Cinema Manager

Gestão de cinemas

Descrição

O objetivo do programa que está a ser desenvolvido é facilitar a gestão de vários aspetos de um cinema, nomeadamente a gestão de clientes, de funcionários (quer trabalhadores ou supervisores), salas e as suas respetivas sessões, filmes em exibição e acompanhamento da venda de bilhetes e registos de limpeza.

Usos

O programa está a ser desenvolvido de forma a poder incluir vários cinemas; dentro de cada cinema todos os funcionários podem utilizar funcionalidades diferentes dependendo da sua função no mesmo. *Os Managers* têm acesso a todas as funcionalidades, enquanto um funcionário de limpeza só tem acesso à gestão de registos de limpeza; por último, os vendedores de bilhetes têm acesso às salas e sessões do seu cinema, aos filmes em exibição e ao histórico de venda de bilhetes, podendo realizar a venda dos mesmos. O programa também é esperado oferecer a possibilidade do público consultar informações sobre os filmes em exibição e as sessões a decorrer.

Análise de requisitos

Dentro de um cinema existem diferentes tipos de funcionários, nomeadamente vendedores de bilhetes, funcionário de limpeza e gerentes. Cada gerente vai estar encarregue de um cinema; este cinema possui várias sessões de filmes; essas ditas sessões têm várias instâncias possíveis, sendo cada uma dessas associada a uma sala do cinema; um vendedor pode vender um bilhete para uma sessão de sessão a um cliente. Por último, entre as sessões os funcionários de limpeza devem limpar as salas do cinema, deixando depois um registo de limpeza.

Cada funcionário deve ter a si associado um número único, o seu nome, turno e função.
Os filmes que o cinema exibe devem estar associados à sua key respectiva de imdb, o seu nome oficial, tempo total e realizador.
As sessões do filmes têm um ID único e é explícita a data em que começam e o número de semanas que devem estar em exibição. As instâncias dessas sessões devem explicitar a hora que começam.
Os bilhetes vendidos pelos vendedores têm a si associados um ID e o seu preço. Os clientes que os compram também vão ter um ID único e também o seu nome completo, email e data de nascimento.
Por último, os registos de limpeza devem indicar a data e hora da limpeza realizada.

