Projeto Banco de Dados, 2021.1
Professor: Marcelo Iury
Filipe Trindade - Matrícula: 20190019912

Introdução

Esse documento é referente ao trabalho da cadeira de Banco de Dados, no período 2022.1, ministrada pelo professor Marcelo Iury, também autor da proposta do projeto.

No Documento, constam informações sobre o projeto e sua execução.

A proposta do Projeto

A proposta do Projeto, como divulgada pelo professor Marcelo Iuri, era a de fazer um Projeto de Banco de Dados de um cinema fictício chamado "Cinema Sauro". A descrição do Cinema Sauro foi dada da seguinte forma:

O Cinema Sauro funciona como 45 Salas Multiplex em um luxuoso Shopping Center de Sousa.

• Para cada sala do cinema existe um cronograma de filmes a serem exibidos na mesma, podendo um mesmo filme estar alocado em mais de uma sala, e podendo uma mesma sala exibir mais de um filme, em horários diferentes.

• Existe um cadastro de próximas estreias, e um sistema de compra antecipada de ingresso para estes filmes. Obviamente que também o ingresso para o mesmo dia é possível. Esta operação é realizada através de cartão de crédito, e é cobrada uma taxa extra de 10% em cima do valor da bilheteria

• O valor do ingresso varia de acordo com o dia da semana, e com as seguintes categorias: adulto, estudante, infantil, idoso e flamenguista. Neste caso, o flamenguista possui entrada gratuita, o infantil paga apenas 25%, o adulto paga entrada inteira, os demais pagam meia entrada.

• Uma pessoa pode comprar mais de um ingresso. E o este conjunto de ingressos podem ser para filmes, salas e horários distintos.

• Cada sala possui sua capacidade e a venda de ingressos deve estar limitada a isso.

• Os filmes dispõem de informações sobre os atores principais, a censura, a categoria (comédia, romance, ficção, ação, suspense...), sua censura e sua duração, a Empresa Produtora e se é Nacional ou não.

• O sistema tb deve dar suporte no que tange a compra de itens na lanchonete do cinema.

• Durante o processo de compra de ingresso, devem ser listadas as ofertas da lanchonete do dia. O comprador poderá incluir uma ou mais ofertas para a sua compra.

• Para cada ingresso ou oferta comprada, devem ser impressos vouchers ao final da compra.

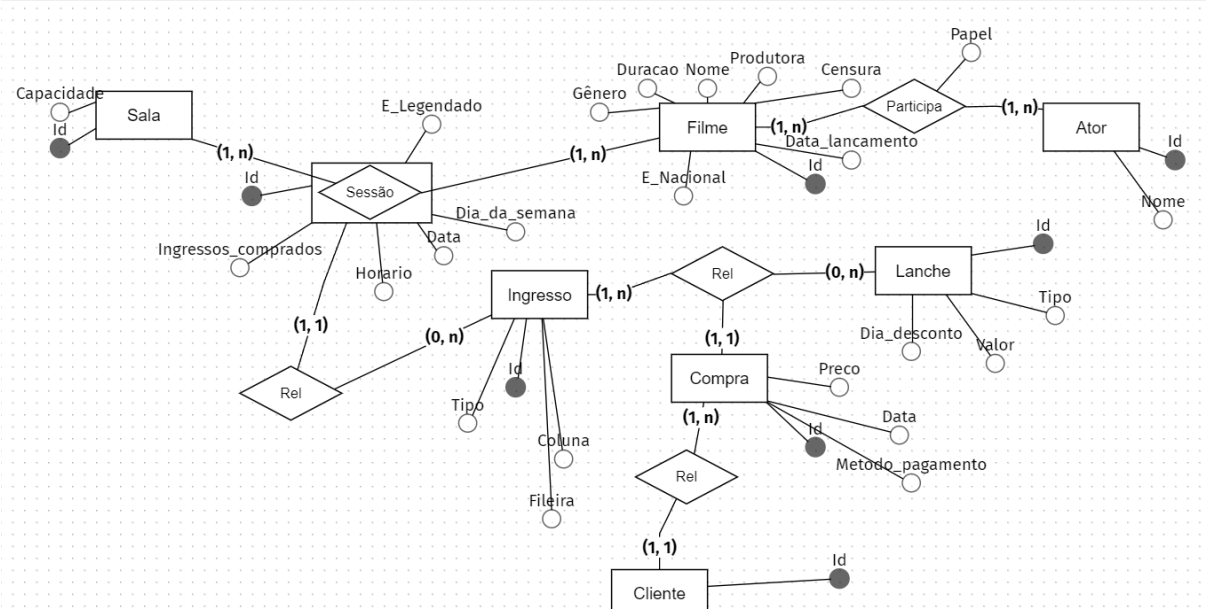
O Projeto seria então dividido em duas partes. A primeira delas seria referente à criação dos modelos conceitual e relacional do banco de dados, assim como fatores como consultas e etc. Já a seguinte etapa dizia respeito à implementação do sistema, assim como de dados necessários a consultas eventuais e exemplares, que permitam o uso do sistema. Era necessário a existência de interface para o uso do banco, e ele deveria estar bem documentado.

