

Projeto de Bases de Dados

Parte 3

| Nome | Número de Aluno | Percentagem de Contribuição | Esforço em horas |
|----------------------------------|------------------------|------------------------------------|-------------------------|
| <i>Joana Maria Leal Coutinho</i> | 87666 | 33.3% | 20h |
| <i>João Rafael Pinto Soares</i> | 87675 | 33.3% | 20h |
| <i>Pedro M. S. P. Rodrigues</i> | 87696 | 33.3% | 20h |

Grupo Nº 3

Turno BD817957L08 (6ª feira, 8:30 – 10:00)

Professor André Vasconcelos

Consultas SQL:

1.

```
SELECT numProcessoSocorro
FROM acciona
GROUP BY numProcessoSocorro
HAVING count(distinct (numMeio, nomeEntidade)) >= all(
    SELECT count(distinct (numMeio, nomeEntidade))
    FROM acciona
    GROUP BY numProcessoSocorro);
```

2.

```
SELECT nomeentidade
FROM EventoEmergencia NATURAL JOIN Acciona
WHERE instanteChamada BETWEEN '2018-06-21 00:00' AND '2018-09-23 23:59'
GROUP BY nomeentidade
HAVING count(*) >= all(
    SELECT count(*)
    FROM EventoEmergencia NATURAL JOIN Acciona
    WHERE instanteChamada BETWEEN '2018-06-21 00:00' AND '2018-09-23 23:59'
    GROUP BY nomeentidade);
```

3.

```
SELECT distinct e.numProcessoSocorro
FROM EventoEmergencia as e, acciona as a
WHERE instanteChamada BETWEEN '2018-01-01 00:00' AND '2018-12-31 23:59'
    AND moradaLocal LIKE 'Oliveira do Hospital'
    AND (a.numProcessoSocorro, a.numMeio, a.nomeEntidade) NOT IN (
        SELECT numProcessoSocorro, numMeio, nomeEntidade
        FROM Audita)
    AND a.numProcessoSocorro = e.numProcessoSocorro;
```

4.

```
SELECT count(*) as numSegmentos
FROM SegmentoVideo NATURAL JOIN video NATURAL JOIN vigia
WHERE duracao > '00:01:00'
    AND moradaLocal LIKE 'Monchique'
    AND dataHoralInicio BETWEEN '2018-08-01 00:00:00' AND '2018-08-31 23:59:00'
```

AND dataHoraFim BETWEEN '2018-08-01 00:00:00' AND '2018-08-31 23:59:00';

5.

```
SELECT *  
FROM MeioCombate  
WHERE (numMeio, nomeEntidade) NOT IN (  
    SELECT numMeio, nomeEntidade  
    FROM alocado);
```

6.

```
SELECT nomeentidade  
FROM acciona NATURAL JOIN MeioCombate  
GROUP BY nomeentidade  
HAVING count(distinct numProcessoSocorro) = (  
    SELECT count(distinct numProcessoSocorro) FROM Acciona);
```

Criação da Base de Dados:

create table Camara

```
(numCamara numeric not null check(numCamara >= 0),  
 constraint pk_Camara primary key(numCamara));
```

create table Video

```
(dataHoraInicio timestamp not null,  
 dataHoraFim timestamp not null check(dataHoraFim > dataHoraInicio),  
 numCamara numeric not null,  
 constraint pk_Video primary key(dataHoraInicio, numCamara),  
 constraint fk_Video_Camara foreign key(numCamara) references Camara(numCamara));
```

create table SegmentoVideo

```
(numSegmento numeric not null check(numSegmento >= 0),  
 duracao time not null,  
 dataHoraInicio timestamp not null,  
 numCamara numeric not null,  
 constraint pk_SegmentoVideo primary key(numSegmento, dataHoraInicio, numCamara),  
 constraint fk_SegmentoVideo_Video foreign key(dataHoraInicio, numCamara) references  
 Video(dataHoraInicio, numCamara));
```

