

Implementação de Banco de Dados

Atividade Estruturada 1/2

Título: Museu de Obras de Arte sobre Computação – modelo de dados

Produto/Resultado:

```
CREATE TABLE Autores (  
    Codigo NUMERIC(5) PRIMARY KEY,  
    Nome VARCHAR(20) NOT NULL,  
    Nacionalidade VARCHAR(15) NOT NULL );  
  
CREATE TABLE Funcionarios (  
    RG NUMERIC(10) PRIMARY KEY, /* adicionar fk */  
    Nome VARCHAR(50) NOT NULL,  
    Salario NUMERIC(6,2) NOT NULL,  
    Turno CHAR(1) NOT NULL,  
    Funcao VARCHAR(9) NOT NULL );  
  
CREATE TABLE Lotacoes (  
    RG NUMERIC(10) NOT NULL UNIQUE, /* adicionar fk */  
    Numero NUMERIC(3) NOT NULL UNIQUE, /* adicionar fk */  
    HoraEntrada TIMESTAMP NOT NULL,  
    HoraSaida TIMESTAMP NOT NULL,  
    PRIMARY KEY(RG,Numero,HoraEntrada) ); /* 3 pks de uma vez */  
  
CREATE TABLE Saloes (  
    Numero NUMERIC(3) PRIMARY KEY, /* adicionar fk */  
    Andar CHAR(1) NOT NULL,  
    Area NUMERIC(4,2) NOT NULL );  
  
CREATE TABLE Obras (  
    Codigo NUMERIC(10) PRIMARY KEY,  
    Titulo VARCHAR(50) NOT NULL,  
    Ano CHAR(4) NOT NULL,  
    Autor NUMERIC(5), /* adicionar fk */  
    Salao NUMERIC(3) ); /* adicionar fk */  
  
CREATE TABLE Esculturas (  
    Codigo NUMERIC(10) PRIMARY KEY, /* adicionar fk */  
    Altura NUMERIC(5,2) NOT NULL,  
    Peso NUMERIC(6,2) NOT NULL,  
    Material VARCHAR(15) NOT NULL );  
  
CREATE TABLE Pinturas (  
    Codigo NUMERIC(10) PRIMARY KEY, /* adicionar fk */  
    Estilo VARCHAR(20) NOT NULL,
```

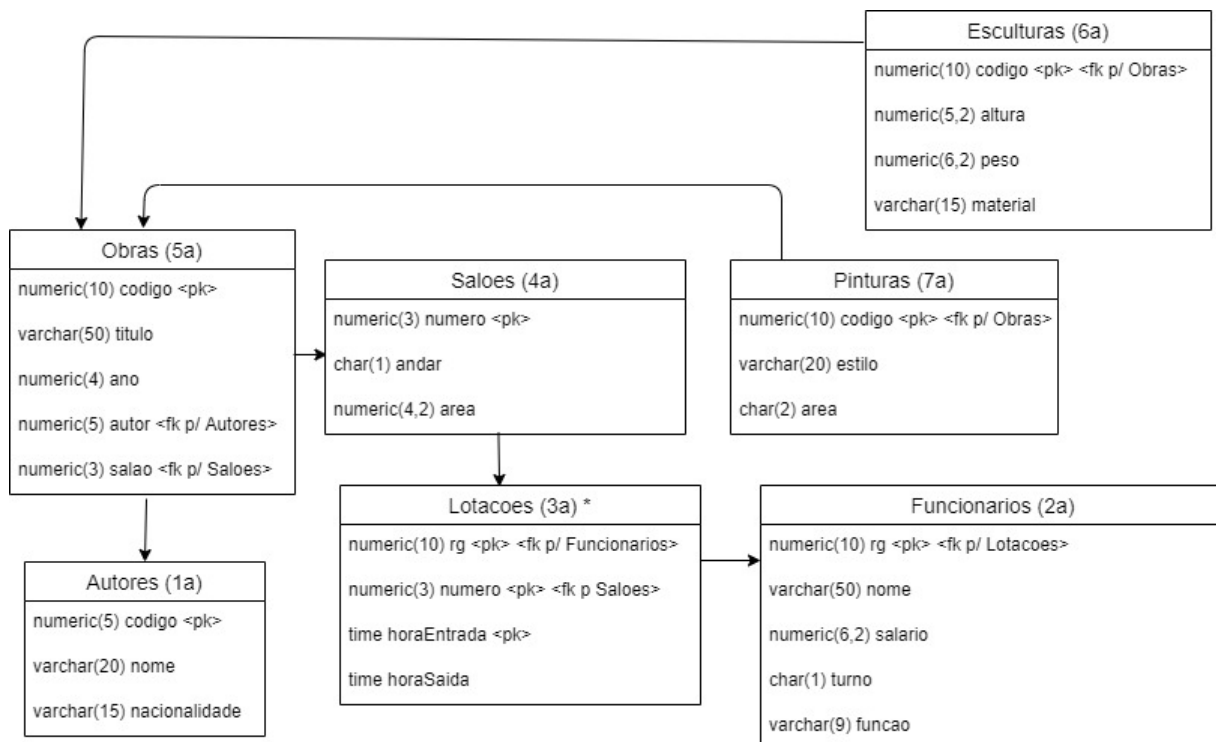
Area CHAR(2) NOT NULL);

