Guillaume Minette de Saint Martin **October 2015**

Technology of Large Databases

Practical Work 1

Table of contents

Subject

1. practical work. Objec-relational DB storage structures

Creation of object – relational database data storage structures, data input, metadata output and data extraction (queries).

1. Table with row type objects.
2. Table with object column.
3. Table with object collection.
4. Object view.
5. Object reference 1 : 1 and 1 : M (many).
6. Use of special object functions in SQL queries.
7. Conclusions.

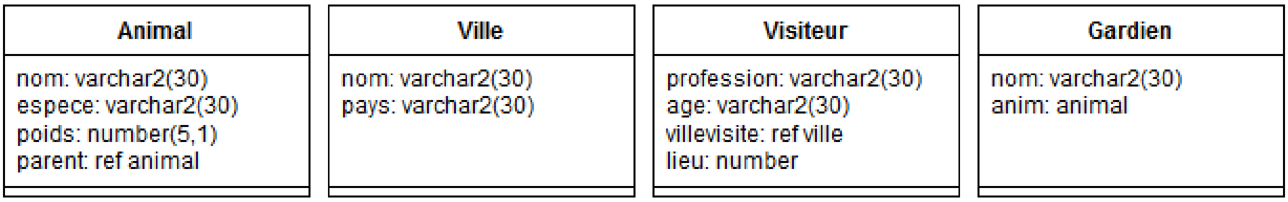
Dictionary

French	English
animal	animal
nom	name
poids	weight
parent	parent
pays	country
profession	job
age	age
villevisite	city visited
ville	city
lieu	place
equipe	team
environement	environment
milieu	original place to live
temperature	temperature
habitant	inhabitant

French Zoos

The idea is to create a database on the model of a zoo with animals, environments, guardians and visitors.

Therefore I created different object types related to theses data.



I also created an other type: **equipe**, it is a table of **gardien** that will be used by the object collection table.

- Animal:

```
create or replace type animal as object(
nom varchar2(30),
espece varchar2(30),
poids number(5,1),
parent ref animal);
```

- Ville:

```
create or replace type ville as object(
nom varchar2(30),
pays varchar2(30));
```

- Visiteur:

```
create or replace type visiteur as object(
profession varchar(30),
age number(3),
villevisite ref ville,
lieu number);
```

- Gardien:

```
create or replace type gardien as object(
nom varchar2(30),
anim animal);
```

- Equipe:

```
create or replace type equipe as table of gardien;
```

Created tables

Tables with row type object

Table visitors is the table that gather all the visitors of the zoos.

```
create table visiteurs of visiteur;
```

Table animaux is the table that gather all the animals.

```
create table animaux of animal;
```

Table parents is the table that gather all the animal's parents.

```
create table parents of animal;
```

Table villes is the table that gather all the cities.

```
create table villes of ville;
```

Table with object type column

Table environements is a table that contains animals and refers it to an environment, that is to say a place to live (ocean, desert, forest), a temperature and city to identify witch city's zoo is able to provide it.

```
create table environements(
id number(4) primary key,
ville ref ville,
milieu varchar(50),
temperature number(3),
habitant animal);
```

Object collection table

Table zoo is a collection that links a city (ville) with a guardian team (equipes_gardiens).

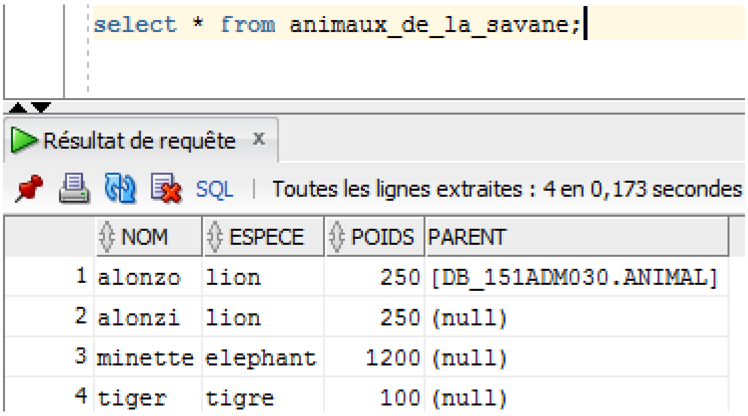
```
create table zoos(
ville ref ville,
equipes_gardiens equipe)
nested table equipes_gardiens store as pets;
```

