



Universidade do Porto
Faculdade de Engenharia
FEUP

OBJECT RELATIONAL ASSIGNMENT

OR1

Tecnologias de Bases de Dados
4º ano do Mestrado Integrado em Engenharia Informática e
Computação

Elementos do grupo F:

Catarina Ramos - up201406219 - up201406219@fe.up.pt
Inês Gomes - up201405778 - up201405778@fe.up.pt
Mário Fernandes - up201201705 - up201201705@fe.up.pt

19 de Maio de 2018

Questão 1

Design an object-relational data model for this situation, exploiting the SQL3 extensions. The model may be drawn schematically and then in actual DDL and implemented.

Para desenvolver este modelo de base de dados o grupo experimentou várias arquiteturas optando pela representada na *fig 1*. As arquiteturas exploradas tinham diferenças essencialmente a nível de ligação entre as tabelas, sendo as seguintes as mais pertinentes:

- Utilização de uma tabela extra que aglomerava todos os distritos, concelhos e freguesias numa só, permitindo facilmente fazer uma pesquisa hierárquica utilizando o código e o pai de cada elemento. Este esquema tinha um ponto negativo de duplicar muita informação ou de necessitar de ligações extras entre tabelas que se conseguiam evitar.
- Utilização de referências entre tabelas em vez de tabelas aninhadas. Nesta alternativa verificou-se que se necessitava de fazer operações desnecessárias em cada query que exigisse a pesquisa em duas ou mais tabelas.

