

# Database Technology - Lab assignment 2

Mariana Monteiro  
up202003480

Vasco Gomes  
up201906617

May 26, 2023

## Contents

<b>1</b>	<b>Introduction</b>	<b>3</b>
<b>2</b>	<b>Object-Relational data model</b>	<b>3</b>
2.1	Model . . . . .	3
2.2	SQL . . . . .	4
<b>3</b>	<b>Populate</b>	<b>5</b>
<b>4</b>	<b>Methods</b>	<b>6</b>
4.1	Area . . . . .	6
4.1.1	partido_mais_votos() . . . . .	6
4.1.2	votos_partido(sigla) . . . . .	6
4.1.3	votos_total() . . . . .	6
4.1.4	SQL . . . . .	6
4.2	Distrito . . . . .	7
4.2.1	votos_partido(sigla) . . . . .	7
4.2.2	votos_total() . . . . .	7
4.2.3	integro() . . . . .	7
4.2.4	mandatos_partido(sigla) . . . . .	7
4.2.5	SQL . . . . .	7
4.3	Concelho . . . . .	8
4.3.1	votos_partido(sigla) . . . . .	8
4.3.2	votos_total() . . . . .	8
4.3.3	SQL . . . . .	8
4.4	Freguesia . . . . .	9
4.4.1	votos_partido(sigla) . . . . .	9
4.4.2	votos_total() . . . . .	9
4.4.3	SQL . . . . .	9
4.5	Partido . . . . .	9
4.5.1	mandatos_total() . . . . .	9
4.5.2	votos_total() . . . . .	9
4.5.3	SQL . . . . .	9
4.6	Lista . . . . .	10
4.6.1	compare() . . . . .	10
4.6.2	SQL . . . . .	10
<b>5</b>	<b>Queries</b>	<b>10</b>
5.1	A . . . . .	10
5.1.1	SQL . . . . .	10
5.1.2	Results . . . . .	11
5.2	B . . . . .	11
5.2.1	SQL . . . . .	11
5.2.2	Results . . . . .	12

5.3	C . . . . .	12
5.3.1	SQL . . . . .	12
5.3.2	Results . . . . .	13
5.4	D . . . . .	13
5.4.1	SQL . . . . .	13
5.4.2	Results . . . . .	13
5.5	E . . . . .	13
5.5.1	SQL . . . . .	13
5.5.2	Results . . . . .	14
5.6	F . . . . .	14
5.6.1	SQL . . . . .	14
5.6.2	Results . . . . .	14
5.7	G . . . . .	14
5.7.1	SQL . . . . .	14
5.7.2	Results G.1 . . . . .	15
5.7.3	SQL . . . . .	15
5.7.4	Results G.2 . . . . .	15
<b>6</b>	<b>Conclusion</b>	<b>15</b>

# 1 Introduction

Within the scope of the course Database Technologies, this assignment has the goal of implementing an object-relational data model, exploiting the SQL 3 extensions. Some of the advantages of using this type of schema are the use of user defined types, with objects combining data structures and the functions to manipulate them, inheritance, nested tables and vectors, object references and comparison and sorting methods.

## 2 Object-Relational data model

### 2.1 Model

Figure 1 represents the object-relational data model, obtained from the given relational model, to exploit SQL 3 extensions.