Diagrama Entidade/Relação

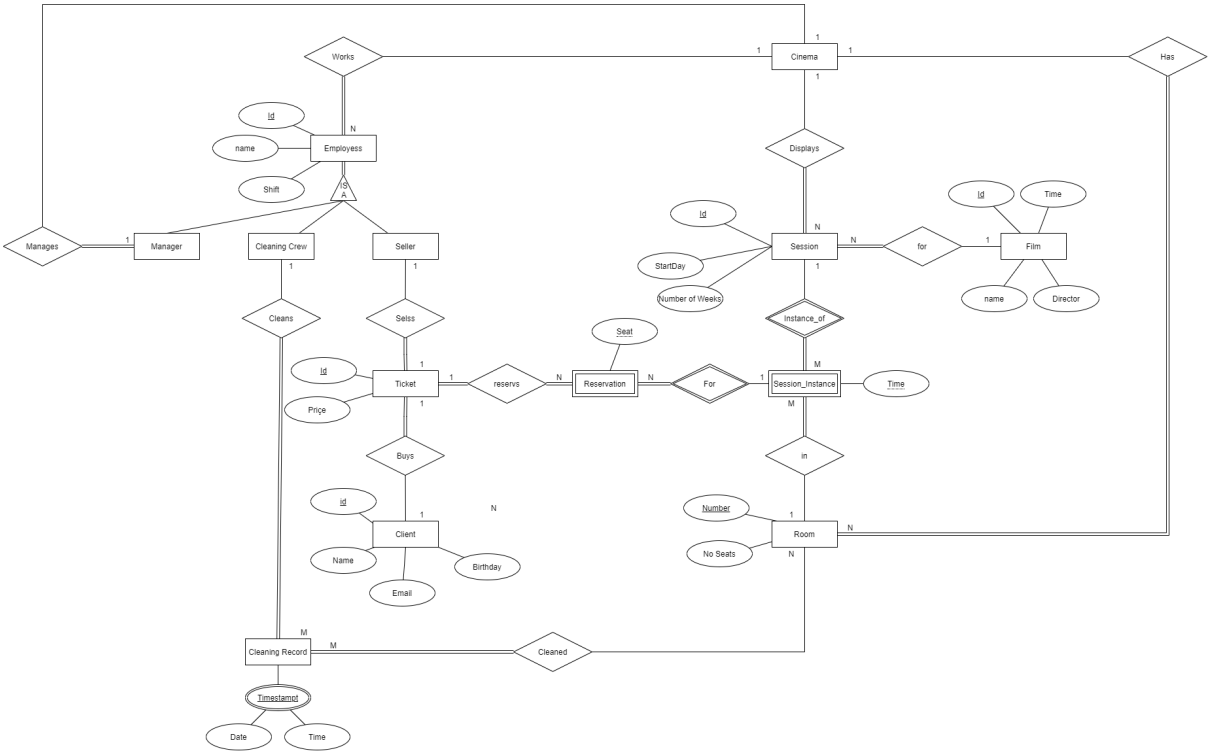
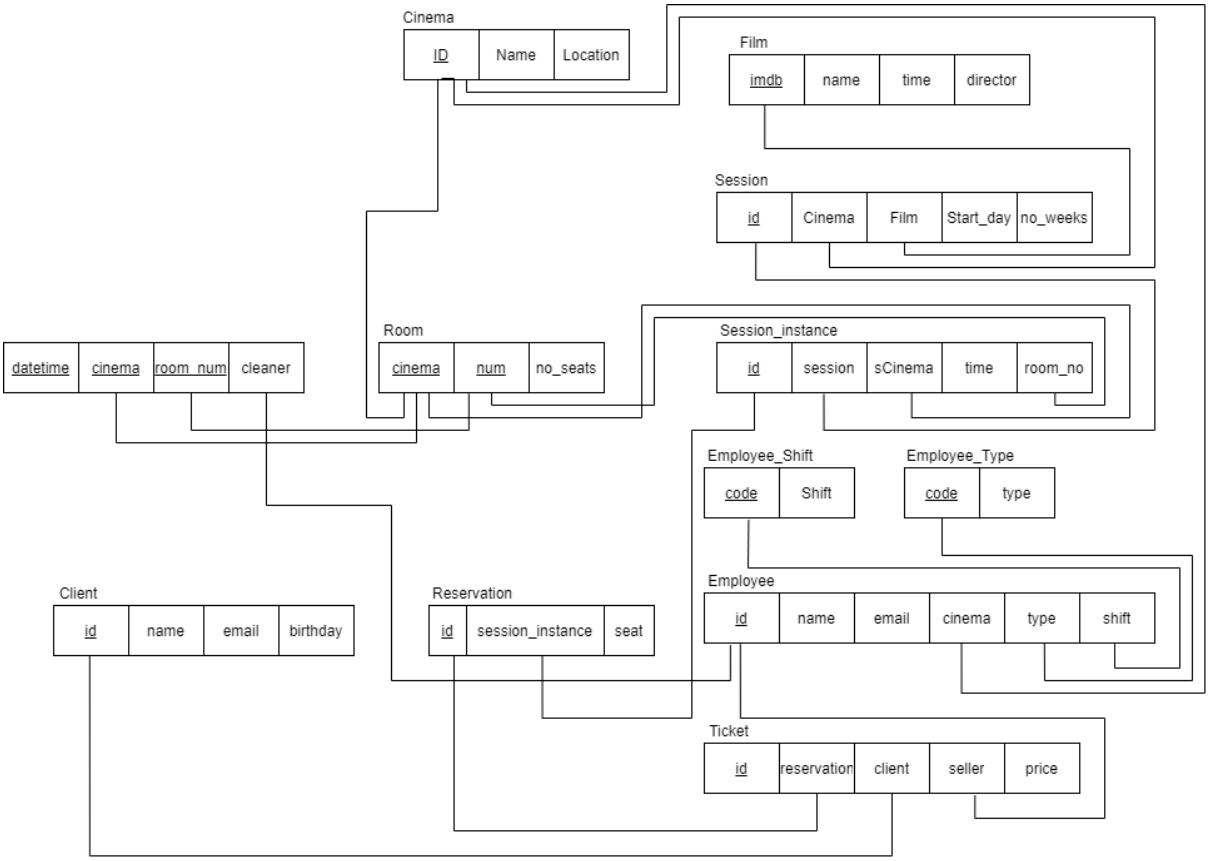