/* adicionando chaves estrangeiras */

```
ALTER TABLE Funcionarios ADD FOREIGN KEY (RG) REFERENCES Lotacoes(RG);
ALTER TABLE Lotacoes ADD FOREIGN KEY (RG) REFERENCES Funcionarios(RG);
ALTER TABLE Lotacoes ADD FOREIGN KEY (Numero) REFERENCES Saloes(Numero);
ALTER TABLE Obras ADD FOREIGN KEY (Autor) REFERENCES Autores(Codigo);
ALTER TABLE Obras ADD FOREIGN KEY (Salao) REFERENCES Saloes(Numero);
ALTER TABLE Esculturas ADD FOREIGN KEY (Codigo) REFERENCES Obras(Codigo);
ALTER TABLE Pinturas ADD FOREIGN KEY (Codigo) REFERENCES Obras(Codigo);
```

/* adicionando indexes */

```
CREATE INDEX idx_saloes ON Saloes (Area);
CREATE INDEX idx_autores ON Autores (Nacionalidade);
CREATE INDEX idx_obras ON Obras (Ano,Autor);
CREATE INDEX idx_pinturas ON Pinturas (Estilo,Area);
CREATE INDEX idx_esculturas ON Esculturas (Material);
CREATE INDEX idx_funcionarios ON Funcionarios (Turno,Funcao);
CREATE INDEX idx_lotacoes ON Lotacoes (HoraEntrada);
```



Implementação de Banco de Dados

Atividade Estruturada 2/2

Título: Museu de Obras de Arte sobre Computação – modelo de dados

Notas: *Ao tentar preencher as tabelas encontrei um problema quanto às chaves. Da forma como o BD foi projetado não era possível inserir tuplas na tabela Funcionarios, Lotacoes e Saloes, pois a chave estrangeira de uma não permitia atributos repetidos na outra e vice-versa(pk e fk estavam marcadas na mesma coluna pra Funcionarios e Lotacoes). Tive que remover a constraint das chaves estrangeiras para conseguir inserir dados e testar as consultas.*

Os comandos de inserção de dados e de alteração de constraint não estão descritos no código deste DOC, pois não foram pedidos e achei que atrapalhariam a visualização das respostas. Acho estranho que sempre que eu quisesse inserir dados eu teria que sempre remover constraint, usar insert, depois refazer as constraints...

