



**FEUP** FACULDADE DE ENGENHARIA  
UNIVERSIDADE DO PORTO

# **Modelo Objeto-Relacional**

## **Trabalho Prático nº 2**

Tecnologia de Bases de Dados  
MIEIC - 2017/2018

Elementos do Grupo:

Andreia Rodrigues - up201404691 - [up201404691@fe.up.pt](mailto:up201404691@fe.up.pt)

Eduardo Leite - gei12068 - [gei12068@fe.up.pt](mailto:gei12068@fe.up.pt)

# Index

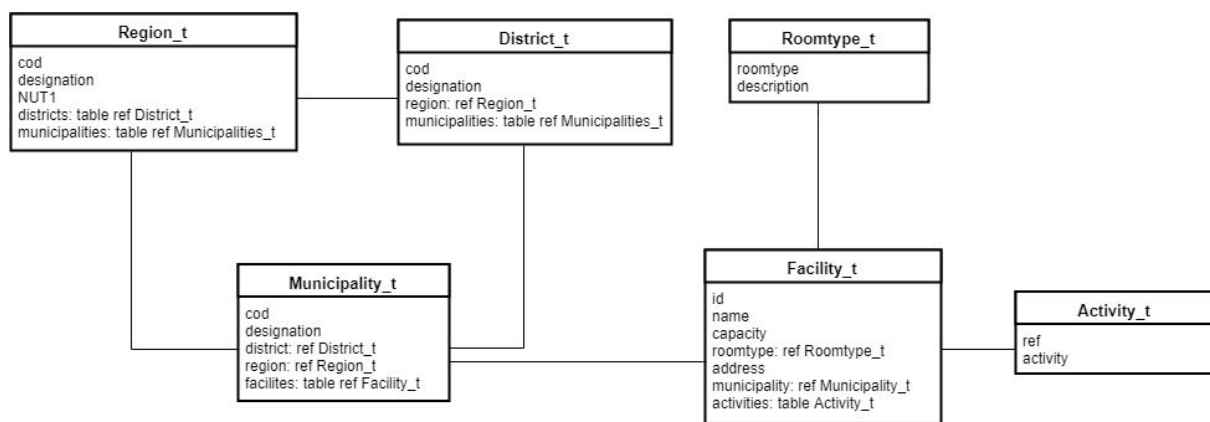
<b>1. Objetivo do Trabalho</b>	<b>2</b>
<b>2. Modelo de Dados Objeto-Relacional</b>	<b>2</b>
2.1 Modelo UML da Base de Dados	2
2.2 Código SQL de Criação da Base de Dados	3
2.3 Código SQL de Povoação do Modelo de Dados	4
2.4 Métodos Comuns Inseridos	4
<b>3. Queries à Base de Dados</b>	<b>5</b>
3.1 Query 1	5
3.1.1 Pergunta	5
3.1.2 Resposta	5
3.1.3 Resultados	5
3.2 Query 2	6
3.2.1 Pergunta	6
3.2.2 Resposta	6
3.2.3 Resultados	6
3.3 Query 3	7
3.3.1 Pergunta	7
3.3.2 Resposta	7
3.3.3 Resultados	7
3.4 Query 4	7
3.4.1 Pergunta	7
3.4.2 Resposta	8
3.4.3 Resultados	9
3.5 Query 5	9
3.5.1 Pergunta	9
3.5.2 Resposta	9
3.5.3 Resultados	10
3.6 Query 6	10
3.6.1 Pergunta	10
3.6.2 Resposta	10
3.6.3 Resultados	11
<b>4. Anexos</b>	<b>12</b>
4.1 Código SQL de Criação da Base de Dados	12
4.2 Código SQL de Povoação do Modelo de Dados	15
4.3 Métodos Comuns Inseridos	16

