



Base de Dados 2018/2019 Taguspark

Projeto de Bases de Dados, Parte 3

Grupo 36

Turno BD817957L05

Professora Raquel Casteleiro

Número	Nome	% esforço por aluno	Esforço em horas
71060	Bruno Rodrigues	31%	10
85254	Alexandre Caldeira	31%	10
87665	Iulian Puscasu	38%	12

Criação da base de dados

```
create table camara(  
  num_camara integer not null,  
  constraint pk_camara primary key(num_camara));
```

```
create table video(  
  data_hora_inicio timestamp not null,  
  data_hora_fim timestamp not null,  
  num_camara integer not null ,  
  constraint pk_video primary key(data_hora_inicio, num_camara) ,  
  constraint fk_video_camara foreign key(num_camara) references  
camara(num_camara) ON DELETE CASCADE ON UPDATE CASCADE);
```

```
create table segmentodevideo(  
  num_segmento integer not null,  
  duracao real not null,  
  data_hora_inicio timestamp not null,  
  num_camara integer not null,  
  constraint pk_segmentodevideo primary key(num_segmento, data_hora_inicio,  
num_camara),  
  constraint fk_segmentodevideo_video foreign key(data_hora_inicio, num_camara)  
references video(data_hora_inicio, num_camara)ON DELETE CASCADE ON  
UPDATE CASCADE);
```

```
create table local(  
  morada varchar(255) not null,  
  constraint pk_local primary key(morada));
```

```
create table vigia(  
  num_camara integer not null,  
  morada varchar(255) not null,  
  constraint pk_vigia primary key(morada, num_camara),  
  constraint fk_vigia_local foreign key(morada) references local(morada),  
  constraint fk_vigia_camara foreign key(num_camara) references  
camara(num_camara)ON DELETE CASCADE ON UPDATE CASCADE);
```

```
create table processosocorro(  
  num_processo_socorro integer not null,  
  constraint pk_processosocorro primary key(num_processo_socorro));
```

```
create table entidademeio(  
  nome_entidade varchar(255) not null,  
  constraint pk_entidademeio primary key(nome_entidade));
```

```

create table eventoemergencia(
    num_telefone char(9) not null,
    instante_chamada timestamp not null,
    nome_pessoa varchar(255) not null,
    morada varchar(255) not null,
    num_processo_socorro integer,
    constraint u_eventoemergencia unique(num_telefone, nome_pessoa),
    constraint pk_eventoemergencia primary key(num_telefone, instante_chamada),
    constraint fk_eventoemergencia_local foreign key(morada) references
local(morada)ON DELETE CASCADE ON UPDATE CASCADE,
    constraint fk_eventoemergencia_processosocorro foreign
key(num_processo_socorro) references processosocorro(num_processo_socorro)ON
DELETE CASCADE ON UPDATE CASCADE);

```

```

create table meio(
    num_meio integer,
    nome_meio varchar(255),
    nome_entidade varchar(255),
    constraint u_meio unique(num_meio, nome_entidade),
    constraint pk_meio primary key(num_meio),
    constraint fk_meio_entidademeio foreign key(nome_entidade) references
entidademeio(nome_entidade)ON DELETE CASCADE ON UPDATE CASCADE);

```

```

create table meiocombate(
    num_meio integer,
    nome_entidade varchar(255),
    constraint pk_meiocombate primary key(num_meio, nome_entidade),
    constraint fk_meiocombate_meio foreign key(num_meio, nome_entidade)
references meio(num_meio, nome_entidade)ON DELETE CASCADE ON UPDATE
CASCADE);

```

```

create table meioapoio(
    num_meio integer,
    nome_entidade varchar(255),
    constraint pk_meioapoio primary key(num_meio, nome_entidade),
    constraint fk_meioapoio_meio foreign key(num_meio, nome_entidade) references
meio(num_meio, nome_entidade)ON DELETE CASCADE ON UPDATE
CASCADE);