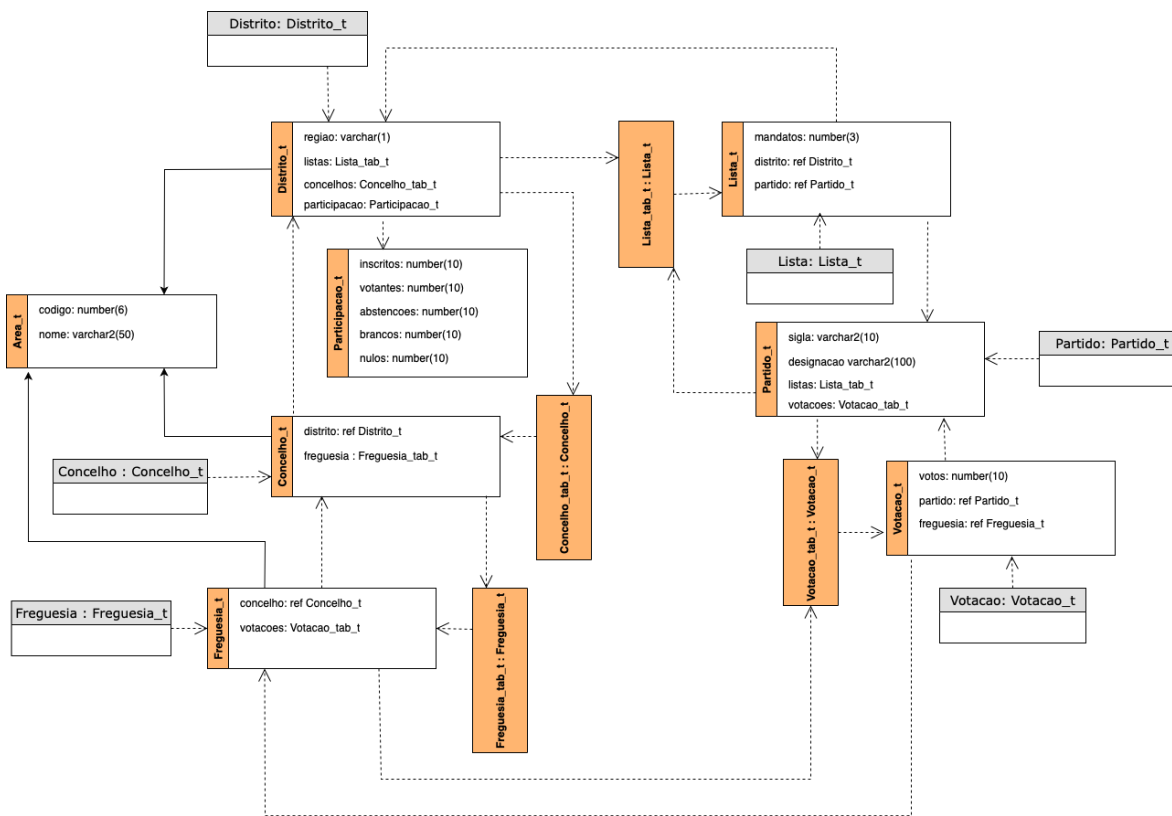


Figure 1: Object-Relational data model.

Given that the relationship between Participacoes and Distritos was one-to-one, we could simply create a type for Participacao and add it to the Distrito type as an attribute. We decided to do this since Participacao itself is only useful with Distrito and with this implementation, we do not need to create a table for Participacao.

The relationships between Distritos, Concelhos and Freguesias were all many-to-one. We decided to use symmetrical references (nested table of references in the one side and a reference in the many side) since there were a lot of relationships with other tables and it would be very difficult and complex to use the inclusion alternative.

The same reason holds for the relationships between Freguesias and Votacoes, Votacoes and Partidos, Partidos and Listas, and Listas and Distritos, given they all are many-to-one.

We also created a new type Area, that would represent administrative areas and had the attributes codigo and nome. Distrito, Concelho and Freguesia were all subtypes of Area.

