**Projeto de Bases de Dados**

**Parte 3**

|  |  |  |  |
| --- | --- | --- | --- |
| **Nome** | **Número de Aluno** | **Percentagem de Contribuição** | **Esforço em horas** |
| Joana Maria Leal Coutinho | 87666 | 33.3% | 20h |
| João Rafael Pinto Soares | 87675 | 33.3% | 20h |
| Pedro M. S. P. Rodrigues | 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 dataHoraInicio 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**,

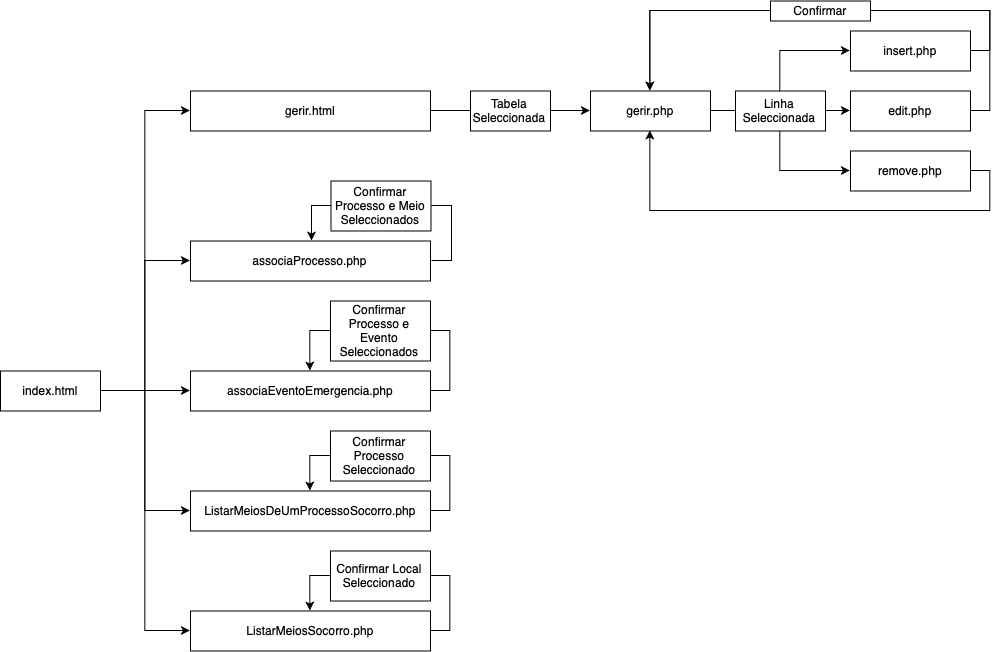
dataHoraInicio timestamp **not null**,

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 selecionada 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 selecionados 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.