
Bases de Données - Valeur C

Examen Partiel (6 Points)

10 Fvrier 1998

1 Modle Orient-Objet (10 points)

Un muse d'instruments de musique veut grer son inventaire avec le SGBD O2. Soit donn le schma suivant:

```
import schema o2kit class Text;
class Instrument inherit Object private type
    tuple(numro: integer,
          annee: integer,
          emplacement: Emplacement,
          description: Text
    )
end;
class Cordes inherit Instrument
    tuple(accord: string,
          constructeur: string)
end;
class Vent inherit Instrument
    tuple(accord: string)
end;
class Percussion inherit Instrument end;

class Emplacement inherit Object private type
end;
class tagre inherit Emplacement public type
    tuple(salle: Salle,
          repre: integer)
end;
class Salle inherit Emplacement public type
    tuple(dsignation: string)
end;
```

Question 1 (Modle O₂/2 Points) *Dcrivez ce schma:*

1. (0.5 points) *Comment peut on ajouter le fait qu'on voudrait distinguer, parmi les instruments vent, entre les bois et les cuivres?*
2. (0.5 points) *O sont situs les instruments dans le muse?*
3. (0.5 points) *Est-ce qu'un instrument peut se trouver des endroits differents?*
4. (0.5 points) *Comment peut-on rendre les instruments et les emplacements persistants?*

Question 2 (O₂C/5 Points) *Donnez les mthodes pour chacun des traitements suivants. Prcisez pour chaque mthode son nom, sa signature, son corps et la classe dans laquelle elle est dfinie. Utilisez le langage O₂C de*

prference.

1. (1.5 points) Les instruments qui se trouvent dans une salle donne. Attention, il ne faut pas oublier les instruments sur les tagres.

Solution :

```
method body donne_instruments: unique set(Instruments)
in class Salle {
  o2 unique set(Instrument) sinstr = unique set();
  o2 Instrument instr;
  o2 Etagere etag;
  for (instr in Les_Instruments where instr.emplacement == self)
    sinstr += unique set(instr);
  for (etag in Les_Etageres where etag.salle == self)
    for (instr in Les_Instruments where instr.emplacement == etag)
      sinstr += unique set(instr);
  return sinstr;
}
```

2. (1 point) Le plus vieux instrument dans une salle.

Solution :

```
method body plus_vieux: Instrument
in class Salle {
  o2 Instrument instr, vieux = nil;
  for (instr in self->donne_instruments)
    if (vieux = nil)
      vieux = instr;
    else if (instr.anne < vieux.anne)
      vieux = instr;
  return vieux;
}
```

3. (2.5 points) L'emplacement (numros de salle et d'tagre) d'un instrument. Si l'instrument n'est pas sur une tagre (par exemple un piano), le numro de l'tagre est gal 0.

Solution :

```
/* Instruments */
method body donne_emplacement:tuple(salle: string, etagere:integer)
in class Instrument {
  return tuple(salle: self->emplacement->donne_salle(),
              etagere: self->emplacement->donne_etagere());
}

/* Emplacements */
method donne_salle:string in class Emplacement;

method body donne_etagere:integer
in class Emplacement {
  return 0;
}

/* Surchargement Salle */
```

```
method body donne_salle:string
in class Salle {
  return self->dsignation;
}

/* Surchargeement tagre */
method body donne_salle:string
in class tagre {
  return self->salle->donne_salle();
}
method body donne_etagere:integer
in class tagre {
  return self->repre;
}
```

Question 3 (OQL/3 Points) Essayer de rpondre aux questions suivantes par des requetes OQL (sans utiliser des mthodes). Ajoutez les racines de persistance ncessaires :

1. (0.5 points) L'emplacement des instruments percussion.

Solution :

Il faut ajouter une racine de persistance Les_Percussions qui contiennent tous les instruments percussion.

```
select p.emplacement
  from p in Les_Percussions
```

2. (1 point) Les instruments qui ne se trouvent pas sur des tagres.

Solution :

```
select distinct instr
  from instr in Les_instruments
  where instr->emplacement in Les_salles
```

3. (1.5 points) Les instruments dans la salle "Instruments Vents XVIII". Attention, il ne faut pas oublier les instruments sur les tagres.

