

PROJETO DE BASES DE DADOS

Parte 3

Grupo 42 - Turno L04 - Segunda-Feira 12:30	
Professor: Taras Lykhenko	
Aluno	Esforço
João Palet - 86447	19h (33,3%)
Miguel Grilo - 86489	19h (33,3%)
Simão Nunes - 86512	19h (33,3%)

CRIAÇÃO DA BASE DE DADOS

```
DROP TABLE IF EXISTS segmentoVideo;  
DROP TABLE IF EXISTS audita;  
DROP TABLE IF EXISTS acciona;  
DROP TABLE IF EXISTS alocado;  
DROP TABLE IF EXISTS transporta;  
DROP TABLE IF EXISTS vigia;  
DROP TABLE IF EXISTS eventoEmergencia;  
DROP TABLE IF EXISTS local;  
DROP TABLE IF EXISTS meioCombate;  
DROP TABLE IF EXISTS meioApoio;  
DROP TABLE IF EXISTS meioSocorro;  
DROP TABLE IF EXISTS meio;  
DROP TABLE IF EXISTS entidadeMeio;  
DROP TABLE IF EXISTS solicita;  
DROP TABLE IF EXISTS coordenador;  
DROP TABLE IF EXISTS video;  
DROP TABLE IF EXISTS camara;  
DROP TABLE IF EXISTS processoSocorro;
```

```
CREATE TABLE camara(  
    numCamara INT NOT NULL,  
    PRIMARY KEY(numCamara)  
);
```

```
CREATE TABLE video(  
    dataHoraInicio TIMESTAMP NOT NULL,  
    dataHoraFim TIMESTAMP NOT NULL,  
    numCamara INT NOT NULL,  
    PRIMARY KEY(dataHoraInicio, numCamara),  
    FOREIGN KEY(numCamara) REFERENCES camara(numCamara) ON DELETE  
CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE segmentoVideo (  
    numSegmento INT NOT NULL,  
    duracao INT NOT NULL,  
    dataHoraInicio TIMESTAMP NOT NULL,  
    numCamara INT NOT NULL,  
    PRIMARY KEY(numSegmento, dataHoraInicio, numCamara),  
    FOREIGN KEY(dataHoraInicio, numCamara) REFERENCES video(dataHoraInicio,  
numCamara) ON DELETE CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE local(  
    moradaLocal VARCHAR(255),  
    PRIMARY KEY(moradaLocal)  
);
```

```
CREATE TABLE vigia(  
    moradaLocal VARCHAR(255),  
    numCamara INT NOT NULL,  
    PRIMARY KEY(moradaLocal, numCamara),  
    FOREIGN KEY(moradaLocal) REFERENCES local(moradaLocal) ON DELETE  
CASCADE ON UPDATE CASCADE,  
    FOREIGN KEY(numCamara) REFERENCES camara(numCamara) ON DELETE  
CASCADE ON UPDATE CASCADE  
);
```

```
CREATE TABLE processoSocorro(  
    numProcessoSocorro INT NOT NULL,  
    PRIMARY KEY(numProcessoSocorro)  
);
```

```

CREATE TABLE eventoEmergencia(
    numTelefone VARCHAR(13) NOT NULL,
    instanteChamada TIMESTAMP NOT NULL,
    nomePessoa VARCHAR(255),
    moradaLocal VARCHAR(255),
    numProcessoSocorro INT NOT NULL,
    PRIMARY KEY(numTelefone, instanteChamada),
    UNIQUE(nomePessoa, moradaLocal),
    FOREIGN KEY(moradaLocal) REFERENCES local(moradaLocal) ON DELETE
CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(numProcessoSocorro) REFERENCES
processoSocorro(numProcessoSocorro) ON DELETE CASCADE ON UPDATE
CASCADE
);

```

```

CREATE TABLE entidadeMeio(
    nomeEntidade VARCHAR(255) NOT NULL,
    PRIMARY KEY(nomeEntidade)
);

```

```

CREATE TABLE meio(
    numMeio INT NOT NULL,
    nomeMeio VARCHAR(255) NOT NULL,
    nomeEntidade VARCHAR(255) NOT NULL,
    PRIMARY KEY(numMeio, nomeEntidade),
    FOREIGN KEY(nomeEntidade) REFERENCES entidadeMeio(nomeEntidade) ON
DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE meioCombate(
    numMeio INT NOT NULL,
    nomeEntidade VARCHAR(255) NOT NULL,
    PRIMARY KEY(numMeio, nomeEntidade),