```

```

create table meiosocorro(
    num_meio integer,
    nome_entidade varchar(255),
    constraint pk_meiosocorro primary key(num_meio, nome_entidade),
    constraint fk_meiosocorro_meio foreign key(num_meio, nome_entidade)
references meio(num_meio, nome_entidade)ON DELETE CASCADE ON UPDATE
CASCADE);

```

```

create table transporta(
    num_meio integer not null,
    nome_entidade varchar(255) not null,
    num_vitimas integer,
    num_processo_socorro integer not null,
    constraint pk_transporta primary key(num_meio, nome_entidade,
num_processo_socorro),
    constraint fk_transporta_meiosocorro foreign key(num_meio, nome_entidade)
references meiosocorro(num_meio, nome_entidade)ON DELETE CASCADE ON
UPDATE CASCADE,
    constraint fk_transporta_processosocorro foreign key(num_processo_socorro)
references processosocorro(num_processo_socorro)ON DELETE CASCADE ON
UPDATE CASCADE);

```

```

create table alocado(
    num_meio integer not null,
    nome_entidade varchar(255) not null,
    num_horas integer not null,
    num_processo_socorro integer,
    constraint pk_alocado primary key(num_meio, nome_entidade,
num_processo_socorro),
    constraint fk_alocado_meioapoio foreign key(num_meio, nome_entidade)
references meioapoio(num_meio, nome_entidade)ON DELETE CASCADE ON
UPDATE CASCADE,
    constraint fk_alocado_processosocorro foreign key(num_processo_socorro)
references processosocorro(num_processo_socorro)ON DELETE CASCADE ON
UPDATE CASCADE);

```

```

create table acciona(
    num_meio integer not null,
    nome_entidade varchar(255) not null,
    num_processo_socorro integer,
    constraint pk_acciona primary key(num_meio, nome_entidade,
num_processo_socorro),
    constraint fk_acciona_meio foreign key(num_meio, nome_entidade) references
meio(num_meio, nome_entidade)ON DELETE CASCADE ON UPDATE
CASCADE,
    constraint fk_acciona_processosocorro foreign key(num_processo_socorro)
references processosocorro(num_processo_socorro)ON DELETE CASCADE ON
UPDATE CASCADE);

```

```

create table coordenador(
    id_coordenador integer not null,
    constraint pk_coordenador primary key(id_coordenador));

```

```

create table audita(
    id_coordenador integer not null,
    num_meio integer not null,
    nome_entidade varchar(255) not null,
    num_processo_socorro integer not null,
    data_hora_inicio timestamp not null,
    data_hora_fim timestamp not null,
    data_auditoria date not null,
    texto text not null,
    constraint ck_audita_hora check(data_hora_inicio < data_hora_fim),
    constraint ck_data_audita check(data_auditoria <= now()),
    constraint pk_audita primary key(id_coordenador, num_meio, nome_entidade,
num_processo_socorro),
    constraint fk_audita_acciona foreign key(num_meio, nome_entidade,
num_processo_socorro) references acciona(num_meio, nome_entidade,
num_processo_socorro)ON DELETE CASCADE ON UPDATE CASCADE,
    constraint fk_audita_coordenador foreign key(id_coordenador) references
coordenador(id_coordenador)ON DELETE CASCADE ON UPDATE CASCADE);

```

```

create table solicita(
    id_coordenador integer not null,
    data_hora_inicio_video timestamp not null,
    num_camara integer not null,
    data_hora_inicio timestamp not null,
    data_hora_fim timestamp not null,
    constraint ck_solicita_hora check( data_hora_inicio < data_hora_fim),
    constraint ck_solicita_hora_video check( data_hora_inicio_video <
data_hora_inicio),
    constraint pk_solicita primary key(id_coordenador, data_hora_inicio_video,
num_camara),
    constraint fk_solicita_coordenador foreign key(id_coordenador) references
coordenador(id_coordenador)ON DELETE CASCADE ON UPDATE CASCADE,
    constraint fk_solicita_video foreign key(data_hora_inicio_video, num_camara)
references video(data_hora_inicio, num_camara)ON DELETE CASCADE ON
UPDATE CASCADE);