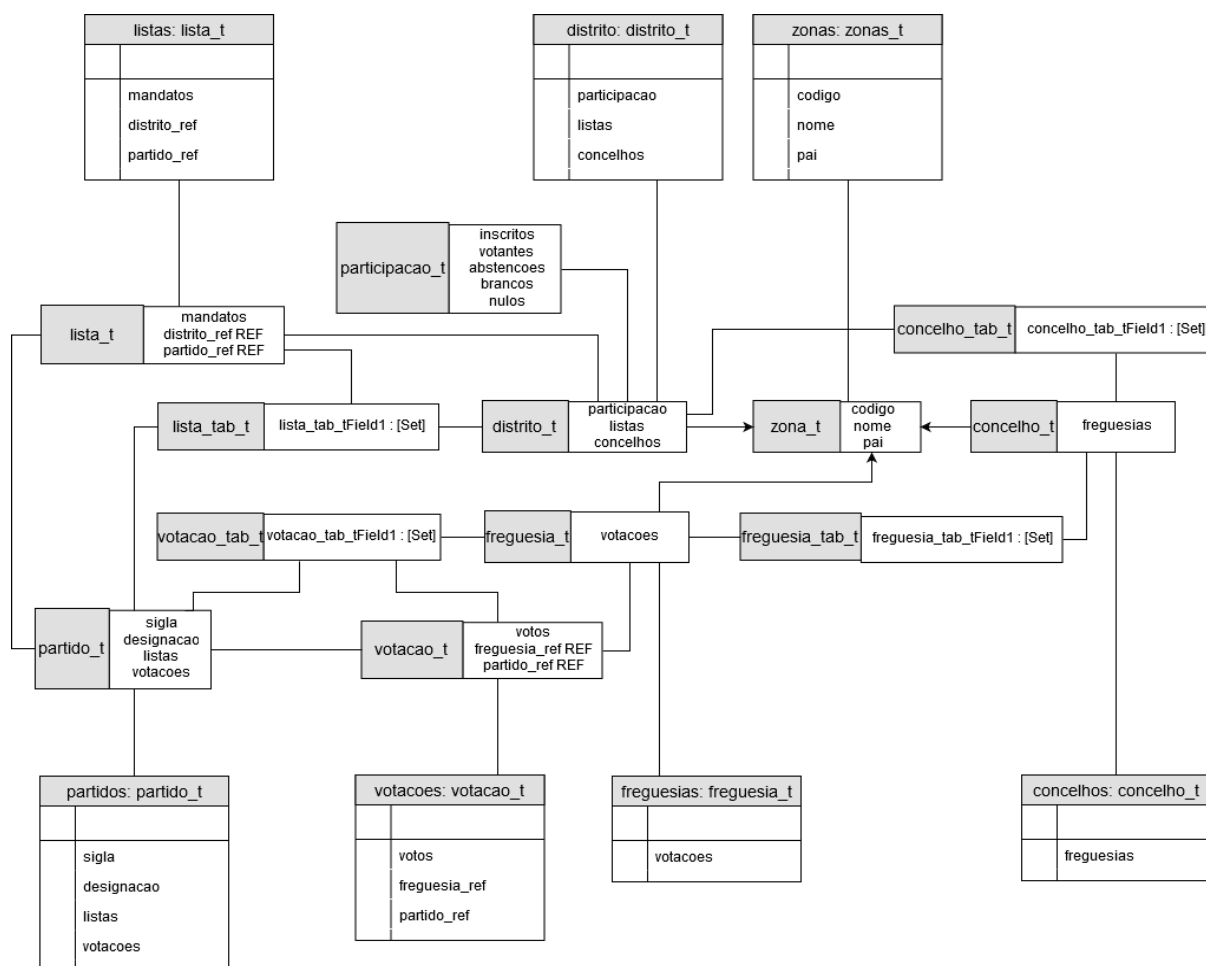


Fig 1: Modelo objeto-relacional do projeto

Código SQL referente à implementação dos tipos e tabelas na figura 1.

1. Tipos

```
create type zona_t as object(  
    codigo number(6),  
    nome varchar2(50),  
    pai number(4)) NOT FINAL;  
  
create type participacao_t as object (  
    inscritos number(10),  
    votantes number(10),  
    abstencoes number(10),  
    nulos number(10),  
    brancos number(10));  
  
create type distrito_t under zona_t (  
    regiao varchar2(1),  
    participacao participacao_t);  
  
create type concelho_t under zona_t ();  
  
create type freguesia_t under zona_t();  
  
create type partido_t as object(  
    sigla varchar2(10),  
    designacao varchar2(100),  
    map member function mandatos_totais return integer);  
  
create type lista_t as object(  
    distrito_ref ref distrito_t,  
    partido_ref ref partido_t,  
    mandatos number(3));  
  
create type votacao_t as object(  
    freguesia_ref ref freguesia_t,  
    partido_ref ref partido_t,  
    votos number(10));
```

2. Tabelas Aninhadas

```
--tabela de referências para lista  
create type lista_tab_t as table of ref lista_t;  
--tabela de referências para votacao  
create type votacao_tab_t as table of ref votacao_t;  
--tabela de referências para concelho
```

```

create type concelho_tab_t as table of ref concelho_t;
--tabela de referências para freguesia
create type freguesia_tab_t as table of ref freguesia_t;

alter type distrito_t add attribute listas lista_tab_t cascade;
alter type distrito_t add attribute concelhos concelho_tab_t cascade;
alter type concelho_t add attribute freguesias freguesia_tab_t cascade;
alter type freguesia_t add attribute votacoes votacao_tab_t cascade;
alter type partido_t add attribute listas lista_tab_t cascade;
alter type partido_t add attribute votacoes votacao_tab_t cascade;

```

3. Criação de Tabelas

```

create table distritos of distrito_t nested table listas store as
listas_distritos_tab nested table concelhos store as concelhos_distritos_tab;

create table concelhos of concelho_t nested table freguesias store as
freguesias_concelhos_tab;

create table freguesias of freguesia_t nested table votacoes store as
votacoes_freguesias_tab;

create table partidos of partido_t nested table listas store as
listas_partidos_tab nested table votacoes store as votacoes_partidos_tab;

create table listas of lista_t;

create table votacoes of votacao_t;

```

Questão 2

Populate the object relational model with the data in the relational database.

O grupo utilizou a informação contida no utilizador *GTD7* das tabelas apresentadas no esquema relacional do enunciado deste trabalho para a população da base de dados.

A inserção foi feita com as queries apresentadas abaixo que selecionam a informação diretamente das tabelas do utilizador.

