

Projeto de Base de Dados – Parte 3

Pedro Moraes 93607

Francisco Bento 93581

Francisco Rosa 93578

Em termos de contribuição:

Pedro Moraes – 33% / 80 horas

Francisco Bento – 33% / 80 horas

Francisco Rosa – 33% / 80 horas

Número do grupo: 6

Turno: 6ª às 11:00

Docente: Paulo Carreira

Comandos da criação da base de dados(schema.sql)

```
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;

-----
-- Table Creation
-----

-- Named constraints are global to the database.
-- Therefore the following use the following naming rules:
-- 1. pk_table for names of primary key constraints
-- 2. fk_table_another for names of foreign key constraints

create table local_publico
(
    latitude decimal(9,6) not null,
    longitude decimal(9,6) not null,
    nome varchar(80) not null,
    constraint pk_local_publico primary key(latitude,longitude),
    constraint chk_lat check(latitude between -90 and 90),
    constraint chk_lon check(longitude between -180 and 180));

create table item
(
    id numeric(10) not null,
    descricao text not null,
    localizacao varchar(80) not null,
    latitude decimal(9,6) not null,
    longitude decimal(9,6) not null,
    constraint pk_item primary key(id),
    constraint fk_item_local_publico foreign key(latitude,longitude) references local_publico(latitude,longitude) on delete cascade on update cascade,
    constraint chk_lat check(latitude between -90 and 90),
    constraint chk_lon check(longitude between -180 and 180));
```

```
create table anomalia
(
    id numeric(10) not null,
    zona varchar(16) not null,
    imagem varchar(60) not null,
    lingua varchar(20) not null,
    ts timestamp not null,
    descricao text not null,
    tem_anomalia_redacao boolean not null,
    constraint pk_anomalia primary key(id),
    constraint chk_imagem check(imagem like '%.jpeg'),
    constraint chk_redacao check(tem_anomalia_redacao in('true','false')));

create table anomalia_traducao
(
    id numeric(10) not null,
    zona2 varchar(16) not null,--(_____,_____,_____\n)
    lingua2 varchar(20) not null,
    constraint pk_anomalia_traducao primary key(id),
    constraint fk_anomalia_traducao_anomalia foreign key(id) references anomalia(id) on delete cascade on update cascade);

create table duplicado
(
    item1 numeric(10) not null,
    item2 numeric(10) not null,
    constraint pk_duplicado primary key (item1,item2),
    constraint fk_duplicado_item1 foreign key (item1) references item(id) on delete cascade on update cascade,
    constraint fk_duplicado_item2 foreign key (item2) references item(id) on delete cascade on update cascade);

create table utilizador
(
    email varchar(80) not null,
    password varchar(20) not null,
    constraint pk_utilizador primary key (email),
    constraint chk_email check(email like '%@%'));

create table utilizador_qualificado
(
    email varchar(80) not null,
    constraint pk_utilizador_qualificado primary key (email),
    constraint fk_utilizador_qualificado_utilizador foreign key (email) references utilizador(email) on delete cascade on update cascade,
    constraint chk_email check(email like '%@%'));
```

```

create table utilizador_regular
(email varchar(80) not null,
constraint pk_utilizador_regular primary key (email),
constraint fk_utilizador_regular_utilizador foreign key (email) references utilizador(email) on delete cascade on update cascade,
constraint chk_email check(email like '%@%'));

create table incidencia
(anomalia_id numeric(10) not null,
item_id numeric(10) not null,
email varchar(80) not null,
constraint pk_incidencia primary key (anomalia_id),
constraint fk_incidencia_anomalia foreign key (anomalia_id) references anomalia(id) on delete cascade on update cascade,
constraint fk_incidencia_item foreign key (item_id) references item(id) on delete cascade on update cascade,
constraint fk_incidencia_utilizador foreign key (email) references utilizador(email));

create table proposta_de_correcao
(email varchar(80) not null,
nro numeric(10) not null,
data_hora timestamp not null,
texto text not null,
constraint pk_proposta_de_correcao primary key (email,nro),
constraint fk_proposta_de_correcao_utilizador_qualificado foreign key (email) references utilizador_qualificado(email),
constraint chk_email check(email like '%@%'));

create table correcao
(email varchar(80) not null,
nro numeric(10) not null,
anomalia_id numeric(10) not null,
constraint pk_correcao primary key (email,nro,anomalia_id),
constraint fk_correcao_proposta_de_correcao foreign key (email,nro) references proposta_de_correcao(email,nro) on delete cascade on update cascade,
constraint fk_correcao_incidencia foreign key (anomalia_id) references incidencia(anomalia_id) on delete cascade on update cascade,
constraint chk_email check(email like '%@%'));

```

Consultas em SQL (queries.sql)

```

--1. Qual o local publico onde estao registadas menos anomalias? VERIFICADA

select nome,count(*) from local_publico
natural join item it
join incidencia i on it.id=i.item_id
group by nome
having count(*) <= all(select count(*)
from local_publico
natural join item it
join incidencia i on it.id=i.item_id
group by nome);

--2. Qual o local publico com mais anomalias de traducao no 1º semestre de 2020?

select distinct nome,count(*) from local_publico
natural join item it
join incidencia i on i.item_id=it.id
join anomalia a on i.anomalia_id=a.id
join anomalia_traducao atd on a.id=atd.id
where ts between '2020-01-01 00:00:00' and '2020-06-30 23:59:59'
group by nome
having count(*) >= all(select count(*)
from local_publico
natural join item it
join incidencia i on i.item_id=it.id
join anomalia a on i.anomalia_id=a.id
join anomalia_traducao atd on a.id=atd.id
where ts between '2020-01-01 00:00:00' and '2020-06-30 23:59:59'
group by nome);

