

**Cours de Bases de données            L3 informatique**  
**Examen de première session        Année 2012-2013**  
**Durée 2 heures et demie**  
**Seul document autorisé : 3 feuilles A4 recto-verso.**

*Les différentes parties de l'examen sont toutes indépendantes.*

## Partie 1 : requêtes SQL – un peu d'histoire

On considère une partie d'une base de données sur l'histoire. Cette base regroupe des événements historiques (règne, guerre, élection, indépendance,...), des personnages historiques qui y ont éventuellement pris part et des livres racontant ces événements.

Dans les tables suivantes, tous les attributs sont différents de `NULL`. Comme on peut s'y attendre, les identifiants sont de type `int`, les dates de type `date`, et les noms et titre sont de type `text`.

- **personnage(\*id\_pers, nom\_pers, d\_nais, d\_mort)** : la liste des personnages (identifiant, nom, date de naissance, date de mort) ;
- **evenement(\*id\_evn, nom\_evn, d\_debut, d\_fin)** : la liste des événements historiques (identifiant, nom, date de début, date de fin). Bien sûr, si l'événement a lieu un seul jour (par exemple, signature d'un accord), la date de début est égale à la date de fin ;
- **a\_pris\_part(\*id\_evn#, \*id\_pers#)** : indique à quel(s) événement(s) a pris part quel personnage ;
- **livre(\*id\_liv, tit\_liv, nb\_page)** : les livres d'histoires (identifiant, titre, nombre de pages (`int`)) ;
- **raconte(\*id\_liv#, \*id\_evn#)** : indique quel(s) livre(s) relatent quel(s) événement(s).

## Questions

Écrivez les requêtes SQL correspondant aux questions suivantes. On rappelle qu'on peut comparer les dates avec les opérateurs classiques `=`, `<`, ...

1. (assez facile) Les couples de personnages (`nom_per1`, `nom_pers2`) tels que le personnage 1 est né le jour de la mort du personnage 2 et qui ont pris part à un même événement ;
2. (moyen) pour chaque événement auquel a pris part au moins 5 personnages, la moyenne du nombre de pages des livres relatant cette événement (tableau résultat : (`nom de l'événement`, `moyenne`)) ;
3. (difficile) les personnages (leurs noms) ayant pris part à tous les événements dont ils sont contemporains. (Un personnage est dit contemporain à un événement s'il était vivant au moins un jour durant de cet événement.)

## Partie 2 : Algèbre relationnelle – magasin de chaussures

Une partie de la base de données d'un magasin de chaussures de sport comporte les tables suivantes :

- **modele(\*id\_mod, nom\_mod, prix)** : la liste des modèles de chaussures (identifiant, nom, prix)
- **sport(\*id\_sport, nom\_sport, genre)** : la liste des sports (identifiant, nom, genre) où genre peut être, par exemple, “glisse”, “raquette”, “combat”,...
- **adaptéA(\*id\_mod#, \*id\_sport#)** : indique à quel(s) sport(s) est adapté un modèle de chaussure
- **existeEn(\*id\_mod#, \*taille, \*couleur)** : indique les tailles et couleurs disponibles pour chaque modèle.

### Questions

Pour chaque question, donnez la requête en algèbre relationnelle qui y répond. Pour plus de lisibilité, vous pouvez faire la requête en plusieurs étapes, par exemple :

$$\begin{aligned}A &= \pi_{val}(R) \bowtie S \\B &= \sigma_{att=5}(T \cup R) \\réponse &= A \cap B\end{aligned}$$

Par ailleurs, si vous avez besoin de distinguer deux utilisations d'une même table, utilisez un numéro pour distinguer les deux utilisations de la table, par exemple, si on utilise deux fois la table  $T$  :

$$C = \sigma_{att=5}(T_1 \bowtie_{T_1.val <> T_2.val} T_2)$$

Donnez les requêtes algébriques correspondant aux ensembles suivants :

1. (facile) les modèles de chaussures (identifiant, nom) à moins de 100 € adapté à au moins un sport de combat (combat est un genre de sport).
2. (moyen) les modèles de chaussures (identifiant, nom) qui existent en deux couleurs différentes et qui conviennent pour la randonnée (nom de sport).
3. (difficile) Les sports pour lesquels tous les modèles sont adaptés.

## Partie 3 : Modélisation

Dans cet exercice, on vous demande une modélisation entité-relation sur le thème des magazines. Vous êtes libre de faire ce que vous voulez dans la mesure où vous respectez les faits suivants :

Pour chaque magazine on doit savoir son type (magazine de sport, économique, politique...). On suppose qu'un magazine n'a qu'un seul type. On doit aussi savoir quels journalistes écrivent régulièrement dans un magazine donné.

Si vous manquez d'idées pour compléter votre base, voici quelques suggestions :

- Donner le type des rubriques qu'il contient (rubrique santé, consommation, reportage,...);
- Donner le rédacteur en chef du magazine.

## Question 1

Donnez une modélisation entité-relation contenant au moins 4 tables et 3 relations. Vous respecterez la norme **crowfoot** (celle du cours). Vous justifierez aussi les choix qui pourraient ne pas être évidents.

## Question 2

Donnez ensuite une traduction en tables de votre modélisation.

# 1 Partie 4 : Contrôle de la concurrence

On a la base de données bancaire suivante :

**client**(\*no\_client, nom\_client) : table des clients ;  
**cheque**(\*no\_compte, no\_client#, montant) : table des compte-chèques ;  
**livret**(\*no\_compte, no\_client#, montant) : table des livrets d'épargne ;  
**pel**(\*no\_compte, no\_client#, montant) : table des comptes PEL ;

Un client qui a un compte-chèque et un livret a donc 2 numéros de compte éventuellement différents.

On considère les 3 transactions suivantes :

**Transaction 1** Martin vide son livret (compte numéro 1) et transfère cet argent sur son compte-chèque (compte numéro 100).

Pour cela, il fait les opérations suivantes :

1. `select montant from livret where no_compte = 1 ;`  
La valeur affichée est alors stockée dans une variable  $x$ .
2. `update livret set montant = 0 where no_compte = 1 ;`
3. `update cheque set montant = montant + x where no_compte = 100 ;` où  $x$  est la valeur récupérée en 1.

**Transaction 2** Jones verse 1500 € sur son PEL (compte numéro 200).

Pour cela, il fait l'opération suivante :

1. `update pel set montant = montant + 1500 where no_compte = 200 ;`

**Transaction 3** EDF rembourse un trop-perçu (100 €) en le versant sur le livret de Martin.

Pour cela, il fait l'opération suivante :

1. `update livret set montant = montant + 100 where no_compte = 1 ;`

## Questions

1. On suppose que ces trois transactions sont faites de manière quasi-simultanée. Donnez un exemple d'ordonancement qui provoque une anomalie. Expliquez brièvement en quoi c'est une anomalie. **Note :** On considère que chaque commande SQL est atomique, c'est-à-dire qu'elle se fait en une seule fois sans que d'autres opérations puissent s'intercaler.
2. Modélisez les transactions précédentes avec les opérations  $r_i(X)$  et  $w_i(X)$  (lecture et écriture dans la table  $X$  par la transaction  $T_i$ ).
3. On applique le verrouillage en 2 phases sur la transaction  $T_1$ . Indiquez le résultat. Notation :  $rl_i(X)$  (resp.  $wl_i(X)$ ) pour le verrouillage en lecture (resp. écriture) de la table  $X$  par la transaction  $T_i$  et  $ul_i(X)$  pour le déverrouillage.