Universidade Federal de Minas Gerais
Instituto de Ciéncias Exatas
Graduação em Ciéncia da Computação
Disciplina de Introdução a Banco de Dados

Trabalho Prático

Banco de dados para consumo de arte

dos requisitos à implementação final

Aluna: Giulia Monteiro Silva Gomes Vieira
Matrícula: 2016006492

Data: 1 de dezembro de 2023

Conteúdo

1 Trabalho Prático 01

1.1 Requisitos de dados

Para projetar um banco de dados usado em um sisema de gerenciamento de consumo de arte para alocação de recursos para artistas devemos considerar quais informações são pertinentes serem armazenadas sobre cada tipo de arte e artista, e qual o padrão de consumo de arte dos usuários, para que este aponte os artistas que produzem objetos mais acessados.

A divisão tradicional dos tipos de arte é: Música, Escultura, Pintura, Literatura, Arquitetura, Performática (teatro e dança), e Filme.

Sendo assim, podemos identificar como requisitos para este sisema as seguintes diferenciações:

- Objeto artístico (generalização):
  - Música:
    - ID: identificador único no sistema
    - Nome: nome da música
    - Tema: álbum
    - Ano: ano em que a música foi publicada
    - Autor: banda ou artista que escreveu a música
    - Executor: banda ou artista que toca a música
    - Detentor: quem possui os direitos sobre a música
    - Gênero: genero percentente
  - Escultura:
    - ID: identificador único no sistema
    - Nome: nome da escultura

    * Tema: temática da escultura
    * Material: material de que é feita
    * Ano: ano em que a escultura foi terminada
    * Autor: quem produziu a escultura
    * Detentor: quem é o dono da escultura
    * Gênero: genero percentente
    * Local: onde está a escultura
- Pintura:
    * ID: identificador único no sistema
    * Nome: nome da pintura
    * Tema: temática da pintura
    * Material: material de que é feita
    * Ano: ano em que a pintura foi terminada
    * Autor: quem produziu a pintura
    * Detentor: quem é o dono da pintura
    * Gênero: genero percentente
    * Local: onde está a pintura
- Literatura:
    * ID: identificador único no sistema
    * Nome: nome do livro
    * Tema: temática do livro
    * Ano: ano em que o livro foi terminada

    * Autor: quem escreveu o livro
    * Detentor: quem detém os direitos sobre o livro
    * Gênero: movimento artístico ao qual pertence
- Arquitetura:
    * ID: identificador único no sistema
    * Nome: nome da contrução
    * Material: material da construção
    * Ano: ano em que foi contruído
    * Autor: quem desenhou o projeto
    * Executor: quem construiu
    * Detentor: quem é o dono da construção
    * Local: onde está
    * Gênero: genero percentente
- Performance:
    * ID: identificador único no sistema
    * Nome: nome da performance
    * Tema: temática da performance
    * Tipo: teatro ou dança
    * Ano: ano em que foi publicada
    * Autor: quem escreveu ou coreografou a performance
    * Executor: quem está performando
    * Detentor: quem tem os direitos sobre a performance

        * Gênero: genero percentente
        * Local: onde está acontecendo
    - Filme:
        * ID: identificador único no sistema
        * Nome: nome do filme
        * Tema: temática do filme
        * Gênero: genero cinematográfico percentente
        * Ano: ano em que foi publicado
        * Autor: quem escreveu o filme
        * Diretor: quem dirigiu o filme
        * Executor: quem está performando o filme
        * Detentor: quem tem os direitos sobre o filme
- Artista:
    - ID: identificador único no sistema
    - Nome: nome da pessoa ou grupo
    - AnoInício: ano nascimento ou inicio do grupo
    - AnoFim: ano de morte ou fim do grupo
- Usuário:
    - ID: identificador único no sistema
    - Nome: nome do usuário
    - email: email do usuário
- Logs:

