

Nom :	Prénom :	<i>page 1</i>
--------------	-----------------	---------------

Module Bases de Données et Web

Examen réparti du 13 novembre 2009

Version CORRIGEE

Les documents sont autorisés – Durée : 2h.

Répondre aux questions sur la feuille du sujet dans les cadres appropriés. La taille des cadres suggère celle de la réponse attendue. Utiliser le dos de la feuille précédente si la réponse déborde du cadre. Le barème est donné à titre indicatif. La qualité de la rédaction sera prise en compte. Ecrire à l'encre bleue ou noire. Ne pas dégrafer le sujet.

Exercice 1 : Requêtes avancées sur la base MONDIAL

6pts

Le schéma relationnel suivant est extrait de la base MONDIAL vu en TD:

Continent (Name, Area)

Country (Name, Code, Capital, Province, Area, Population) // *Province* est la région de la capitale

Encompasses (Country, continent, percentage) // *pourcentage de 0 à 100*

Borders (Country1, Country2, length) // *cette relation est asymétrique. On a (F, I, ...) pour France-Italie et (E,F,...) pour Espagne-France. Country1 < Country2 selon l'ordre lexicographique.*

Language (Country, Name, Percentage) // *pourcentage de 0 à 100*

Religion (Country, Name, Percentage) // *pourcentage de 0 à 100*

Sea (Name, Depth)

Geo_Sea (Sea, Country, Province)

Ecrire en SQL les requêtes suivantes

1. Le(s) pays ayant la plus grande frontière avec la france (**sans group by**) (1pt)

```
select c2.name, b.length as longueur_frontiere
from country c1, country c2, borders b
where c1.name = 'France'
and
(
(c1.code = b.country1
and c2.code = b.country2)
or
(c1.code = b.country2
and c2.code = b.country1)
)
and b.length >= ALL (select b1.length
                     from country c3, borders b1
                     where
                     (
                     (c1.code = b1.country1
                      and c3.code = b1.country2)
                     or
                     (c1.code = b1.country2
                      and c3.code = b1.country1)
                     ))
;
```

-- la même mais sans ALL

```
select c2.name, b.length as longueur_frontiere
from country c1, country c2, borders b
where c1.name = 'France'
and
(
(c1.code = b.country1
and c2.code = b.country2)
or
(c1.code = b.country2
and c2.code = b.country1)
)
and not exists (select *
                from country c3, borders b1
                where b1.length > b.length and
                (
                (c1.code = b1.country1
                 and c3.code = b1.country2)
                or
                (c1.code = b1.country2
                 and c3.code = b1.country1)
                ))
;
```

2. Les pays frontaliers de la France avec, pour chacun, le nombre de pays voisins (0,5pt)

```
select c2.name, count(c3.code) as nbvoisins
from country c1, country c2, country c3, borders b, borders b1
where c1.name = 'France'
and
(
(c1.code = b.country1
and c2.code = b.country2)
or
(c1.code = b.country2
and c2.code = b.country1)
)
and
(
(c3.code = b1.country1
and c2.code = b1.country2)
or
(c3.code = b1.country2
and c2.code = b1.country1)
)
group by c2.name;
```

3. Combien de gens dans le monde ont le français pour langue officielle ? (0,5pt)

```
select SUM(c1.population * l.percentage / 100)
from country c1, language l
where l.country=c1.code and l.name = 'French';
```

4. Quelle(s) religion(s) sont pratiquées dans tous les continents ? (1pt)

```
select distinct r.name
from religion r
where not exists
(select * from continent co where not exists
(select * from country c, encompasses e, religion r1
where e.country=c.code and e.continent = co.name
and r1.name=r.name and r1.country=c.code));
```

5. Quel(s) continent(s) a(ont) une densité moyenne supérieure à 30 ? (1pt)

```
select e.continent
from country c, encompasses e
where e.country=c.code
group by e.continent
having SUM(c.population)/SUM(c.area) > 30 ;
```

6. Quelle est la mer la plus profonde qui baigne l'Afrique ? (1pt)

```
select distinct s.name, s.depth, c.name
from sea s, geo_sea g, encompasses e, country c
where e.country=c.code and e.continent='Africa' and g.country=c.code and g.sea=s.name
and s.depth =
(select MAX(s1.depth)
from sea s1, geo_sea g1, encompasses e1, country c1
where e1.country=c1.code and e1.continent='Africa' and g1.country=c1.code and g1.sea=s1.name);
```

7. Couples de pays voisins partageant au moins une langue à plus de 30% (1pt)

```
select distinct c1.name, c2.name
from country c1, country c2, borders b, language l1, language l2
where b.country1=c1.code and b.country2=c2.code and l1.country=c1.code and l2.country=c2.code and
l1.percentage > 30 and l2.percentage > 30 and l1.name=l2.name order by c1.name ;
```

Exercice 2. SQL3 : modélisation

3 pts

On considère le schéma relationnel suivant extrait de la base MONDIAL (vue en TD):

Continent (Name, Area)

Country (Name, Code, Capital, Province, Area, Population) // Province est la région de la capitale

Province (Name; Country, Population, Area, Capital)

City (Name, Country, Province, Population, Longitude, Latitude)

Encompasses (Country, continent, percentage)

Borders (Country1, Country2, length) // cette relation est asymétrique. On a (F, I, ...) pour France-Italie et (E,F,...) pour Espagne-France. Country1 < Country2 selon l'ordre lexicographique.