```

```

--3. Quais sao os utilizadores que registaram em 2020 propostas de correcao em todos os locais
--publicos situados a sul de Rio Maior (Portugal)?

select distinct email from proposta_de_correcao p
where not exists (
    select it.id
    from item it
    join incidencia i on it.id=i.item_id
    where it.latitude < 39.336775
    except
    select inc.item_id
    from incidencia inc
    where p.email = inc.email)
and p.data_hora between '2020-01-01 00:00:00' and '2020-12-31 23:59:59';

--4. Quais sao os utilizadores que apresentaram propostas de correcao para todas anomalias por eles
--registadas no ano de 2020?

select distinct email from correcao c
where not exists(
    select i.email
    from incidencia i join anomalia a on a.id=i.anomalia_id
    where a.ts between '2020-01-01 00:00:00' and '2020-12-31 23:59:59'
    except
    select c2.email
    from correcao c2
    join incidencia i2 on i2.anomalia_id=c2.anomalia_id
    where c2.email = c.email);

```

Restrições de Integridade (RI.sql)

```
-----RI-1, RI-2-----
drop trigger if exists anomalia_traducao_bit on anomalia_traducao;

create or replace function anomalia_traducao_bif()
    returns trigger as
$$
    declare newx1 decimal default 0;
           newy1 decimal default 0;
           newx2 decimal default 0;
           newy2 decimal default 0;
           oldx1 decimal default 0;
           oldy1 decimal default 0;
           oldx2 decimal default 0;
           oldy2 decimal default 0;
           old text[];
           coortostr text[];

    begin
        if new.id in (select id from anomalia) then
            coortostr = regexp_split_to_array(new.zona2,'');
            newx1=cast(coortostr[1] as decimal);
            newy1=cast(coortostr[2] as decimal);
            newx2=cast(coortostr[3] as decimal);
            newy2=cast(coortostr[4] as decimal);

            Select (regexp_split_to_array(zona,'')) into old
            from anomalia
            where new.id=id;

            oldx1=cast(old[1] as decimal);
            oldy1=cast(old[2] as decimal);
            oldx2=cast(old[3] as decimal);
            oldy2=cast(old[4] as decimal);

            if new.lingua2 <> (select lingua from anomalia where new.id = id) and
                not ((newx1 < oldx2 and newx2 > oldx2) or (newx2 > oldx1 and newx1 < oldx1)
                    or (newy1 < oldy2 and newy2 > oldy2) or (newy2 > oldy1 and newy1 < oldy1)) then
                return new;
            end if;
        end if;
    end;

$$ language plpgsql;

create trigger anomalia_traducao_bit
before insert on anomalia_traducao
for each row execute procedure anomalia_traducao_bif();
```

```

-----RI-4,5,6-----
drop trigger if exists utilizador_bit on utilizador;

create or replace function utilizador_bif()
    returns trigger as

$$
begin
    raise Exception 'Nao foi possivel inserir o utilizador';
end;
$$ language plpgsql;

create trigger utilizador_bit
before insert on utilizador
for each row execute procedure utilizador_bif();

drop function utilizador_qualificado_insert(email varchar(80), password varchar(20));

create or replace function utilizador_qualificado_insert(email varchar(80), password varchar(20))
    returns void as

$$
begin
    if email not in (select email from utilizador_regular) then
        alter table utilizador disable trigger utilizador_bit;
        insert into utilizador values(email, password);
        insert into utilizador_qualificado values(email);
        alter table utilizador enable trigger utilizador_bit;
    else
        raise exception 'Este utilizador ja e regular';
    end if;

end;
$$ language plpgsql;

```

```

drop function utilizador_regular_insert(email varchar(80), password varchar(20));

create or replace function utilizador_regular_insert(email varchar(80), password varchar(20))
    returns void as

$$
begin
    if email not in (select email from utilizador_qualificado) then
        alter table utilizador disable trigger utilizador_bit;
        insert into utilizador values(email, password);
        insert into utilizador_regular values(email);
        alter table utilizador enable trigger utilizador_bit;
    else
        raise exception 'Este utilizador ja e qualificado';
    end if;

end;
$$ language plpgsql;

```

Explicação da arquitetura da aplicação PHP e relações entre os diversos ficheiros

No que consta a arquitetura PHP, optámos primeiramente por criar uma homepage que permitisse o acesso a cada uma das alíneas.

Considerámos que qualquer alteração provocada em qualquer registo tem consequências na base de dados globalmente.

Ou seja, por exemplo, se o utilizador quiser inserir uma anomalia, terá que preencher apenas os dados referentes a essa anomalia. No entanto, se pretender inserir um novo item no sistema terá que, para além de registar a descrição e a localização do mesmo, inserir os dados associados a uma nova anomalia correspondente a esse item.

Para inserir um novo local público, considerámos que apenas faria sentido se registássemos uma nova incidência composta por um novo local público, um novo item e uma nova anomalia.

Dito isto, juntámos as alíneas a) e d) numa só página.

De seguida, apresentamos uma figura que ilustra a nossa estrutura PHP e mapeia as relações entre as diferentes páginas, para ser mais clara a navegação.

Optámos também por mostrar o conteúdo relevante das tabelas (respetivo a cada exercício ao utilizador para conseguir analisar os dados mais facilmente e torná-lo mais consciente do manuseamento da aplicação.

Aqui se encontra o link da Homepage, web.ist.utl.pt/ist193581/homepage.php.