Código SQL referente à população das tabelas.

```
-- Partidos
insert into partidos(sigla,designacao)
  select p.sigla, p.designacao
  from GTD7.PARTIDOS p;

-- Distritos
insert into distritos (CODIGO, NOME, PAI, PARTICIPACAO,REGIAO)
  select d.CODIGO, d.NOME, 0, participacao_t(p.INSCRITOS, p.VOTANTES,
p.ABSTENCOES, p.NULOS, p.BRANCOS), d.REGIAO
  from GTD7.DISTRITOS d
  join GTD7.PARTICIPACOES p on d.CODIGO = p.DISTRITO;

-- Listas
insert into listas(DISTRITO_REF, PARTIDO_REF, MANDATOS)
  select ref(d), ref(p), l.MANDATOS
  from distritos d
  join GTD7.listas l on d.codigo = l.distrito
  join partidos p on p.sigla = l.partido;

update distritos d set LISTAS = cast( multiset (
(select ref(l) from LISTAS l where DISTRITO_REF =
(select ref(s) from DISTRITOS s where s.codigo = d.codigo) ) ) as LISTA_TAB_T);

update partidos p set LISTAS = cast( multiset (
(select ref(l) from LISTAS l where PARTIDO_REF =
(select ref(s) from PARTIDOS s where s.sigla = p.sigla) ) ) as LISTA_TAB_T);

-- Concelhos
insert into concelhos(codigo,nome,pai)
  select *
  from GTD7.CONCELHOS c;

update DISTRITOS d set CONCELHOS = cast( multiset(
select ref(c) from CONCELHOS c where c.PAI = d.CODIGO) as CONCELHO_TAB_T);
```

```

-- Freguesias
insert into freguesias(codigo,nome,pai)
  select f.CODIGO, f.nome, f.CONCELHO
  from GTD7.FREGUESIAS f;

update CONCELHOS c set FREGUESIAS = cast( multiset(
select ref(f) from FREGUESIAS f where f.PAI = c.CODIGO) as FREGUESIA_TAB_T);

-- Votações
insert into votacoes(FREGUESIA_REF,PARTIDO_REF,votos)
  select ref(f), ref(p), GTD7.VOTACOES.VOTOS
  from GTD7.VOTACOES
  join FREGUESIAS f on f.CODIGO = GTD7.VOTACOES.FREGUESIA
  join PARTIDOS p on p.SIGLA = GTD7.VOTACOES.PARTIDO;

update freguesias f set VOTACOES = cast( multiset (
(select ref(v) from VOTACOES v where FREGUESIA_REF =
(select ref(s) from FREGUESIAS s where s.codigo = f.codigo))) as VOTACAO_TAB_T);

update partidos p set VOTACOES = cast( multiset (
(select ref(v) from VOTACOES v where PARTIDO_REF=
(select ref(s) from PARTIDOS s where s.sigla = p.sigla) ) ) as VOTACAO_TAB_T);

```

Questão 3

Add some methods that may be useful for some common SQL queries.

1. Freguesia

- a. Retorna o número de votos total
- b. Retorna o número de votos do partido vencedor
- c. Retorna uma referência para o partido vencedor na freguesia.

```
alter type freguesia_t add MEMBER FUNCTION votos_total RETURN INTEGER cascade;
alter type freguesia_t add MEMBER FUNCTION votos_partido_vencedor RETURN INTEGER
cascade;
alter type freguesia_t add MEMBER FUNCTION partido_vencedor RETURN ref partido_t
cascade;

create or replace TYPE BODY freguesia_t AS
-- a)
MEMBER FUNCTION votos_total RETURN INTEGER IS vts INTEGER;
BEGIN
    vts := 0;
    SELECT SUM(v.votos) INTO vts
    FROM VOTACOES v
    WHERE v.FREGUESIA_REF.CODIGO = self.CODIGO;
    return vts;
END votos_total;

-- b)
MEMBER FUNCTION votos_partido_vencedor RETURN INTEGER IS vts INTEGER;
BEGIN
    vts := 0;
    SELECT MAX(v.votos) INTO vts
    FROM VOTACOES v
    WHERE v.FREGUESIA_REF.CODIGO = self.CODIGO;
    return vts;
END votos_partido_vencedor;

-- c)
MEMBER FUNCTION partido_vencedor RETURN ref partido_t IS partido ref
partido_t;
BEGIN
    SELECT v.COLUMN_VALUE.PARTIDO_REF INTO partido
    FROM FREGUESIAS f, table(f.VOTACOES) v
    WHERE f.CODIGO = self.CODIGO AND v.COLUMN_VALUE.VOTOS =(
        SELECT MAX(v.column_value.votos)
        FROM FREGUESIAS f, table(f.VOTACOES) v
        WHERE f.CODIGO = self.CODIGO
    );
    return partido;
```

```
END partido_vencedor;  
END;
```