Created Views

Object view

View **animaux_de_la_savane** (savannah animals) displays all the animals from table animaux (animals) that usually live in savannah (information found in table environements).

```
create or replace view animaux_de_la_savane of animal with object id(nom)as
select a.*
from animaux a, environements b
where a.nom=b.habitant.nom and
b.milieu='savane'
;
```



```
select * from animaux_de_la_savane;
```

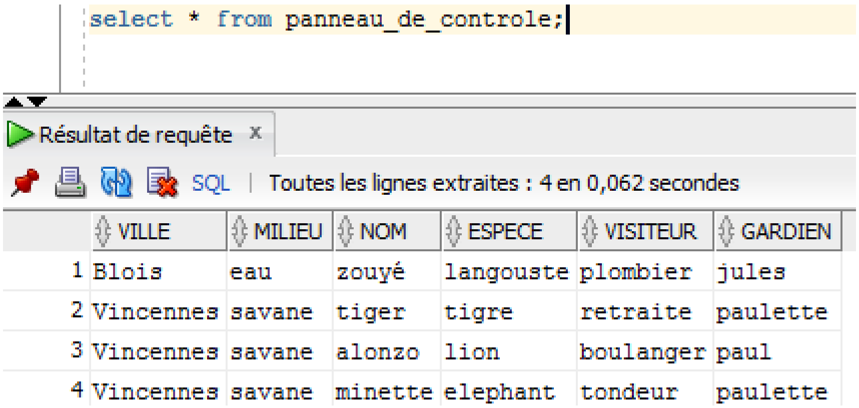
Toutes les lignes extraites : 4 en 0,173 secondes

	NOM	ESPECE	POIDS	PARENT
1	alonzo	lion	250	[DB_151ADM030.ANIMAL]
2	alonzi	lion	250	(null)
3	minette	elephant	1200	(null)
4	tiger	tigre	100	(null)

Traditional view

View **panneau_de_controle** (control panel) displays for each tuple with available information the name of the city where the zoo is located (ville), the focused place (milieu), the name of the animal (nom), its type (espece), the visitor next to it (visiteur) and the associated guardian (gardien).

```
create or replace view panneau_de_controle as
select v.nom as ville, e.milieu as milieu, e.habitant.nom as nom,
e.habitant.espece as espece, a.profession as visiteur, value(u).nom as gardien
from villes v, visiteurs a, environements e, zoos z, table(z.equipes_gardiens) u
where a.villevisite.nom=v.nom and
e.ville.nom=v.nom and
a.lieu=e.id and
u.anim.nom=e.habitant.nom
order by v.nom
;
```



```
select * from panneau_de_controle;
```

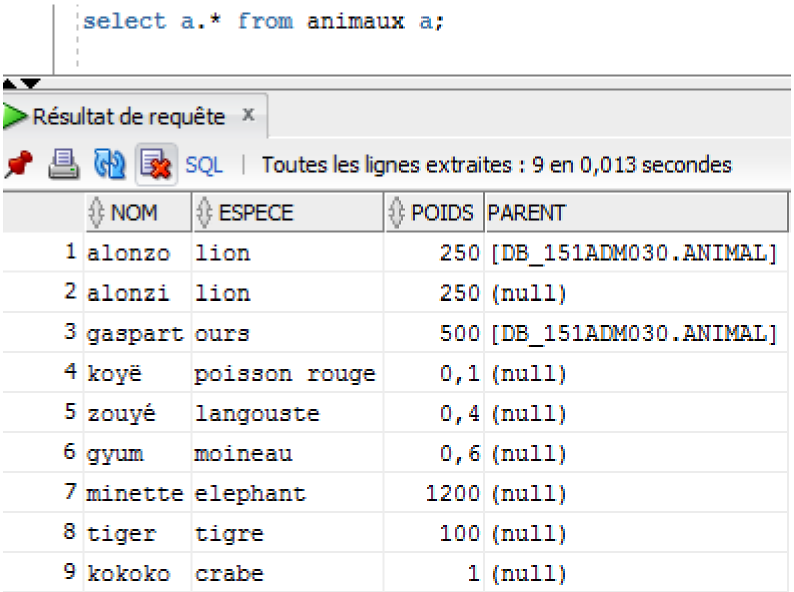
Toutes les lignes extraites : 4 en 0,062 secondes