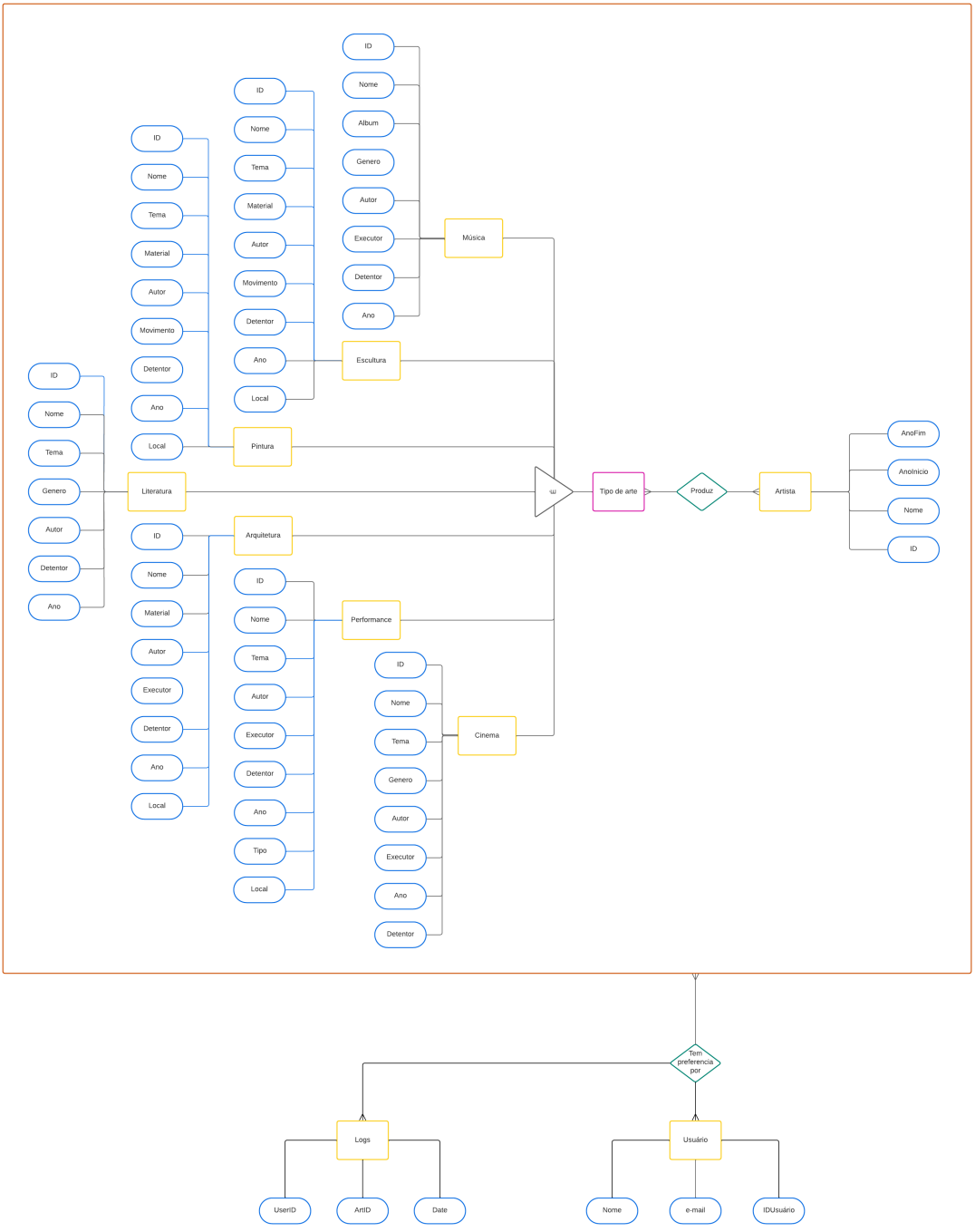
- IDUsuário: identificador do usuário
- IDObjeto: objeto artístico consumido
- Data: data em que foi consumido

Todas as entidades citadas têm um identificador único porque estamos supondo que todas as características podem se repetir e não queremos utilizar chave composta.

Estas entidades se relacionam da seguinte maneira:
Um objeto artístico pode ser produzido por vários artistas e cada artista de produzir vários objetos artísticos de tipos diferentes. Sendo assim, temos uma relação M:N entre estas categorias.
Um usuário pode consumir diversos objetos artísticos, e cada objeto pode ser consumido por vários usuários. Sendo assim, esta relanção também é M:N.
Um usuário pode ter preferência por diversos artistas, e cada artista pode ser preferido o por vários usuários. Sendo assim, esta relanção também é M:N.
Um usuário pode ser registrado nos logs diversas vezes, e os logs dizes respeito a todos os usuários. Portanto a relação também é M:N.
Um objeto pode ser acessado nos logs diversas vezes, e os logs dizes respeito a todos os usuários. Portanto a relação também é M:N.

1.2 Esquema de entidades e relacionamentos estendido

Dadas as restrições apresentadas anteriormente, portanto, é possível desenhar um esquema de entidades e relacionamentos estendido da seguinte maneira:

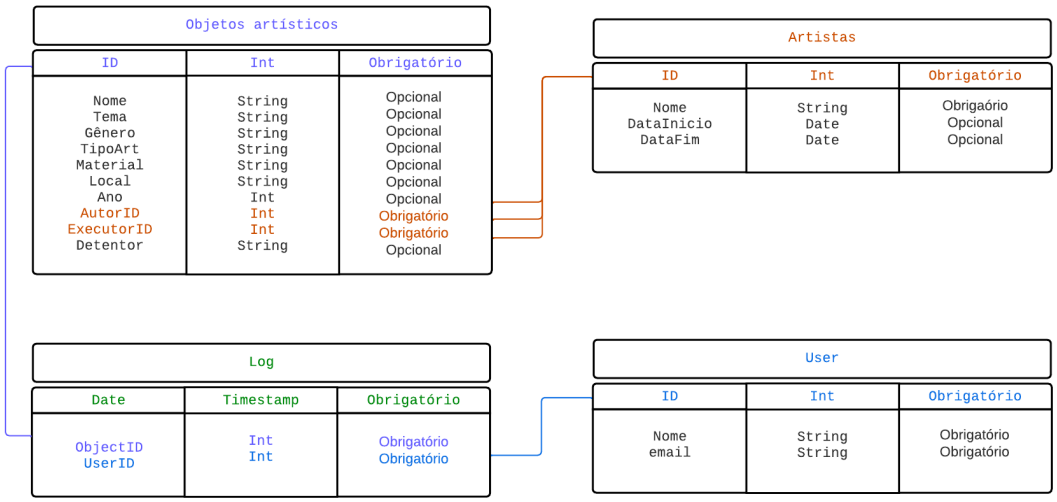


2 Trabalho Prático 02

2.1 Esquema relacional

Prosseguindo com o desenvolvimento iniciado no trabalho anterior, aqui estabeleceremos o esquema relacional derivado do esquema de entidade e relacionamentos estendido.
Como podemos perceber no esquema da sessão anterior, as formas de arte são especificações do tipo geral objeto artístico e seus atributos são muito parecidos. Por este motivo, decidi que a super-entidade Arte será uma tabela só, onde algumas colunas serão opcionais.

Na tabela a seguir, os atributos chave de cada tabela estão coloridos, e suas relações como chave estrangeira demarcadas com conectores da mesma cor. O domínio de cada um está na segunda coluna, e a possibilidade de aceitar-se valor NULL (ou seja, obrigatoriedade ou não do campo ser preenchido) na terceira coluna.



2.2 Script para geração do banco de dados

```
-- Table 01: Artistic Object
CREATE TABLE ArtisticObject (
    ID INT PRIMARY KEY,
    Name VARCHAR(255),
    Theme VARCHAR(255),
    Genre VARCHAR(255),
    Year VARCHAR(4),
    TypeArt VARCHAR(255),
    Material VARCHAR(255),
    AuthorID INT NOT NULL,
    ExecutorID INT NOT NULL,
    Owner VARCHAR(255),
    FOREIGN KEY (AuthorID) REFERENCES Artists(ID),
    FOREIGN KEY (ExecutorID) REFERENCES Artists(ID)
);

-- Table 02: Artists
CREATE TABLE Artists (
    ID INT PRIMARY KEY,
    Name VARCHAR(255),
    DateInit DATE,
    DateEnd DATE
);

-- Table 03: Users
CREATE TABLE Users (
    ID INT PRIMARY KEY,
    Name VARCHAR(255),
    Email VARCHAR(255)
);

-- Table 04: Logs
CREATE TABLE Logs (
    Date TIMESTAMP PRIMARY KEY,
```

```
    ObjectID INT NOT NULL,
    ArtistID INT NOT NULL,
    FOREIGN KEY (ObjectID) REFERENCES ArtisticObject(ID),
    FOREIGN KEY (ArtistID) REFERENCES Artists(ID)
);
```

Este script foi desenvolvido pensando no modelo SQLite. Em outros modelos os CONSTRAINTS de FOREIGN KEY devem ser estabelecidos posteriormente, visto que a tabela referenciada deve existir para ser referenciada. Neste caso, as linhas iniciadas com FOREIGN KEY seriam subsitituídas pela adição das seguintes linhas ao final do script:

```
-- Add Foreign Key Constraints

-- Add foreign key constraints to ArtisticObject table
ALTER TABLE ArtisticObject
ADD FOREIGN KEY (AuthorID) REFERENCES Artists(ID),
ADD FOREIGN KEY (ExecutorID) REFERENCES Artists(ID);

-- Add foreign key constraints to Logs table
ALTER TABLE Logs
ADD FOREIGN KEY (ObjectID) REFERENCES ArtisticObject(ID),
ADD FOREIGN KEY (ArtistID) REFERENCES Artists(ID);
```

2.3 Protótipo do sistema

Para criar o banco:

```
sqlite3 db.db < create_db.sql
```

Para popular o banco:

```
python populate_db.py
```

Para operar o banco: a saïida desta parte deve ser (1) uma lista de todos os artistas vivos até 2019, (2) uma lista vazia de todos os artistas vivos até 2019, gerada após editarmos a data de morte de todos estes artistas para 2018, e (3) o nome do artista mais visitado, com o número de visitas.

```
python operate_db.py
```