Primeira Etapa

A Primeira etapa consiste em, pela sua descrição:

“Primeira etapa: fazer o modelo e-r completo do sistema selecionado, definindo todas as entidades e relacionamentos necessários ao sistema, suas restrições de integridade e cardinalidades. Também devem especificar o modelo relacional do sistema selecionado, tomando como base o modelo conceitual e-r já previamente definido, mostrando os atributos chaves das tabelas, como também, demonstrando a normalização de todas elas. As restrições de integridade, domínio e entidade devem ser documentadas e apresentadas nesta etapa. Também devem estar listadas quais são as possíveis consultas e quais deverão ser implementadas como stored-procedures. “

Dada a descrição do Cinema Sauro, foi possível pensar nas seguintes entidades (assim como seus respectivos atributos) para um esquema conceitual:



Explicação do modelo entidade relacionamento:

Na prática, ainda que o Cinema Sauro pudesse ser uma entidade do sistema (ele possui salas e exibe filmes), não é necessário o colocar aqui. Sua existência já é implícita, e por ou não tal entidade não afetará o funcionamento do banco de dados.

A entidade Ator corresponde a um ator qualquer (visto que, na especificação, é necessário informar os tais de cada filme). Cada ator possui um id identificador (único) e também um nome. Atores podem se relacionar com a entidade Filme: um filme pode ter vários atores, e um ator pode estar em vários filmes. Em cada filme, um ator desempenha um papel (personagem que ele interpreta no filme).

A entidade Filme representa os filmes do Cinema Sauro. Cada filme possui um id identificador (único), um nome (título do filme), atributos booleanos que indicam se um filme é ou não legendado e é ou não nacional. Além disso, cada filme possui uma produtora, uma duração (em tempo), uma censura (livre, 10 anos, etc.) e também um gênero (que é um atributo multivalorado, visto que cada filme pode ter mais de um gênero). O filme também deve salvar a sua data de lançamento e sua data final para o Cinema Sauro.

Salas possuem um id identificador (que é simplesmente o número da sala, único) e também um atributo que indica a sua capacidade. Sala e Filme se relacionam quando ocorre uma Sessão. Sessão, que pode ser interpretada como uma entidade relacionamento. Ela possuirá de atributos um id identificador (único), uma data, um horário de início, uma referência ao dia da semana (pois ingressos para sessões em determinados dias são mais baratos) e também o número de ingressos comprados (para garantir que a quantidade vendida de ingressos não exceda a capacidade da sala).

A entidade Ingressos possui um id identificador (único), uma referência ao seu tipo (flamenguista, infantil, etc), um atributo para marcar a coluna e a fileira do ingresso (local na sala). Todo ingresso deve necessariamente pertencer a uma sessão, podendo uma mesma sessão vender múltiplos ingressos.