create table Local

```
(moradaLocal varchar(80) not null, constraint pk_Local primary key(moradaLocal));
```

create table Vigia

```
(moradaLocal varchar(80) not null,  
numCamara numeric not null,  
constraint pk_Vigia primary key(moradaLocal, numCamara),  
constraint fk_Vigia_Local foreign key(moradaLocal) references Local(moradaLocal) on delete  
cascade,  
constraint fk_Vigia_Camara foreign key(numCamara) references Camara(numCamara));
```

create table ProcessoSocorro

```
(numProcessoSocorro numeric not null check(numProcessoSocorro >= 0),  
constraint pk_ProcessoSocorro primary key(numProcessoSocorro));
```

create table EventoEmergencia

```
(numTelefone numeric(9) not null check(numTelefone >= 0),  
instanteChamada timestamp not null,  
nomePessoa varchar(80) not null,  
moradaLocal varchar(80) not null,  
numProcessoSocorro numeric,  
unique(numTelefone, nomePessoa),  
constraint pk_EventoEmergencia primary key(numTelefone, instanteChamada),  
constraint fk_EventoEmergencia_Local foreign key(moradaLocal) references  
Local(moradaLocal) on delete cascade,  
constraint fk_EventoEmergencia_ProcessoSocorro foreign key(numProcessoSocorro)  
references ProcessoSocorro(numProcessoSocorro) on delete cascade);
```

create table EntidadeMeio

```
(nomeEntidade varchar(80) not null,  
constraint pk_EntidadeMeio primary key(nomeEntidade));
```

create table Meio

```
(numMeio numeric not null check(numMeio >= 0),  
nomeMeio varchar(80) not null,  
nomeEntidade varchar(80) not null,  
constraint pk_Meio primary key(numMeio, nomeEntidade),  
constraint fk_Meio_EntidadeMeio foreign key(nomeEntidade) references  
EntidadeMeio(nomeEntidade) on delete cascade);
```

create table MeioCombate

```
(numMeio numeric not null,  
nomeEntidade varchar(80) not null,
```

constraint pk_MeioCombate **primary key**(numMeio, nomeEntidade),

constraint fk_MeioCombate_Meio **foreign key**(numMeio, nomeEntidade) **references** Meio(numMeio, nomeEntidade) **on delete cascade**);

create table MeioApoio

(numMeio *numeric not null*,

nomeEntidade *varchar(80) not null*,

constraint pk_MeioApoio **primary key**(numMeio, nomeEntidade),

constraint fk_MeioApoio_Meio **foreign key**(numMeio, nomeEntidade) **references** Meio(numMeio, nomeEntidade) **on delete cascade**);

create table MeioSocorro

(numMeio *numeric not null*,

nomeEntidade *varchar(80) not null*,

constraint pk_MeioSocorro **primary key**(numMeio, nomeEntidade),

constraint fk_MeioSocorro_Meio **foreign key**(numMeio, nomeEntidade) **references** Meio(numMeio, nomeEntidade) **on delete cascade**);

create table Transporta

(numMeio *numeric not null*,

nomeEntidade *varchar(80) not null*,

numVitimas *numeric not null check*(numVitimas >= 0),

numProcessoSocorro *numeric not null*,

constraint pk_Transporta **primary key**(nomeEntidade, numMeio, numProcessoSocorro),

constraint fk_Transporta_MeioSocorro **foreign key** (numMeio, nomeEntidade) **references** MeioSocorro(numMeio, nomeEntidade) **on delete cascade**,

constraint fk_Transporta_ProcessoSocorro **foreign key** (numProcessoSocorro) **references** ProcessoSocorro(numProcessoSocorro) **on delete cascade**);

create table Alocado

(numMeio *numeric not null*,

nomeEntidade *varchar(80) not null*,

numHoras *numeric not null check*(numHoras >= 0),

numProcessoSocorro *numeric not null*,

constraint pk_Alocado **primary key**(nomeEntidade, numMeio, numProcessoSocorro),

