

PROJETO DE BASES DE DADOS

PARTE 3

Pedro Manuel Moreira Vaz Antunes de Sousa

GRUPO 63 - L 07 – QUARTA 9:30		
ALUNO	HORAS	% ESFORÇO
Andreia Matos 89413	20	33.3
Inês Correia 89464	20	33.3
João Marques 89473	20	33.3

CRIAÇÃO DA BASE DE DADOS

```
drop table correcao cascade;
drop table proposta_de_correcao cascade;
drop table incidencia cascade;
drop table utilizador_regular cascade;
drop table utilizador_qualificado cascade;
drop table utilizador cascade;
drop table duplicado cascade;
drop table anomalia_traducao cascade;
drop table anomalia cascade;
drop table item cascade;
drop table local_publico cascade;

create table local_publico (
    latitude float not null,
    longitude float not null,
    unique(latitude, longitude),
    nome char(20) not null,
    primary key (latitude, longitude)
);

create table item (
    id serial not null unique,
    descricao text not null,
    localizacao char(14) not null,
    latitude float not null,
    longitude float not null,
    foreign key(latitude, longitude) references local_publico(latitude, longitude) ON DELETE
CASCADE ON UPDATE CASCADE,
    primary key(id)
);

create table anomalia (
    id serial not null unique,
    zona char(10) not null,
    imagem char(10) not null,
    lingua char(12) not null,
    ts timestamp not null,
    descricao text not null,
    tem_anomalia_redacao boolean not null,
    primary key (id)
);

create table anomalia_traducao (
    id serial not null,
    zona2 char(10) not null,
    lingua2 char(12) not null,
    foreign key(id) references anomalia(id) ON DELETE CASCADE ON UPDATE CASCADE
);

create table duplicado (
```

```

    item1 serial not null,
    item2 serial not null,
    primary key(item1, item2),
    foreign key(item1) references item(id) ON DELETE CASCADE ON UPDATE CASCADE,
    foreign key(item2) references item(id) ON DELETE CASCADE ON UPDATE CASCADE,
    unique(item1,item2),
    check (item1 < item2)
);

create table utilizador (
    email char(20) not null unique,
    password char(14) not null,
    primary key(email)
);

create table utilizador_qualificado (
    email char(20) not null unique,
    foreign key(email) references utilizador(email) ON DELETE CASCADE ON UPDATE CASCADE
);

create table utilizador_regular (
    email char(20) not null unique,
    foreign key(email) references utilizador(email) ON DELETE CASCADE ON UPDATE CASCADE
);

create table incidencia (
    anomalia_id serial not null unique,
    item_id serial not null,
    email char(20) not null,
    foreign key(anomalia_id) references anomalia(id) ON DELETE CASCADE ON UPDATE CASCADE,
    foreign key(item_id) references item(id) ON DELETE CASCADE ON UPDATE CASCADE,
    foreign key(email) references utilizador(email) ON DELETE CASCADE ON UPDATE CASCADE,
    primary key(anomalia_id)
);

create table proposta_de_correcao (
    email char(20) not null,
    nro serial not null unique,
    data_hora timestamp not null,
    texto text not null,
    foreign key(email) references utilizador_qualificado(email) ON DELETE CASCADE ON UPDATE
CASCADE,
    primary key(email, nro)
);

```

```

create table correcao (
    email char(20) not null,
    nro serial not null unique,
    anomalia_id serial not null,
    foreign key(email, nro) references proposta_de_correcao(email, nro) ON DELETE CASCADE ON
UPDATE CASCADE,
    foreign key(anomalia_id) references incidencia(anomalia_id) ON DELETE CASCADE ON UPDATE
CASCADE,
    primary key(email, nro, anomalia_id)
);

```

CONSULTAS EM SQL

1.

```

SELECT nome FROM local_publico
NATURAL JOIN item
INNER JOIN incidencia on item.id = incidencia.item_id
GROUP BY nome
HAVING count((latitude,longitude)) >=
ALL( SELECT COUNT((latitude,longitude)) FROM local_publico
NATURAL JOIN item INNER JOIN incidencia ON item.id = incidencia.item_id GROUP BY nome);

```

2.

```

select email from (anomalia
natural join anomalia_traducao
inner join incidencia on anomalia.id = incidencia.anomalia_id
natural join utilizador
natural join utilizador_regular)
where extract(month from ts) < '07' and extract(year from ts) = '2019'
group by email having count(email) >=
ALL(select count(email) from anomalia
natural join anomalia_traducao
inner join incidencia on anomalia.id = incidencia.anomalia_id
natural join utilizador
natural join utilizador_regular
where extract(month from ts) < '07' and extract(year from ts) = '2019' group by email);

```

3.

```

select email from (utilizador
natural join incidencia
inner join item on item.id = incidencia.item_id
inner join anomalia on anomalia.id = incidencia.anomalia_id)
where extract(year from ts) = 2019 and latitude > 39.336775;

```

4.

```
select email from
(select utilizador_qualificado.email from
(select email from (select count(email) as countp, counti, email from
(select * from (select * from (select count(email) as countI, email from incidencia
natural join utilizador_qualificado group by email) as TempTable natural join incidencia) as
TempTable2
natural join correcao where TempTable2.email = correcao.email and TempTable2.anomalia_id =
correcao.anomalia_id) as TempTable3
group by email, counti) as TempTable4 where counti = countp) as TempTable5
inner join utilizador_qualificado on TempTable5.email <> utilizador_qualificado.email) as
TempTable6
natural join incidencia inner join item on incidencia.item_id = item.id
inner join anomalia on anomalia.id = anomalia_id where extract(year from ts) = '2018' and
latitude < 39.336775 group by email;
```

ARQUITETURA DA APLICAÇÃO PHP RELAÇÃO ENTRE OS DIVERSOS FICHEIROS

Pode aceder à aplicação PHP através de: <http://web.tecnico.ulisboa.pt/ist189473/index.html>

A aplicação PHP (Translate Right) é composta por um menu inicial com várias operações: **Inserir, Registrar, Remover, Editar, Listar**. Estas permitem manipular o conteúdo da base de dados em causa, sendo que esta já se encontra inicialmente preenchida com alguns valores.

Na operação **Inserir** estão disponíveis as seguintes especificações: **Local Público, Item, Anomalia, Proposta de Correção**. Em cada uma destas, é pedido ao utilizador que introduza os dados necessários para a criação do elemento escolhido, sendo este adicionado à base de dados caso os dados inseridos sejam válidos. ´

É também possível **Remover** qualquer um destes elementos da base de dados, através da identificação do mesmo, quer pelo nome quer pelo número que o distingue dos restantes.

Pode ainda **Registrar, Incidências** ou **Duplicados**, inserindo os respetivos parâmetros.

A seguinte operação disponível é **Editar**, sendo o único elemento sobre o qual se pode realizar as **Propostas de Correção**, que possuem um texto passível a alteração por parte do utilizador, modificando-o assim na respetiva proposta de correção na base de dados.

Por fim, o utilizador pode **Listar** um dos seguintes: **Utilizadores, Anomalias entre 2 Locais** ou **Anomalias por Área nos Últimos 3 Meses**. No primeiro caso, são listados todos os utilizadores existentes na base de dados, identificados pelo respetivo email. Nos outros dois, o utilizador especifica os valores a serem considerados para a procura de anomalias na situação desejada, sendo que estas são então listadas pelo seu ID.

Todas as páginas se relacionam entre si através de uma hiperligação ou um botão, que ilustram claramente a função em causa, por exemplo: **Voltar** , **Validar**.