	VILLE	MILIEU	NOM	ESPECE	VISITEUR	GARDIEN
1	Blois	eau	zouyé	langouste	plombier	jules
2	Vincennes	savane	tiger	tigre	retraite	paulette
3	Vincennes	savane	alonzo	lion	boulanger	paul
4	Vincennes	savane	minette	elephant	tondeur	paulette

Object References

One to one references

Table animal is linked with the parent table by a one to one reference, each animal has a attribute **parent** that is a reference to an other animal from table parent. Here each animal has 0 or 1 parent.

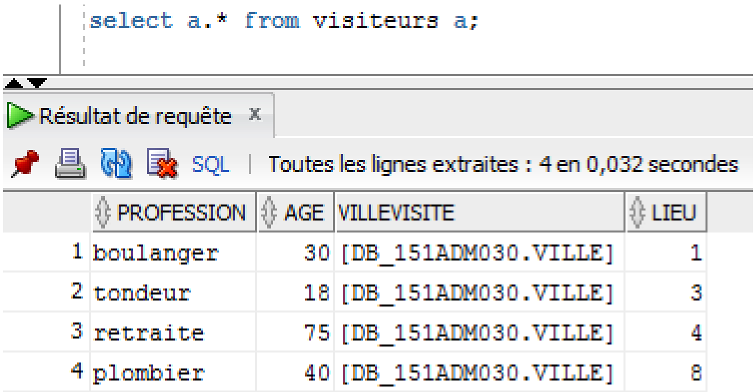


```
select a.* from animaux a;
```

Toutes les lignes extraites : 9 en 0,013 secondes

	NOM	ESPECE	POIDS	PARENT
1	alonzo	lion	250	[DB_151ADM030.ANIMAL]
2	alonzi	lion	250	(null)
3	gaspart	ours	500	[DB_151ADM030.ANIMAL]
4	koyë	poisson rouge	0,1	(null)
5	zouyé	langouste	0,4	(null)
6	gyum	moineau	0,6	(null)
7	minette	elephant	1200	(null)
8	tiger	tigre	100	(null)
9	kokoko	crabe	1	(null)

Table visiteur (visitors) also uses one to one references to table villes (cities).



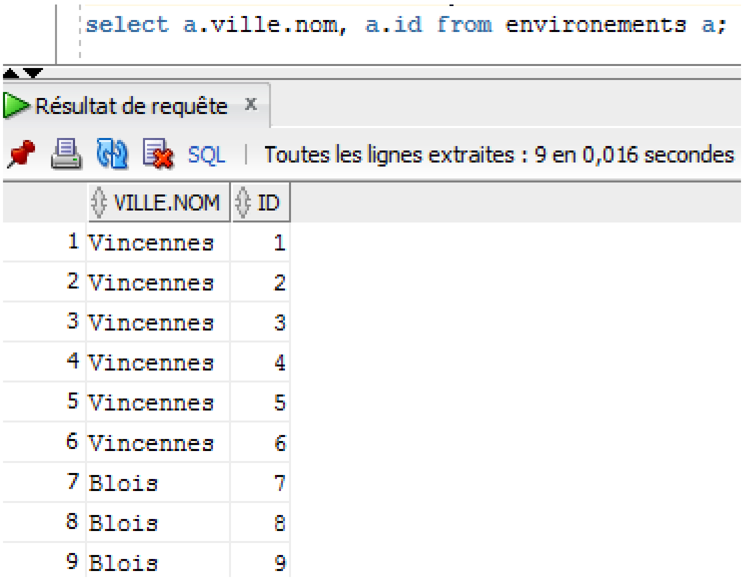
```
select a.* from visiteurs a;
```

Toutes les lignes extraites : 4 en 0,032 secondes

	PROFESSION	AGE	VILLEVISITE	LIEU
1	boulanger	30	[DB_151ADM030.VILLE]	1
2	tondeur	18	[DB_151ADM030.VILLE]	3
3	retraite	75	[DB_151ADM030.VILLE]	4
4	plombier	40	[DB_151ADM030.VILLE]	8

