Projeto de Bases de Dados, Parte 3

Trabalho realizado por:

Nome	Número	Esforço
Bernardo Valente	87521	15 horas
Francisco Machado	87530	15 horas
João Felício	87542	15 horas

Grupo: 34

Turno: Quarta-feira, 15h - BD817957L05

Docente: Raquel Casteleiro

Projeto de Bases de Dados, Parte 3

Criação e População da Base de Dados

Criação da base de dados feita no ficheiro “schema.php”. Foram criadas as tabelas dadas no Anexo A (modelo relacional) do enunciado do projecto.

Certas *foreign keys* estão seguidas de ON DELETE CASCADE. Deve-se ao facto de quando removemos uma entrada de uma tabela queremos que todas as outras referências a essa entrada sejam eliminadas. Por exemplo, quando removemos um local, os eventos de emergência associados a esse local são eliminados.

Para a população da base de dados usámos o ficheiro “gen.py” de forma a automatizar as inserções nas tabelas. No fim do populate.sql que foi gerado tivemos ainda de adicionar umas entradas extras para que nenhuma das queries apresentasse resultados nulos.

SCHEMA.SQL

```
CREATE TABLE Camara(
numCamara integer not null unique,
primary key (numCamara));

CREATE TABLE Video(
dataHoraInicio timestamp not null,
dataHoraFim timestamp not null,
numCamara integer not null,
primary key (dataHoraInicio, numCamara),
foreign key (numCamara) references Camara);

CREATE TABLE SegmentoVideo(
numSegmento integer not null unique,
duracao integer not null,
dataHoraInicio timestamp not null,
numCamara integer not null ,
primary key (numSegmento, dataHoraInicio, numCamara),
foreign key (numCamara) references Camara);

CREATE TABLE Local(
moradaLocal varchar(80) not null unique,
primary key (moradaLocal));

CREATE TABLE Vigia(
moradaLocal char(80) not null ,
numCamara integer not null ,
primary key(moradaLocal, numCamara),
foreign key(moradaLocal) references Local ON DELETE CASCADE,
foreign key(numCamara) references Camara);

CREATE TABLE ProcessoSocorro(
numProcessoSocorro integer not null unique,
primary key (numProcessoSocorro));

CREATE TABLE EventoEmergencia(
numTelefone integer not null unique,
instanteChamada timestamp not null ,
nomePessoa varchar(80) not null unique,
moradaLocal varchar(80) not null,
numProcessoSocorro integer,
primary key (numTelefone, instanteChamada),
foreign key (moradaLocal) references Local ON DELETE CASCADE,
foreign key (numProcessoSocorro) references ProcessoSocorro ON
DELETE CASCADE);

CREATE TABLE EntidadeMeio(
nomeEntidade varchar(80) not null unique,
primary key (nomeEntidade));

CREATE TABLE Meio(
numMeio integer not null unique,
nomeMeio varchar(80) not null,
nomeEntidade varchar(80) not null unique,
primary key(numMeio, nomeEntidade),
foreign key(nomeEntidade) references EntidadeMeio ON DELETE
CASCADE);

CREATE TABLE MeioCombate(
numMeio integer not null unique,
nomeEntidade varchar(80) not null unique,
primary key(numMeio,nomeEntidade),
foreign key(numMeio) references Meio(numMeio) ON DELETE
CASCADE,
```

Projeto de Bases de Dados, Parte 3

foreign key(nomeEntidade) references EntidadeMeio(nomeEntidade)
ON DELETE CASCADE);

CREATE TABLE MeioApoio(
numMeio integer not null unique,
nomeEntidade varchar(80) not null unique,
primary key(numMeio, nomeEntidade),
foreign key(numMeio) references Meio(numMeio) ON DELETE CASCADE,
foreign key(nomeEntidade) references Meio(nomeEntidade) ON DELETE CASCADE);

CREATE TABLE MeioSocorro(
numMeio integer not null unique,
nomeEntidade varchar(80) not null unique,
primary key(numMeio, nomeEntidade),
foreign key(numMeio) references Meio(numMeio) ON DELETE CASCADE,
foreign key(nomeEntidade) references Meio(nomeEntidade) ON DELETE CASCADE);

CREATE TABLE Transporta(
numMeio integer not null unique,
nomeEntidade varchar(80) not null unique,
numVitimas integer not null,
numProcessoSocorro integer not null ,
primary key(numMeio, nomeEntidade, numProcessoSocorro),
foreign key(numMeio, nomeEntidade) references MeioSocorro ON DELETE CASCADE,
foreign key(numProcessoSocorro) references ProcessoSocorro(numProcessoSocorro) ON DELETE CASCADE);

CREATE TABLE Alocado(
numMeio integer not null ,
nomeEntidade varchar(80) not null ,
numVitimas integer not null, --TODO pode ser null?
numProcessoSocorro integer not null ,
primary key(numMeio, nomeEntidade, numProcessoSocorro),
foreign key(numMeio, nomeEntidade) references MeioApoio ON DELETE CASCADE,
foreign key(numProcessoSocorro) references ProcessoSocorro ON DELETE CASCADE);

CREATE TABLE Acciona(
numMeio integer not null ,
nomeEntidade varchar(80) not null ,
numProcessoSocorro integer not null ,
primary key(numMeio, nomeEntidade, numProcessoSocorro),
foreign key(numMeio) references Meio(numMeio) ON DELETE CASCADE,
foreign key(nomeEntidade) references Meio(nomeEntidade) ON DELETE CASCADE,
foreign key(numProcessoSocorro) references ProcessoSocorro(numProcessoSocorro) ON DELETE CASCADE);