# 1. Objetivo do Trabalho

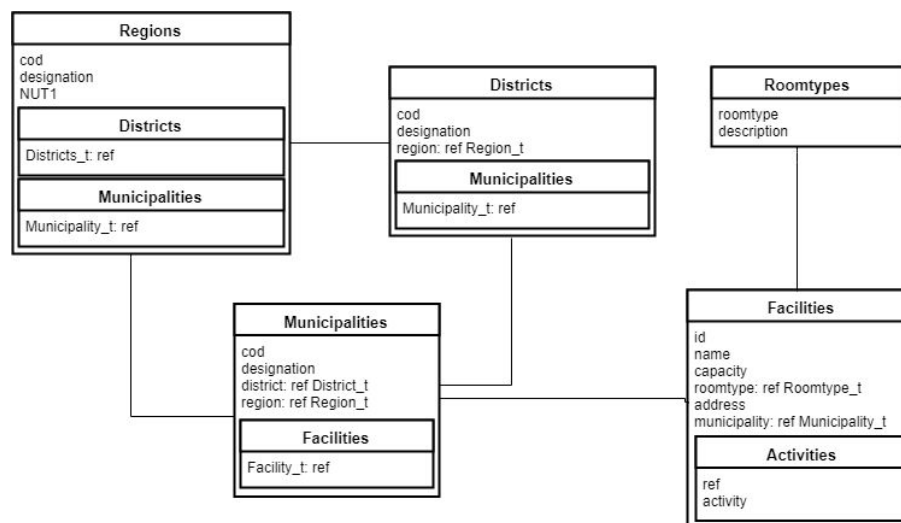
## 2. Modelo de Dados Objeto-Relacional

### 2.1 Modelo UML da Base de Dados

Modelo UML dos Tipos a Utilizar nas Tabelas da Base de Dados



Modelo UML das Tabelas da Base de Dados



## 2.2 Código SQL de Criação da Base de Dados

Refletindo o diagrama representado na secção acima, definimos os seguintes tipos para serem usados na criação das tabelas da base de dados:

- **Region\_t**: Representa uma região do país, contém duas tabelas de referências, uma para os seus distritos e outra para os seus municípios.
- **District\_t**: Representa um distrito do país, contém uma tabela de referências para os seus municípios e uma referência para a região onde se insere (pode ser nulo).
- **DistrictsRef\_tab\_t**: Representa uma tabela de referências para distritos para ser usada no tipo Region\_t.
- **Municipality\_t**: Representa um município, contém uma tabela de referências para as suas instalações, uma referência para o distrito a que pertence e outra para a região onde se insere.
- **MunicipalitiesRef\_tab\_t**: Representa uma tabela de referências para municípios para ser usada nos tipos Region\_t e District\_t.
- **RoomType\_t**: Representa um tipo de quarto.
- **Facility\_t**: Representa uma instalação, contém uma tabela de actividades, uma referência para o município onde se insere e uma referência para o seu tipo de quarto.
- **FacilitiesRef\_tab\_t**: Representa uma tabela de referências para instalações para ser usada no tipo Municipality\_t.
- **Activity\_t**: Representa uma actividade.
- **Activities\_tab\_t**: Representa uma tabela de actividades para ser usada no tipo Facility\_t.

Foram criadas as seguintes tabelas:

- **Regions**
- **Districts**
- **Municipalities**
- **RoomTypes**
- **Facilities**

O código SQL de criação da base de dados e respetivos tipos a utilizar está em anexo (secção 4.1).

## 2.3 Código SQL de Povoação do Modelo de Dados

O código SQL de povoação do modelo de dados com a informação presente na base de dados relacional está em anexo (secção 4.2).

## 2.4 Métodos Comuns Inseridos

Os métodos criados com a finalidade de abstrair alguns acessos/operações que considerámos úteis para utilizar em perguntas comuns à base de dados foram os seguintes:

- **GetCapacity** no tipo `Municipality_t`: retorna a capacidade total de um município, ou seja, a soma total das capacidades das instalações pertencentes a esse município.
- **GetFacilityCount** no tipo `Municipality_t`: retorna o número total de instalações pertencentes ao município.
- **GetMaxCARatio** no tipo `Municipality_t`: rácio máximo entre a capacidade de uma instalação e o número de atividades diferentes que lá se podem realizar das instalações presentes no município.
- **DescContains(text string)** no tipo `RoomType_t`: retorna 'true' caso a descrição contenha a string especificada, caso contrário retorna 'false'.
- **CARatio** no tipo `Facility_t`: rácio entre a capacidade de uma instalação e o número de atividades diferentes que lá se podem realizar.

O código SQL de criação dos métodos na base de dados está em anexo (secção 4.3).