Produto/Resultado:

```
/* 1 código e título de obras de 1965 a 1975 */
SELECT codigo, titulo FROM Obras
WHERE ano BETWEEN 1965 AND 1975 AND salao = 36;

/* 2 código e título de obras do autor Pablo Picasso que estão no 3º andar */
SELECT O.codigo, titulo FROM Obras O, Autores A, Saloes S
WHERE O.autor = A.codigo AND O.salao = S.numero
      AND A.nome = 'Pablo Picasso' AND S.andar = '3';

/* 3 código e título de obras impressionistas cujo material é argila */
(SELECT O.codigo, O.titulo FROM Obras O, Esculturas E
WHERE O.codigo = E.codigo AND E.material = 'argila')
UNION
(SELECT O.codigo, O.titulo FROM Obras O, Pinturas P
WHERE O.codigo = P.codigo AND P.estilo = 'impressionista');

/* 4.a nome e nacionalidade dos autores que possuem obras no museu -
usando subconsulta */
SELECT nome, nacionalidade FROM Autores
WHERE codigo in(SELECT autor FROM Obras);

/* 4.b usando Join*/
SELECT DISTINCT nome, nacionalidade FROM autores A
JOIN obras O ON O.autor = A.codigo;

/* 5.a nome e nacionalidade dos autores que não têm obras no museu -
usando subconsulta */
SELECT nome, nacionalidade FROM Autores
```

```
WHERE codigo NOT IN(SELECT autor FROM Obras);
```

```
/* 5.b usando join */
```

```
SELECT O.codigo, nome, nacionalidade FROM Obras O  
RIGHT JOIN Autores A ON A.codigo = O.autor  
WHERE O.codigo IS null;
```

```
/* 6.a nome e nacionalidade dos autores que têm apenas pinturas -  
usando subconsulta */
```

```
SELECT nome, nacionalidade FROM autores A  
WHERE A.codigo IN (  
    SELECT autor FROM Obras O  
    WHERE O.codigo IN (  
        SELECT P.codigo FROM Pinturas P));
```

```
/* 6.b usando join */
```

```
SELECT DISTINCT A.nome, A.nacionalidade FROM Pinturas P  
JOIN Obras O ON P.codigo = O.codigo  
JOIN Autores A ON O.autor = A.codigo;
```

```
/* 7.a nome e nacionalidade dos autores que têm tanto pinturas quanto esculturas -  
usando subconsulta */
```

```
SELECT nome,nacionalidade FROM autores  
WHERE codigo IN ((  
    SELECT autor FROM obras  
    WHERE codigo IN (  
        SELECT codigo FROM esculturas))  
INTERSECT  
(SELECT autor FROM obras  
WHERE codigo IN (  
    SELECT codigo FROM pinturas))));
```

```
/* 7.b usando join */
```

```
(SELECT A.nome,A.nacionalidade FROM autores A  
JOIN Obras O ON A.codigo = O.autor  
JOIN Esculturas E ON O.codigo = E.codigo)  
INTERSECT  
(SELECT A.nome,A.nacionalidade FROM autores A  
JOIN Obras O ON A.codigo = O.autor  
JOIN Pinturas P ON O.codigo = P.codigo)
```

```
/* 8a nome e nacionalidade dos autores que têm ou apenas pinturas  
ou apenas esculturas - usando subconsulta */
```

```
SELECT nome,nacionalidade FROM autores  
WHERE codigo IN ((  
    SELECT autor FROM obras  
    WHERE codigo IN (  
        SELECT codigo FROM esculturas))  
UNION  
(SELECT autor FROM obras  
WHERE codigo IN (  
    SELECT codigo FROM pinturas))));
```

/* 8.b usando join */

```
(SELECT A.nome,A.nacionalidade FROM autores A
JOIN Obras O ON A.codigo = O.autor
JOIN Esculturas E ON O.codigo = E.codigo)
UNION
(SELECT A.nome,A.nacionalidade FROM autores A
JOIN Obras O ON A.codigo = O.autor
JOIN Pinturas P ON O.codigo = P.codigo)
```

/* 9.a codigo dos autores que não têm obras no museu - usando conjuntos */

```
(SELECT codigo FROM autores)
EXCEPT
(SELECT autor FROM Obras)
```

/* 9.b codigo dos autores que possuem apenas pinturas - usando conjuntos */

```
(SELECT codigo FROM autores)
INTERSECT
(SELECT autor FROM Obras
WHERE codigo IN(
SELECT codigo FROM Pinturas))
```