2. Concelho

- a. Retorna uma referência para o partido vencedor na freguesia.
- b. Retorna o número total de votos do partido vencedor.
- c. Retorna o número total de votos.

```
alter type concelho_t add MEMBER FUNCTION votos_total RETURN INTEGER cascade;  
alter type concelho_t add MEMBER FUNCTION votos_partido_vencedor RETURN INTEGER  
cascade;  
alter type concelho_t add MEMBER FUNCTION partido_vencedor RETURN ref partido_t  
cascade;  
create or replace TYPE BODY concelho_t AS  
  -- a)  
  MEMBER FUNCTION votos_total RETURN INTEGER IS vts INTEGER;  
  BEGIN  
    vts := 0;  
    SELECT SUM(f.COLUMN_VALUE.votos_total()) INTO vts  
      FROM CONCELHOS c, table(c.FREGUESIAS) f  
      WHERE c.CODIGO = self.CODIGO;  
    return vts;  
  END votos_total;  
  
  -- b)  
  MEMBER FUNCTION votos_partido_vencedor RETURN INTEGER IS vts INTEGER;  
  BEGIN  
    vts := 0;  
    SELECT MAX(f.COLUMN_VALUE.votos_partido_vencedor()) INTO vts  
      FROM CONCELHOS c, table(c.FREGUESIAS) f  
      WHERE c.CODIGO = self.CODIGO;  
    return vts;  
  END votos_partido_vencedor;  
  
  -- c)  
  MEMBER FUNCTION partido_vencedor RETURN ref partido_t IS partido ref  
partido_t;  
  BEGIN  
    SELECT f.COLUMN_VALUE.partido_vencedor() INTO partido  
      FROM CONCELHOS c, table(c.FREGUESIAS) f  
      WHERE c.CODIGO = self.CODIGO  
      AND f.COLUMN_VALUE.votos_partido_vencedor() = (  
        SELECT MAX(f.COLUMN_VALUE.votos_partido_vencedor())  
          FROM CONCELHOS c, table(c.FREGUESIAS) f  
          WHERE c.CODIGO = self.CODIGO  
        );  
    return partido;  
  END partido_vencedor;  
END;
```


3. Distrito

- a. Participações correctas
- b. Retorna uma referência para o partido vencedor na freguesia.
- c. Retorna o número total de votos do partido vencedor.
- d. Retorna o número total de votos.

```
alter type distrito_t add MEMBER FUNCTION participacao_correcta RETURN BOOLEAN
cascade;
alter type distrito_t add MEMBER FUNCTION votos_total RETURN INTEGER cascade;
alter type distrito_t add MEMBER FUNCTION votos_partido_vencedor RETURN INTEGER
cascade;
alter type distrito_t add MEMBER FUNCTION partido_vencedor RETURN ref partido_t
cascade;
create or replace TYPE BODY distrito_t AS
-- a)
MEMBER FUNCTION votos_total RETURN INTEGER IS vts INTEGER;
BEGIN
    vts := 0;
    SELECT SUM(value(v).votos) INTO vts
        FROM freguesias f, table(f.votacoes) v;
    return vts;
END votos_total;

-- b)
MEMBER FUNCTION votos_partido_vencedor RETURN INTEGER IS vts INTEGER;
BEGIN
    vts := 0;
    SELECT MAX(c.COLUMN_VALUE.votos_partido_vencedor()) INTO vts
        FROM DISTRITOS d, table(d.CONCELHOS) c
        WHERE d.CODIGO = self.CODIGO;
    return vts;
END votos_partido_vencedor;

-- c)
MEMBER FUNCTION partido_vencedor RETURN ref partido_t IS partido ref
partido_t;
BEGIN
    SELECT c.COLUMN_VALUE.partido_vencedor() INTO partido
        FROM DISTRITOS d, table(d.CONCELHOS) c
        WHERE d.CODIGO = self.CODIGO
        AND c.COLUMN_VALUE.votos_partido_vencedor() = (
            SELECT MAX(c.COLUMN_VALUE.votos_partido_vencedor())
            FROM DISTRITOS d, table(d.CONCELHOS) c
            WHERE d.CODIGO = self.CODIGO
        );
    return partido;
END partido_vencedor;

-- d)
```

```

    MEMBER FUNCTION participacao_correcta RETURN INTEGER IS distrito ref
    distrito_t;
    BEGIN
        distrito := null;
        SELECT ref(d) INTO distrito
            FROM DISTRITOS d
            WHERE d.CODIGO = self.CODIGO
                AND d.participacao.inscritos = d.PARTICIPACAO.votantes +
d.participacao.abstencoes + d.participacao.brancos + d.participacao.nulos +
d.votos_total();
        if(distrito is null) then return 1;
        else return 0;
        end if;
    END participacao_correcta;
END;

