deti universidade de aveiro
departamento de eletrónica,
telecomunicações e informática

MIECT - Bases de Dados

Projeto

Gestão de Eventos para uma Sala de Concertos

João Gameiro nº 93097 Pedro Abreu nº 93240

P2G5

Análise de Requisitos

Para o estudo sobre este tema falámos com vários músicos e tentamos identificar alguns problemas com a organização e planeamento de eventos.

Problema

A organização de eventos pode ser um verdadeiro problema, tanto para organizadores como para os artistas que vão atuar. Muitas vezes ocorrem problemas como por exemplo: os organizadores não sabem o material que os músicos vão trazer, os artistas não saberem a que horas vão tocar, entre outros. Um evento tem de ser algo bem planeado em que todos os seus intervenientes sabem o que se está a passar a qualquer hora do mesmo, o que muitas vezes não acontece, originando atrasos e problemas que prejudicam a experiência do público.

Oportunidade

Surge assim uma oportunidade de desenvolver uma base de dados para uma sala de concertos aonde se podem realizar eventos. Esta base de dados permitirá guardar todos os dados necessários para que um evento decorra sem qualquer tipo de problemas. Deve permitir que todos os intervenientes do evento a possam consultar com o objetivo de se manterem informados em relação ao que acontece no evento e minimizar a probabilidade de atrasos e problemas.

Requisitos

- Os utilizadores deste sistema devem conseguir:
    - Inserir/Remover concertos e toda a informação associada aos mesmos (hora de começo, soundcheck, artistas, ...)
    - Inserir/Remover informação relativa ao material que os artistas trazem
    - Inserir/Remover informação relativamente às comitivas que acompanham os artistas
    - Inserir/Remover informação relativamente às refeições que vão ser servidas e as empresas de catering que o vão fazer
    - Inserir/Remover informação relativamente ao stage manager e promotor do evento
    - Configurar informação do próprio evento (número de bilhetes vendidos, datas, ...)
    - Consultar toda a informação inserida.