```

```

FOREIGN KEY(numMeio, nomeEntidade) REFERENCES meio(numMeio,
nomeEntidade) ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE meioApoio(
    numMeio INT NOT NULL,
    nomeEntidade VARCHAR(255) NOT NULL,
    PRIMARY KEY(numMeio, nomeEntidade),
    FOREIGN KEY(numMeio, nomeEntidade) REFERENCES meio(numMeio,
nomeEntidade) ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE meioSocorro(
    numMeio INT NOT NULL,
    nomeEntidade VARCHAR(255) NOT NULL,
    PRIMARY KEY(numMeio, nomeEntidade),
    FOREIGN KEY(numMeio, nomeEntidade) REFERENCES meio(numMeio,
nomeEntidade) ON DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE transporta(
    numMeio INT NOT NULL,
    nomeEntidade VARCHAR(255) NOT NULL,
    numVitimas INT,
    numProcessoSocorro INT NOT NULL ,
    PRIMARY KEY(numMeio, nomeEntidade, numProcessoSocorro),
    FOREIGN KEY(numMeio, nomeEntidade) REFERENCES meioSocorro(numMeio,
nomeEntidade) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(numProcessoSocorro) REFERENCES
processoSocorro(numProcessoSocorro) ON DELETE CASCADE ON UPDATE
CASCADE
);

```

```

CREATE TABLE alugado(
    numMeio INT NOT NULL,
    nomeEntidade VARCHAR(255) NOT NULL,
    numHoras INT NOT NULL,
    numProcessoSocorro INT NOT NULL,
    FOREIGN KEY(numMeio, nomeEntidade) REFERENCES meioApoio(numMeio,
nomeEntidade) ON DELETE CASCADE ON UPDATE CASCADE,
    FOREIGN KEY(numProcessoSocorro) REFERENCES
processoSocorro(numProcessoSocorro) ON DELETE CASCADE ON UPDATE
CASCADE
);

```

```

CREATE TABLE acciona(
    numMeio INT NOT NULL,
    nomeEntidade VARCHAR(255) NOT NULL,
    numProcessoSocorro INT NOT NULL,
    PRIMARY KEY(numMeio, nomeEntidade, numProcessoSocorro),
    FOREIGN KEY(numMeio, nomeEntidade) REFERENCES meio(numMeio,
nomeEntidade) ON DELETE CASCADE,
    FOREIGN KEY(numProcessoSocorro) REFERENCES
processoSocorro(numProcessoSocorro) ON DELETE CASCADE ON UPDATE
CASCADE
);

```

```

CREATE TABLE coordenador(
    idCoordenador INT NOT NULL UNIQUE,
    PRIMARY KEY(idCoordenador)
);

```

```

CREATE TABLE audita(
    idCoordenador INT NOT NULL UNIQUE,
    numMeio INT NOT NULL,
    nomeEntidade VARCHAR(255) NOT NULL,
    numProcessoSocorro INT NOT NULL,
    dataHoralInicio TIMESTAMP NOT NULL,
    dataHoraFim TIMESTAMP NOT NULL,
    dataAuditoria TIMESTAMP NOT NULL,
    texto TEXT NOT NULL,
    PRIMARY KEY(idCoordenador, numMeio, nomeEntidade,
numProcessoSocorro, dataHoralInicio, dataHoraFim, dataAuditoria, texto),
    FOREIGN KEY(numMeio, nomeEntidade, numProcessoSocorro) REFERENCES
acciona(numMeio, nomeEntidade, numProcessoSocorro) ON DELETE CASCADE
ON UPDATE CASCADE,
    FOREIGN KEY(idCoordenador) REFERENCES coordenador(idCoordenador) ON
DELETE CASCADE ON UPDATE CASCADE
);

```

```

CREATE TABLE solicita(
    idCoordenador INT NOT NULL UNIQUE,
    dataHoralInicioVideo TIMESTAMP NOT NULL,
    numCamara INT NOT NULL,
    dataHoralInicio TIMESTAMP NOT NULL,
    dataHoraFim TIMESTAMP NOT NULL,
    PRIMARY KEY(idCoordenador, dataHoralInicioVideo, numCamara),
    FOREIGN KEY(idCoordenador) REFERENCES coordenador(idCoordenador) ON
DELETE CASCADE,
    FOREIGN KEY(dataHoralInicioVideo, numCamara) REFERENCES
video(dataHoralInicio, numCamara) ON DELETE CASCADE ON UPDATE CASCADE
);