## 3. Queries à Base de Dados

### 3.1 Query 1

#### 3.1.1 Pergunta

*Which are the facilities where the room type description contains 'touros' and have 'teatro' as one of their activities? Show the id, name, description and activity.*

#### 3.1.2 Resposta

##### Solução 1

```
select f.id, f.name, f.roomtype.description, a.activity
from facilities f, table(f.activities) a
where
a.activity = 'teatro' and
f.roomtype.DescContains('touros') = 'true';
```

##### Solução 2 - Mostrando todas as atividades

```
select f.id, f.name, f.roomtype.description, a.activity
from facilities f, table(f.activities) a
where f.id
in
(
    select f.id
    from facilities f, table(f.activities) a
    where
        a.activity = 'teatro' and
        f.roomtype.DescContains('touros') = 'true'
);
```

#### 3.1.3 Resultados

##### Solução 1

	ID	NAME	ROOMTYPE.DESCRPTION	ACTIVITY
1	940	ARENA DE ÉVORA - EX PRAÇA DE TOIROS	Praça de touros multiusos	teatro
2	957	COLISEU DE REDONDO - EX PRAÇA DE TOIROS	Praça de touros multiusos	teatro
3	916	COLISEU JOSÉ RONDÃO DE ALMEIDA-EX PRAÇA DE TOIROS	Praça de touros multiusos	teatro

## Solução 2

	ID	NAME	ROOMTYPE.DESCRPTION	ACTIVITY
1	940	ARENA DE ÉVORA - EX PRAÇA DE TOIROS	Praça de touros multiusos	dança
2	940	ARENA DE ÉVORA - EX PRAÇA DE TOIROS	Praça de touros multiusos	música
3	940	ARENA DE ÉVORA - EX PRAÇA DE TOIROS	Praça de touros multiusos	tauromaquia
4	940	ARENA DE ÉVORA - EX PRAÇA DE TOIROS	Praça de touros multiusos	teatro
5	957	COLISEU DE REDONDO - EX PRAÇA DE TOIROS	Praça de touros multiusos	dança
6	957	COLISEU DE REDONDO - EX PRAÇA DE TOIROS	Praça de touros multiusos	música
7	957	COLISEU DE REDONDO - EX PRAÇA DE TOIROS	Praça de touros multiusos	tauromaquia
8	957	COLISEU DE REDONDO - EX PRAÇA DE TOIROS	Praça de touros multiusos	teatro
9	916	COLISEU JOSÉ RONDÃO DE ALMEIDA-EX PRAÇA DE TOIROS	Praça de touros multiusos	dança
10	916	COLISEU JOSÉ RONDÃO DE ALMEIDA-EX PRAÇA DE TOIROS	Praça de touros multiusos	música
11	916	COLISEU JOSÉ RONDÃO DE ALMEIDA-EX PRAÇA DE TOIROS	Praça de touros multiusos	tauromaquia
12	916	COLISEU JOSÉ RONDÃO DE ALMEIDA-EX PRAÇA DE TOIROS	Praça de touros multiusos	teatro

## 3.2 Query 2

### 3.2.1 Pergunta

*How many facilities with 'touros' in the room type description are there in each region?*

### 3.2.2 Resposta

```
select r.cod, r.designation, r.NUT1, Coalesce(aux.countVal, 0)
from regions r
left join
(
select r.cod, r.designation, r.NUT1, count(value(f).ID) countVal
from regions r, table(r.municipalities) m, table(value(m).facilities) f
where
value(f).roomtype.DescContains('touros') = 'true'
group by r.cod, r.designation, r.NUT1
) aux on r.cod = aux.cod
order by r.cod asc;
```

### 3.2.3 Resultados

	COD	DESIGNATION	NUT1	COALESCE(AUX.COUNTVAL,0)
1	1	Norte	Continente	3
2	2	Centro	Continente	11
3	3	Lisboa	Continente	6
4	4	Alentejo	Continente	43
5	5	Algarve	Continente	1
6	6	Açores	Açores	0
7	7	Madeira	Madeira	0

### 3.3 Query 3

#### 3.3.1 Pergunta

*How many municipalities do not have any facility with an activity of 'cinema'?*

#### 3.3.2 Resposta

```
select count(*)
from municipalities m
where cod not in
(
select distinct m.cod
from municipalities m, table(m.facilities) f, table(value(f).activities) a
where a.activity = 'cinema'
);
```

#### 3.3.3 Resultados

	COUNT(*)
1	100

### 3.4 Query 4

#### 3.4.1 Pergunta

*Which is the municipality with more facilities engaged in each of the six kinds of activities?  
Show the activity, the municipality name and the corresponding number of facilities.*



### 3.4.2 Resposta

Problema "Greatest n per group"