```

4. Partido

- a. Número de mandatos total
- b. Número de votos total

```

alter type partido_t add MEMBER FUNCTION votos_totais RETURN INTEGER cascade;

create or replace TYPE BODY partido_t AS
    -- a)
    MAP MEMBER FUNCTION mandatos_totais RETURN INTEGER IS mandatos INTEGER;
    BEGIN
        mandatos := 0;
        SELECT SUM(value(l).mandatos) INTO mandatos
            FROM partidos p, table(p.listas) l
            WHERE p.sigla = self.sigla;
        return mandatos;
    END mandatos_totais;

    -- b)
    MEMBER FUNCTION votos_totais RETURN INTEGER IS vts INTEGER;
    BEGIN
        vts := 0;
        SELECT SUM(value(v).votos) INTO vts
            FROM partidos p, table(p.votacoes) v
            WHERE p.sigla = self.sigla;
    END votos_totais;
END;

```

Questão 4

Execute some queries on the OR DB.

Alínea a)

Calculate the total number of Members of the Parliament that each party has get.

```
select p.sigla, p.mandatos_totais() as mandatos_totais
from partidos p
order by mandatos_totais desc;
```

	SIGLA	MANDATOS_TOTAIS
1	PS	112
2	PPDPSD	80
3	PCPPEV	17
4	CDSPP	15
5	BE	2
6	PCTPMRPP	0
7	PH	0
8	MPT	0
9	POUS	0
10	PPM	0
11	PSN	0
12	PDA	0

Fig 2: Resultado da execução da query da alínea a

Alínea b)

In each district, how many votes got each party?

```
create view votes_district_party as
select d.codigo, d.nome, sum(value(v).votos) as nvotos,
value(v).PARTIDO_REF.sigla as siglapartido
from distritos d, table(concelhos) c, table(value(c).freguesias) f,
table(value(f).votacoes) v
group by d.codigo, d.nome, value(v).PARTIDO_REF.sigla
order by d.codigo asc;

select * from votes_district_party;
```

Existem 182 resultados, pelo que a figura 3 é apenas um excerto da resposta.

	CODIGO	NOME	NVOTOS	SIGLAPARTIDO
1	1	Aveiro	4676	BE
2	1	Aveiro	49183	CDSPP
3	1	Aveiro	847	MPT
4	1	Aveiro	12797	PCPPEV
5	1	Aveiro	1511	PCTPMRPP
6	1	Aveiro	968	PH
7	1	Aveiro	138686	PPDPSD
8	1	Aveiro	1148	PPM
9	1	Aveiro	145575	PS
10	1	Aveiro	660	PSN
11	2	Beja	1316	BE
12	2	Beja	3315	CDSPP
13	2	Beja	279	MPT
14	2	Beja	24077	PCPPEV
15	2	Beja	1664	PCTPMRPP
16	2	Beja	12308	PPDPSD
17	2	Beja	393	PPM
18	2	Beja	39728	PS
19	2	Beja	207	PSN
20	3	Braga	5164	BE
21	3	Braga	39027	CDSPP
22	3	Braga	973	MPT
23	3	Braga	23821	PCPPEV
24	3	Braga	2947	PCTPMRPP
25	3	Braga	873	PH

Fig 3: Excerto do resultado da execução da query da alínea b

Alínea c)

Indicate the winning party on each municipality.

```
create view winning_party_municipality as
  select codigo, nome, max(nvotos) as votostotais, siglapartido
  from
    (select c.codigo, c.nome, sum(value(v).votos) as nvotos,
value(v).PARTIDO_REF.sigla as siglapartido
    from concelhos c, table(c.freguesias) f, table(value(f).votacoes) v
    group by c.codigo, c.nome, value(v).PARTIDO_REF.sigla
    order by c.nome DESC ) s
  group by codigo, nome, siglapartido
  order by codigo asc;
```

```
select * from winning_party_municipality;
```

Existem 2817 resultados, pelo que a figura 4 é apenas um excerto da resposta.

