

Projeto de Bases de Dados, Parte 3

Trabalho realizado por grupo nº49:

Joana Teodoro Nº86440

João Andrade Nº86441

Taíssa Ribeiro Nº86514

| | <i>Horas de trabalho</i> | <i>Percentagem relativa de esforço</i> |
|-----------------------|--------------------------|--|
| <i>Joana Teodoro</i> | 18 | 33,3% |
| <i>João Andrade</i> | 18 | 33,3% |
| <i>Taíssa Ribeiro</i> | 18 | 33,3% |

Expressão SQL – queries.sql:

- 1. SELECT** numProcessoSocorro
FROM (**SELECT** numProcessoSocorro, **COUNT**(*)
 FROM acciona
 GROUP BY numProcessoSocorro
 HAVING COUNT(*) >= all(**SELECT COUNT**(*)
 FROM acciona
 GROUP BY numProcessoSocorro)) **AS** a;
- 2. SELECT** nomeEntidade
FROM (**SELECT** nomeEntidade, **COUNT**(*)
 FROM acciona **NATURAL JOIN** eventoEmergencia
 WHERE instanteChamada **BETWEEN** **TIMESTAMP** '2018-06-21 00:00:00' **AND**
 TIMESTAMP '2018-09-21 23:59:59'
 GROUP BY nomeEntidade
 HAVING COUNT(*) >= all(**SELECT COUNT**(*)
 FROM acciona **NATURAL JOIN** eventoEmergencia
 WHERE instanteChamada **BETWEEN** **TIMESTAMP**
 '2018-06-21 00:00:00' **AND** **TIMESTAMP** '2018-09-21
 23:59:59'
 GROUP BY nomeEntidade)) **AS** a;
- 3. SELECT DISTINCT** numProcessoSocorro
FROM((**SELECT** numProcessoSocorro, nomeEntidade, numMeio
 FROM acciona **NATURAL JOIN** (**SELECT** numProcessoSocorro
 FROM eventoEmergencia
 WHERE instanteChamada **BETWEEN**
 TIMESTAMP '2018-01-01 00:00:00' **AND**
 TIMESTAMP '2018-12-31 23:59:59' **AND**
 moradaLocal = 'Oliveira do Hospital') **AS** a)
EXCEPT
(**SELECT** numProcessoSocorro, nomeEntidade, numMeio
 FROM audita)) **AS** b;
- 4. SELECT COUNT**(*)
FROM (**SELECT** *
 FROM segmentoVideo **NATURAL JOIN** vigia
 WHERE dataHoraInicioVideo **BETWEEN** **TIMESTAMP** '2018-08-01 00:00:00' **AND**
 TIMESTAMP '2018-08-31 23:59:59' **AND** duracao > 60 **AND** moradaLocal =
 'Monchique') **AS** a;
- 5. (SELECT** numMeio, nomeEntidade
 FROM acciona **NATURAL JOIN** meioCombate)
EXCEPT
(**SELECT** numMeio, nomeEntidade
 FROM acciona **NATURAL JOIN** meioApoio);
- 6. SELECT DISTINCT** nomeEntidade
FROM acciona **NATURAL JOIN** meioCombate **AS** a
WHERE NOT EXISTS (**SELECT** numProcessoSocorro

```

FROM acciona
EXCEPT
SELECT numProcessoSocorro
FROM acciona NATURAL JOIN meioCombate AS b
WHERE a.nomeEntidade = b.nomeEntidade);

```

Base de Dados – schema.sql:

```

drop table if exists camara cascade;
drop table if exists video cascade;
drop table if exists segmentoVideo cascade;
drop table if exists local cascade;
drop table if exists vigia cascade;
drop table if exists eventoEmergencia cascade;
drop table if exists processoSocorro cascade;
drop table if exists entidadeMeio cascade;
drop table if exists meio cascade;
drop table if exists meioCombate cascade;
drop table if exists meioApoio cascade;
drop table if exists meioSocorro cascade;
drop table if exists transporta cascade;
drop table if exists alocado cascade;
drop table if exists acciona cascade;
drop table if exists coordenador cascade;
drop table if exists audita cascade;
drop table if exists solicita cascade;