#### **Solução 1** - *Join with Group-Identifier Max-value-in-Group Subquery*

```
select t1.*
from
(
    select a.ACTIVITY, m.COD, m.DESIGNATION, count(value(f).ID) CountValue
    from municipalities m, table(m.facilities) f, table(value(f).activities) a
    group by a.ACTIVITY, m.COD, m.DESIGNATION
) t1
inner join
(
    select ACTIVITY, max(CountValue) MaxCountValue
    from
    (
        select a.ACTIVITY, m.COD, count(value(f).ID) CountValue
        from municipalities m, table(m.facilities) f,
        table(value(f).activities) a
        group by a.ACTIVITY, m.COD
    )
    group by ACTIVITY
) t2 on t1.ACTIVITY = t2.ACTIVITY and t1.CountValue = t2.MaxCountValue
order by t1.ACTIVITY;
```

#### **Solução 2** - *Left Join Self using join conditions*

```
select t1.*
from
(
    select a.ACTIVITY, m.COD, m.DESIGNATION, count(value(f).ID) CountValue
    from municipalities m, table(m.facilities) f, table(value(f).activities) a
    group by a.ACTIVITY, m.COD, m.DESIGNATION
) t1
left outer join
(
    select a.ACTIVITY, m.COD, m.DESIGNATION, count(value(f).ID) CountValue
    from municipalities m, table(m.facilities) f, table(value(f).activities) a
    group by a.ACTIVITY, m.COD, m.DESIGNATION
) t2 ON (t1.ACTIVITY = t2.ACTIVITY AND t1.CountValue < t2.CountValue)
WHERE t2.ACTIVITY IS NULL
```

```
ORDER BY t1.ACTIVITY;
```

### 3.4.3 Resultados

	ACTIVITY	COD	DESIGNATION	COUNTVALUE
1	cinema	1106	Lisboa	96
2	circo	1106	Lisboa	2
3	dança	1106	Lisboa	47
4	música	1106	Lisboa	77
5	tauromaquia	210	Moura	4
6	teatro	1106	Lisboa	66

## 3.5 Query 5

### 3.5.1 Pergunta

*Which are the codes and designations of the districts with facilities in all the municipalities?*

### 3.5.2 Resposta

#### Solução 1 - Dupla negação, sem usar métodos

```
select d.COD, d.DESIGNATION
from districts d
where d.COD not in
(
  select distinct d.COD
  from districts d, table(d.municipalities) m
  where (d.DESIGNATION, value(m).DESIGNATION) not in
  (
    select d.DESIGNATION, value(m).DESIGNATION
    from districts d, table(d.municipalities) m,
    table(value(m).facilities) f
  )
)
order by COD;
```

#### Solução 2 - Usando métodos (execução mais rápida)

```
select d.COD, d.DESIGNATION
```

```

from districts d
where d.COD not in
(
    select distinct d.COD
    from districts d, table(d.municipalities) m
    where value(m).GetFacilityCount() = 0
)
order by COD;

```

### 3.5.3 Resultados

	COD	DESIGNATION
1	7	Évora
2	11	Lisboa
3	12	Portalegre
4	15	Setúbal

## 3.6 Query 6

### 3.6.1 Pergunta

*Add a query that illustrates the use of OR extensions*

### 3.6.2 Resposta

**Solução 1** - *What is the total capacity of each Municipality? (sum of the capacity of all facilities in that municipality).*

*Show the municipality code, name and total capacity.*

```

select m.COD, m.DESIGNATION, Coalesce(m.GetCapacity(), 0) TotalCapacity
from municipalities m
order by TotalCapacity desc;

```

**Solução 2** - *In each municipality, what is the facility with the best capacity/activity ratio, 2 decimals.*

```

select m.COD, value(f).ID, round(value(f).CARatio(), 2) MaxCARatio
from municipalities m, table(m.facilities) f
where value(f).CARatio() = m.GetMaxCARatio()
order by m.COD;

```

### 3.6.3 Resultados

#### Solução 1

	◇ COD	◇ DESIGNATION	◇ TOTALCAPACITY
1	1106	Lisboa	77339
2	1312	Porto	16374
3	1416	Santarém	13993
4	1207	Elvas	11402
5	1304	Gondomar	11086
6	811	Portimão	11013
7	308	Guimarães	10469
8	1512	Setúbal	9691
9	314	Vizela	8339
10	1507	Montijo	8232
11	705	Évora	8045
12	605	Figueira da Foz	7914
13	210	Moura	7414
14	603	Coimbra	7230
15	1105	Cascais	7219
16	1308	Matosinhos	7206
17	1317	Vila Nova de Gaia	7088
18	1409	Coruche	6448
19	1313	Póvoa de Varzim	6197
20	1403	Almeirim	5758
21	1214	Portalegre	5534
22	1506	Moita	5353

(...)