One to many references

Tables environements (environments) and villes (cities) are linked by a one to many relation: one city to many environments.



```
select a.ville.nom, a.id from environements a;
```

Toutes les lignes extraites : 9 en 0,016 secondes

	VILLE.NOM	ID
1	Vincennes	1
2	Vincennes	2
3	Vincennes	3
4	Vincennes	4
5	Vincennes	5
6	Vincennes	6
7	Blois	7
8	Blois	8
9	Blois	9

Object Functions

Data insert

To insert data, the following object functions are used:

- **REF ( ) :**

```
declare
par   ref   animal;
begin

select ref(A) into par  from  parents  A  where A.nom =
'alonzoTheFirst';
update  animaux B set B.parent = par  where B.nom ='alonzo';

select ref(A) into par  from  parents  A  where A.nom =
'gaspartTheFirst';
update  animaux B set B.parent = par  where B.nom ='gaspart';

end;
```

- **CURSOR** and **FOR IN LOOP**

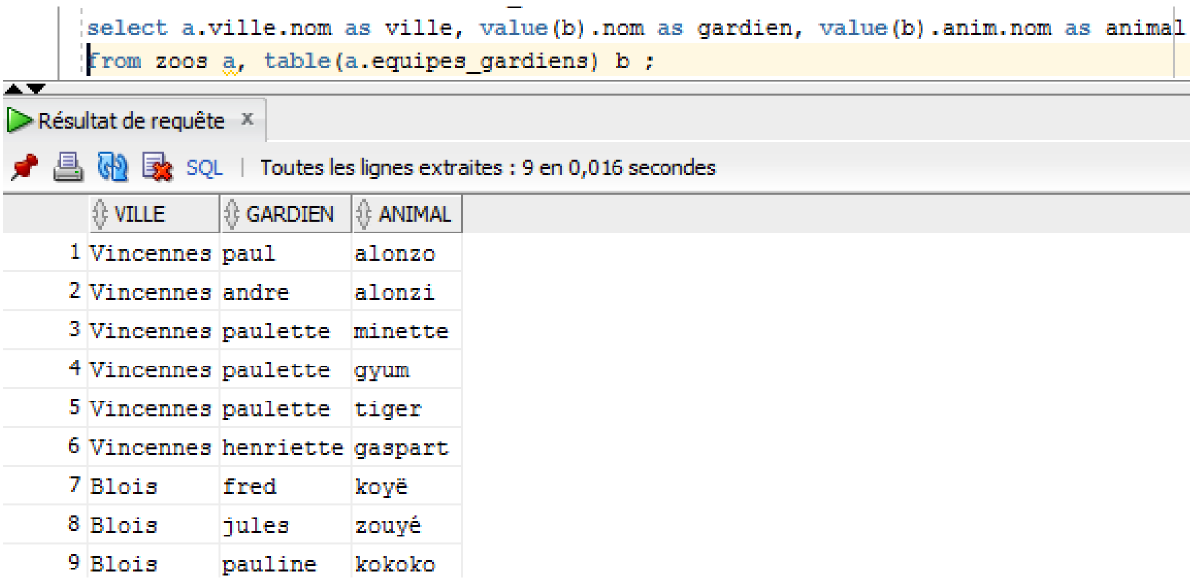
```
declare
  cursor cursor_animaux1 is select * from animaux a where
a.espece='lion' or a.espece='tigre' or a.espece='elephant';
  cursor cursor_animaux2 is select * from animaux a where
a.espece='ours' or a.espece='moineau';
  cursor cursor_animaux3 is select * from animaux a where
a.espece='poisson rouge' or a.espece='langouste' or a.espece='crabe';
  i number default 1;
  refville ref ville;
  a1 animal;
  a2 animal;
  refa3 animal;
  refa4 animal;
begin

--environements
select ref(a) into refville from villes a where a.nom='Vincennes';
for anim in cursor_animaux1 loop
  insert into environements values (i,refville, 'savane', 25,
animal(anim.nom,anim.espece,anim.poids,anim.parent));
  i:=i+1;
end loop;

for anim in cursor_animaux2 loop
  insert into environements values (i,refville, 'foret', 14,
animal(anim.nom,anim.espece,anim.poids,anim.parent));
```

```
  i:=i+1;
end loop;

select ref(a) into refville from villes a where a.nom='Blois';
for anim in cursor_animaux3 loop
  insert into environements values (i,refville, 'eau', 15,
animal(anim.nom,anim.espece,anim.poids,anim.parent));
  i:=i+1;
end loop;

end;
```