## 2.2 SQL

```
create or replace type Participacao_t as object(
    inscritos number(10),
    votantes number(10),
    abstencoes number(10),
    brancos number (10),
    nulos number (10));
/

create or replace type Area_t as object(
    codigo number(6),
    nome varchar(250),
    not instantiable member function votos_partido (sigla varchar2) return number,
    not instantiable member function votos_total return number)not instantiable not final;
/

create or replace type Distrito_t under Area_t(
    regiao varchar(1),
    participacao Participacao_t,
    overriding member function votos_partido (sigla varchar2) return number,
    overriding member function votos_total return number);
/

create or replace type Concelho_t under Area_t(
    distrito ref Distrito_t,
    overriding member function votos_partido (sigla varchar2) return number,
    overriding member function votos_total return number);
/

create or replace type Concelho_tab_t as table of ref Concelho_t;
/

create or replace type Freguesia_t under Area_t(
    concelho ref Concelho_t,
    overriding member function votos_partido (sigla varchar2) return number,
    overriding member function votos_total return number);
/

create or replace type Freguesia_tab_t as table of ref Freguesia_t;
/

create or replace type Partido_t as object(
    sigla varchar2(10),
    designacao varchar2(100));
/

create or replace type Lista_t as object(
    mandatos number(3),
    distrito ref Distrito_t,
    partido ref Partido_t);
/

create or replace type Lista_tab_t as table of ref Lista_t;
/

create or replace type Votacao_t as object(
    votos number(10),
```

```

    partido ref Partido_t,
    freguesia ref Freguesia_t);
/

create or replace type Votacao_tab_t as table of ref Votacao_t;
/

alter type Distrito_t add attribute (listas Lista_tab_t, 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, votacoes Votacao_tab_t) cascade;
/

create table Distrito of Distrito_t
    nested table listas store as listas_distrito_tab
    nested table concelhos store as concelhos_tab;
/

create table Concelho of Concelho_t
    nested table freguesias store as freguesia_tab;
/

create table Freguesia of Freguesia_t
    nested table votacoes store as votacoes_freguesia_tab;
/

create table Partido of Partido_t
    nested table listas store as listas_partido_tab
    nested table votacoes store as votacoes_partido_tab;
/

create table Lista of Lista_t;
/

create table Votacao of Votacao_t;
/

```

### 3 Populate

```

insert into Distrito(codigo, nome, regiao, participacao)
    select d.codigo, d.nome, d.regiao, Participacao_t(p.inscritos, p.votantes,
        p.abstencoes, p.branco, p.nulos)
    from GTD11.distritos d join GTD11.participacoes p on d.codigo = p.distrito;

insert into Partido(sigla, designacao)
    select p.sigla, p.designacao
    from GTD11.partidos p;

insert into Lista(distrito, partido, mandatos)
    select ref(d), ref(p), l.mandatos
    from Distrito d
    join GTD11.listas l on d.codigo = l.distrito

```

```

    join Partido p on p.sigla = l.partido;

insert into Concelho(nome, codigo, distrito)
select c.nome, c.codigo, ref(d)
from GTD11.concelhos c
join Distrito d on d.codigo = c.distrito;

insert into Freguesia(codigo, nome, concelho)
select f.codigo, f.nome, ref(c)
from GTD11.freguesias f
join Concelho c on c.codigo=f.concelho;

insert into Votacao(freguesia, partido, votos)
select ref(f), ref(p), v.votos
from Freguesia f
join GTD11.votacoes v on v.freguesia = f.codigo
join Partido p on v.partido = p.sigla;

update Distrito d
set d.listas = cast(multiset(select ref(l) from Lista l where l.distrito.codigo =
    d.codigo) as Lista_tab_t),
d.concelhos = cast(multiset(select ref(c) from Concelho c where c.distrito.codigo =
    d.codigo) as Concelho_tab_t);

update Partido p
set p.listas = cast(multiset(select ref(l) from Lista l where l.partido.sigla =
    p.sigla) as Lista_tab_t),
p.votacoes = cast(multiset(select ref(v) from Votacao v where v.partido.sigla =
    p.sigla) as Votacao_tab_t);

update Concelho c
set c.freguesias = cast(multiset(select ref(f) from Freguesia f where
    f.concelho.codigo = c.codigo) as Freguesia_tab_t);

update Freguesia f
set f.votacoes = cast(multiset(select ref(v) from Votacao v where v.freguesia.codigo =
    f.codigo) as Votacao_tab_t);

```

## 4 Methods

### 4.1 Area

#### 4.1.1 partido\_mais\_votos()

This method returns a reference of Partido\_t, the party with the most votes in the administrative area. This function uses the function votos\_partido() that will be overridden for each subtype.

#### 4.1.2 votos\_partido(sigla)

This method returns the number of votes of a given party in the administrative area. Since this method is not instantiable, it was declared in the *create.sql* file and will be overridden by its subtypes.

#### 4.1.3 votos\_total()

This method returns the total number of votes in the administrative area. Since this method is not instantiable, it was declared in the *create.sql* file and will be overridden by its subtypes.

