



# Projeto de Base de Dados

## Translate Right - Parte 3 Grupo 48

TURNO: Quarta-feira, 08:00, L13

PROFESSOR: Pedro Manuel Moreira Vaz Antunes de Sousa

Número de Aluno	Nome do Aluno	Tempo disponibilizado
89421	Bruno Meira	15 (25%)
89513	Nikoletta Matur	15 (25%)
89276	Rui Baeta	15 (25%)
89559	Vasco Faria	15 (25%)

## 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_correcao  cascade;  
drop table correcao           cascade;
```

```
create table local_publico (  
    lp_latitude decimal(8, 6) not null,  
    lp_longitude decimal(9, 6) not null,  
    lp_nome varchar(80) not null,  
    constraint pk_local_publico primary key (lp_latitude, lp_longitude)  
);
```

```
create table item (  
    i_id serial not null unique,  
    i_descricao varchar(80) not null,  
    i_localizacao varchar(80) not null,  
    lp_latitude decimal(8, 6) not null,  
    lp_longitude decimal(9, 6) not null,  
    constraint pk_item primary key (i_id),  
    constraint fk_i_local_publico foreign key (lp_latitude, lp_longitude)  
        references local_publico(lp_latitude, lp_longitude)  
        on delete cascade on update cascade  
);
```

```
create table utilizador (  
    u_email varchar(80) not null unique,  
    u_password varchar(80) not null,  
    constraint pk_utilizador primary key (u_email)  
);
```

```
create table utilizador_qualificado (  
    u_email    varchar(80) not null unique,  
    constraint pk_utilizador_qualificado primary key (u_email),  
    constraint fk_uq_utilizador          foreign key (u_email)  
        references utilizador(u_email)  
        on delete cascade on update cascade  
);
```

```
create table utilizador_regular (  
    u_email    varchar(80) not null unique,  
    constraint pk_utilizador_regular primary key (u_email),  
    constraint fk_ur_utilizador      foreign key (u_email)  
        references utilizador(u_email)  
        on delete cascade on update cascade  
);
```

```
create table anomalia (  
    a_id                serial          not null unique,  
    a_zona              varchar(8) not null,  
    a_imagem            varchar(80) not null,  
    a_lingua            varchar(80) not null,  
    a_ts               timestamp default current_timestamp(0),  
    a_descricao         varchar(80) not null,  
    a_tem_anomalia_redacao boolean    not null,  
    constraint pk_anomalia primary key (a_id),  
    constraint ck_zone    check (substring(a_zona, 1, 3)::int8 >= 0  
        and substring(a_zona, 4, 2) = ', '  
        and substring(a_zona, 6, 3)::int8 >=0)  
);
```

```
create function fn_Check_Zone (zone text)  
returns integer as $$ DECLARE a_count integer;  
BEGIN SELECT COUNT(*) into a_count  
        FROM anomalia WHERE a_zona = zone;  
RETURN a_count;  
END $$ LANGUAGE plpgsql;
```

```
create function fn_Check_Lingua (lingua text)  
returns integer as $$ DECLARE a_count integer;  
BEGIN SELECT COUNT(*) into a_count  
        FROM anomalia WHERE a_lingua = lingua;  
RETURN a_count; END $$ LANGUAGE plpgsql;
```

```
create table anomalia_traducao (  
    a_id      smallint    not null unique,  
    at_zona2  varchar(8) not null,  
    at_lingua2 varchar(80) not null,  
    constraint pk_anomalia_traducao primary key (a_id),  
    constraint fk_at_anomalia      foreign key (a_id)  
        references anomalia(a_id)  
        on delete cascade on update cascade,  
    constraint ck_zone2 check (substring(at_zona2, 1, 3)::int8 >= 0  
        and substring(at_zona2, 4, 2) = ', '  
        and substring(at_zona2, 6, 3)::int8 >=0),  
    constraint ck_zone_diff check (fn_Check_Zone(at_zona2) = 0),  
    constraint ck_lingua_diff check (fn_Check_Lingua(at_lingua2) = 0)  
);
```

```
create table duplicado (  
    i_id1 smallint not null,  
    i_id2 smallint not null,  
    constraint pk_duplicado primary key (i_id1, i_id2),  
    constraint fk_d_item1   foreign key (i_id1)  
        references item(i_id)  
        on delete cascade on update cascade,  
    constraint fk_d_item2   foreign key (i_id2)  
        references item(i_id)  
        on delete cascade on update cascade,  
    constraint check_items check (i_id1 < i_id2)  
);
```

```
create table incidencia (  
    a_id      smallint    not null,  
    i_id      smallint    not null,  
    u_email   varchar(80) not null,  
    constraint pk_incidencia primary key (a_id),  
    constraint fk_ic_anomalia foreign key (a_id)  
        references anomalia(a_id)  
        on delete cascade on update cascade,  
    constraint fk_ic_item     foreign key (i_id)  
        references item(i_id)  
        on delete cascade on update cascade,  
    constraint fk_ic_utilizador foreign key (u_email)  
        references utilizador(u_email)  
        on delete cascade on update cascade);
```

