

Université Paris 7

Licence 3 Informatique et Master 1 ISIFAR, Bases de données.

23 mai 2011

Durée : 3 heures. Documents manuscrits, notes de cours, notes de TD/TP autorisés. Livres interdits.

Le sujet comporte 4 pages.

## Comprendre les requêtes

**Exercice 1** Les tables X(A,B) et Y(B,C) initialement vides sont alimentées par

```
insert into X values          et insert into Y
(1,2),                       values
(8,4),                       (2,3),
(6,5),                       (2,4),
(8,5),                       (5,10),
(12,NULL),                   (null,4),
(1,0);                       (0,0);
```

Donner les résultats des requêtes :

```
select count(*) as D, B      et select sum(b-c) as E
from X natural left join Y   from X natural join Y
where a>b                    where a + 1= any(select b
group by B                   from Y
having max(A) > 6 and b<> 0   )
order by D;                  group by a;
```

---

## Écrire les scripts SQL

**Exercice 2** Une base de données d'un site d'enchère en ligne contient les tables suivantes :

```
client(id_client, nom, prenom, date_naiss, adresse)
produit(id_produit, nom_produit, description)
objet(id_objet, id_produit, id_vendeur, date_ouverture, date_cloture, prix)
offre(id_acheteur, id_objet, quand, valeur)
vendu(id_objet, id_acheteur, prix_achat)
```

- (A) La table `client` contient tous les clients du site, aussi bien les acheteurs que les vendeurs. L'attribut `id_client` sert de clé, tous les attributs sont non null, `date_naiss` est de type `date`, `nom`, `prenom`, `adresse` sont de chaînes de caractères.
- (B) La table `produit` énumère tous les produits qui peuvent être vendus. L'attribut `id_produit` sert de clé, `nom_produit` est non null. Pour donner un exemple, un produit ça peut être CD Led Zeppelin In Through the Out Door.
- (C) La table `objet` contient tous les objets mis en vente. Chaque objet est associé à un produit, par exemple pour chaque exemplaire du CD de Led Zeppelin ci-dessus qui est mis en vente on crée un enregistrement dans la table `objet`.  
`id_objet` c'est la clé primaire, `id_produit` c'est une clé étrangère vers `produit`, `id_vendeur` c'est une clé étrangère vers l'attribut `id_client` de `client`, `date_ouverture` donne la

date d'ouverture de l'enchère de cet objet, `date_cloture` donne la date de clôture de l'enchère pour l'objet (c'est le seul attribut qui peut être NULL). L'attribut `prix` donne le prix minimal de l'objet, le vendeur refuse le prix inférieur à celui-ci, ce prix doit être supérieur à 0.

- (D) `offre` c'est la table qui donne toutes les offres pour les objets mis en enchères. La table contient toutes les offres, chaque offre est gardée pour toujours (même si on a reçu une offre supérieure). `id_acheteur` c'est une clé étrangère vers l'attribut `id_client` de la table `client`. La valeur de cet attribut permet d'identifier le client qui soumet l'offre (l'acheteur potentiel). `id_objet` c'est une clé étrangère vers l'objet que l'acheteur veut acquérir. L'attribut `quand` de type `timestamp` donne la date et l'heure du moment où l'offre est arrivé. L'attribut `valeur` est un nombre décimal avec deux chiffres après virgule. Cette attribut donne le prix proposé par l'acheteur. La valeur doit être supérieure à 0. La clé primaire de la table `offre` c'est le triplet `id_acheteur, id_objet, quand`.
- (E) La table `vendu` contient tous les objets vendus (ces objets sont toujours maintenus dans la table `objet`). Les attributs `id_objet, id_acheteur` sont des clés étrangères vers `objet` et `client`. `id_client`, ce couple forme la clé primaire. On devine à quoi sert `prix_achat`.

Si la date d'aujourd'hui est supérieure à la date `objet.date_cloture` alors l'enchère pour cet objet est terminée.

**Les questions suivantes sont indépendantes. Sauf si c'est demandé explicitement les requêtes ne doivent pas créer des tables ou des vues auxiliaires (la création de tables auxiliaires diminue la note).**

**Question 1:** Écrire les requêtes SQL qui construisent toutes les tables avec les contraintes d'intégrité suivant la description ci-dessus.

**Question 2:** Écrire une commande permettant d'insérer *d'un seul coup* deux objets dans la table `objet` :

(4, 2, 1, aujourd'hui, dans 10 jours, 30) et

(5, 3, 1, aujourd'hui, dans 5 jours, 45) (l'ordre d'attributs est le même que dans la spécification de la table).

À la place de `aujourd'hui` il faut mettre la date d'aujourd'hui. A la place `dans 10 jours` il faut mettre la date dans 10 jours, pareil pour `dans 5 jours`.

**Mais attention, on préfère que toutes les dates soient calculées avec les fonctions et expressions SQL qui conviennent, les dates mises *en dur* diminuent la note de cet exercice.**

**Question 3:** Pour l'objet `id_objet=4` trouver l'offre la plus élevée.

**Question 4:** Trouver le nombre d'objets pour lesquels l'enchère est terminée sans qu'il y ait une offre. (L'enchère est terminée si la date d'aujourd'hui est supérieure à la date `objet.date_cloture`.)