Imagem grande: https://imgur.com/4mxWkLL

Esquema Relacional



SQL DDL

Schemas e as suas tabelas

- **Data:** Film, Session
- **Locations:** Cinema
- **Operations:** Cleaning_Record, Client, Reservation, Session_instance, Ticket
- **Management:** Employee, Employee_Shift, Employee_Type, Room

Exemplos de criação de tabelas

```
create table Location.Cinema(
        id              int             Primary Key IDENTITY(0,1),
        name    varchar(max) not null,
        location varchar(max) not null
);
```

Schema: Location, Tabela: Cinema

```
create table Management.Room(
        num             int not null,
        cinema  int not null,
        nSeats  int not null,

        Primary Key(cinema, num),

        CONSTRAINT FK_room_cinema FOREIGN KEY (cinema)
      REFERENCES Location.Cinema(id)
      ON UPDATE CASCADE
          ON DELETE Cascade
);
```

Schema: Management, Tabela: Room

```
create table Data.Session(
        id              int Primary Key IDENTITY(0,1),
        cinema  int not null,
        filmId  int not null,
        wDay    int not null, -- represents one day of the week
        dTime   time not null
        CONSTRAINT FK_session_filmid FOREIGN KEY (filmId)
      REFERENCES Data.Film(imdb)
      ON UPDATE CASCADE,
          -- on delete

        CONSTRAINT FK_session_cinema FOREIGN KEY (cinema)
      REFERENCES Location.Cinema(id)
      ON UPDATE CASCADE
          On Delete Cascade
);
```

Schema: Data, Tabela: Session

```
create table Operations.Reservation(
        seat            int     not null,
        session         int not null,
        sCinema         int not null,
        sNum            int not null,

        Primary Key(seat, session, sCinema, sNum),

        CONSTRAINT FK_reservation_session FOREIGN KEY (session, sCinema, sNum)
      REFERENCES Operations.Session_instance(session, sCinema, sNum)
      ON UPDATE CASCADE
          ON DELETE Cascade,
);
```

Schema: Operations, Tabela: Reservation

Eliminação de tabelas

```
drop table operations.Ticket;
drop table operations.Client;
drop table operations.Reservation;
drop table operations.Session_instance;
drop table operations.Cleaning_Record;


drop table management.Room;
drop table management.Employee;
drop table management.Employee_Type;
drop table management.Employee_Shift;

drop table data.Session;
drop table data.Film;

drop table location.Cinema;
```

SQL DML

Exemplos de Inserts

```
insert into location.Cinema values ('Glicinias Aveiro', 'Aveiro');
insert into location.Cinema values ('Palacio do gelo', 'Viseu');
insert into location.Cinema values ('Forum Viseu', 'Viseu');
insert into location.Cinema values ('Forum Coimbra', 'Coimbra');
```

Inserção de alguns cinemas, indicando o seu nome e a sua localização.

```
insert into data.Film values (9599292, 'O Pai', 7, 'Florian Zeller');
insert into data.Film values (11083552, 'Um Homem Furioso', 119, 'Guy Ritchie');
insert into data.Film values (1321510, 'Ao Ritmo de Washington Heigths', 143, 'Jon M. Chu');
insert into data.Film values (8368512, 'O Espião Inglês', 112, 'Dominic Cooke');
insert into data.Film values (11169050, 'Supernova', 95, 'Harry Macqueen');
```