CREATE TABLE Coordenador(
idCoordenador integer not null unique,
primary key(idCoordenador));

CREATE TABLE Audita(
idCoordenador integer not null unique,
numMeio integer not null ,
nomeEntidade varchar(80) not null ,
numProcessoSocorro integer not null ,
datahoraInicio timestamp not null ,
datahoraFim timestamp not null ,
primary key(idCoordenador, numMeio, nomeEntidade, numProcessoSocorro),
foreign key(numMeio, nomeEntidade, numProcessoSocorro) references Acciona ON DELETE CASCADE,
foreign key(idCoordenador) references Coordenador);

CREATE TABLE Solicita(
idCoordenador integer not null unique,
dataHoraInicioVideo timestamp not null ,
numCamara integer not null ,
dataHoraInicio timestamp not null,
dataHoraFim timestamp not null ,
primary key (idCoordenador,dataHoraInicioVideo,numCamara),
foreign key(idCoordenador) references Coordenador,
foreign key(dataHoraInicioVideo,numCamara) references Video);

SQL QUERIES

1.
```
SELECT numProcessoSocorro
FROM Acciona
GROUP BY numProcessoSocorro
HAVING COUNT(numProcessoSocorro) =
        (SELECT max(c)
        FROM
```

Projeto de Bases de Dados, Parte 3

```
                (SELECT numProcessoSocorro, COUNT(numProcessoSocorro) AS c
                FROM Acciona
                GROUP BY numProcessoSocorro) t);
2.
SELECT nomeEntidade
FROM Acciona NATURAL JOIN EventoEmergencia
WHERE instanteChamada > '2018-06-21 00:00:00'
AND instanteChamada < '2018-09-23 00:00:00'
GROUP BY nomeEntidade
HAVING COUNT(numprocessosocorro) =
        (SELECT MAX(c)
        FROM
                (SELECT nomeEntidade, COUNT(numProcessoSocorro) c
                FROM Acciona NATURAL JOIN EventoEmergencia
                WHERE instanteChamada > '2018-06-21 00:00:00'
                AND instanteChamada < '2018-09-23 00:00:00'

                GROUP BY nomeEntidade ) t);

3.
SELECT numProcessoSocorro
From Acciona NATURAL JOIN EventoEmergencia
WHERE numProcessoSocorro NOT IN
        (SELECT numProcessoSocorro
        FROM Audita)
AND moradaLocal = 'Oliveira do Hospital'
AND instanteChamada > '2018-01-01 00:00:00'
AND instanteChamada < '2018-12-31 00:00:00';

4. SELECT COUNT (numSegmento)
        FROM vigia NATURAL JOIN SegmentoVideo
        WHERE duracao > 60
                AND moradaLocal = 'Monchique'
                AND dataHoraInicio >= '2018-08-01 00:00:00'
                AND dataHoraInicio <= '2018-08-31 00:00:00';

5.
SELECT C.numMeio, C.nomeEntidade
FROM MeioCombate C
Where (c.numMeio, c.nomeEntidade) not in
        (select A.numMeio, A.nomeEntidade
        From MeioApoio A NATURAL JOIN Acciona);

6.
SELECT C.nomeEntidade
FROM MeioCombate C
WHERE NOT EXISTS(
        (SELECT S.nomeEntidade
        FROM Acciona S)
        EXCEPT
```

Projeto de Bases de Dados, Parte 3

```
(SELECT N.nomeEntidade
FROM Acciona NATURAL JOIN MeioCombate N));
```

Desenvolvimento da aplicação

Existem 10 ficheiros ".php" que formam a aplicação web desenvolvida. Na página *"index.php"* podemos aceder às páginas das alíneas A a F.

A estrutura das páginas "*A.php"* e "*B.php"* têm uma estrutura semelhante.

Pode se observar as tabelas pedidas e todo o seu conteúdo. Existe ainda uma coluna adicional com um link "*remove*". Este link redireciona o utilizador para uma página que confirma a remoção da entrada selecionada.

Para se inserir novas entradas o utilizador pode preencher a primeira linha com os atributos desejados e selecionar "inserir".

Em *"C.php"* podemos observar todo o conteúdo das tabelas ProcessoSocorro e Meio.

Em *"D.php"* é apresentado ao utilizador as tabelas Processo Socorro e Meios para consulta e as tabelas Acciona e Evento Emergencia para associação de processos de socorro a meios e processos de socorro a eventos de emergência, respetivamente.

Nas alíneas E e F, o utilizador insere o input pretendido na caixa de texto que se encontra em *"index.php"* e clica "Ir". É redirecionado para uma página onde se encontra uma tabela com o resultado. No caso da alínea E o utilizador insere um número de processo de socorro e é apresentado todos os meios acionados por este. Na alínea F o utilizador insere um local de incêndio e é apresentado os meios de socorro acionados em processos de socorro nesse local.

Criámos dois ficheiros auxiliares: o *"inserir.php"* recebe como argumento o nome de uma tabela e valores para uma entrada nessa tabela. Utiliza um switch case para selecionar a tabela certa e insere os valores recebidos. O "*remove.php"* recebe como argumentos o nome de uma tabela e um conjunto de chaves primarias dessa tabela. Utiliza tambem um switch case para selecionar a tabela desejada e apaga a entrada cujas chaves primarias sao iguais aos valores recebidos.