constraint fk_Alocado_MeioApoio **foreign key** (numMeio, nomeEntidade) **references** MeioApoio(numMeio, nomeEntidade) **on delete cascade**,

constraint fk_Alocado_ProcessoSocorro **foreign key** (numProcessoSocorro) **references** ProcessoSocorro(numProcessoSocorro) **on delete cascade**);

create table Acciona

(numMeio *numeric not null*,

nomeEntidade *varchar(80)* **not null**,
numProcessoSocorro *numeric* **not null**,
constraint pk_Acciona **primary key**(nomeEntidade, numMeio, numProcessoSocorro),
constraint fk_Acciona_Meio **foreign key** (numMeio, nomeEntidade) **references** Meio(numMeio, nomeEntidade) **on delete cascade**,
constraint fk_Acciona_ProcessoSocorro **foreign key** (numProcessoSocorro) **references** ProcessoSocorro(numProcessoSocorro) **on delete cascade**);

create table Coordenador

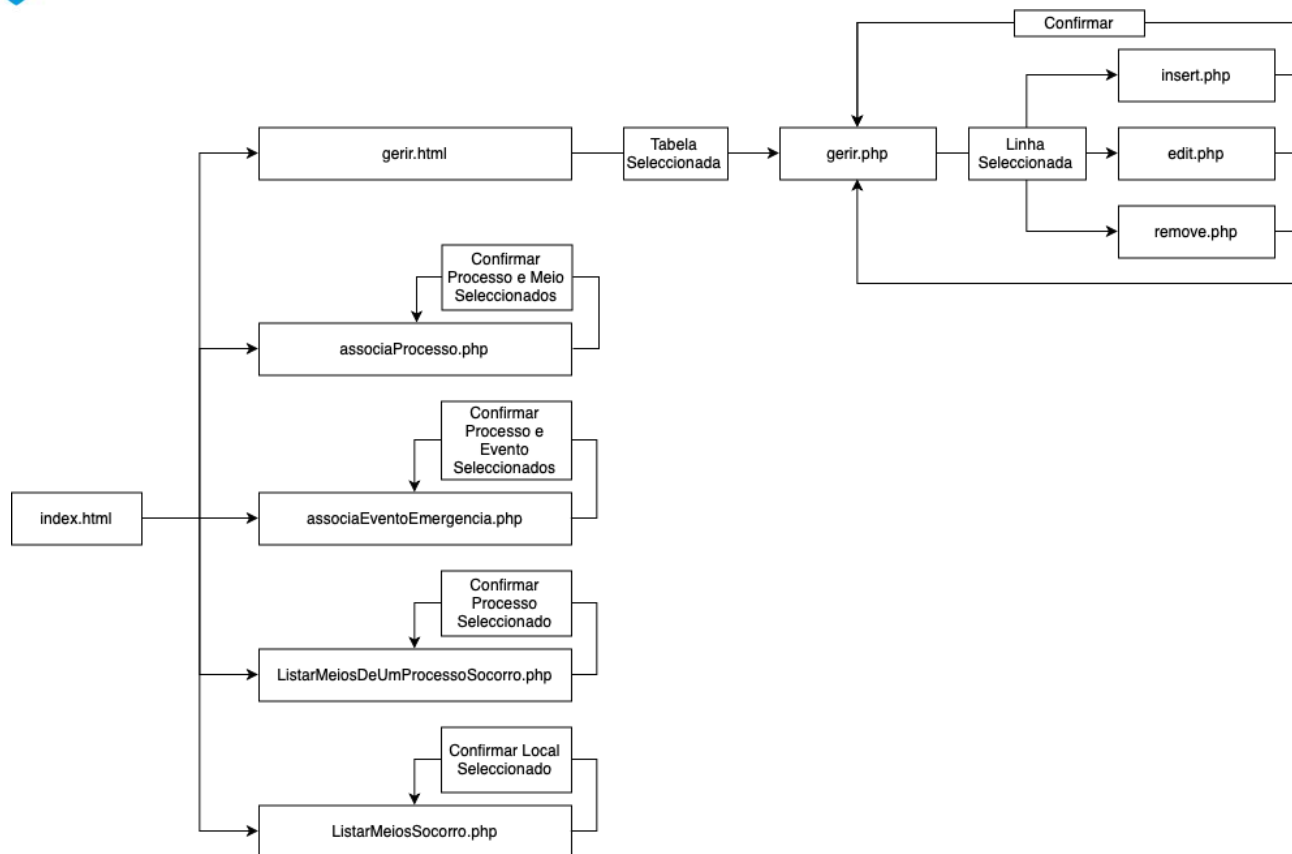
(idCoordenador *numeric* **not null check**(idCoordenador >= 0),
constraint pk_Coordenador **primary key**(idCoordenador));