A entidade Lanche possui um id identificador (corresponde a uma unidade de lanche vendida, sendo esse único), um tipo (pipoca, refrigerante, etc.), um valor e um dia da semana que corresponde ao seu dia de desconto (seja segunda, terça, etc.).

Uma Compra é uma entidade que se relaciona com Lanche e Ingresso. Toda Compra deve conter ao menos 1 ingresso e pode conter lanche(s). Compras possuem como atributo um id identificador (único), data de ocorrência da mesma, o valor pago (preço) e também o método de pagamento (pois compras via cartão são mais caras). Toda Compra é feita por um único Cliente, representado por um atributo identificador id.

Já para as tabelas do Modelo Relacional, após normalização , onde as linhas sublinhadas são as chaves primárias:

Sala (id, capacidade)
Produtora (produtora_id, nome_produtora)
Filme (filme_id, nome_filme, censura, duração, produtora_id, e_nacional, data_lancamento, data_final)
    produtora_id referencia Produtora
Genero(genero_id, nome_genero)
Filme_Genero(genero_id,filme_id)
    genero_id referencia Genero

filme_id referencia Filme

Ator (ator_id, nome_ator)

Ator_Filme (ator_id, filme_id , papel)

ator_id referencia Ator

filme_id referencia Filme

Sessão (sessao_id, data, horario,filme_id, sala_id, n_ingressos_comprados,e_legendado)

filme_id referencia Filme

sala_id referencia Sala

Ingresso_tipo (tipo_id, nome_tipo, preço, valor_desconto, dia_desconto)

Ingresso (ingresso_id, fileira, coluna, tipo_id, sessao_id)

sessao_id referencia Sessão

tipo_id referencia Ingresso_tipo

Cliente (cliente_id)

Compra_Pagamento (pagamento_id, tipo_compra)

Compra (compra_id, data, horário,valor,forma_pagamento, cliente_id)

forma_pagamento referencia Compra_Pagamento

cliente_id referencia Cliente

Lanche_tipo (tipo_id, nome, preço, valor_desconto, dia_desconto)

Lanche (tipo_id, valor, lanche_id)

tipo_id referencia Lanche_tipo

Compra_Possui (compra_id , ingresso_id , lanche_id)

compra_id referencia Compra

lanche_id referencia Lanche

ingresso_id referencia Ingresso

A normalização torna necessário o surgimento de novas tabelas;

Como um filme pode ter múltiplos gêneros, foi necessário a criação de outras tabelas, uma para armazenar os diferentes gêneros possíveis, e outra para indicar o relacionamento de um filme e um gênero.

Criou-se uma tabela para armazenar informações de Produtora (nome, no caso), usada de referência em filme para indicar a produtora de cada filme. A tabela Lanche Tipo surge com um propósito semelhante, pois cada tipo de lanche possuirá um nome, um preço, um valor de desconto e um dia de desconto. Lógica também semelhante sendo usada para Ingresso_tipo e Compra_pagamento.

A tabela Ator_Filme indica o relacionamento entre um ator e um filme.

Dentre as consultas possíveis, das que se fazem relevantes para as funcionalidades do Cinema Sauro, podem ser indicadas as seguintes:

Vou evitar aqui repetir procedimentos que sejam muito parecidos entre si, pois eles seriam triviais e redundantes de apresentar, o que acho que foge um pouco do ponto de apresentação do trabalho.

Primeiramente, podem ser usados stored procedures que facilitem a inserção nas tabelas. Particularmente, no que se refere a não ter que, manualmente, escolher IDs para cada nova inserção. Podem ser criados procedimentos assim para boa parte dos itens da tabela, ainda que os parâmetros para cada um vão variar. Basicamente, busca-se via Select

o maior id na tabela, e o novo id será o maior + 1. Um exemplo disso, usando a tabela Ator, seria:

```
create procedure insert_ator(nome char)
language 'sql'
as $body$
declare
novo_id integer;
begin
    novo_id :=0;
    novo_id := (select max(id_ator) from Ator)+1;
    insert into Ator(novo_id,nome);
end;
```