#### 4.1.4 SQL

```

alter type Area_t add member function partido_mais_votos return ref Partido_t cascade;

create or replace type body Area_t as

    member function partido_mais_votos return ref Partido_t is
    r ref partido_t;
begin
    select ref(p1) into r from partido p1 where
        not exists (select * from partido p2
            where (p1.sigla != p2.sigla and self.votos_partido(p1.sigla) <
                self.votos_partido(p2.sigla)));
    return r;
end partido_mais_votos;

end;

```

## 4.2 Distrito

### 4.2.1 votos\_partido(sigla)

This method returns the number of votes of a given party in the district. Since this method is overriding, it was declared in the *create.sql* file.

### 4.2.2 votos\_total()

This method returns the total number of votes in the district. Since this method is overriding, it was declared in the *create.sql* file.

### 4.2.3 integro()

This method returns 1 if the district follows the 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 and 0 otherwise.

### 4.2.4 mandatos\_partido(sigla)

This method returns the number of mandates of a given party in the district.

### 4.2.5 SQL

```

alter type Distrito_t add member function integro return number cascade;
alter type Distrito_t add member function mandatos_partido(sigla varchar2) return number
    cascade;

create or replace type body Distrito_t as

    overriding member function votos_partido(sigla varchar2) return number is
    r number;
begin
    select nvl(sum(value(v).votos),0) into r
    from table(self.concelhos) c, table(value(c).freguesias) f,
        table(value(f).votacoes) v
    where value(v).partido.sigla = sigla;
    return r;
end votos_partido;

    overriding member function votos_total return number is
    r number;
begin

```

```

        select sum(value(v).votos) into r
        from table(self.concelhos) c, table(value(c).freguesias) f,
            table(value(f).votacoes) v;
        return r;
    end votos_total;

member function integro return number is
r number;
begin
    if (self.votos_total() + self.participacao.branco + self.participacao.nulos +
        self.participacao.abstencoes) = self.participacao.inscritos then
        r:=1;
    else
        r:=0;
    end if;
    return r;
end integro;

member function mandatos_partido(sigla varchar2) return number is
r number;
begin
    select nvl(value(l).mandatos,0) into r from table(self.listas) l where
        value(l).partido.sigla=sigla;
    return r;
end mandatos_partido;

end;

```

## 4.3 Concelho

### 4.3.1 votos\_partido(sigla)

This method returns the number of votes of a given party in the municipality. Since this method is overriding, it was declared in the *create.sql* file.

### 4.3.2 votos\_total()

This method returns the total number of votes in the municipality. Since this method is overriding, it was declared in the *create.sql* file.

### 4.3.3 SQL

```

create or replace type body Concelho_t as

    overriding member function votos_partido(sigla varchar2) return number is
    r number;
    begin
        select nvl(sum(value(v).votos),0) into r
        from table(self.freguesias) f, table(value(f).votacoes) v
        where value(v).partido.sigla = sigla;
        return r;
    end votos_partido;

    overriding member function votos_total return number is
    r number;
    begin
        select sum(value(v).votos) into r
        from table(self.freguesias) f, table(value(f).votacoes) v;
        return r;
    end votos_total;

```



```
end;
```

## 4.4 Freguesia

### 4.4.1 votos\_partido(sigla)

This method returns the number of votes of a given party in the parish. Since this method is overriding, it was declared in the *create.sql* file.

### 4.4.2 votos\_total()

This method returns the total number of votes in the parish. Since this method is overriding, it was declared in the *create.sql* file.

### 4.4.3 SQL

```
create or replace type body Freguesia_t as

    overriding member function votos_partido(sigla varchar2) return number is
    r number;
    begin
        select nvl(sum(value(v).votos),0) into r
        from table(self.votacoes) v
        where value(v).partido.sigla = sigla;
        return r;
    end votos_partido;

    overriding member function votos_total return number is
    r number;
    begin
        select sum(value(v).votos) into r
        from table(self.votacoes) v;
        return r;
    end votos_total;

end;
```

## 4.5 Partido

### 4.5.1 mandatos\_total()

This method returns the total number of mandates of the party.