```

```

-----
-- Table Creation
-----

```

```

create table camara
(numCamara integer not null unique,
 constraint pk_camara primary key(numCamara));

create table video
(numCamara integer not null,
 dataHoraInicioVideo timestamp not null,
 dataHoraFimVideo timestamp not null,
 constraint fk_video_camara foreign key(numCamara) references camara(numCamara) on delete
 cascade on update cascade,
 constraint pk_video primary key(numCamara, dataHoraInicioVideo));

create table segmentoVideo
(numCamara integer not null,
 dataHoraInicioVideo timestamp not null,
 numSegmento integer not null,
 duracao numeric(16,4) not null, -- see if its an integer or numeric or varchar
 constraint fk_segmentoVideo_video foreign key(numCamara, dataHoraInicioVideo) references
 video(numCamara, dataHoraInicioVideo) on delete cascade on update cascade,
 constraint pk_segmentoVideo primary key(numSegmento, dataHoraInicioVideo, numCamara));

create table local
(moradaLocal varchar(80) not null unique,
 constraint pk_local primary key(moradaLocal));

```

```

create table vigia
(numCamara integer not null unique,
 moradaLocal varchar(80) not null,
 constraint fk_vigia_local foreign key(moradaLocal) references local(moradaLocal),
 constraint fk_vigia_camara foreign key(numCamara) references camara(numCamara),
 constraint pk_vigia primary key (numCamara, moradaLocal));

create table processoSocorro
(numProcessoSocorro integer not null unique,
 constraint pk_processoSocorro primary key(numProcessoSocorro)); -- ver ri

create table eventoEmergencia
(numTelefone integer not null unique,
 instanteChamada timestamp not null unique,
 nomePessoa varchar(80) not null,
 moradaLocal varchar(80) not null,
 numProcessoSocorro integer not null,
 constraint pk_numTelefone primary key(numTelefone, instanteChamada),
 constraint fk_eventoEmergencia_local foreign key(moradaLocal) references local(moradaLocal),
 constraint fk_eventoEmergencia_processoSocorro foreign key(numProcessoSocorro) references
processoSocorro(numProcessoSocorro),
 unique(numTelefone, nomePessoa));

create table entidadeMeio
(nomeEntidade varchar(80) not null unique,
 constraint pk_entidadeMeio primary key(nomeEntidade));

create table meio
(numMeio integer not null,
 nomeMeio varchar(80) not null,
 nomeEntidade varchar(80) not null,
 constraint fk_meio_entidadeMeio foreign key(nomeEntidade) references
entidadeMeio(nomeEntidade) on delete cascade on update cascade,
 constraint pk_meio primary key(numMeio, nomeEntidade));

create table meioCombate
(numMeio integer not null,
 nomeEntidade varchar(80) not null,
 constraint fk_meioCombate_meio foreign key(nomeEntidade, numMeio) references
meio(nomeEntidade, numMeio) on delete cascade on update cascade,
 constraint pk_meioCombate primary key(numMeio, nomeEntidade));

create table meioApoio
(numMeio integer not null,
 nomeEntidade varchar(80) not null,
 constraint fk_meioApoio_meio foreign key(nomeEntidade, numMeio) references
meio(nomeEntidade, numMeio) on delete cascade on update cascade,
 constraint pk_meioApoio primary key(numMeio, nomeEntidade));

create table meioSocorro
(numMeio integer not null,
 nomeEntidade varchar(80) not null,
 constraint fk_meioSocorro_meio foreign key(nomeEntidade, numMeio) references
meio(nomeEntidade, numMeio) on delete cascade on update cascade,
 constraint pk_meioSocorro primary key(numMeio, nomeEntidade));

create table transporta
(numMeio integer not null,