#### Solução 2

	COD	VALUE(F).ID	MAXCARATIO
1	101	280	171,75
2	102	281	251,5
3	103	285	69,25
4	104	227	200
5	105	303	305
6	106	211	50
7	107	72	250
8	108	306	127
9	109	229	469,33
10	110	308	166
11	111	312	122
12	113	230	230,33
13	114	313	120
14	115	315	1800
15	116	233	473,5
16	117	317	80,75
17	118	318	58
18	201	972	2000
19	203	974	50
20	204	975	60,75
21	205	978	3822
22	206	980	84,5

(...)

## 4. Anexos

### 4.1 Código SQL de Criação da Base de Dados

```

-----

----- Type Declaration -----

-----

-- Region_t
create or replace type Region_t as object
(
    COD          NUMBER(4,0),
    DESIGNATION   VARCHAR2(50 BYTE),
    NUT1          VARCHAR2(50 BYTE)
);

-- District_t
create or replace type District_t as object

```

```

(
    COD          NUMBER(4,0),
    DESIGNATION   VARCHAR2(50 BYTE),
    REGION ref Region_t
);

-- DistrictsRef_tab_t
create or replace type DistrictsRef_tab_t as table of ref District_t;

-- Municipality
create or replace type Municipality_t as object
(
    COD          NUMBER(4,0),
    DESIGNATION   VARCHAR2(50 BYTE),
    DISTRICT ref District_t,
    REGION ref Region_t
);

-- MunicipalitiesRef_tab_t
create or replace type MunicipalitiesRef_tab_t as table of ref
Municipality_t;

-- RoomType
create or replace type RoomType_t as object
(
    ROOMTYPE NUMBER(4,0),
    DESCRIPTION VARCHAR2(50 BYTE)
);

-- Facility
create or replace type Facility_t as object
(
    ID number(4,0),
    NAME varchar2(80),
    CAPACITY number(8,0),
    ROOMTYPE ref RoomType_t,
    ADDRESS varchar2(80),
    MUNICIPALITY ref Municipality_t
);

-- FacilitiesRef_tab_t
create or replace type FacilitiesRef_tab_t as table of ref Facility_t;

-- Activity
create or replace type Activity_t as object
(
    REF varchar2(20),
    ACTIVITY varchar2(20)
);

-- Activities_tab_t

```

```

create or replace type Activities_tab_t as table of Activity_t;

-- Region Districts
alter type Region_t
add attribute (DISTRICTS DistrictsRef_tab_t) CASCADE;

-- Region Municipalities
alter type Region_t
add attribute (MUNICIPALITIES MunicipalitiesRef_tab_t) CASCADE;

-- District Municipalities
alter type District_t
add attribute (MUNICIPALITIES MunicipalitiesRef_tab_t) CASCADE;

-- Municipality Facilities
alter type Municipality_t
add attribute (FACILITIES FacilitiesRef_tab_t) CASCADE;

-- Facility Activities
alter type Facility_t
add attribute (ACTIVITIES Activities_tab_t) CASCADE;

-----

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

-----

-- Regions
create table Regions of Region_t
    nested table Districts store as Region_Districts,
    nested table Municipalities store as Region_Municipalities;

-- Districts
create table Districts of District_t
    nested table Municipalities store as District_Municipalities;

-- Municipalities
create table Municipalities of Municipality_t
    nested table Facilities store as Municipality_Facilities;

-- RoomTypes
create table RoomTypes of RoomType_t;

-- Facilities
create table Facilities of Facility_t
    nested table Activities store as Facility_Activities;