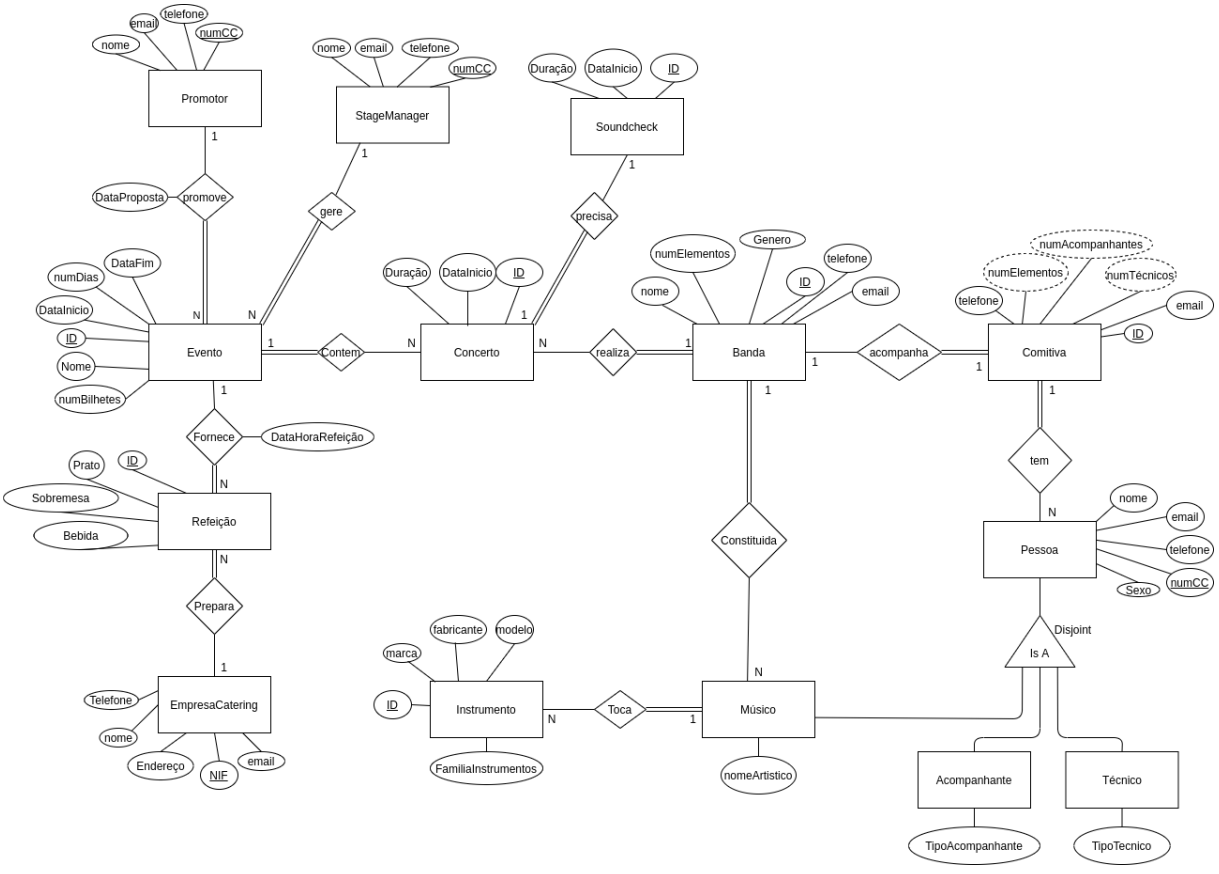
Diagrama Entidade-Relacionamento

- Um **Evento** é caracterizado por um ID, nome, datas de início e fim e tem a si associado um número de bilhetes vendidos.
- Um **StageManager** é responsável pela gestão e coordenação de tudo o que acontece em palco durante o evento.
- Um **Promotor** promove um evento e tal como **StageManager** e **Pessoa** é caracterizado por nome, telefone, email e número de Cartão de Cidadão.
- Um **Concerto** é caracterizado por uma duração, data de início. É dado por uma **Banda** que é caracterizada por um nome, número de elementos, ID, telefone, email e género musical (Rock,

Pop, Rap, Metal, Jazz, Clássico). Um concerto precisa de um **Soundcheck** que tem a si associado uma duração, data início e ID.

- Uma **Banda** pode ser acompanhada por uma **Comitiva**, que é constituída por **Pessoas** e caracterizada por um ID, email, telefone e número de elementos.
- Uma **Pessoa** pode ser um **Músico**, um **Acompanhante** ou um **Técnico** e é caracterizado por um nome, email, número de Cartão de Cidadão, telefone e sexo.
- Um **Músico** tem um nome artístico e toca um ou mais I**nstrumentos** caracterizados por um ID, fabricante, marca ou família (Sopro, Percussão ou Cordas).
- Um **Acompanhante** pode ser familiar, amigo ou namorada(o). Um **Técnico** por ser de Som, de Guitarras, de Baixos, Bateria ou de Luzes.
- Um **Evento** fornece aos seus intervenientes várias **Refeições** caracterizadas por um ID, prato principal (carne, peixe, vegetariano, vegan), sobremesa (bolo, fruta, gelado, café) e uma Bebida (cerveja, água, refrigerante).
- As refeições são preparadas por uma **Empresa de Catering** caracterizada por um nome, NIF, endereço, email e telefone.

