

Projeto de Bases de Dados - Parte 3

Grupo 2 - BD4L03
Professor Paulo Carreira

Fábio Sousa	93577	33%	12 horas
João Pereira	93587	33%	12 horas
Pedro Godinho	93608	33%	12 horas

Comandos da criação da base de dados:

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

```
create table local_publico  
(  
    latitude numeric(9,6) not null,  
    longitude numeric(9,6) not null,  
    nome varchar(40) not null,  
    constraint pk_coordinates primary key(latitude, longitude)  
);
```

```
create table item  
(  
    id INT not null,  
    descricao varchar(100) not null,  
    localizacao varchar(40) not null,  
    latitude numeric(9,6) not null,  
    longitude numeric(9,6) not null,  
    constraint pk_item_id primary key(id),  
    constraint fk_item_coordinates foreign key(latitude, longitude) references  
local_publico(latitude, longitude)  
);
```

```
create table anomalia  
(  
    id INT not null,  
    x1 INT not null,  
    y1 INT not null,  
    x2 INT not null,  
    y2 INT not null,  
    imagem varchar(100) not null,
```

```
lingua varchar(20) not null,  
ts TIMESTAMP not null,  
descricao varchar(100) not null,  
tem_anomalia_redacao boolean not null,  
constraint pk_anomalia_id primary key(id)  
);
```

create table anomalia_traducao

```
(  
id INT not null,  
x3 INT not null,  
y3 INT not null,  
x4 INT not null,  
y4 INT not null,  
lingua2 varchar(20) not null,  
constraint fk_anomalia_traducao_id foreign key(id) references anomalia(id),  
constraint pk_anomalia_traducao_id primary key(id)  
);
```

create table duplicado

```
(  
id1 INT not null,  
id2 INT not null,  
constraint fk_id1 foreign key(id1) references item(id),  
constraint fk_id2 foreign key(id2) references item(id),  
constraint pk_duplicado_id1_id2 primary key(id1, id2)  
);
```

create table utilizador

```
(  
email varchar(50) not null,  
password varchar(50) not null,  
constraint pk_utilizador_email primary key(email)  
);
```

create table utilizador_qualificado

```
(  
email varchar(50) not null,  
constraint fk_utilizador_qualificado_email foreign key(email) references  
utilizador(email),  
constraint pk_utilizador_qualificado_email primary key(email)  
);
```

```
create table utilizador_regular
(
    email varchar(50) not null,
    constraint fk_utilizador_regular_email foreign key(email) references
utilizador(email),
    constraint pk_utilizador_regular_email primary key(email)
);
```

```
create table incidencia
(
    anomalia_id INT not null,
    item_id INT not null,
    email varchar(50) not null,
    constraint pk_incidencia_anomalia_id primary key(anomalia_id),
    constraint fk_incidencia_anomalia_id foreign key(anomalia_id) references
anomalia(id),
    constraint fk_incidencia_item_id foreign key(item_id) references item(id),
    constraint fk_incidencia_email foreign key(email) references utilizador(email)
);
```

```
create table proposta_de_correcao
(
    email varchar(50) not null,
    nmr INT not null,
    data_hora TIMESTAMP not null,
    texto varchar(50) not null,
    constraint pk_proposta_de_correcao_email_nmr primary key(email, nmr),
    constraint fk_proposta_de_correcao_email foreign key(email) references
utilizador_qualificado(email)
);
```

```
create table correcao
(
    email varchar(50) not null,
    nmr INT not null,
    anomalia_id INT not null,
    constraint pk_correcao_email_nmr_anomalia_id primary key(email, nmr,
anomalia_id),
    constraint fk_correcao_email_nmr foreign key(email, nmr) references
proposta_de_correcao(email, nmr),
```

```
constraint fk_correcao_anomalia_id foreign key(anomalia_id) references  
incidencia(anomalia_id)  
);
```

SQL:

1.

```
select latitude, longitude, count(*)  
from incidencia i left outer join item t on i.item_id=t.id left outer join anomalia a on  
i.anomalia_id=a.id  
group by latitude, longitude  
having count(*) <= all (  
    select count(*)  
    from incidencia i left outer join item t on i.item_id=t.id left outer join anomalia a  
on i.anomalia_id=a.id  
    group by latitude, longitude);
```

2.

```
select latitude, longitude  
from incidencia i left outer join anomalia a on a.id=i.anomalia_id left outer join  
anomalia_traducao at on at.id=a.id left outer join item it on it.id=i.item_id  
where ts >= '2020-01-01 00:00:00' and ts <= '2020-06-30 23:59:59' and  
tem_anomalia_redacao='f'  
group by latitude, longitude  
having count(*) >= all (  
    select count(*)  
    from incidencia i left outer join anomalia a on a.id=i.anomalia_id left outer join  
anomalia_traducao at on at.id=a.id left outer join item it on it.id=i.item_id  
    where ts >= '2020-01-01 00:00:00' and ts <= '2020-06-30 23:59:59' and  
tem_anomalia_redacao='f'  
    group by latitude, longitude);
```

