

Support de cours autorisé – Durée : 2 h – Barème fourni à titre indicatif – Questions indépendantes

Exercice 1 : Algèbre relationnelle (2 points)

Soit la base de données relationnelle suivante.

CLIENT (IDcli, Nom, Prénom, Rue, CP, Ville)
FACTURE (IDFac, DateFac, IDcli#)
PRODUIT (IDProd, Libellé, PrixHT)
LIGNEFAC (IDFac#, IDProd#, Quantité)

Clés primaires
Clés étrangères#

Formuler en algèbre relationnelle la requête : « coordonnées complètes des clients qui n'ont reçu aucune facture ». Vous pouvez utiliser indifféremment la notation mathématique, fonctionnelle ou graphique.

Exercice 2 : SQL (12 points)

Soit la base de données « événementiel » dont le schéma relationnel est donné ci-dessous.

MANIFESTATION (id, nom, date, tarif, id_organisme_organisateur#, id_lieu#)
ORGANISME (id, nom, type adresse)
LIEU (id, nom, type, adresse)
INTERVENANT (id, nom, prénom, coordonnées, id_organisme#)
FEEDBACK_VISITEUR (id, texte, id_manifestation#)
INTERVENIR (id_manifestation#, id_intervenant#, nb_heures)

Clés primaires
Clés étrangères#

Formuler en SQL les requêtes suivantes.

- Liste des manifestations indiquant les noms de l'organisme organisateur et du lieu (et non leurs identifiants), triée par ordre chronologique inverse.
- Liste des manifestations ayant lieu entre le 1^{er} juillet 2011 et le 31 août 2011 inclus.
- Tarifs minimum, moyen et maximum des manifestations.
- Nombre de retours visiteurs (*feedback*) par manifestation (indiquer le nom de la manifestation).
- Prénom et coordonnées des intervenants dont le nom est saisi au clavier.
- Identifiant, nom, prénom et nom d'organisme des « intervenants » qui n'interviennent dans aucune manifestation.
- Identifiant, nom, prénom (sans doublon) des intervenants qui :
 - soit font partie d'un organisme qui a organisé au moins une manifestation ;
 - soit sont intervenus sur au moins une manifestation pendant plus de 6 heures.
- Par manifestation (indiquer le nom des manifestations), identifiant, nom, prénom et nombre d'interventions des intervenants dont le nombre d'heures d'intervention total est supérieur ou égal à un seuil saisi au clavier.

9. Nom du ou des lieux les plus utilisés par des manifestations.

10. Nom des organismes qui interviennent dans toutes les manifestations, qu'ils soient ou non organisateurs. Règle de gestion : des intervenants d'un organisme organisateur d'une manifestation interviennent nécessairement lors de cette manifestation.

11. Créer un « groupe d'utilisateurs » nommé SAISIE.

12. Donner accès en lecture et en écriture (sauf suppression) à la table FEEDBACK_VISITEUR pour les utilisateurs du groupe SAISIE.

Exercice 3 : Requêtes XQuery (6 points)

Soit le document XML `livres.xml`, qui représente le catalogue d'ouvrages d'une librairie en ligne et dont l'arborescence d'éléments et d'attributs (préfixés par @) est représentée ci-dessous. Le caractère * indique que l'élément concerné peut apparaître plusieurs fois dans le document.

```
catalogue
... livre *
... .. @identifiant
... .. auteur
... .. titre
... .. genre
... .. prix
... .. date_publication
... .. resume
```

Formuler à l'aide du langage XQuery les requêtes suivantes (utiliser, de la syntaxe XPath ou FLWOR, la plus appropriée).

- Liste des noms d'auteurs, chaque auteur ne devant apparaître qu'une fois.
- Nombre de livres publiés avant 2001.
- Résumé des ouvrages de genre informatique ou science fiction.
- Liste des ouvrages d'informatique au format `<livre id="ID">TITRE</livre>`.
- Nombre de livres par genre.
- Moyenne des prix des livres par auteur et par genre, triée par prix moyen décroissant.