Um procedimento para selecionar as informações (elenco, produtora, etc) de um certo filme, a partir de seu id dado:

```
create procedure select_filme_viaid(id_buscado integer)
language 'sql'
as $body$
declare
a2 integer;
begin
    Select (nome_filme, duração, censura, e_nacional) from Filme where filme_id = id_buscado;
    Select ator_id, papel from Ator_Filme where filme_id = id_buscado;
    a2 = Select produtora_id from Filme where filme_id = id;
    Select * from Produtora where produtora_id = A2 ;
end
```

Um procedimento que mostra quais filmes estarão em cartaz, para uma certa data:

```
create procedure filmes_em_cartaz(data_buscada date)
language 'sql'
as $body$
declare
a1 integer; variavel record;
begin
    a1 select filme_id from Filme where data_lancamento<= data_buscada and data_final >= data_buscada ;
    for variavel in Select * from a1 loop
        select_sessao_viaid(variavel.sessao_id);
    end loop
end;
```

- Podem ser também criados procedimentos que façam busca via nome, para tabelas que possuem um char de nome. Isso não é recomendável, pois nomes podem ser partilhados, o que poderia retornar inúmeros resultados. Ainda sim, tal operação poderia ser útil para fazer busca em diversas tabelas. Um possível exemplo disso, usando a tabela ator, seria:

```
create procedure select_ator_vianome(nome_buscado char)
language 'sql'
as $body$
declare
begin
    select * from Ator where nome_ator like %nome_buscado%;
end
```

- Selecionar as informações de uma certa sessão, a partir do id dela:

```
create procedure select_sessao_viaid(id_buscado integer)
language 'sql'
as $body$
declare
a1 integer;
varRegistro record;
begin
    a1 = Select filme_id from Sessao where sessao_id = id_buscado;
    Select (nome_filme, duração, censura) from Filme where filme_id = a1;
    Select sala_id, data, horario, e_legendado from Sessao where sessao_id = id_buscado;
end
```

- Exibir todos as sessões do Cinema Sauro, para um certo dia:

```
create procedure exibe_sessoes_dia(data_buscada date)
language 'sql'
as $body$
declare
a1 integer; variavel record;
begin
    a1 = Select sessao_id from Sessao where data = data_buscada;
    for variavel in Select sessao_id from a1 loop
        select_sessao_viaid(variavel.sessao_id);
    end loop
end;
```

- O mesmo do anterior, mas sessões de um filme específico:

```
create procedure exibe_sessoes_dia_filme(data_buscada date, id_filme integer)
language 'sql'
as $body$
declare
a1 integer;variavel record;
begin
    a1 = Select sessao_id from Sessao where data = data_buscada and filme_id = id_filme;
    for variavel in Select sessao_id from a1 loop
        select_sessao_viaid(variavel.sessao_id);
    end loop
end;
```

- Filtre as sessões de um determinado dia com filmes nacionais:

```
create procedure exibe_sessoes_dia_nacionais (data_buscada date)
language 'sql'
as $body$
declare
a1 integer; a2 integer; variavel record;
begin
    a1 = select filme_id from Filme where e_nacional = 1;
    for variavel in select* from a1 loop
        exibe_sessoes_dia_filme (data_buscada, variavel);
    end loop
end
```

- Filtrar sessões com filmes legendados (pode-se mudar facilmente o procedimento para ter filmes não legendados):

```
create procedure exibe_sessoes_dia_legendado (data_buscada date)
language 'sql'
as $body$
declare
a1 integer; variavel record;
begin
    a1 = select sessao_id from Sessao where e_legendado = 1;
    for variavel in select * from a1 loop
        select_sessao_viaid(variavel.sessao_id);
    end loop
end
```

- Exibir as sessões de um certo dia que se encaixam em certa classificação indicativa:

```
create procedure exibe_filme_classificacao (data_buscada date, idade integer)
language 'sql'
as $body$
declare
a1 integer;  variavel record;
begin
    a1 = select filme_id from Filme where censura <= idade;
    for variavel in select * from a1 loop
        exibe_sessoes_dia_filme(data_buscada,variavel);
    end loop
end
```