### 4.5.2 votos\_total()

This method returns the total number of votes of the party.

### 4.5.3 SQL

```
alter type Partido_t add member function mandatos_total return number cascade;
alter type Partido_t add member function votos_total return number cascade;

create or replace type body Partido_t as

    member function mandatos_total return number is
    r number;
    begin
        select nvl(sum(value(l).mandatos),0) into r from table(self.listas) l;
```

```

        return r;
    end mandatos_total;

    member function votos_total return number is
    r number;
    begin
        select nvl(sum(value(v).votos),0) into r from table(self.votacoes) v;
        return r;
    end votos_total;
end;

```

## 4.6 Lista

### 4.6.1 compare()

This method is used to order a Lista, based on the number of mandates.

### 4.6.2 SQL

```

alter type Lista_t add order member function compare (l Lista_t) return number cascade;

create or replace type body Lista_t as

    order member function compare (l Lista_t) return number is
    r number;
    begin
        if mandatos < l.mandatos then return -1;
        elsif mandatos > l.mandatos then return 1;
        else return 0;
        end if;
    end compare;
end;

```

## 5 Queries

### 5.1 A

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

#### 5.1.1 SQL

```

select p.sigla, p.designacao, p.mandatos_total() as "Mandatos"
from Partido p order by value(p) DESC;

```

### 5.1.2 Results

	⚡ SIGLA	⚡ DESIGNACAO	⚡ MANDATOS
1	PS	Partido Socialista	112
2	PPDPSD	Partido Social Democrata	80
3	PCPPEV	Partido Comunista Português	17
4	CDSPP	Partido Popular	15
5	BE	Bloco de Esquerda	2
6	PCTPMRPP	Partido Comunista dos Trabalhadores Portugueses	0
7	PH	Partido Humanista	0
8	MPT	Movimento Partido da Terra	0
9	POUS	Partido Operário de Unidade Socialista	0
10	PPM	Partido Popular Monárquico	0
11	PSN	Partido Solidariedade Nacional	0
12	PDA	Partido Democrático do Atlântico	0

Figure 2: Results of Query A.

## 5.2 B

**Query:** In each district, how many votes got each party?

### 5.2.1 SQL

```
select d.nome, p.sigla, d.votos_partido(p.sigla) as votos
from Distrito d, Partido p
order by p.sigla,d.nome;
```

### 5.2.2 Results

	⚡ NOME	⚡ SIGLA	⚡ VOTOS
1	Açores	BE	992
2	Aveiro	BE	4676
3	Beja	BE	1316
4	Braga	BE	5164
5	Bragança	BE	679
6	Castelo Branco	BE	1520
7	Coimbra	BE	4545
8	Évora	BE	1394
9	Faro	BE	4106
10	Guarda	BE	1088
11	Leiria	BE	3843
12	Lisboa	BE	55113
13	Madeira	BE	1480
14	Portalegre	BE	876
15	Porto	BE	21384
16	Santarém	BE	4700

Figure 3: Part of the results of Query B (240 total values).

## 5.3 C

**Query:** Indicate the winning party on each municipality.

### 5.3.1 SQL

```
|| select c.nome, c.partido_mais_votos().sigla as "Partido com mais votos"  
|| from Concelho c order by c.nome;
```

### 5.3.2 Results

	NOME	Partido com mais votos
1	Abrantes	PS
2	Águeda	PS
3	Aguiar da Beira	PPDPSD
4	Alandroal	PS
5	Albergaria-a-Velha	PPDPSD
6	Albufeira	PS
7	Alcácer do Sal	PS
8	Alcanena	PS
9	Alcobaça	PPDPSD
10	Alcochete	PS
11	Alcoutim	PS
12	Alenquer	PS
13	Alfandega da Fé	PPDPSD
14	Alijó	PS
15	Aljezur	PS
16	Aljustrel	PS

Figure 4: Part of the results of Query C (308 total values).

## 5.4 D

**Query:** 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.

### 5.4.1 SQL

```
select d.nome
from Distrito d
where d.integro() = 0;
```

### 5.4.2 Results

	NOME
1	Açores

Figure 5: Results of query D.