create table Audita

(idCoordenador *numeric* **not null**,
numMeio *numeric* **not null**,
nomeEntidade *varchar(80)* **not null**,
numProcessoSocorro *numeric* **not null**,
datahoraInicio *timestamp* **not null**,
datahoraFim *timestamp* **not null check**(datahoraFim > datahoraInicio),
dataAuditoria *timestamp* **not null check**(dataAuditoria >= CURRENT_TIMESTAMP),
texto *varchar(255)* **not null**,
constraint pk_Audita **primary key**(nomeEntidade, numMeio, numProcessoSocorro, idCoordenador),
constraint fk_Audita_Coordenador **foreign key** (idCoordenador) **references** Coordenador(idCoordenador),
constraint fk_Audita_Acciona **foreign key** (nomeEntidade, numMeio, numProcessoSocorro) **references** Acciona(nomeEntidade, numMeio, numProcessoSocorro));

create table Solicita

(idCoordenador *numeric* **not null**,
dataHoraInicioVideo *timestamp* **not null**,
numCamara *numeric* **not null**,
dataHoraInicio *timestamp* **not null**,
dataHoraFim *timestamp* **not null check**(datahoraFim > datahoraInicio),
constraint pk_Solicita **primary key**(idCoordenador, dataHoraInicioVideo, numCamara),
constraint fk_Solicita_Coordenador **foreign key** (idCoordenador) **references** Coordenador(idCoordenador),
constraint fk_Solicita_Video **foreign key** (dataHoraInicioVideo, numCamara) **references** Video(dataHoraInicio, numCamara));



Para além dos ficheiros representados no diagrama, existe ainda o ficheiro `functions.php`, que contém:

- Quatro arrays, um com os nomes das colunas de cada tabela, um com a versão a apresentar desses nomes (Por exemplo o primeiro array contém “numProcessoSocorro” e o segundo “Numero do Processo de Socorro”), um com quais tabelas permitem editar linhas e outro com o nome das colunas cuja representação em queries SQL é um inteiro (e, consequentemente, uma string);
- Funções genéricas que permitem mostrar em HTML os conteúdos passados como parâmetros e ainda converter uma comparação entre elementos para a representação correta para que possa ser utilizada em queries (por exemplo a moradalocal Oliveira do Hospital passa a ‘Oliveira do Hospital’);
- Os parâmetros de acesso à base de dados.

O ficheiro `gerir.php` providencia, utilizando os ficheiros indicados no esquema, simultaneamente as tarefas de listar e de inserir, remover e editar, dependendo da tabela seleccionada em `gerir.html`.

As funcionalidades de listar os Meios acionados num Processo de Socorro listar os Meios de Socorro acionados em Processos de Socorro originados num dado Local de incêndio são dadas respetivamente pelos ficheiros `ListarMeiosDeUmProcessoSocorro.php` e `ListarMeiosSocorro.php`, após serem seleccionados um Processo de Socorro ou um Local.

As funcionalidades de Associar Processos de Socorro a Meios e Processos de Socorro a Eventos de Emergência são providenciadas, respetivamente, pelos ficheiros `associaProcesso.php` e `associaEventoEmergencia.php`.