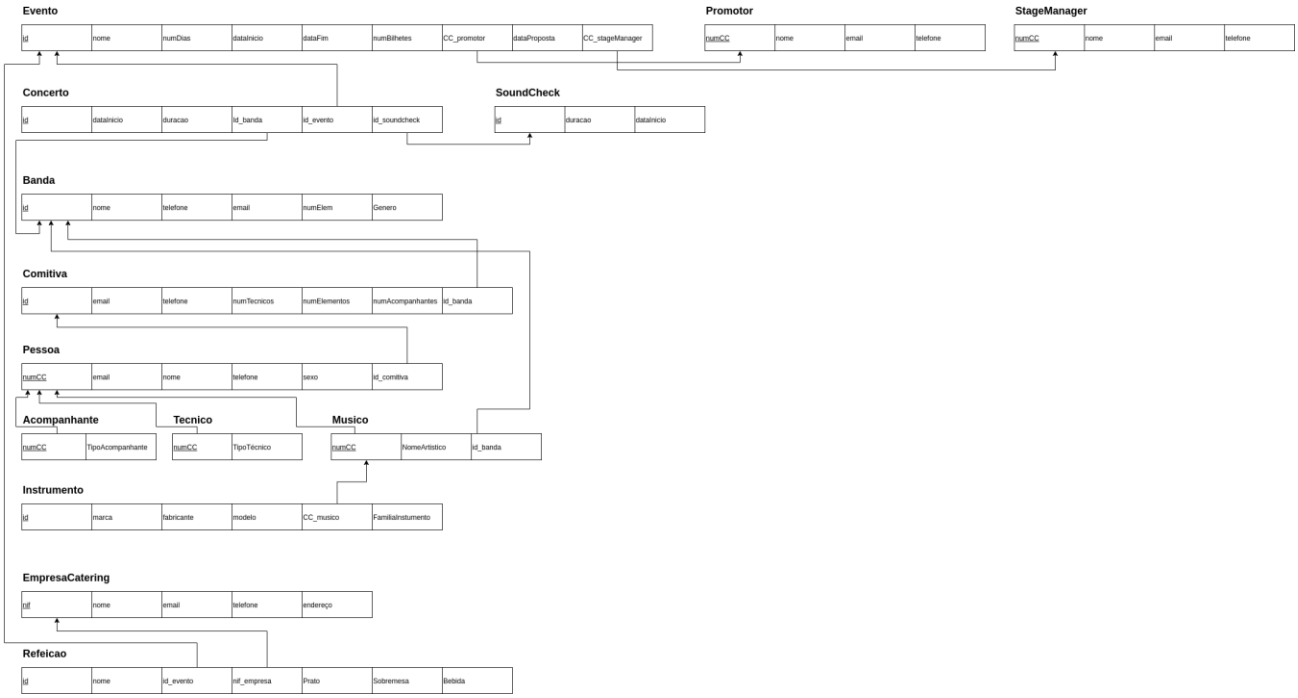
Devido ao facto de o diagrama ser de tamanho elevado, segue em anexo um ficheiro com extensão jpg com o mesmo para que possa ser possível a sua consulta de forma mais detalhada.



Após a primeira entrega algumas alterações foram feitas ao diagrama entidade-relacionamento, entre as quais:

- Remover todos os atributos multi-valor e após uma discussão, transformá-los em atributos normais
    - Prato, Sobremesa, Bebida em Refeição
    - Género em Banda
    - TipoAcompanhante em Acompanhante
    - TipoTécnico em Técnico
    - FamiliaInstrumentos em Instrumento
- Transformar atributo numAcompanhantes e numTécnicos na tabela Comitiva em atributos derivados

Esquema Relacional



Após a 1ª entrega, as alterações feitas no diagrama entidade-relacionamente resultaram em alterações ao diagrama relacional entre as quais:

- Remover todas as tabelas originadas pelos atributos multi-valor
    - Remover tabelas Prato, Sobremesa Bebida
    - Remover tabela Género
    - Remover tabela TipoAcompanhante
    - Remover tabela TipoTécnico
    - Remover tabela FamiliaInstrumentos
- Adicionar atributos correspondentes aos atributos multi-valor eliminados nas tabelas

Normalização

Após uma discussão e análise do esquema relacional chegámos à conclusão que não seriam necessárias alterações ao diagrama relacional. Pois a maioria das das tabelas por opção de desenho do SGDB foram defenidas pela chave primária id constituída por caracteres alfanuméricos. Nas tabelas EmpresaCatering(NIF), Pessoa(numCC) nós achamos que a tabela estava numa forma estável, e não sofreu nenhuma alteração.