```

## 4.2 Código SQL de Povoação do Modelo de Dados

```
-----  
  
----- Data Migration -----  
  
-----  
  
-- Regions  
delete from Regions;  
insert into Regions (COD, DESIGNATION, NUT1)  
select r.COD, r.DESIGNATION, r.NUT1  
from gtd8.Regions r;  
  
-- Districts  
delete from Districts;  
insert into Districts (COD, DESIGNATION, REGION)  
select d.COD, d.DESIGNATION, (select ref(r) from Regions r where r.COD =  
d.REGION)  
from gtd8.Districts d;  
  
-- Municipalities  
delete from Municipalities;  
insert into Municipalities (COD, DESIGNATION, DISTRICT, REGION)  
select m.COD, m.DESIGNATION, (select ref(d) from Districts d where d.COD =  
m.DISTRICT), (select ref(r) from Regions r where r.COD = m.REGION)  
from gtd8.Municipalities m;  
  
-- RoomTypes  
delete from RoomTypes;  
insert into RoomTypes (ROOMTYPE, DESCRIPTION)  
select rt.ROOMTYPE, rt.DESCRPTION  
from gtd8.RoomTypes rt;  
  
-- Facilities  
delete from Facilities;  
insert into Facilities (ID, NAME, CAPACITY, ROOMTYPE, ADDRESS, MUNICIPALITY)  
select f.ID, f.NAME, f.CAPACITY, (select ref(rt) from RoomTypes rt where  
rt.ROOMTYPE = f.ROOMTYPE), f.ADDRESS, (select ref(m) from Municipalities m  
where m.COD = f.MUNICIPALITY)  
from gtd8.Facilities f;  
  
-----  
  
----- Nested Migration -----
```



```

-----

-- Facilities Activities
update Facilities f
set f.Activities =
cast(multiset(select a.ref, a.activity from gtd8.Activities a join gtd8.Uses
u on a.ref = u.ref where u.id = f.id) as Activities_tab_t);

-- Municipalities Facilities
update Municipalities m
set m.Facilities =
cast(multiset(select ref(f) from Facilities f where m.cod =
f.municipality.cod) as FacilitiesRef_tab_t);

-- Districts Municipalities
update Districts d
set d.Municipalities =
cast(multiset(select ref(m) from Municipalities m where d.cod =
m.district.cod) as MunicipalitiesRef_tab_t);

-- Regions Districts
update Regions r
set r.Districts =
cast(multiset(select ref(d) from Districts d where r.cod = d.region.cod) as
DistrictsRef_tab_t);

-- Regions Municipalities
update Regions r
set r.Municipalities =
cast(multiset(select ref(m) from Municipalities m where r.cod = m.region.cod)
as MunicipalitiesRef_tab_t);

```

## 4.3 Métodos Comuns Inseridos

```

-----

----- Function Declaration -----

-----

alter type Municipality_t
add member function GetCapacity return integer cascade;

alter type Municipality_t
add member function GetFacilityCount return integer cascade;

alter type Municipality_t
add member function GetMaxCARatio return float cascade;

```

```
alter type RoomType_t
add member function DescContains(text string) return string cascade;
```

```
alter type Facility_t
add member function CARatio return float cascade;
```

```
-----
----- Function Definition -----
-----
```

```
create or replace type body Municipality_t as
```

```
    member function GetCapacity return integer is
```

```
        totalCapacity integer;
```

```
    begin
```

```
        select sum(f.capacity) into totalCapacity
```

```
        from facilities f
```

```
        where f.municipality.cod = self.cod;
```

```
    return totalCapacity;
```

```
    end GetCapacity;
```

```
---
```

```
    member function GetFacilityCount return integer is
```

```
        facilityCount integer;
```

```
    begin
```

```
        select count(*) into facilityCount
```

```
        from municipalities m, table(m.facilities) f
```

```
        where m.cod = self.cod;
```

```
    return facilityCount;
```

```
    end GetFacilityCount;
```

```
--
```

```
    member function GetMaxCARatio return float is
```

```
        bestRatio float;
```

```

begin

select max(value(f).CARatio()) into bestRatio
from municipalities m, table(m.facilities) f
where m.COD = self.COD
group by COD;

return bestRatio;

end GetMaxCARatio;

end;

create or replace type body RoomType_t as

member function DescContains(text string) return string is

expression varchar(50);

begin
    expression := '%' || text || '%';

    if description like expression then
        return 'true';
    else
        return 'false';
    end if;

end DescContains;

end;

create or replace type body Facility_t as

member function CARatio return float is

activityCount integer;

begin

select count(*) into activityCount
from facilities f, table(f.activities) a
where f.ID = self.ID;

if activityCount = 0 then
    return 0;
else
    return self.CAPACITY/activityCount;
end if;

end CARatio;

```

end;