```

Consultas SQL

1. Qual o processo de socorro que envolveu maior numero de meios distintos.

```

SELECT num_processo_socorro, SUM(n_meios_por_entidade) AS n_meios
FROM (
    SELECT num_processo_socorro, nome_entidade, count(DISTINCT num_meio)
    FROM acciona
    GROUP BY num_processo_socorro, nome_entidade
) AS table1(num_processo_socorro, nome_entidade, n_meios_por_entidade)
GROUP BY num_processo_socorro
ORDER BY n_meios DESC
LIMIT 1;

```

2. Qual a entidade fornecedora de meios que participou em mais processos de socorro no Verão de 2018.

```
SELECT nome_entidade, mycount
FROM(
  SELECT nome_entidade, COUNT(DISTINCT num_processo_socorro)
  FROM eventoemergencia NATURAL JOIN acciona
  WHERE instante_chamada >= '2018-06-21 00:00:00'
  AND instante_chamada <= '2018-09-23 00:00:00'
  GROUP BY nome_entidade
)AS table1(nome_entidade, mycount)
GROUP BY nome_entidade, mycount
ORDER BY mycount DESC
LIMIT 1;
```

3. Quais são os processos de socorro, referente a eventos de emergência de 2018 de Oliveira do Hospital, onde existe pelo menos um acionamento de meios que não foi alvo de auditoria.

```
SELECT num_processo_socorro
FROM eventoemergencia
WHERE morada = 'Oliveira do Hospital'
AND instante_chamada >= '2018-06-21 00:00:00'
AND instante_chamada <= '2018-09-23 00:00:00'
AND num_processo_socorro IN (
  SELECT acciona.num_processo_socorro
  FROM acciona LEFT OUTER JOIN audita
  ON acciona.num_Meio = audita.num_meio
  AND acciona.nome_entidade = audita.nome_entidade
  WHERE audita.num_processo_socorro IS NULL);
```

4. Quantos segmentos de vídeo com duração superior a 60 segundos, foram gravados em câmaras de vigilância de Monchique durante o mês de Agosto de 2018.

```
SELECT COUNT(num_segmento)
FROM segmentodevideo NATURAL JOIN vigia
WHERE morada = 'Monchique'
AND data_hora_inicio >= '2018-08-01 00:00:00'
AND data_hora_inicio <= '2018-08-31 00:00:00'
AND duracao > 60;
```

5. Liste os Meios de combate que não foram usados como Meios de Apoio em nenhum processo de socorro.

```
(SELECT num_meio, nome_entidade FROM meiocombate NATURAL JOIN
acciona)
EXCEPT
(SELECT num_meio, nome_entidade FROM meiosocorro NATURAL JOIN acciona)
```

6. Liste as entidades que forneceram meios de combate a todos os Processos de socorro que acionaram meios.queries.php

```
SELECT DISTINCT nome_entidade
FROM acciona
WHERE nome_entidade IN(
    SELECT nome_entidade
    FROM acciona NATURAL JOIN meiocombate
    GROUP BY nome_entidade
    HAVING COUNT(DISTINCT num_processo_socorro) = (
        SELECT COUNT(DISTINCT num_processo_socorro)
        FROM acciona));
```

PHP

Há três componentes principais na estrutura php:

1. index.html: o ficheiro inicial com todos os links para as procuras que o utilizador deseja.
2. -request.php: ficheiro com os formulários para obter os argumentos necessários para as queries.
3. -folder queries: contém ficheiros que executam as queries, havendo 1 ficheiro por cada query.

Quando o utilizador clica num link no index.html, vai abrir o ficheiro request.php com um certo argumento. O argumento é uma variável cujo valor depende de qual link foi carregado. Dentro do request.php há um switch com todos os argumentos possíveis. Dependendo do argumento, vamos mostrar o formulário associado à operação (ou mostrar logo o resultado, se não for necessário formulário). Depois do formulário estar preenchido e ser submetido, é executado um dos ficheiros php dentro do folder queries. Cada ficheiro corresponde a uma operação diferente e os argumentos com que são executados são aqueles fornecidos pelo formulário.