SQL Scripts

Todo o código SQL desenvolvido para o projeto encontra-se presente no diretório SQL_scripts.

Setup

O ficheiro setup_EVENTMANAGEMENT.sql contém o código sql que permite a criação das tabelas do projeto, descritas anteriormente no diagrama relacional.

Drop

O ficheiro drop_EVENTMANAGEMENT.sql contém o código sql que permite a eliminação das tabelas do projeto, descritas anteriormente no diagrama relacional.

Inserts

O ficheiro inserts_EVENTMANAGEMENT.sql contém o código sql que permite a inserção de dados nas tabelas do projeto, descritas anteriormente no diagrama relacional.

Querys

Numa fase inicial do projeto foi desenvolvido um conjunto de querys. Estes querys estão descritos no ficheiro querys_EVENTMANAGEMENT.sql.

Views

O ficheiro views.sql contém o código sql que permite a criação das views do projeto.

Foram defenidas 4 views, V_OVERVIEW, V_CONCERTOS, V_BANDAS, V_MUSICOS, para juntar atributos de várias tabelas, por exemplo a V_OVERVIEW foi criada para ter uma noção geral da sala de espetáculos.

Triggers

O código SQL que contém os triggers do projeto encontra-se no ficheiro trigger_EVENTMANAGEMENT.sql.

Os triggers desenvolvidos para a Base de Dados foram:

- simultaneous_events (não podem existir dois eventos a acontecer ao mesmo tempo)
    - Tabela EVENTO
    - Tipo Instead Of Insert, caso exista sobreposição de datas, é lançada uma exceção e o processo de inserção não ocorre.
    - Implementado com recurso a um cursor para iterar sobre todas as datas de todos os eventos para a verificação se são simultâneas à que vai ser inserida ou não
- simultaneous_concerts (não podem existir dois concertos a acontecer ao mesmo tempo)
    - Tabela CONCERTO
    - Tipo Instead Of Insert, e tal como o anterior é lançada exceção se houver sobreposição de datas.
    - Implementado com recurso a um cursor para iterar sobre todas as datas de todos os concertos
    - Também é verificado se a data do concerto é dentro dos limites do evento (concerto tem de estar entre a data início do evento e data fim)
- simultaneous_soundchecks (não podem existir dois soundchecks a acontecer ao mesmo tempo)
    - Tabela SOUNDCHECK
    - Tipo Instead Of Insert, e tal como o anterior é lançada exceção se houver sobreposição de datas.
    - Implementado com recurso a um cursor para iterar sobre todas as datas de todos os soundchecks
- concert_soundcheck (soundcheck tem de ser pelo menos duas horas antes do concerto)
    - Tabela CONCERTO
    - Tipo After Insert
    - Se não for pelo menos duas horas antes do concerto, o soundcheck é apagado
- soundcheck_duration (duração de um soundcheck tem de ser menor que duas horas)
    - Tabela SOUNDCHECK
    - Tipo After Insert
    - Se a duração for maior que duas horas, é ajustada para duas horas
- delete_event (Apagar um evento, apaga todos os seus concertos)
    - Tabela EVENTO
    - Tipo Intead of Delete
- delete_concert (Apagar um concerto, apaga todos o seu soundcheck)
    - Tabela CONCERTO
    - Tipo After Delete

Stored Procedures

O ficheiro sps_EVENTMANAGEMENT.sql contém o código sql que permite a criação de todos os procedures do projeto.

Os procedures do projeto foram criados com a intenção de que as operações de alteração dos dados da BD nunca deveriam ser executadas diretamente sobre a mesma, e em vez disso através da chamada a um conjunto de procedures que criavam uma espécie de camada de abstração entre a BD e as opções/atividades selecionadas através da interface de modo a não pôr em risco a integridade do conteúdo da base de dados.