**Question 5:** Quel est le nombre d'objets avec les enchères en cours (non terminées)?

**Question 6:** Pour le produit "chaussures Nike" trouver tous les objets correspondants et pour chaque des objets trouvés afficher `id_objet`, le timestamp `quand` de l'offre la plus récente et la valeur de cette offre.

**Question 7:** Trouver la valeur la plus élevée d'une offre pour un objet correspondant au produit "chaussures adidas". (On regarde tous les objets pour ce produits et toutes les offres pour ces objets.)

**Question 8:** Trouver tous les vendeurs qui ont mis en vente au moins 10 objets. Pour chaque de ces vendeurs afficher nom, prénom, date de naissances, et le nombre d'objets mis en vente. Le résultat doit être trié dans l'ordre décroissant par le nombre d'objets mis en vente.

**Question 9:** Trouver le nombre de vendeurs. (Le vendeur c'est un client qui a mis en vente au moins un objet).

**Question 10:** Trouver le nombre de clients qui n'ont jamais rien mis en vente (qui ne sont pas vendeurs).

**Question 11:** Supprimer de la table `offre` toutes les offres qui concernent les objets pour lesquels l'enchère est terminée.

**Question 12:** Quel est le nombre d'offres pour chaque objet ? Pour chaque objet afficher : nom de produit correspondant, la date de clôture de l'enchère, la date d'ouverture, le nombre d'offres. Trier le résultat par le nombre d'offres.

**Question 13:** Quel est l'âge moyen des clients ?

**Question 14:** Quel est le nombre maximal d'offres soumises par un seul acheteur ? On prendra en compte toutes les offres de chaque acheteur, c'est-à-dire tous les enregistrements de la table `offre` concernant chaque acheteur. Par exemple si la table `offre` contient 4 offres de l'acheteur X, 5 offres de Y, et 10 offres de Z, la requête retournera 10.

**Question 15:** Pour le client `id_client=3` donner la liste des objets pour lesquels ce client a fait au moins 3 offres. On affichera uniquement les objets pour lesquels l'enchère n'est pas terminée.

**Question 16:** Créer une vue avec tous les objets pour lesquels l'enchère n'est pas terminée.

---

## JDBC

L'exercice suivant ne concerne pas les étudiant ISIFAR. **Exercice 3** La base de données utilisée dans cet exercice est la même que dans l'exercice précédent.

Nous avons la classe `Enchere` :

```
public class Enchere(  
    private Connection connect;  
    .....  
    public boolean ajouterOffre(int id_acheteur, int id_objet, int valeur){  
  
    }  
}
```

On suppose que la classe est complètement écrite (constructeurs, méthodes) sauf la méthode `ajouterOffre`. Le constructeur de la classe établit la connexion avec la base et on suppose que l'attribut `connect` contient la référence vers l'objet `Connection` déjà correctement initialisé. Il reste à écrire la méthode `ajouterOffre` — c'est le but de cet exercice. Cette méthode doit ajouter une nouvelle offre dans la table `offre`. Mais auparavant la méthode doit faire quelques vérifications :

1. si c'est la première offre pour cet objet il faut vérifier si `valeur` est supérieure au prix minimal `objet.prix` fixé par le vendeur,
2. s'il y a déjà d'autres offres pour l'objet alors `valeur` doit être supérieure à toutes les valeurs d'offres enregistrées pour l'objet,
3. la date d'aujourd'hui doit être inférieure ou égale à la date `objet.date_cloture` qui termine l'enchère pour l'objet.

Si une de conditions ci-dessus n'est pas satisfaites la méthode retourne `false` et l'offre ne sera pas enregistrée. Sinon l'offre doit être enregistrée, avec les valeurs d'attributs données en paramètres. La valeur de l'attribut `quand` sera donnée par `current_timestamp` (une fonction de SQL).

Écrire la méthode `ajouterOffre` en respectant les contraintes imposées dans cet exercice.

---

## Conception d'une BD

**Exercice 4** Le but de cet exercice est de concevoir une base de données de gestion d'urgences d'un hôpital.

Les entités principales : patient, médecin, et sans doute d'autres entités nécessaires que vous devez découvrir vous mêmes.

La BD permettra de récupérer les informations suivantes :

- l'heure d'arrivée du patient,
- le médecin qui a traité le patient,
- l'heure à laquelle avait lieu le premier examen par un médecin (en général il y a un temps d'attente entre le moment d'arrivée et le moment de l'examen),
- les examens et les soins donnés au patient, chaque examen a un coût fixe stocké dans la base,
- la décision du médecin soignant (patient retourne à la maison ou il est admis à l'hôpital, dans ce dernier cas il faut savoir dans quel service le patient est envoyé),

**Question 1:** Proposez un Modèle Conceptuel de Données pour cette BD, avec une taille raisonnable d'au moins 3 entités avec des attributs appropriés, plusieurs relations. N'oubliez pas de bien marquer les cardinalités.

**Question 2:** En déduire le schéma de la base, c'est-à-dire quelles tables vous obtenez à partir de votre MCD. Écrire les commandes `CREATE TABLE` pour les tables. N'oubliez pas la clé, les contraintes référentielles s'il y en a (clés étrangères) etc.

---

## Dépendances fonctionnelles

**Exercice 5** Pour la table  $T(A, B, C, D, E)$  avec les dépendances :  $A \rightarrow BC$  et  $CD \rightarrow E$  trouver toutes les clés candidates.

---