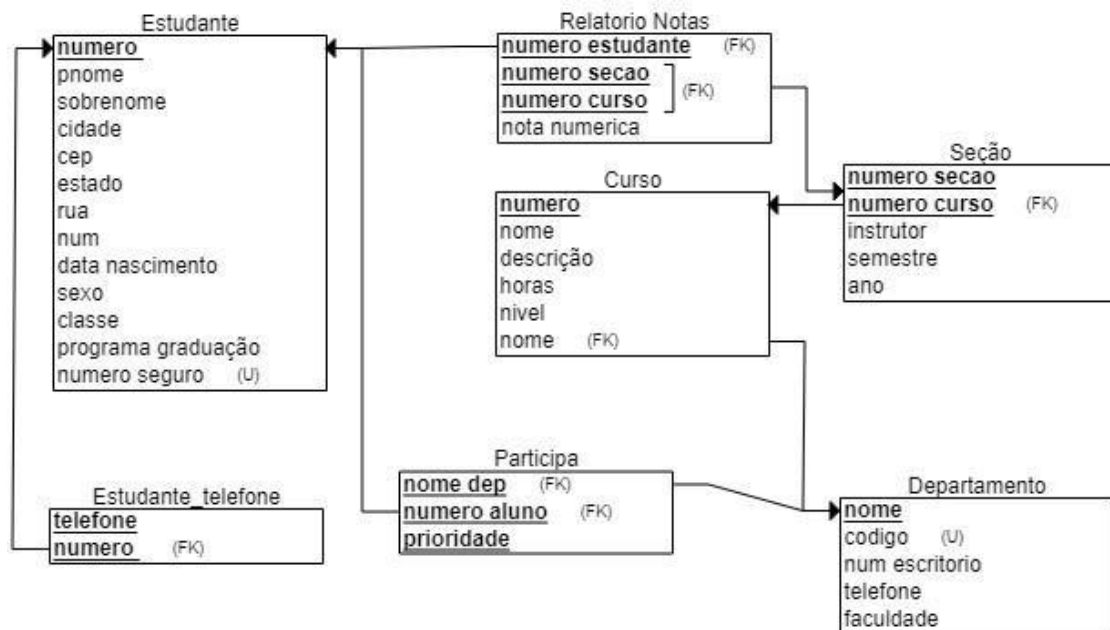


Universidade Federal do Ceará
Fundamentos de bancos de dados
Profa. Livia Almada

Considere o diagrama do esquema relacional apresentado a seguir.



Parte 1) Criação do banco de dados.

1.1) Descreva as chaves primárias, as chaves únicas indicando quais atributos que as compõem.

Tabela Estudante: numero (chave primária), numero seguro (chave única)

Tabela Estudante_telefone: telefone (chave primária)

Tabela Relatorio_Notas: numero_estudante (chave primária)

Tabela Seção: numero_secao (chave primária)

Tabela Curso: numero (chave primária)

Tabela Participa: numero_aluno (chave primária)

Tabela Departamento: nome(chave primária), codigo(chave única)

1.2) Descreva as chaves estrangeiras de cada relação, indicando o atributo que é chave estrangeira e qual relação e atributo cada chave estrangeira referencia.

Tabela Estudante_telefone: chave estrangeira numero que faz referência ao atributo numero na tabela Estudante .

Tabela Relatorio_Notas: chave estrangeira numero_estudante que faz referência ao atributo numero na tabela Estudante. Chave estrangeira

numero_secao e numero_curso que faz referência ao atributo numero_secao na tabela Seção.

Tabela Curso: chave estrangeira nome que faz referência ao atributo nome na tabela Departamento.

Tabela Participa: chave estrangeira nome_dep que faz referência ao atributo nome na tabela Departamento. Chave estrangeira numero_aluno que faz referência ao atributo numero na tabela Estudante.

Tabela Seção: chave estrangeira numero_curso que faz referência ao atributo numero na tabela Curso.

1.3) Crie o banco de dados e em seguida, crie as tabelas indicando as chaves primárias, únicas e estrangeiras (quando for possível). Observe a ordem de dependência/referência entre tabelas para determinar a ordem de criação. Adicione as chaves estrangeiras que ficaram faltando. *Utilize os comandos: CREATE TABLE e ALTER TABLE*

Para as chaves estrangeiras especifique o comportamento para alteração ou exclusão da referência.

- *On delete cascade (ou) on delete no action (ou) on delete set null (ou) on delete set default*
- *On update cascade (ou) on update no action (ou) on update set null (ou) on update set default*

```
CREATE TABLE Estudante (  
    numero INT PRIMARY KEY,  
    pnome VARCHAR(100) NOT NULL,  
    cidade VARCHAR(100),  
    cep VARCHAR(10),  
    estado VARCHAR(100),  
    rua VARCHAR(100) NOT NULL,  
    num INT,  
    data_nascimento DATE,  
    sexo VARCHAR(20),  
    classe VARCHAR(20),  
    programa_graduacao VARCHAR(50),  
    numero_seguro INT  
);
```

```
CREATE TABLE Estudante_telefone(  
    telefone VARCHAR(100) PRIMARY KEY,  
    numero INT,  
    FOREIGN KEY (numero)  
    REFERENCES Estudante(numero)  
    ON DELETE CASCADE  
);
```

```
CREATE TABLE Departamento (  
    nome VARCHAR(50) PRIMARY KEY,  
    codigo INTEGER UNIQUE,  
    num_escritorio INTEGER,  
    telefone VARCHAR(50),  
    faculdade VARCHAR(50)  
);
```

```

CREATE TABLE Curso (
    numero INT PRIMARY KEY,
    nome VARCHAR(50),
    descricao VARCHAR(200),
    horas INT,
    nivel INT,
    nome_departamento VARCHAR(50),
    FOREIGN KEY (nome_departamento)
    REFERENCES Departamento(nome)
    ON DELETE CASCADE
    ON UPDATE CASCADE
);