## 5.5 E

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

### 5.5.1 SQL

```
select sigla,
round(((p.mandatos_total()/mandatos_nacionais)-(p.votos_total()/votos_nacionais))*100,2)
as "Diferenca entre mandatos e votos (%)"
```

```

from Partido p, (select sum(l.mandatos) mandatos_nacionais from Lista l), (select
sum(d.participacao.votantes) votos_nacionais from Distrito d) order by "Diferenca
entre mandatos e votos (%)" DESC;

```

### 5.5.2 Results

SIGLA	Diferenca entre mandatos e votos (%)
1 PS	5,56
2 PPDPSP	3,07
3 PDA	-0,01
4 POUS	-0,08
5 PH	-0,14
6 PSN	-0,21
7 PPM	-0,3
8 MPT	-0,36
9 PCTPMRPP	-0,74
10 PCPPEV	-1,5
11 BE	-1,57
12 CDSPP	-1,74

Figure 6: Results of query E.

## 5.6 F

**Query:** Which parties got Members of the Parliament in every district?

### 5.6.1 SQL

```

select p.sigla
from Partido p
where p.sigla
not in (
select p.sigla
from Partido p, Distrito d
where d.mandatos_partido(p.sigla)=0
);

```

### 5.6.2 Results

SIGLA
1 PS

Figure 7: Results of query F.

## 5.7 G

**Query 1:** What were the top 10 lists with more mandates?

### 5.7.1 SQL

```

select l.distrito.nome, l.partido.sigla, mandatos
from Lista l
order by value(1) DESC
fetch first 10 rows only;

```

This query shows the usage of a sorting method in a user defined type, Lista.t.

### 5.7.2 Results G.1

⚡	DISTRITO.NOME	⚡	PARTIDO.SIGLA	⚡	MANDATOS
1	Lisboa		PS		23
2	Porto		PS		19
3	Lisboa		PPDPSD		14
4	Porto		PPDPSD		13
5	Braga		PS		8
6	Setúbal		PS		8
7	Aveiro		PS		7
8	Braga		PPDPSD		7
9	Coimbra		PS		6
10	Aveiro		PPDPSD		6

Figure 8: Results of query G.1.

**Query 2:** Indicate the winning party on each district.

### 5.7.3 SQL

```
select d.nome, d.partido_mais_votos().sigla as "Partido com mais votos"
from Distrito d order by d.nome;
```

This query shows the ease of reusing code with inheritance. The query is very similar to c) and uses pretty much the same code since both Distrito.t and Concelho.t are subtypes of Area.t and both can use the function partido\_mais\_votos() and override the function votos\_partido(sigla).

### 5.7.4 Results G.2

⚡	NOME	⚡	Partido com mais votos
1	Açores		PS
2	Aveiro		PS
3	Beja		PS
4	Braga		PS
5	Bragança		PPDPSD
6	Castelo Branco		PS
7	Coimbra		PS
8	Évora		PS
9	Faro		PS
10	Guarda		PS
11	Leiria		PPDPSD
12	Lisboa		PS
13	Madeira		PPDPSD
14	Portalegre		PS
15	Porto		PS
16	Santarém		PS
17	Setúbal		PS
18	Viana do Castelo		PS
19	Vila Real		PPDPSD
20	Viseu		PPDPSD

Figure 9: Results of Query G.2

## 6 Conclusion

In conclusion, this project, which utilizes SQL3 extensions has showcased the immense capabilities and benefits of leveraging an object-relational database system. By incorporating object-oriented concepts into the traditional relational model, Oracle SQL with SQL3 extensions provides a robust platform for managing complex and diverse data.

Throughout this project, we have deepened the various features and advantages offered by the SQL3 extensions. We have explored the flexibility and extensibility provided by user-defined types, enabling the modelling of complex data structures and facilitating more intuitive data manipulation.

Furthermore, the project has demonstrated the power of object-relational database features such as inheritance, nested tables, and methods. By utilizing these capabilities, we have achieved improved data organization, enhanced code reuse, and increased productivity in database development.