

Exercícios CREATE TABLE e INSERT

Considere o esquema lógico a seguir.

```
cidade (codigo, nome, uf, regioao)
professor (codigo, nome, RG, sexo, codCid)
           codcid referencia cidade (codigo)
aluno (codigo, nome, raca, dtaNasc, sexo, RG, codCid)
       codcid referencia cidade (codigo)
tipo (codigo, nome) /* indica o tipo de aula */
aula (codAlu, codTip, codProf, dtaInicio, dtaFim, aproveitamento)
      codAlu referencia aluno (codigo)
      codTip referencia tipo (codigo)
      codProf referencia professor (codigo)
rendimento (codAlu, codTip, codProf, dtaInicio, observacoes, aproveitamento)
            (codAlu, codTip, codProf, dtaInicio) referencia
                                                    aula (codAlu, codTip, codProf, dtaInicio)
```

1. Durante a criação das tabelas no Banco de Dados, cuide a definição das seguintes restrições para as tabelas:
 - a. todas as tabelas devem ser criadas com **CONSTRAINT** para PK e FK.
 - b. cidade:
 - i. crie-a de duas formas: usando a definição da PK e FK ao final do CREATE TABLE e outra criando-as na mesma linha de definição dos atributos.
 - ii. ao executar a criação da tabela, comente uma das duas CREATE TABLE.
 - c. Professor e aluno
 - i. RG deve ser único (mas não será PK)
 - ii. atributo sexo deve ter um default e imponha a restrição de que apenas os valores F e M devem ser aceitos
 - d. Rendimento
 - i. o atributo aproveitamento só pode aceitar os valores 'Excelente', 'Bom' e 'Ruim'. Escolha um deles para ser o default.

```
CREATE TABLE cidade(
    codigo integer not null,
    nome varchar(200),
    uf char(2),
    regioao varchar(200),
    CONSTRAINT pk_cidade PRIMARY KEY (codigo)
);
```

```
CREATE TABLE tipo(
    codigo integer not null,
    nome varchar(200),
    CONSTRAINT pk_tipo PRIMARY KEY (codigo)
);
```

```
CREATE TABLE aluno(
    codigo integer not null,
    nome varchar(200),
    raca varchar(200),
    dataNasc date,
    codcid integer,
    sexo char(1) default 'M' check(sexo in ('M', 'F')),
    rg integer,
```

```
CONSTRAINT pk_aluno PRIMARY KEY (codigo),
CONSTRAINT fk_aluno_cidade FOREIGN KEY(codcid) REFERENCES cidade(codigo),
unique(RG)
);

CREATE TABLE professor(
    codigo integer not null,
    nome varchar(200),
    sexo char(1) default 'M' check(sexo in ('M', 'F')),
    rg integer,
    codcid integer,

    CONSTRAINT pk_professor PRIMARY KEY (codigo),
    CONSTRAINT fk_professor_cidade FOREIGN KEY(codcid) REFERENCES cidade(codigo),
    unique(RG)
);

CREATE TABLE aula(
    codAlu integer not null,
    codTip integer not null,
    codProf integer not null,
    dtaInicio date not null,
    dtaFim date,
    aproveitamento varchar(10) DEFAULT 'ruim' check(aproveitamento in ('excelente',
'ruim', 'bom')),
    CONSTRAINT pk_aula PRIMARY KEY (codAlu, codTip, codProf,dtaInicio),
    CONSTRAINT fk_aula_aluno FOREIGN KEY (codAlu) REFERENCES aluno(codigo),
    CONSTRAINT fk_aula_professor FOREIGN KEY (codProf) REFERENCES professor(codigo),
    CONSTRAINT fk_aula_tipo FOREIGN KEY (codTip) REFERENCES tipo(codigo)
);

CREATE TABLE rendimento(
    codAlu integer not null,
    codTip integer not null,
    codProf integer not null,
    dtaInicio date not null,
    observacoes varchar(200),
    aproveitamento varchar(10) DEFAULT 'ruim' check(aproveitamento in ('excelente',
'ruim', 'bom')),
    CONSTRAINT pk_rendimento PRIMARY KEY (codAlu, codTip, codProf,dtaInicio),
    CONSTRAINT fk_aula_rendimento FOREIGN KEY (codAlu,codTip,codProf,dtaInicio)
REFERENCES aula(codAlu,codTip,codProf,dtaInicio)
);
```

2. Efetue as seguintes alterações:

a. Adicione a coluna contato na tabela **cliente**

```
ALTER TABLE aluno
    add contato varchar(200);
```

b. Altere a tabela **cidade** (usar um único comando de **ALTER TABLE**)

- definindo o valor default 'SC' na coluna UF
- deletando a coluna chamada "regiao"
- adicionando uma coluna chamada "região_nacional".

```
ALTER TABLE cidade
    alter uf set default 'SC',
    drop regiao,
    add regiao_nacional varchar(200);
```

3. Insira dados nas tabelas:

- a. 3 cidades, 2 alunos, 2 professores, 2 tipos, 2 aulas e 2 rendimentos

```
INSERT INTO cidade VALUES (1, 'Palhoça', 'SC', 'Sul');  
INSERT INTO cidade VALUES (2, 'Flóripa', 'SC', 'Sul');  
INSERT INTO cidade VALUES (3, 'Porto Alegre', 'SS', 'Sul');
```

```
INSERT INTO aluno(codigo, nome, raca, dtaNasc, sexo, RG, codCid) VALUES  
(1, 'Joao', NULL, '1/1/2000', 'M', 123123123, 1),  
(2, 'Juca', NULL, '1/1/2010', 'M', 1, 123123323, 2);
```

```
INSERT INTO professor(codigo, nome, RG, sexo, codCid) VALUES  
(1, 'Maria', 123123123, 'F', 1),  
(2, 'Ana', 123123323, 'F', 1),  
(3, 'Lia', 133123123, 'F', 1);
```

```
INSERT INTO tipo VALUES  
(1, 'Regular'),  
(2, 'Treinamento');
```

```
INSERT INTO aula (codAlu, codTip, codProf, dtaInicio, dtaFim, aproveitamento)  
VALUES
```

```
(1, 1, 1, '1/1/2000', '1/10/2000', 'ruim'),  
(1, 2, 1, '1/1/2010', '1/11/2010', 'ruim');
```

```
INSERT INTO rendimento (codAlu, codTip, codProf, dtaInicio, observacoes,  
aproveitamento) VALUES
```

```
(1, 1, 1, '1/1/2000', 'falta de atenção', 'ruim'),  
(1, 2, 1, '1/1/2010', 'falta de material', 'ruim');
```

Para visualizar os dados inseridos:

```
SELECT *  
FROM <nome_da_tabela>
```

4. Altere dados nas seguintes tabelas criadas acima:

- a) altere o valor de UF de uma cidade.

```
UPDATE cidade  
SET uf = 'RS'  
WHERE codigo = 3
```

- d) delete uma linha da tabela cidade.

```
DROP cidade  
WHERE codigo = 3
```