funcionario VALUES  
(10, 'José da Silva', 1800, 'm', 3, '2000-10-12', null),  
(11, 'Benedito Almeida', 1200, 'm', 5, '2001-09-01', null);
```

```
INSERT INTO departamento VALUES  
(110, 'RH', 3, '2018-11-10');
```