Data extraction

- **TABLE ( ) and VALUE ( ) :**

```
select a.ville.nom as ville, value(b).nom as gardien, value(b).anim.nom as
animal
from zoos a, table(a.equipes_gardiens) b ;
```



```
select a.ville.nom as ville, value(b).nom as gardien, value(b).anim.nom as animal
from zoos a, table(a.equipes_gardiens) b ;
```

Toutes les lignes extraites : 9 en 0,016 secondes

	VILLE	GARDIEN	ANIMAL
1	Vincennes	paul	alonzo
2	Vincennes	andre	alonzi
3	Vincennes	paulette	minette
4	Vincennes	paulette	gyum
5	Vincennes	paulette	tiger
6	Vincennes	henriette	gaspart
7	Blois	fred	koyë
8	Blois	jules	zouyé
9	Blois	pauline	kokoko

Conclusion

Object databases allows to store extremely complex structures easily. It is a very efficient way to store information like graphs, sounds or geographic data.

It can be used to optimize big complex data storage structures and allow fast response. It also allows to minimize the memory usage of data storage comparing to conventional ways to store objects, that is to say serialization into files and compression.

However because of the new possibilities it allows, this type of database is more complex to set up and requires additional qualifications comparing to standard relational databases.

Annexes: SQL scripts

Create types:

```
create or replace type animal as object(
nom varchar2(30),
espece varchar2(30),
poids number(5,1),
parent ref animal);

create or replace type ville as object(
nom varchar2(30),
pays varchar2(30));

create or replace type visiteur as object(
profession varchar(30),
age number(3),
villevisite ref ville,
lieu number
);

create or replace type gardien as object(
nom varchar2(30),
anim animal);

create or replace type equipe as table of gardien;
```

Create tables:

```
--row type
create table visiteurs of visiteur;
create table animaux of animal;
create table parents of animal;
create table villes of ville;

--column object type
create table environements(
id number(4) primary key,
ville ref ville,
milieu varchar(50),
temperature number(3),
habitant animal);


--collection
create table zoos(
ville ref ville,
equipes_gardiens equipe)
nested table equipes_gardiens store as pets;
```

Create views:

```
--object view
create or replace view animaux_de_la_savane of animal with object id(nom)
as
select a.*
from animaux a, environements b
where a.nom=b.habitant.nom and
b.milieu='savane';


--simple view
create or replace view panneau_de_controle as
select v.nom as ville, e.milieu as milieu, e.habitant.nom as nom,
e.habitant.espece as espece, a.profession as visiteur, value(u).nom as gardien
from villes v, visiteurs a, environements e, zoos z, table(z.equipes_gardiens) u
where a.villevisite.nom=v.nom and
e.ville.nom=v.nom and
a.lieu=e.id and
u.anim.nom=e.habitant.nom
order by v.nom
;
```

Insert data:

Script 1:

```
delete from environements;
delete from animaux;
delete from visiteurs;
delete from parents;
delete from villes;
delete from zoos;

--villes
insert into villes values(ville('Vincennes','France'));
insert into villes values(ville('Blois','France'));

--parents
insert into parents values(animal('alonzoTheFirst','lion','250,0',null));
insert into parents values(animal('gaspartTheFirst','ours','500,0',null));

--animaux
insert into animaux values(animal('alonzo','lion','250,0',null));
insert into animaux values(animal('alonzi','lion','250,0',null));
insert into animaux values(animal('gaspart','ours','500,0',null));
insert into animaux values(animal('koyë','poisson rouge','0,1',null));
insert into animaux values(animal('zouyé','langouste','0,4',null));
insert into animaux values(animal('gyum','moineau','0,6',null));
insert into animaux values(animal('minette','elephant','1200',null));
insert into animaux values(animal('tiger','tigre','100',null));
insert into animaux values(animal('kokoko','crabe','1',null));

--parents 2
declare
```

```
par   ref   animal;
begin

select ref(A) into par  from  parents  A  where A.nom = 'alonzoTheFirst';
update  animaux B set B.parent = par  where B.nom ='alonzo';

select ref(A) into par  from  parents  A  where A.nom = 'gaspartTheFirst';
update  animaux B set B.parent = par  where B.nom ='gaspart';

end;
```

Script 2:

```
declare
  cursor cursor_animaux1 is select * from animaux a where a.espece='lion' or
a.espece='tigre' or a.espece='elephant';
  cursor cursor_animaux2 is select * from animaux a where a.espece='ours' or
a.espece='moineau';
  cursor cursor_animaux3 is select * from animaux a where a.espece='poisson
rouge' or a.espece='langouste' or a.espece='crabe';
  i number default 1;
  refville ref ville;
begin

--environements
select ref(a) into refville from villes a where a.nom='Vincennes';
for anim in cursor_animaux1 loop
  insert into environements values (i,refville, 'savane', 25,
animal(anim.nom,anim.espece,anim.poids,anim.parent));
  i:=i+1;
end loop;

for anim in cursor_animaux2 loop
  insert into environements values (i,refville, 'foret', 14,
animal(anim.nom,anim.espece,anim.poids,anim.parent));
  i:=i+1;
end loop;

select ref(a) into refville from villes a where a.nom='Blois';
for anim in cursor_animaux3 loop
  insert into environements values (i,refville, 'eau', 15,
animal(anim.nom,anim.espece,anim.poids,anim.parent));
  i:=i+1;
end loop;

end;
```

Script 3:

```
declare
  refville ref ville;
  refa1 animal;
  refa2 animal;
  refa3 animal;
  refa4 animal;
  refa5 animal;
  refa6 animal;
```

```
begin
--equipes and visiteurs
select ref(a) into refville from villes a where a.nom='Vincennes';
select a.habitant into refa1 from environements a where a.habitant.nom='alonzo';
select a.habitant into refa2 from environements a where a.habitant.nom='alonzi';
select a.habitant into refa3 from environements a where
a.habitant.nom='gaspart';
select a.habitant into refa4 from environements a where
a.habitant.nom='minette';
select a.habitant into refa5 from environements a where a.habitant.nom='gyum';
select a.habitant into refa6 from environements a where a.habitant.nom='tiger';
insert into zoos values(
refville,
equipe(gardien('paul',refa1),gardien('andre',refa2),gardien('paulette',refa4),ga
rdien('paulette',refa5),gardien('paulette',refa6),gardien('henriette',refa3))
);

insert into visiteurs values(visiteur('boulanger',30,refville,1));
insert into visiteurs values(visiteur('tondeur',18,refville,3));
insert into visiteurs values(visiteur('retraite',75,refville,4));

select a.habitant into refa1 from environements a where a.habitant.nom='koyë';
select a.habitant into refa2 from environements a where a.habitant.nom='zouyé';
select a.habitant into refa3 from environements a where a.habitant.nom='kokoko';

select ref(a) into refville from villes a where a.nom='Blois';
insert into zoos values(
refville,
equipe(gardien('fred',refa1),gardien('jules',refa2),gardien('pauline',refa3))
);

insert into visiteurs values(visiteur('plombier',40,refville,8));
end;
```

Select queries:

```
select * from villes;

select * from zoos;

select a.profession, a.age, a.VILLEVISITE.nom from visiteurs a;

select * from animaux_de_la_savane;

select * from panneau_de_controle;

select a.* from visiteurs a;

select a.* from animaux a;

select a.ville.nom, a.id from environements a;

select a.ville.nom as ville, value(b).nom as gardien, value(b).anim.nom as
animal
from zoos a, table(a.equipes_gardiens) b ;
```