```

```

CREATE TABLE Secao (
    numero_secao INT PRIMARY KEY,
    numero_curso INT,
    instrutor_nome VARCHAR(50),
    ano INTEGER,
    FOREIGN KEY (numero_curso) REFERENCES Curso(numero) ON DELETE
    CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE Relatorio_Notas (
    numero_estudante INT,
    numero_secao INT,
    numero_curso INT,
    nota_numerica INTEGER,
    FOREIGN KEY (numero_estudante) REFERENCES Estudante(numero) ON
    DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (numero_secao) REFERENCES Secao(numero_secao) ON
    DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY (numero_curso) REFERENCES Curso(numero) ON DELETE
    CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE Participa (
    nome_dep VARCHAR(50),
    numero_aluno INT,
    prioridade VARCHAR(50),
    FOREIGN KEY (nome_dep)
    REFERENCES
    Departamento(nome)
    ON DELETE CASCADE
);

```

- Insira 3 tuplas em cada tabela. Execute o comando de consulta para verificar as tuplas inseridas: `SELECT * FROM <NOME DA TABELA>`

- `UPDATE` é um comando da DML para fazer alteração nos valores de tuplas já inseridas. A sintaxe geral do comando é:
`UPDATE <NOME DA TABELA> SET <ATRIBUTO> = <VALOR> WHERE <CONDIÇÃO>.`
 Por exemplo, o comando a seguir altera o valor da coluna cidade para Quixadá para o aluno cujo numero é 1.

`UPDATE ESTUDANTE SET CIDADE = 'QUIXADÁ' WHERE NUMERO=1.`

Agora é a sua vez! Modifique 1 tupla de três tabelas diferentes utilizando o comando `UPDATE`.

- **Fiz um update no atributo pnome de João para Jeferson Aires, na tabela estudante**
`UPDATE estudante`
`SET pnome = 'Jeferson Aires'`
`WHERE numero = 1;`
- **Fiz um update no atributo nota_numerica de 75 para 100, na tabela relatorio_notas**
`UPDATE relatorio_notas`
`SET nota_numerica = 100`
`WHERE numero_estudante = 3`
- **Fiz um update no atributo telefone de '(22) 2222-2222' para '(88) 4002-8922', na tabela departamento**
`UPDATE departamento`
`SET telefone = '(88)4002-8922'`
`WHERE nome = 'Departamento B'`

- `DELETE` é um comando da DML para apagar tuplas já inseridas. A sintaxe geral do comando é:

`DELTE FROM <NOME DA TABELA> WHERE <CONDIÇÃO>.`

Por exemplo, o comando a seguir apaga a tupla do aluno cujo numero é 1. `DELETE FROM ESTUDANTE WHERE NUMERO=1.`

Agora é a sua vez! Apague 1 tupla de três tabelas diferentes utilizando o comando `DELETE`.

- **Excluí da tabela de departamento pelo o atributo nome.**
 Conteúdo antes da exclusão: ('Departamento A', 101, 1010, '(11) 1111-1111', 'Faculdade A').
`DELETE FROM departamento`
`WHERE nome = 'Departamento A';`
- **Excluí da tabela de participa pelo o atributo numero_aluno.**
 Conteúdo antes da exclusão: ('Departamento B', 2, 'Média').
`DELETE FROM participa`
`WHERE numero_aluno = 2;`

- **Excluí da tabela de curso pelo o atributo numero.**
Conteúdo antes da exclusão: (3, 'Curso C', 'Descrição C', 75, 1, 'Departamento C');
DELETE FROM curso
WHERE numero = 3;

Parte 3 - Restrições de integridade

- Execute um comando de inserção que infrija uma restrição de unicidade. O que o SGBD retorna? Explique.

Eu havia realizado a inserção de uma tupla e tentei novamente fazer essa inserção, na tabela de departamento. Meu SGBD me retornou a seguinte mensagem:
INSERT INTO departamento (nome, codigo, num_escritorio, telefone, faculdade)
VALUES

('Departamento B', 102, 2020, '(22) 2222-2222', 'Faculdade B')

ERROR: Chave (nome)=(Departamento B) já existe.duplicar valor da chave viola a restrição de unicidade "departamento_pkey"

O SGBD não permite a inserção de dados duplicados para manter a integridade dos dados na tabela e garantir que os valores exclusivos sejam respeitados.

- Execute um comando de alteração que infrija uma restrição referencial. O que o SGBD retorna?Explique.

Tentei inserir um número de telefone com um "numero" que não existe em "Estudante"

INSERT INTO Estudante_telefone (telefone, numero)
VALUES ('123-456-7890', 9999);

Chave (numero)=(9999) não está presente na tabela "estudante".inserção ou atualização em tabela "estudante_telefone" viola restrição de chave estrangeira "estudante_telefone_numero_fkey"

O SGBD não permite insirir um valor na coluna de chave estrangeira ("numero") que não tenha uma correspondência na tabela referenciada ("Estudante").

- Execute um comando de exclusão que infrija uma restrição referencial. O que o SGBD retorna?Explique.

Tentei executar um comando de exclusão e a exclusão não foi realizada, pois eu estava tentando executar uma ação em uma tabela referenciada por outra tabela por meio de uma chave estrangeira, mas essa ação violaria a integridade referencial porque ainda existem registros na tabela filha que dependem dos registros na tabela pai