Stored Procedure	Descrição
create_evento	Criar um evento. Recebe como parâmetros de entrada os dados necessários para criar uma entrada na tabela EVENTO. Implementado sem Transaction pois dentro do procedure é efetuado apenas um insert numa tabela.
create_concerto	Criar um concerto. Recebe como parâmetros de entrada os dados necessários para criar uma entrada na tabela CONCERTO e na tabela SOUNDCHECK. Implementado com uma Transaction pois são efetuados dois Inserts em duas tabelas diferentes.
delete_evento	Apagar um evento. Recebe como argumento o id do evento a apagar. Foi implementado com recurso a uma transação pois a instrução de apagar um evento apaga também os seus concertos associados, o que torna importante a atomicidade da operação.
delete_concerto	Apagar um concerto. Recebe como argumento o id do concerto e é implementado com recurso a uma transação pois a operação de delete evento apaga também o soundcheck associado.
alter_evento	Alterar um evento. Recebe como argumento todos os parâmetros de um evento. Aqueles que diferirem da entrada original são atualizados. Implementado com recurso a uma transação pois pode ser necessário fazer mais do que um update.
alter_concerto	Alterar um concerto. Recebe como argumento todos os parâmetros de um concerto. Só são alterados os que diferem da entrada original. Implementado com recurso a uma transação.

UDFs

O ficheiro udfs_EVENTMANAGEMENT.sql contém o código sql que permite a criação de todas as udfs usadas no projeto.

Tal como no caso anterior, foram criadas funções de consulta para que a interação com a base de dados não seja feita através de querys diretos sobre a base de dados, mas sim através da chamada a funções que os executam.

UDF	Descrição
getEventosById	Devolver eventos dado um ID de um evento

getEventosByNome	Devolver eventos dado um nome de um evento
getEventosByPromotor	Devolver Eventos dado o CC de um promotor
getEventosByStageManager	Devolver Eventos dado o CC de um StageManager
getConcertosByIDEvento	Devolver Concertos dado o ID de um evento
getConcertosSoundchecksByIdEvento	Devolver Concertos e respetivos soundchecks dado o ID de um evento
getBandasByIDEvento	Devolver Bandas dado o ID de um evento
getConcertosByIDBanda	Devolver Concertos dado o ID de uma banda
getConcertoByNomeBanda	Devolver Concertos dado o nome de uma banda
getMusicoByCC	Devolver Músicos dado um CC
getMusicoByName	Devolver Músicos dado um Nome
getBandaById	Devolver Bandas dado o ID de uma banda
getBandaByGenre	Devolver Bandas dado um género de música
getMusicosByIDBanda	Devolver Músicos dado o ID de uma banda
getEventosInBetween	Dadas duas datas, devolver os eventos que se encontram no intervalo formado pelas duas datas
getConcertosDuracaoInBetween	Dadas duas durações, devolver os concertos cuja duração se encontra no intervalo fornecido

Estas funções de consulta são usadas nos formulários de pesquisa quando se selecionam os filtros de pesquisa.

Indexes

O ficheiro indexes_EVENTMANAGEMENT.sql contém o código sql que permite a criação de todos os indexes usados no projeto.

Para este projeto foram criados 5 indexes

- Index na tabela Evento, coluna Id
- Index na tabela Concerto, coluna Id
- Index em Banda, coluna Id
- Index na tabela Banda, coluna Genre
- Index na tabela Músico, coluna Id

Para a escolha dos indexes, tivemos em conta pesquisas que são feitas frequentemente e se os atributos permitem valores repetidos. Como são feitas pesquisas pelas chaves primárias de várias tabelas (Id), criamos indexes em cada uma dessas. Finalmente decidimos criar também um index na coluna Genre da tabela Banda pois é um atributo que permite possui valores repetidos e são efetuadas pesquisas pelo mesmo.

Interface

Para a interface do projeto decidimos utilizar apenas um conjunto de tabelas que consideramos como sendo mais importantes ao invés de utilizar todas as do projeto. As tabelas escolhidas foram a EVENTO, CONCERTO, BANDA, MUSICO, SOUNDCHECK, STAGEMANAGER e PROMOTOR.