3.

```
select latitude, longitude
from incidencia i left outer join anomalia a on a.id=i.anomalia_id left outer join
anomalia_traducao at on at.id=a.id left outer join item it on it.id=i.item_id
where ts >= '2020-01-01 00:00:00' and ts <= '2020-06-30 23:59:59' and
tem_anomalia_redacao='f'
group by latitude, longitude
having count(*) >= all (
    select count(*)
    from incidencia i left outer join anomalia a on a.id=i.anomalia_id left outer join
anomalia_traducao at on at.id=a.id left outer join item it on it.id=i.item_id
    where ts >= '2020-01-01 00:00:00' and ts <= '2020-06-30 23:59:59' and
tem_anomalia_redacao='f'
    group by latitude, longitude);
```

4.

```
select i.email
from correcao c join incidencia i on c.anomalia_id = i.anomalia_id
where c.email = i.email;
```

RIs:

drop trigger if exists email_utilizador_regular_trigger on utilizador_regular;

create or replace function email_utilizador_regular_verifier()

returns trigger as

\$\$

begin

if new.email in (Select email from utilizador_qualificado) then

raise exception 'Este Utilizador já é um utilizador qualificado.';

end if;

return new;

end;

\$\$ language plpgsql;

create trigger email_utilizador_regular_trigger

before insert on utilizador_regular

for each row execute procedure email_utilizador_regular_verifier();

drop trigger if exists email_verifier_trigger on utilizador_qualificado;

create or replace function email_utilizador_qualificado_verifier()

returns trigger as

\$\$

begin

if new.email in (Select email from utilizador_regular) then

raise exception 'Este Utilizador já é um utilizador regular.';

end if;

return new;

end;

\$\$ language plpgsql;

create trigger email_utilizador_qualificado_trigger

before insert on utilizador_qualificado

for each row execute procedure email_utilizador_qualificado_verifier();

drop trigger if exists zona_sobreposta_trigger on anomalia_traducao;

create or replace function zona_sobreposta()

returns trigger as

```

$$
begin
    if new.x3 >= all (Select x1 from anomalia where id=new.id) and new.x3 <= all
    (Select x2 from anomalia where id=new.id) then
        if new.y3 >= all (Select y1 from anomalia where id=new.id) and new.y3 <= all
        (Select y2 from anomalia where id=new.id) then
            return null;
        end if;
    end if;
    if new.x4 >= all (Select x1 from anomalia where id=new.id) and new.x4 <= all
    (Select x2 from anomalia where id=new.id) then
        if new.y4 >= all (Select y1 from anomalia where id=new.id) and new.y4 <= all
        (Select y2 from anomalia where id=new.id) then
            return null;
        end if;
    end if;
    return new;
end;
$$ language plpgsql;

```

```

create trigger zona_sobreposta_trigger
before insert on anomalia_traducao
for each row execute procedure zona_sobreposta();

```


Desenvolvimento da aplicação:

O menu principal da aplicação encontra-se no ficheiro `./index.php`. Este ficheiro contém apenas links para os ficheiros responsáveis pelas tarefas. As tarefas pedidas em a) encontra-se nas pastas `./local`, `./item` e `./anomalia`. Em cada uma destas pastas existe um ficheiro `add.php` e um ficheiro `update.php`. Add é responsável por mostrar o form com que o utilizador interagirá e passar os inputs para `update.php`, que faz a query. A exceção é a anomalia, que tem um ficheiro extra `update_trad.php`, chamado por `update.php` caso se trate de uma anomalia de tradução.

As tarefas pedidas em b) encontra-se em `./correcao` e `./proposta_de_correcao`, nas quais temos os ficheiros `gerir.php`, `edit.php`, `update_add.php`, `update_edit.php`, e `update_remove.php`. Os ficheiros com o prefixo `update` são os que fazem as queries, enquanto os outros são responsáveis apenas pelos forms. Em `./correcao` são omitidos os ficheiros `edit.php` e `update_edit.php`, uma vez que editar uma correção impossível, sendo que a correção tem apenas campos de primary key, sendo que a única forma de o fazer será remover uma correção e voltar a adicionar. A tarefa pedida em c) encontra-se em `./utilizadores/list.php`.

As tarefas pedidas em e) e em f) encontra-se nas pastas `./find_by_place` e `./find_by_coord`, respetivamente. Em ambos os casos, há um ficheiro (`./list.php` e `./coord.php`) chamado pelo `./index.php` que mostrará o form e fará o POST request ao outro ficheiro na diretoria, `show.php`, que fará a query em questão e mostrará o resultado.

Índices:

4.1

```
ALTER TABLE incidencia ADD PRIMARY KEY(item_id);
```

Ao fazer com que `item_id` seja uma primary key permite com que a pesquisa seja rápida.

4.2

```
CREATE INDEX ts_idx ON anomalia USING HASH (ts);
```

Ao criar um index na anomalia permite que a querie seja mais rápida.