

## Examen

### Bases de Données Avancées (BDA)

Tous documents papier autorisés pour la partie 1, fiche de synthèse au format A4 pour la partie 2. Téléphones et ordinateurs interdits. Le barème est donné à titre indicatif. La durée de l'examen est 2 heures. Les parties 1 & 2 doivent être **rédigées et rendues sur des copies séparées**. Ce sujet comporte 2 pages. Soignez vos réponses et votre orthographe.

#### **PARTIE 1:**

##### **Exercice 1: Modélisation** (6 points)

Un club de danse gère ses activités à l'aide d'une base de données. Les informations conservées sont organisées comme suit.

- Toute personne fréquentant le club doit être *membre*, et chaque membre est caractérisé par un *code unique*, un *nom*, un *prénom* et une *adresse*. Il y a deux types de membres, les *professeurs de danse*, pour lesquels on conserve également le *numéro de téléphone*, et les *membres ordinaires* devant payer une *cotisation annuelle*. Pour chaque membre ordinaire, on conserve la *date d'échéance* de la *cotisation courante*.
- Des *cours de danse* et des *soirées* sont organisés au sein du club. Pour chaque soirée, on conserve dans la base de données les *membres participant à la soirée*.
- Les cours sont caractérisés par le *nom de la danse enseignée*, le *niveau* et le *professeur*, ils consistent en un certain *nombre de séances* dont on précise la *date* et l'*heure*. Pour chaque cours, la *liste des membres inscrits* est conservée. Enfin, pour chaque séance on conserve la *liste des membres présents*. Il est à noter que deux séances du même cours ne peuvent être données en même temps, mais par contre, des séances de cours différents peuvent être prévues au même moment.

**On vous demande donc de :**

- Concevoir un diagramme ER qui capture ces informations. Indiquer les contraintes de clé et de participation (émettre des hypothèses si vous estimez que des détails vous manquent).
- Créer (en SQL) les tables correspondantes en capturant le plus de contraintes possibles. Quand ce n'est pas possible, expliquer pourquoi.

## **Exercice 2: Questions de cours** (4 points)

Répondez aux questions suivantes de manière claire et détaillée :

1. Quelle est la différence entre un bloc (*block*) et une piste (*track*) de disque? Quel est l'impact de la taille du bloc sur les performances d'accès aux données?
2. Citez un mécanisme permettant d'assurer la fiabilité du stockage des données et un mécanisme permettant d'assurer la fiabilité de la manipulation des données.
3. Pourquoi est-il nécessaire d'optimiser les requêtes exécutées sur une base de données?
4. Qu'est-ce que JDBC? Quels sont les avantages de l'utilisation de JDBC face à l'utilisation des API propriétaires des SGBD?

## **PARTIE 2:**

### **Exercice 1 : XML schemas** (2 points)

Question 1.1 : Définissez une DTD qui valide tous les arbres dans l'ensemble suivant et aucun autre:

$$\{ac, a(b)c, a(b, b)c, a(b, b, b)c, \dots\}$$

Il suffit de donner les expressions régulières pour les lettres  $a$ ,  $b$ , et  $c$ . Il n'est pas nécessaire d'élaborer la syntaxe concrète de cette DTD.

### **Exercice 2 : Navigation dans les bases de données XML** (2 points)

Définissez la requête dans les arbres qui sélectionne tous les noeuds qui sont *petit-fils*:

Question 2.1 : par une formule logique du premier ordre.

Question 2.2 : par un chemin de Core XPath 1.0.

### **Exercice 3 : Implication en FO et XPath** (2 points)

Question 3.1 : Soient  $e$  et  $e'$  des formules logiques du premier ordre. Définissez l'implication  $e \rightarrow e'$  par une formule logique du premier ordre avec disjonction, conjonction et/ou négation, mais sans implication.

Question 3.2 : Définissez en Core XPath 1.0 une requête dans les arbres, qui sélectionne tous les noeuds, qui ont un fils étiqueté par  $x$  ayant lui-même un fils étiqueté par  $y$ .

### **Exercice 4 : Quantification en FO et XPath** (2 points)

Question 4.1 : Définissez par une formule logique du premier ordre une requête dans les arbres qui sélectionne tous les noeuds dont tout ascendant est étiqueté par  $a$ .

Question 4.2 : Définissez la même requête en Core XPath 1.0

### **Exercice 5 : Transformations en XQuery** (2 points)

Définissez une transformation en Core XQuery 1.0, qui efface tous les livres dont l'auteur est 'hull' ou 'gray' d'une bibliographie du type suivant:

