

Projeto de Bases de Dados - Parte 3

Trabalho realizado por:

Afonso Vasconcelos, 90698

Pedro Leitão, 90764

Rodrigo Rosa, 90777

Contribuição para o projeto:

90698 | 33.3(3)% | 8 horas

90764 | 33.3(3)% | 8 horas

90777 | 33.3(3)% | 8 horas

Grupo 16

Turno: Sexta-feira, 8:30h (BD8179577L06)

Docente: Paulo Carreira

Criação da Base de Dados:

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

```
create table local_publico(  
    latitude numeric(9,6) not null,  
    longitude numeric(9,6) not null,  
    nome varchar(30) not null,  
    constraint pk_local_publico primary key(latitude, longitude)  
);
```

```
create table item(  
    item_id integer not null,  
    descricao_item text not null,  
    localizacao varchar(30) not null,  
    latitude numeric(9,6) not null,  
    longitude numeric(9,6) not null,  
    constraint pk_item primary key(item_id),  
    constraint fk_item foreign key(latitude, longitude) references local_publico(latitude, longitude)  
);
```

```
create table anomalia(  
    anomalia_id integer not null unique,  
    zona varchar(30) not null,  
    imagem varchar(2083) not null,  
    lingua char(3) not null,  
    ts timestamp not null,  
    descricao_anomalia text not null,  
    tem_anomalia_redacao boolean not null,  
    constraint pk_anomalia primary key(anomalia_id)  
);
```

```
create table anomalia_traducao(  
    anomalia_id integer not null unique,  
    zona2 varchar(30) not null,  
    lingua2 char(3) not null,  
    constraint pk_anomalia_traducao primary key(anomalia_id),  
    constraint fk_anomalia_traducao foreign key(anomalia_id) references anomalia(anomalia_id)  
);
```

```
create table duplicado(  
    item1 integer not null unique,  
    item2 integer not null unique,  
    constraint pk_duplicado primary key(item1, item2),  
    constraint fk_duplicado_item1 foreign key(item1) references item(item_id),  
    constraint fk_duplicado_item2 foreign key(item2) references item(item_id)  
);
```

```
create table utilizador(  
    email varchar(254) not null unique,  
    password varchar(30) not null,  
    constraint pk_utilizador primary key(email)  
);
```

```
create table utilizador_qualificado(  
    email varchar(254) not null unique,  
    constraint pk_utilizador_qualificado primary key(email),  
    constraint fk_utilizador_qualificado foreign key(email) references utilizador(email)  
);
```

```
create table utilizador_regular(  
    email varchar(254) not null unique,  
    constraint pk_utilizador_regular primary key(email),  
    constraint fk_utilizador_regular foreign key(email) references utilizador(email)  
);
```

```
create table incidencia(  
    anomalia_id integer not null unique,  
    item_id integer not null,  
    email varchar(254) not null,  
    constraint pk_incidencia primary key(anomalia_id),  
    constraint fk_incidencia_anomalia foreign key(anomalia_id) references anomalia(anomalia_id),  
    constraint fk_incidencia_item foreign key(item_id) references item(item_id),  
    constraint fk_incidencia_email foreign key(email) references utilizador(email)
```

```
);
```

```
create table proposta_de_correcao(  
    email varchar(254) not null unique,  
    nro serial not null unique,  
    data_hora timestamp not null,  
    texto text not null,  
    constraint pk_proposta_de_correcao primary key(email, nro),  
    constraint fk_proposta_de_correcao foreign key(email) references utilizador_qualificado(email)  
);
```

```
create table correcao(  
    email varchar(254) not null unique,  
    nro int not null unique,  
    anomalia_id integer not null unique,  
    constraint pk_correcao primary key(email, nro, anomalia_id),  
    constraint fk_correcao_email foreign key(email) references proposta_de_correcao(email),  
    constraint fk_correcao_nro foreign key(nro) references proposta_de_correcao(nro),  
    constraint fk_correcao_anomalia foreign key(anomalia_id) references incidencia(anomalia_id)  
);
```