- Retornar a tabela de preço de ingressos:

```
create procedure exibe_ingressos()
language 'sql'
as $body$
declare
begin
    select nome_tipo, preco, valor_desconto, dia_desconto from Ingresso_tipo;
end
```

- Retornar a tabela de preço de itens da lanchonete:

```
create procedure exibe_lanches()
language 'sql'
as $body$
declare
begin
    select nome, preco, valor_desconto, dia_desconto from Lanche_tipo;
end
```

- Retornar quais itens da lanchonete estão em promoção no dia atual:

```
create procedure lanche_promocao_dia(data_atual date)
language 'sql'
as $body$
declare
begin
    select nome, preco, valor_desconto from Lanche_tipo where dia_desconto = data_atual;
end
```

- Criar um ingresso para uma sessão, para um assento específico:

```
create procedure insere_ingresso(sessao integer, coluna_buscada integer, fileira_buscada integer, tipo integer)
language 'sql'
as $body$
declare
a1 integer;
begin
    if (not exists(select * from Ingresso where sessao_id = sessao and coluna = coluna_buscada and fileira = filei
        a1 := select max(ingresso_id) from Ingresso; a1 := a1+1;
        insert into ingresso (a1, fileira, coluna,tipo, sessao);
        raise notice ' ingresso de id %', a1;
    else
        raise notice ' assento ja ocupado '
end
```

- Inserir itens em uma compra:

```
create procedure insere_ingresso_compra(compra integer, ingresso integer)
language 'sql'
as $body$
declare
a1 float; a2 float;
begin
    insert into Compra_Possui (compra, ingresso, null);
    a1 = select valor from Compra where compra_id = compra;
    a2 = select preco from Ingresso where ingresso_id = ingresso;
    insert into Compra(valor) values(a1 + a2) where compra_id = compra;
end

create procedure insere_lanche_compra(compra integer, lanche integer)
language 'sql'
as $body$
declare
a1 float; a2 float;
begin
    insert into Compra_Possui (compra, null,lanche);
    a1 = select valor from Compra where compra_id = compra;
    a2 = select valor from Lanche where lanche_id = lanche;
    insert into Compra(valor) values(a1 + a2) where compra_id = compra;
end
```

```
create procedure insere_ambos_compra(compra integer, ingresso integer, lanche integer)
language 'sql'
as $body$
declare
a1 float; a2 float; a3 float;
begin
    insert into Compra_Possui (compra, ingresso,lanche);
    a1 = select valor from Compra where compra_id = compra;
    a2 = select preco from Ingresso where ingresso_id = ingresso;
    a3 = select valor from Lanche where lanche_id = lanche;
    insert into Compra(valor) values(a1 + a2 + a3) where compra_id = compra;
end
```

- Ver informações de uma compra:

```
create procedure checa_compra(compra integer)
language 'sql'
as $body$
declare
begin
    select * from Compra where compra_id = compra;
    select * from Compra_Possui where compra_id = compra;
end
```

Quanto às restrições do sistema e suas tabelas:

Todas as tabelas apresentadas possuem atributos essenciais para o funcionamento do cinema, e, com exceção de Compra_Possui, não devem conter atributos nulos

Todos os ids (chaves primárias) são únicos, inteiros não negativos

Se uma produtora ou gênero forem removidos, filmes deve ser atualizados de acordo com

Toda sessão só pode ocorrer numa data após a data de lançamento do filme

Todo filme deve possuir ao menos um gênero

Todo filme deve ter uma produtora

As sessões não podem vender mais ingressos do que a capacidade da sala permite

Segunda etapa

A segunda etapa consiste em, pela sua descrição:

“Segunda etapa: o sistema final deverá estar implementado usando uma linguagem de programação e sgbd de sua preferência. Todas as funcionalidades básicas do sistema de compra, pesquisa, remoção e cadastro de filmes/ofertas deverão estar funcionando. Projetos sem a devida aplicação funcional serão rejeitados sumariamente. É verdade esse Bilhete!"