Correction Exercice 1

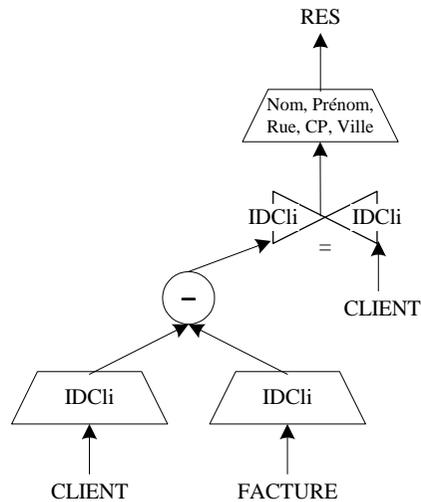
Notation mathématique

$$\text{RES} = \prod \langle \text{Nom, Prénom, Rue, CP, Ville} \rangle$$
$$(\text{CLIENT} \times$$
$$(\prod \langle \text{IDcli} \rangle (\text{CLIENT}) - \prod \langle \text{IDcli} \rangle (\text{FACTURE})))$$

Notation fonctionnelle

$$\text{RES} = \text{PROJECT}(\text{JOIN}(\text{CLIENT}, \text{MINUS}(\text{PROJECT}(\text{CLIENT} / \text{IDcli}),$$
$$\text{PROJECT}(\text{FACTURE} / \text{IDcli}))) / \text{Nom, Prénom, Rue, CP, Ville}$$

Notation graphique



Correction Exercice 2

```
--1
SELECT m.id, m.nom, date, tarif, o.nom, l.nom
FROM MANIFESTATION m, ORGANISME o, LIEU l
WHERE id_organisme_organisateur = o.id
AND id_lieu = l.id
ORDER BY date DESC;
```

```
--2
SELECT *
FROM MANIFESTATION
WHERE date BETWEEN '01/07/2011' AND '31/08/2011';
```

```
--3
SELECT MIN(tarif), AVG(tarif), MAX(tarif)
FROM MANIFESTATION;
```

```
--4
SELECT nom, count(*)
FROM MANIFESTATION, FEEDBACK_VISITEUR
WHERE id_manifestation = id
GROUP BY nom;
```

```
--5
SELECT prénom, coordonnées
FROM INTERVENANT
WHERE nom = '&nom_saisi';
```

```
--6
SELECT i.id, i.nom, prénom, o.nom
FROM INTERVENANT i, ORGANISME o
WHERE id_organisme = o.id
AND i.id NOT IN (
  SELECT id_intervenant
  FROM INTERVENIR);
```

```
--7
SELECT DISTINCT id, nom, prénom
FROM INTERVENANT
WHERE id_organisme IN (
  SELECT id_organisme_organisateur
  FROM MANIFESTATION)
OR id IN (
  SELECT id
  FROM INTERVENIR
  WHERE nb_heures > 6);
```

```
--8
SELECT m.nom, i.id, i.nom, prénom, COUNT(*)
FROM INTERVENANT i, INTERVENIR t, MANIFESTATION m
WHERE i.id = id_intervenant
AND id_manifestation = m.id
GROUP BY m.nom, i.id, i.nom, prénom
HAVING SUM(nb_heures) >= &seuil;
```

```
--9
SELECT l.nom
FROM LIEU l, MANIFESTATION m
WHERE l.id = id_lieu
GROUP BY nom
HAVING COUNT(*) = (
  SELECT MAX(COUNT(id_lieu))
  FROM MANIFESTATION
  GROUP BY id_lieu);
```

```
--10
SELECT nom
FROM ORGANISME o
WHERE NOT EXISTS (
  SELECT *
  FROM MANIFESTATION m
  WHERE NOT EXISTS (
    SELECT *
    FROM INTERVENANT i, INTERVENIR
    WHERE id_intervenant = i.id
    AND id_manifestation = m.id
    AND id_organisme = o.id));
```

```
--11
CREATE ROLE SAISIE;
```

```
--12
GRANT SELECT, INSERT, UPDATE ON FEEDBACK_UTILISATEUR TO SAISIE;
```

Correction Exercice 3

```
(: 1 :)
distinct-values(doc("livres.xml")//auteur)

(: 2 :)
count(doc("livres.xml")//livre[date_publication < "2001-01-01"])

(: 3 :)
doc("livres.xml")//livre[genre = "Informatique"
                        or genre = "Science Fiction"]/resume

(: 4 :)
for $x in doc("livres.xml")//livre
where $x/genre = "Informatique"
return <livre id="{data($x/@identifiant)}">{data($x/titre)}</livre>

(: 5 :)
for $g in distinct-values(doc("livres.xml")//genre)
let $c := count(doc("livres.xml")//livre[genre = $g])
return <groupe>
  <genre>{data($g)}</genre>
  <nb_livres>{$c}</nb_livres>
</groupe>

(: 6 :)
for $a in distinct-values(doc("livres.xml")//auteur),
  $g in distinct-values(doc("livres.xml")//genre)
let $m := avg(doc("livres.xml")//livre[auteur = $a and genre = $g]/prix)
order by $m descending
return <groupe>
  <auteur>{data($a)}</auteur>
  <genre>{data($g)}</genre>
  <prix_moyen>{$m}</prix_moyen>
</groupe>
```