SQL:

1.

```
SELECT nome, latitude, longitude  
FROM local_publico  
NATURAL JOIN item  
NATURAL JOIN incidencia  
NATURAL JOIN anomalia  
GROUP BY nome, latitude, longitude  
HAVING COUNT (nome) = (  
    SELECT MAX(count)  
FROM (  
    SELECT nome, COUNT(nome) AS count  
    FROM local_publico  
    NATURAL JOIN item  
    NATURAL JOIN incidencia  
    NATURAL JOIN anomalia  
    GROUP BY nome  
    ) AS foo  
);
```

2.

```
SELECT email FROM utilizador_regular
NATURAL JOIN incidencia
NATURAL JOIN anomalia
NATURAL JOIN anomalia_traducao
WHERE (ts <= '2019-06-30 23:59:59' AND ts >= '2019-01-01 00:00:00')
GROUP BY email
HAVING count (email) = (SELECT MAX(count) FROM (
  SELECT email, count(email) AS count
  FROM utilizador_regular
  NATURAL JOIN incidencia
  NATURAL JOIN anomalia
  NATURAL JOIN anomalia_traducao
  WHERE (ts <= '2019-06-30 23:59:59' AND ts >= '2019-01-01 00:00:00')
  GROUP BY email)
AS foo
);
```

3.

```
SELECT email FROM utilizador
NATURAL JOIN incidencia
NATURAL JOIN item
NATURAL JOIN anomalia
WHERE (latitude > 39.336775 AND ts >= '2019-01-01 00:00:00' AND ts <= '2019-12-31 23:59:59');
```

4.

```
SELECT email FROM utilizador_qualificado
NATURAL JOIN incidencia
NATURAL JOIN item
NATURAL JOIN anomalia
WHERE latitude < '39.336775'
GROUP BY email
EXCEPT (
  SELECT email FROM utilizador_qualificado
  NATURAL JOIN proposta_de_correcao
  GROUP BY email
);
```

Aplicação:

A **aplicação** é constituída por um menu principal(**index.html**), a partir do qual é possível executar as seguintes funcionalidades do programa, sendo estas:

- **Inserir**(insere um elemento na tabela pretendida);
- **Listar**(mostra uma tabela com os elementos pretendidos);
- **Registar**(regista um elemento na tabela pretendida);
- **Remover**(remove um elemento da tabela selecionada);
- **Editar**(edita um elemento da tabela selecionada);

Para **Inserir/Registar** um elemento da tabela selecionada, a aplicação pede as informações necessárias para a inserção do novo elemento e depois ao ser pressionado o **botão(no ecrã) “Submit”**, o programa adiciona o elemento à tabela(caso as informações inseridas sejam válidas).

Para **Listar** os elementos pedidos basta clicar numa das opções presentes na secção “Listar”, presente no menu principal. Após clicar num desses botões, será levado para uma página que lhe apresentará a tabela desejada.

Para **Remover** um elemento da tabela selecionada, a aplicação pede as informações necessárias para a identificação do elemento a remover e de seguida, caso os dados introduzidos sejam válidos, remove o elemento da tabela.

Para **Editar** um elemento da tabela selecionada, a aplicação pede os dados necessários para identificar o elemento a ser editado e as informações a serem alterados e, caso as informações inseridas sejam válidas, altera os dados do elemento pretendido.

Sempre que uma das instruções termina a sua execução, será apresentado um **botão “Voltar”**, que ao ser pressionado o levará de volta para o menu principal, onde poderá depois executar uma nova tarefa.

Relações entre ficheiros:

Todas as opções no **menu principal(index.html)** possuem uma hiperligação para um ficheiro php que trata de receber o **input** do utilizador(todos os ficheiros php cujo nome comece por “**pede**” desempenha esta função). Após ser inserido os dados pedidos, será então enviado para uma página php que trata de executar a instrução pretendida(ficheiros que comecem por um **verbo** desempenham a ação do mesmo).