```

SQL

1.

```
SELECT numprocessosocorro
FROM acciona
GROUP BY numprocessosocorro
HAVING count(*) >= ALL (
    SELECT count(*)
    FROM acciona
    GROUP BY numprocessosocorro);
```

2.

```
SELECT nomeentidade
FROM (
    SELECT distinct numprocessosocorro, nomeentidade
    FROM acciona NATURAL JOIN eventoemergencia
    WHERE instantechamada BETWEEN '2018-06-21 00:00:00' AND '2018-09-23
23:59:59') as t1
GROUP BY nomeentidade
HAVING count(*) >= ALL (
    SELECT count(*)
    FROM (
        SELECT distinct numprocessosocorro, nomeentidade
        FROM acciona NATURAL JOIN eventoemergencia
        WHERE instantechamada BETWEEN '2018-06-21 00:00:00' AND '2018-
09-23 23:59:59') as t2
    GROUP BY nomeentidade);
```

3.

```
SELECT distinct numprocessosocorro
FROM (
    SELECT numprocessosocorro, nummeio, nomeentidade
    FROM acciona NATURAL JOIN eventoemergencia
    WHERE moradalocal = 'Lisboa' AND instantechamada BETWEEN '2016-01-01
00:00:00' AND '2018-12-31 23:59:59') AS t1
WHERE NOT EXISTS (
    SELECT 1
    FROM audita
    WHERE numprocessosocorro = t1.numprocessosocorro AND nummeio =
t1.nummeio AND nomeentidade = t1.nomeentidade);
```

4.

```
SELECT count(*)  
FROM segmentovideo NATURAL JOIN video NATURAL JOIN vigia  
WHERE duracao > 1 AND moradalocal = 'Monchique' AND datahorainicio BETWEEN  
'2018-08-01 00:00:00' AND '2018-08-31 23:59:59';
```

5.

```
SELECT *  
FROM meiocombate  
EXCEPT  
SELECT nummeio, nomeentidade  
FROM meioapoio NATURAL JOIN acciona;
```

6.

```
SELECT nomeentidade  
FROM meiocombate NATURAL JOIN acciona  
GROUP BY nomeentidade  
HAVING count(distinct numprocessosocorro) = (  
    SELECT count(distinct numprocessosocorro)  
    FROM meiocombate NATURAL JOIN acciona);
```

HTML E PHP

Na pasta *web/* encontram-se 6 pastas, cada uma relativa a uma das opções previstas no enunciado, bem como um ficheiro *index.html*, a página inicial de cada menu correspondente.

Nestes menus encontram-se listadas todas as possíveis operações de manipulação da base de dados relacionadas com as opções referidas no enunciado. Cada uma destas operações contém um ficheiro *php* que será responsável pela execução da mesma, sendo que o utilizador será redirecionado para o ficheiro específico após selecionar a operação que pretende. Estes ficheiros consistem em porções de código *html* intercalado com código *php* onde são efetuadas as *queries* à base de dados em questão, bem como a estruturação da interface.

Maioritariamente, há 4 tipos de operações que são executadas em cada um destes ficheiros:

-Listar, inserir, editar e eliminar informações da base de dados.

Para que as informações possam ser listadas na interface da nossa aplicação, utilizamos uma *table* que gera novas linhas à medida que vai percorrendo (através de um *foreach*) o resultado de uma consulta à base de dados.

Quando pretendemos inserir novas informações, ou editar dados existentes, utilizamos um *form* para que o utilizador possa inserir nas caixas de texto dados fundamentais relativos a cada coluna da tabela presente na base de dados. Quando o *form* é submetido, o ficheiro é corrido de novo e deteta a submissão de novos dados através de um *if*, executando posteriormente uma nova *query* que insere os dados na tabela correspondente.

Para eliminar, pressionamos o ícone do caixote do lixo da entrada correspondente e, de uma forma análoga à inserção, é removida a entrada pretendida.

A aplicação final com a base de dados populada pode ser consultada acedendo ao seguinte link: <http://web.tecnico.ulisboa.pt/~ist186512/projects/database/>