/* 9.c codigo dos autores que possuem apenas esculturas - usando conjuntos */

```
(SELECT codigo FROM autores)
INTERSECT
(SELECT autor FROM Obras
WHERE codigo IN(
SELECT codigo FROM Esculturas))
```

/* 9.d codigo dos autores que possuem tanto pinturas quanto esculturas - usando conjuntos */

```
((SELECT autor FROM obras
WHERE codigo IN (
SELECT codigo FROM pinturas))
INTERSECT
(SELECT autor FROM obras
WHERE codigo IN (
SELECT codigo FROM esculturas)))
INTERSECT
(SELECT codigo FROM autores)
```

/* 9.e codigo dos autores que possuem ou apenas pinturas ou apenas esculturas - usando conjuntos */

/* União */

```
((SELECT DISTINCT autor FROM obras
WHERE codigo IN (
SELECT codigo FROM pinturas))
UNION
(SELECT DISTINCT autor FROM obras
```

```

WHERE codigo IN (
    SELECT codigo FROM esculturas)))
/* fim União*/
EXCEPT /* diferença com */
/* Interseção */
((SELECT DISTINCT autor FROM obras
WHERE codigo IN (
    SELECT codigo FROM pinturas))
INTERSECT
(SELECT DISTINCT autor FROM obras
WHERE codigo IN (
    SELECT codigo FROM esculturas)))
/* fim Interseção */

/* 10 pares seguranças que cuidam do mesmo salão no mesmo período */
/* bloco de resposta */
CREATE VIEW v1 AS
(SELECT F.nome, L.numero, L.horaEntrada, L.horaSaida FROM funcionarios F
JOIN lotacoes L ON F.rg = L.rg
WHERE UPPER(F.funcao) = 'SEGURANÇA');

CREATE VIEW v2 AS
(SELECT F.nome, L.numero, L.horaEntrada, L.horaSaida FROM funcionarios F
JOIN lotacoes L ON F.rg = L.rg
WHERE UPPER(F.funcao) = 'SEGURANÇA');

SELECT v1.nome, v2.nome FROM v1, v2
WHERE v1.horaEntrada = v2.horaEntrada AND v1.horaSaida = v2.horaSaida AND
    v1.numero = v2.numero AND v1.nome != v2.nome
/* fim do bloco de resposta */

/* 11 numero dos salões onde tem apenas esculturas */
SELECT DISTINCT S.numero FROM esculturas E
JOIN obras O ON E.codigo = O.codigo
JOIN saloes S ON O.salao = S.numero;

/* 12 nome e rg dos faxineiros que limpam todos os saloes do 4º andar */
SELECT nome,rg FROM funcionarios
WHERE UPPER(funcao) = 'FAXINEIRO' AND rg IN(
    SELECT rg FROM lotacoes
    WHERE numero IN(
        SELECT numero from saloes
        WHERE andar = '4'));

/* 13 autores brasileiros que têm apenas esculturas */
SELECT * FROM esculturas E
JOIN Obras O ON E.codigo = O.codigo
JOIN Autores A ON O.autor = A.codigo
WHERE UPPER(A.nacionalidade) = 'BRASILEIRO';

```

```
/* 14 nome e rg dos seguranças que cuidam apenas do salão 28 */  
SELECT F.nome, F.rg FROM funcionarios F  
JOIN lotacoes L ON F.rg = L.rg  
JOIN saloes S ON S.numero = L.numero  
WHERE S.numero = 28 AND UPPER(F.funcao) = 'SEGURANÇA';
```

```
/* 16 inserir um novo salao de nº 70, no 20º andar e de 121 m² */  
INSERT INTO saloes (numero,andar,area)  
VALUES (70,'20',121.00);
```

```
/* 17 transferir todas as obras do salao 51 para o salão 70 */  
UPDATE obras  
SET salao = 70  
WHERE salao = 51;
```

```
/* 18 eliminar todas as pinturas do autor Salvador Dali do cadastro */  
DELETE FROM pinturas  
WHERE codigo IN (  
    SELECT codigo FROM obras  
    WHERE autor IN (  
        SELECT codigo FROM autores  
        WHERE UPPER(nome) = 'SALVADOR DALI'));
```