Solution :

```
select distinct instr
  from instr in Les_instruments
  where instr->emplacement in
    (select s
      from s in Les_salles,
      where s->dsignation == ``Instruments Vents XVIII``)
  or instr->emplacement in
    (select e
      from e in Les_tagres,
      where e->salle->dsignation == ``Instruments Vents XVIII``)
```

2 Bases de donnes spatiales (4 points)

Soit la base O2:

```
name Communes: set (Communes)
```

```
Class Dept
  tuple (num-dept: integer,
        nom-reg : string,
        communes : set (Commune))
```

```
Class Commune
  tuple (nom-commune : integer,
        nb-hab: integer,
        departement : Dept,
        geometrie : Zone)
```

```
Class Zone...
```

Question 4 (Requetes/2 points)

1. (1 point) Calculer en OQL le nombre d'habitants de la region de nom Alsace-Lorraine (on se sert de la fonction agrgat sum).

Solution :

```
sum (select c.nb-hab from c in Communes where c.departement.nom-reg="Alsace-Lorraine")
```

2. (1 point) Afficher la gomtrie de l'Alsace-Lorraine (utiliser la fonction agrgat Uniongeo qui prend en entre un ensemble de Zones et en sortie fait l'union gomtrique en liminant les frontieres internes).

Solution :

```
unionGeo (select c.geometrie from c in Communes where c.departement.nom-reg="Alsace-Lorraine")
```

Question 5 (Index spatiaux/2 points) Donner l'algorithme de fenêtrage de la grille rgulire. Utiliser un pseudo-code comme dans les transparents ou le livre ou toute autre prsentation concise et prcise.

3 Relationnel tendu (3 points)

Soit la base relationnelle NINF, avec un type abstrait *Zone*, des dpartements et communes de France de schma :

```
Commune (nom-commune: string, num-dept: integer, geometrie: Zone)
Departement (num-dept, communes : set (Commune), geometrie : Zone)
```

Question 6 (3 points) Calculer la table *Departement* partir de la table *Commune*. Calculer d'abord l'ensemble des communes de chaque departement (table *R1*, 1 point), la gomtrie de chaque dpartement (table *R2*, 1 point), puis le rsultat *Departement* partir de *R1* et *R2* (1 point). On se servira de l'algbre NINF vue en cours, de la fonction agrgat *unionGeo* dfinie plus haut et de l'opration *apply* ($F(A)$, R) qui prend en entre une table R ayant un attribut de nom A et qui remplace dans chaque nuplet la valeur de l'attribut A par la fonction $F(A)$.

Solution :

$$R1 = nest_{num-dept}(\pi_{num-dept, nom-commune}(Commune))$$

$$R3 = nest_{num-dept}(\pi_{num-dept, geometrie}(Commune))$$

$R3$ a un attribut gomtries qui est un ensemble de zones.

$$R2 = apply(unionGeo(geometries), R3)$$

$$Departement = R1 \cap R2$$

4 Optimisation relationnelle (3 points)

Soit la relation G-DEPT (num-dept, geometrie) qui donne pour chaque departement sa geometrie et la relation C-DEPT (num-dept, commune) qui donne pour chaque commune son dpartement.

Question 7 (3 points)

1. (1 point) Donner la requête SQL qui calcule $DEPT = C-DEPT \bowtie G-DEPT$ c'est--dire qui cre la table des dpartements d'attributs num-dept, commune, geometrie.

Solution :

```
Select C-DEPT.num-dept, geometrie, commune
From C-DEPT, G-DEPT
Where C-DEPT.num-dept = G-DEPT.num-dept
```

2. (1 point) On suppose qu'il n'y a pas d'index sur le numero de departement ni dans C-DEPT, ni dans G-DEPT. Donner l'algorithme de jointure (dtaill) utilis par ORACLE, ou le plan d'excution choisi (Outil EXPLAIN).

Solution :

```
0 SELECT STATEMENT
1 MERGE JOIN
2 SORT JOIN
3 TABLE ACCESS FULL C-DEPT
4 SORT JOIN
5 TABLE ACCESS FULL G-DEPT
```

3. (1 point) On suppose qu'il existe un index sur le numero de departement soit dans G-DEPT soit dans C-DEPT. Pour laquelle des tables serait-il le plus intressant d'avoir l'index pour cette requete. Dtailler l'algorithme de jointure choisi par ORACLE ou donner le plan d'excution choisi (EXPLAIN).

Solution :

```
0 SELECT STATEMENT
1 NESTED LOOPS
2 TABLE ACCESS FULL G-DEPT
3 TABLE ACCESS BY ROWID C-DEPT
4 INDEX RANGE SCAN IDX-C-DEPT-num-dept
```