On souhaite migrer cette base vers le relationnel-objet. Dans ce nouveau schéma, les données sont stockées dans **4 tables** (TheContinents, TheCountries, TheProvinces, TheCities) uniquement. **Aucune autre table ne doit être créée.**

Définissez en SQL3 les **types et les tables** permettant cette migration, (en maximisant le partage d'objets dès que possible). Toutes les informations du schéma relationnel doivent être conservées.

```
create type Continent as object (  
    name varchar2(20),  
    Area number );  
  
create type encompasses as object (  
    cont ref Continent,  
    pourcentage number);  
/  
create type EnsEncompasses as table of encompasses;  
/  
create type Country;  
/  
create type City;  
/  
create type Province;  
/  
create type border as object (  
    pays ref Country,  
    length number);  
/  
create type EnsBorder as table of border;  
/  
create type Country as object (  
    name varchar2(30),
```



```
code number,  
capital ref City,  
prov ref Province,  
Area number,  
population number,  
estdans EnsEncompasses,  
frontieres EnsBorder);  
  
/  
create type Province as object (  
    name varchar2(30),  
    pays ref Country,  
    population number,  
    area number,  
    capital ref City );  
  
/  
create type City as object (  
    name varchar2(30),  
    pays ref Country,  
    prov ref Province,  
    population number,  
    longitude number,  
    latitude number);  
  
/  
create table TheContinents of Continent;  
  
create table TheCountries of Country  
nested table estdans store as lescontinents,  
nested table frontieres store as lesfrontieres;  
  
create table TheProvinces of Province;  
  
create table TheCities of City;
```

Exercice 3. SQL3 : mise à jour et interrogation**4 pts**

On considère le schéma objet-relationnel suivant :

```
create type Personne;
```

```
create type Ville as object (  
    nom varchar2(30),  
    codePostal varchar2(5));
```

```
create type Maison as object (  
    numero number,  
    rue varchar2(30),  
    situe ref Ville);
```

```
create type EnsMaisons as table of ref Maison;  
create type EnsParents as varray(2) of ref Personne;
```

```
create type Personne as object (  
    nom varchar2(20),  
    prenom varchar2(20),  
    age number(3),  
    habite EnsMaisons,  
    conjoint ref Personne,  
    parents EnsParents );
```

create table LesPersonnes of Personne
nested table habite store as habitations;
create table LesVilles of Ville;
create table LesMaisons of Maison;

On suppose que la base contient deux personnes, Max Traide et Annie Malle, les villes de Clamecy et de Paris ainsi que les maisons 20 rue du Bac à Paris et 36 rue Chaude à Clamecy.,

Question 1 (2 pts)

De nouvelles informations doivent être ajoutées à la base :

Max Traide et Annie Malle sont mariés. Ils ont un fils, Jean Traide, qui a 30 ans, et qui habite 20 rue du Bac, à Paris 7^{ème}.

Ecrivez les instructions SQL3 permettant d'intégrer ces informations à la base.

Mariage de Max Traide et Annie Malle :

```
update LesPersonnes p
set p.conjoint = (select ref(i)
from LesPersonnes i where i.nom='Traide' and i.prenom='Max')
where p.nom='Malle' and p.prenom='Annie';
```

```
update LesPersonnes p
set p.conjoint = (select ref(i) from LesPersonnes i where i.nom='Malle' and
i.prenom='Annie')
where p.nom='Traide' and p.prenom='Max';
```

insertion de leur fils Jean Traide :

```
insert into LesPersonnes values
```

```
(Personne ('Traide', 'Jean', 30,  
    EnsMaisons((select ref(m) from LesMaisons m where m.rue='du Bac' and m.numero=20  
and m.situe.nom='Paris' and m.situe.co  
dePostal='75007'))),  
    NULL,  
    EnsParents((select ref(p) from LesPersonnes p where p.nom='Malle' a  
nd p.prenom='Annie'), (select ref(p) from LesPersonnes p where p.nom='Traide' and  
p.prenom='Max'))));
```

Question 2 (2 pts)

Ecrivez en SQL3 les requêtes suivantes :

1. Nom et prenom du conjoint de Max

```
select p.nom, p.prenom from LesPersonnes p where p.conjoint.prenom='Max';
```

2. Prenom des personnes qui habitent avec leur conjoint ---

```
select p.prenom from LesPersonnes p where p.habite in p.conjoint.habite;
```

3. Adresse (numero, rue, nom de la ville et code postal) de Jean Traide ---

```
select value(h).numero, value(h).rue, value(h).situe.nom, value(h).situe.codePostal  
from les Personnes p, table(p.habite) h  
where p.nom='Traide' and p.prenom='Jean';
```

4. Nom des personnes qui habitent Paris

```
Select p.nom from LesPersonnes p, table(p.habite) m where  
value(m).situe.nom='Paris';
```