```

create table proposta_correcao (
    pc_nro          smallint      not null unique,
    pc_data_hora    varchar(80) not null,
    pc_texto        varchar(80) not null,
    u_email         varchar(80) not null,
    constraint pk_proposta_correcao      primary key (u_email, pc_nro),
    constraint fk_pc_utilizador_qualificado foreign key (u_email)
        references utilizador_qualificado(u_email)
        on delete cascade on update cascade
);

create table correcao (
    u_email    varchar(80) not null,
    pc_nro     smallint    not null,
    a_id       smallint    not null,
    constraint pk_correcao              primary key (u_email, pc_nro, a_id),
    constraint fk_c_proposta_correcao  foreign key (u_email, pc_nro)
        references proposta_correcao(u_email, pc_nro) on delete cascade,
    constraint fk_c_anomalia           foreign key (a_id)
        references incidencia(a_id) on delete cascade
);

```

## CONSULTAS EM SQL

-- 1.

```

SELECT lp_nome, lp_latitude, lp_longitude FROM (
    local_publico NATURAL JOIN (
        SELECT lp_latitude, lp_longitude FROM (
            item NATURAL JOIN
                (SELECT a_id, i_id FROM incidencia) AS A
        ) GROUP BY lp_latitude, lp_longitude
        HAVING
            COUNT(*) = (
                SELECT MAX(D.CNT) FROM (
                    SELECT COUNT(*) AS CNT FROM (
                        item NATURAL JOIN (SELECT a_id, i_id FROM
                            incidencia) AS B
                    ) AS C GROUP BY (lp_latitude, lp_longitude)
                ) AS D
            ) AS E
    )
);

```

-- 2.

```
SELECT u_email FROM (  
    SELECT u_email FROM (  
        incidencia NATURAL JOIN anomalia NATURAL JOIN  
        anomalia_traducao NATURAL JOIN utilizador_regular  
    ) AS A WHERE a_ts >= '2019-01-01 00:00:00'  
    AND a_ts < '2019-07-01 00:00:00' GROUP BY u_email  
HAVING  
COUNT(*) = (  
    SELECT MAX(C.cnt) FROM (  
        SELECT COUNT(*) AS cnt FROM (  
            incidencia NATURAL JOIN anomalia NATURAL JOIN  
            anomalia_traducao NATURAL JOIN utilizador_regular  
        ) AS B WHERE a_ts >= '2019-01-01 00:00:00'  
        AND a_ts < '2019-07-01 00:00:00' GROUP BY (u_email)  
    ) AS C  
)) AS E;
```

-- 3.

```
SELECT DISTINCT u_email FROM (  
    incidencia NATURAL JOIN  
    (SELECT MAX(i_id) AS i_id, lp_latitude, lp_longitude FROM  
        item WHERE lp_latitude > 39.336775  
        GROUP BY lp_latitude, lp_longitude) AS locals  
    NATURAL JOIN (SELECT * FROM  
        anomalia WHERE a_ts >= '2019-01-01 00:00:00'  
        AND a_ts < '2020-01-01 00:00:00') AS dates  
    ) AS aux GROUP BY (u_email)  
HAVING  
COUNT(*) = (  
    SELECT COUNT(*) AS cnt FROM  
        local_publico WHERE lp_latitude > 39.336775  
    );
```

-- 4.

```
SELECT DISTINCT u_email FROM (  
    incidencia NATURAL JOIN anomalia NATURAL JOIN  
    utilizador_qualificado NATURAL JOIN item  
) WHERE lp_latitude < 39.336775 AND a_ts >= '2019-01-01 00:00:00'  
AND a_ts < '2020-01-01 00:00:00'  
EXCEPT(SELECT DISTINCT incid.u_email FROM (  
    correcao NATURAL JOIN proposta_correcao) AS correcoes  
    NATURAL JOIN (SELECT DISTINCT u_email FROM (  
        incidencia NATURAL JOIN anomalia NATURAL JOIN  
        utilizador_qualificado NATURAL JOIN item  
    ) WHERE lp_latitude < 39.336775  
    AND a_ts >= '2019-01-01 00:00:00'  
    AND a_ts < '2020-01-01 00:00:00') AS incid  
);
```

## EXPLICAÇÃO DA ARQUITETURA DA APLICAÇÃO PHP

Esta aplicação tem como ponto de vista o utilizador que não é um administrador de sistemas e, como tal, os utilizadores não terão acesso total (não poderão, por exemplo, apagar propostas de correção que não sejam da sua autoria). Sendo, desta forma, necessária a criação de uma página de login para a autenticação do mesmo. No login, são validadas as credenciais e guardadas em cookies, para nos permitir gerir as permissões de cada utilizador.

Após o login[1], o utilizador vê um menu que corresponde a todas as ações possíveis na aplicação e que são botões que o redirecionam para o formulário correspondente. Isto é: escolhe-se a categoria que contenha a ação pretendida e preenche-se formulários (no caso de “inserts” e “lists”) e submete-se ou então, seleciona-se o botão que corresponda às ações de editar (editando de antemão o campo que o permita), selecionar ou remover da linha onde o botão se encontra (no caso dos “edits”, “registers” e “removes”).

[1] Efetue o login com [professor@gmail.com](mailto:professor@gmail.com) e password: professor