Inserção de alguns filmes

```
-- quiet place 2: palacio gelo viseu
insert into data.Session values (1, 8332922, 1, '19:50');
insert into data.Session values (1, 8332922, 1, '22:20');
insert into data.Session values (1, 8332922, 2, '19:50');
insert into data.Session values (1, 8332922, 2, '22:20');
```

Inserção de várias sessões da exibição do filme “Quiet Place 2” no cinema "Palácio Gelo, Viseu”"

```
insert into management.Employee_Type values ('Manager');
insert into management.Employee_Type values ('Deputy Manager');
insert into management.Employee_Type values ('Sales');
insert into management.Employee_Type values ('Cleaning');
```

Inserção dos diferentes tipos de empregado

```
    -- full time shifts
insert into management.Employee_Shift values ('8:00', '18:00');

insert into management.Employee_Shift values ('8:00', '17:00');
insert into management.Employee_Shift values ('17:00', '2:00');
    -- part time shifts
insert into management.Employee_Shift values ('8:00', '12:30');
insert into management.Employee_Shift values ('12:30', '17:00');
```

Inserção de alguns dos turnos

```
-- glicinias aveiro
insert into management.Room values (1, 0, 114);
insert into management.Room values (2, 0, 102);
insert into management.Room values (3, 0, 120);
insert into management.Room values (4, 0, 114);
insert into management.Room values (5, 0, 92);
insert into management.Room values (6, 0, 146);
```

Inserção das salas do cinema "Glicinias, Aveiro”

UDFs

Foram criados UDFs (Inline Table) de forma a criar uma camada de abstração no acesso aos dados.
Para todas as entidades foi criada uma udf que retorna os conteúdos nas tabelas.
No entanto foi necessário criar UDFs com mais alguma lógica para o acesso a sessões a decorrer e filmes em exibição em uma data e num cinema

Resumo de alguns udfs mais utilizados:

Data Schema

f_get_film(imdb): retorna o filme com id 'imdb';
f_get_session(sessionId): retorna a sessão cujo id = sessionId;

Locations Schema

f_get_cinemas(): retorna todos os cinemas;

Management Schema:

f_get_employees(): retorna todos os employees;
f_get_employee(id): retorna um employee com id=id;
f_get_sellers(): retorna uma lista de employees que são vendedores;

Operations Schema

f_get_clients(): retorna todos os clientes;
f_get_session_instance_w_seats(date): retorna session_instance com informação relativa ao número de lugares na sala onde a sessão decorre e contagem do número de reservas para a sessão.

Public Access Schema

f_get_open_sessions(date): lista de sessões em cartaz numa determinada data;
f_get_open_sessions_cinema(date, cinema): lista de sessões em cartaz num cinema numa determinada data;
f_get_films_exibition(date): retorna lista de filmes em cartaz;
f_get_films_exibition_cinema(date, cinema): retorna lista de filmes em cartaz num cinema;

Stored Procedures

Os Stored Procedures foram principalmente usados para inserção de dados.

De forma a garantir estabilidade de dados, Procedures que no decorrer da sua execução necessitavam de acesso a mais do que 1 tabela, foram implementados recorrendo a transações.

Data Schema

p_new_film(Film): cria um novo filme na DB;
p_new_sessions(cinema, film, startday, noweeks): cria uma nova sessão para um filme em um cinema que começa em startDay e está em cartaz noweeks semanas na DB;
p_delete_session(id): elimina uma sessão da DB;

Locations Schema

p_new_cinema(Cinema): cria um novo cinema;
p_delete_cinema(id): elimina um cinema;

Management Schema

p_new_employee(Employee):cria um novo funcionario na DB;
p_new_room(Room): cria uma nova sala na DB;

Operations Schema

p_new_client(Client): cria um novo cliente;
p_delete_client(id): elimina um cliente;
p_new_cleaning_record(Cleaning_Record): cria um novo registo de limpeza para uma sala;
p_new_reservation(Reservation): cria uma nova reserva;
p_new_ticket(Ticket): cria um novo bilhete para uma sessão e cliente;

Grupo 8 - Turma 3

- João Martins - 93183 - joaofmartins@ua.pt
- Pedro Coutinho - 93278 - pmacoutinho@ua.pt