```

```

nomeEntidade varchar(80) not null,
numVitimas integer not null,
numProcessoSocorro integer not null,
constraint fk_transporta_meioSocorro foreign key(numMeio, nomeEntidade) references
meioSocorro(numMeio, nomeEntidade),
constraint fk_transporta_processoSocorro foreign key(numProcessoSocorro) references
processoSocorro(numProcessoSocorro),
constraint pk_transporta primary key(numMeio, nomeEntidade, numProcessoSocorro));

```

```

create table alocado
(numMeio integer not null,
nomeEntidade varchar(80) not null,
numHoras integer not null,
numProcessoSocorro integer not null,
constraint fk_alocado_meioApoio foreign key(numMeio, nomeEntidade) references
meioApoio(numMeio, nomeEntidade),
constraint fk_alocado_processoSocorro foreign key(numProcessoSocorro) references
processoSocorro(numProcessoSocorro),
constraint pk_alocado primary key(numMeio, nomeEntidade, numProcessoSocorro));

```

```

create table acciona
(numMeio integer not null,
nomeEntidade varchar(80) not null,
numProcessoSocorro integer not null,
constraint fk_acciona_meio foreign key(numMeio, nomeEntidade) references meio(numMeio,
nomeEntidade),
constraint fk_acciona_processoSocorro foreign key(numProcessoSocorro) references
processoSocorro(numProcessoSocorro),
constraint pk_acciona primary key(numMeio, nomeEntidade, numProcessoSocorro));

```

```

create table coordenador
(idCoordenador integer not null unique,
constraint pk_coordenador 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,
dataAuditoria date not null,
texto text not null,
constraint fk_audita_coordenador foreign key(idCoordenador) references
coordenador(idCoordenador),
constraint fk_audita_acciona foreign key(numMeio, nomeEntidade, numProcessoSocorro)
references acciona(numMeio, nomeEntidade, numProcessoSocorro),
constraint pk_audita primary key(idCoordenador, numMeio, nomeEntidade, numProcessoSocorro),
check(dataHoraInicio < dataHoraFim),
check(dataAuditoria <= current_date));

```

```

create table solicita
(idCoordenador integer not null,
dataHoraInicioVideo timestamp not null,
dataHoraInicio timestamp not null,
dataHoraFim timestamp not null,
numCamara integer not null,

```

```
constraint fk_solicita_coordenador foreign key(idCoordenador) references
coordenador(idCoordenador),
constraint fk_solicita_video foreign key(numCamara, dataHoraInicioVideo) references
video(numCamara, dataHoraInicioVideo),
constraint pk_solicita primary key(idCoordenador, dataHoraInicioVideo, numCamara));
```

Arquitetura:

A arquitetura desenvolvida para a concretização da 3ª entrega baseia-se num menu principal em html, index.html, que contém todas as funcionalidades pedidas pelo enunciado. Para cada comando, posteriormente, é feito um ficheiro PHP que apresenta a opção de Inserir ao lado do nome da tabela e que lista os elementos da tabela apresentando ao lado destes as opções para Remover e Editar. De forma a concretizar esta listagem, o ficheiro PHP de cada comando realiza uma instrução SQL própria para o efeito. É de notar que ficheiros com várias instruções SQL a realizar têm uma linha de código específica para começar e concretizar uma transação por forma a garantir a integridade dos resultados.

Relativamente à associação de processos a meios, existe um menu dedicado para a sua implementação que leva o utilizador para um ficheiro PHP com um form para preencher com a chave de processo e chaves de meio para poder associar estes. A associação de processos a eventos de emergência é feita implicitamente na inserção de qualquer um dos mesmos.

Para cada inserção existe um ficheiro PHP com um form que solicita ao utilizador os atributos necessários para criar uma nova posição na tabela onde se pretende inserir um novo elemento.

Para remover apenas é necessário clicar no link da linha que contém o elemento que o utilizador quer remover.

Para editar um meio específico é apresentado um form para o utilizador preencher com as novas características, nome do meio, que pretende que este meio tenha.

De forma a realizar todas estas funcionalidades, existem vários ficheiros que produzem na base de dados, a partir de uma ou mais instruções SQL, os resultados requeridos.