	CODIGO	NOME	VOTOSTOTAIS	SIGLAPARTIDO
1	101	Águeda	240	BE
2	101	Águeda	3242	CDSPP
3	101	Águeda	52	MPT
4	101	Águeda	850	PCPPEV
5	101	Águeda	104	PCTPMRPP
6	101	Águeda	55	PH
7	101	Águeda	9543	PPDPSD
8	101	Águeda	82	PPM
9	101	Águeda	10104	PS
10	101	Águeda	54	PSN
11	102	Albergaria-a-Velha	86	BE
12	102	Albergaria-a-Velha	2219	CDSPP
13	102	Albergaria-a-Velha	15	MPT
14	102	Albergaria-a-Velha	305	PCPPEV
15	102	Albergaria-a-Velha	38	PCTPMRPP
16	102	Albergaria-a-Velha	25	PH
17	102	Albergaria-a-Velha	4845	PPDPSD
18	102	Albergaria-a-Velha	35	PPM
19	102	Albergaria-a-Velha	4041	PS
20	102	Albergaria-a-Velha	35	PSN
21	103	Anadia	193	BE
22	103	Anadia	2270	CDSPP
23	103	Anadia	36	MPT

Fig 4: Excerto do resultado da execução da query da alínea c

Alínea d)

Check whether any district violates the following integrity rule: the sum of the votes in the several lists, the white and the null votes, plus the number of abstentions must equal the number of enrolled citizens.

```
select d.codigo, d.NOME
from distritos d
where d.participacao_correcta() = 0;
```

Como esperado, esta query não retorna nenhum resultado visto que todos os distritos têm as participações dentro da lei.

Alínea e)

Which are the differences between the percentages of mandates and of votes for each party at the national level?

```
create view mandatos_vs_partidos as
  select p.sigla , p.votos_totais() as votos_totais, p.mandatos_totais() as
mandatos_totais, ROUND(p.votos_totais()/votos,3)*100 AS percentagem_votos,
ROUND(p.mandatos_totais()/mandatos, 3)*100 AS percentagem_mandatos
  from
    partidos p,
    (select sum(v.votos) as votos
     from votacoes v),
    (select sum(l.mandatos) as mandatos
     from listas l)
  order by mandatos_totais desc;

select * from mandatos_vs_partidos;
```

	⚡ SIGLA	⚡ VOTOS_TOTAIS	⚡ MANDATOS_TOTAIS	⚡ PERCENTAGEM_VOTOS	⚡ PERCENTAGEM_MANDATOS
1	PS	2359939	112	44,9	49,6
2	PPDPSD	1733814	80	33	35,4
3	PCPPEV	483716	17	9,2	7,5
4	CDSPP	449310	15	8,5	6,6
5	BE	131840	2	2,5	0,9
6	PCTPMRPP	39467	0	0,8	0
7	PH	7346	0	0,1	0
8	MPT	19511	0	0,4	0
9	POUS	4056	0	0,1	0
10	PPM	16333	0	0,3	0
11	PSN	11358	0	0,2	0
12	PDA	437	0	0	0

Fig 5: Resultado da execução da query da alínea e

Alínea f)

Which parties got Members of the Parliament in every district?

```

select sigla, distritosdist, mandatosdist
from
  (select count(d.codigo) as distritosDist
   from distritos d) x,
  (select sigla, count(sigla) as mandatosDist
   from
     (select p.sigla, value(1).mandatos, value(1).DISTRITO_REF.nome
      from partidos p, table(p.listas) l
      where value(1).mandatos > 0) y
   group by sigla) z
where distritosdist = mandatosdist;

```

	SIGLA	DISTRITOSDIST	MANDATOSDIST
1	PS	20	20

Fig 6: Resultado da execução da query da alínea f

Alínea g)

Add a query that illustrates the use of OR extensions.

Selecionar os distritos cuja abstenção foi maior que 40%.

```

select d.codigo, d.NOME, ROUND((d.PARTICIPACAO.abstencoes /
d.PARTICIPACAO.inscritos),4)*100 as abstencao_per
from distritos d
where ROUND((d.PARTICIPACAO.abstencoes / d.PARTICIPACAO.inscritos),4)*100 > 40
order by abstencao_per desc;

```

	CODIGO	NOME	ABSTENCAO_PER
1	40	Açores	49,75
2	4	Bragança	45,34
3	17	Vila Real	43,24
4	8	Faro	42,51
5	30	Madeira	41,71
6	18	Viseu	41,49
7	2	Beja	41,33
8	9	Guarda	40,4

Fig 7: Resultado da execução da query da alínea g