



**EXAMEN SESSION 1 – HAI502I**  
**Système d'Information et Base de Données 2**

Session : 1	Durée de l'épreuve : 2 heures
Date : 13-janvier-2023	Documents autorisés : tous
Mention Informatique	Matériel utilisé : aucun
Licence 3ème année : SIBD2 (HAI502I)	

**Remarques :** Il y a 4 exercices, le barème est donné à titre indicatif.

**Exercice 1 : Entité-Association (5 points)**

Considérez le schéma relationnel suivant utilisé par une mairie pour gérer ses casernes de pompiers avec leur personnel (des pompiers), leurs matériels (des camions) et leurs interventions :

CASERNE (**id\_caserne**, capaCamions, capaPompiers, #num\_rue, #nom\_rue, #CP, #nom\_ville)  
CAMION (**id\_camion**, #id\_caserne, nbPlaces)  
CITERNE (**#id\_camion**, contenance)  
VILLE (**nom\_ville**, **CP**, nbHabitants)  
POMPIER (**id\_pompier**, #id\_caserne, nom, prenom, #num\_rue, #nom\_rue, #CP, #nom\_ville)  
ADRESSE (**num\_rue**, **nom\_rue**, **#nom\_ville**, **#CP**, typeHabitation, #id\_proche\_caserne, km)  
PROTEGE (**#idcaserne**, **#nom\_ville**, **#CP**)  
INTERVENTION (**idIntervention**, #num\_rue, #nom\_rue, #CP, #nom\_ville, typeIntervention, dateIntervention, dureeIntervention)  
INTERVENTION\_POMPIER (**#idIntervention**, **#idPompier**)  
INTERVENTION\_CAMION (**#idIntervention**, **#idCamion**)

Les clés primaires sont en gras souligné et les clés étrangères sont précédées du caractère #.

La relation CASERNE contient un identifiant par caserne, avec son adresse (numéro et nom de rue, code postal et ville), sa capacité maximale humaine et sa capacité maximale de stockage de camions.

La relation CAMION contient, pour chaque camion, un identifiant attribué par la caserne où il est stocké, ainsi qu'un nombre de place maximum et un nom de modèle.

La relation CITERNE contient des camions particuliers, appelés citernes, pour lesquelles nous connaissons également la contenance.

La relation POMPIER contient, pour chaque pompier, son identifiant attribué par la caserne où il est affecté, ainsi qu'un nom et une adresse (numéro et nom de rue, code postal et ville). Les adresses constituées d'un numéro de rue, un nom de rue, un code postal, un nom de ville, du type d'habitation correspondant à l'adresse (e.g. une ferme, un château, une cité, etc.), d'une estimation du kilométrage entre cette adresse et la plus proche caserne sont stockées dans la relation ADRESSE.




Les villes sont stockées dans la relation VILLE qui contient en plus du nom et du code postal, son nombre d'habitants.

La relation PROTEGE contient les informations concernant la protection des villes par les casernes.

Les relations INTERVENTION, INTERVENTION\_POMPIER et INTERVENTION\_CAMION contiennent des informations permettant de connaître l'identifiant de l'intervention, son type d'intervention, sa date, sa durée, l'adresse où elle s'est effectuée, ainsi que les camions et pompiers impliqués dans cette intervention.

**Question :** Écrire le modèle Entité-Association correspondant au schéma relationnel

### **Exercice 2 : Dépendances fonctionnelles et normalisation (7 points)**

Soit la relation R (A, B, C, D, E, ) où chaque attribut ne peut avoir qu'une seule valeur et l'ensemble de dépendances fonctionnelles suivant

$$F = \{ A \rightarrow C ; B \rightarrow D ; AD \rightarrow E ; AC \rightarrow D ; CD \rightarrow E \}.$$

**Question 1 :** Quelle est la couverture minimale de DF ? Justifier votre réponse.

**Question 2 :** Quelles sont la ou les clés candidates ? Justifier votre réponse.

**Question 3 :** En quelle forme normale est la relation R ? Justifier votre réponse.

**Question 4 :** Proposer une décomposition telles que toutes les relations soient en 3NF et qu'il n'y ait pas de perte d'information. Justifier votre réponse.

**Question 5 :** Vérifier avec l'algorithme de Chase (méthode des alphas) que la décomposition précédente ne perd pas d'information

### **Exercice 3 : SQL (6 points)**

Vous disposez d'une base de données d'une plateforme de vente d'occasion en ligne où les clients achètent des produits (à une date et un certain prix) qu'ils peuvent revendre à d'autres clients par la suite (à une autre date et à un autre prix). Le modèle relationnel de cette base de données est le suivant :

Produit (**idProduit**, désignation)

Client (**idClient**, nom, prénom)

Vente (**#idProduit**, **#idClient**, prix, dateVente)

**Question 1 :** Écrire en SQL la requête suivante : « quel est le ou les produits (identifiant, la désignation) qui ont été vendus le plus cher ? »



**Question 2 :** Écrire en SQL la requête suivante : « pour chaque produit, quel est le ou les clients qui ont acheté ce produit le plus cher ? (indiquer le nom, prénom du ou des clients, ainsi que l'identifiant et la désignation de chaque produit) »

**Question 3 :** Écrire en SQL la requête suivante : « quel est le ou les produits (identifiant et désignation) qui ont été achetés le plus de fois . »

En algèbre relationnel, la division peut s'exprimer de la manière suivante. Soient  $R(X,Y)$  et  $S(Y)$ , la division de  $R$  par  $S$  est :  $R \div S = \pi_{[X]} R - \pi_{[X]} ( ( \pi_{[X]} R ) \otimes S ) - R )$  où  $\pi_{[X]} R$  correspond à la projection de l'attribut  $X$  pour la relation  $R$  ;  $-$  correspond à la différence entre 2 relations et  $\otimes$  correspond au produit cartésien.

L'interprétation est la suivante :

- $\pi_{[X]} R$  retourne tous les  $X$  de  $R$  ;
- $(( \pi_{[X]} R ) \otimes S )$  retourne le produit cartésien de « tous les  $X$  de  $R$  par  $S$  » donc toutes les combinaisons possibles de  $X$  avec les  $Y$  (aussi bien les tuples qui ont une valeur dans  $R$  que les autres) ;
- $(( ( \pi_{[X]} R ) \otimes S ) - R )$  supprime de toutes les combinaisons possibles celles qui sont dans  $R$ , en d'autres termes on supprime ici tous les valeurs qui sont dans  $R$  (donc celles recherchées) ;
- $\pi_{[X]} ( ( ( \pi_{[X]} R ) \otimes S ) - R )$  permet de projeter le résultat sur  $X$  afin pouvoir faire la différence finale ;
- la différence  $\pi_{[X]} R - \pi_{[X]} ( ( ( \pi_{[X]} R ) \otimes S ) - R )$  permet donc de supprimer de la partie gauche tous les tuples qui ne sont pas dans  $R$ .

**Question 4 :** Écrire en algèbre relationnel puis en SQL la requête suivante : « quels sont les produits qui ont été achetés par tous les clients ? » en utilisant l'expression de la division algébrique (indiquer les identifiants des produits uniquement). En d'autres termes vous ne devez pas utiliser de double « not exists » ni de « group by » mais vous devez traduire en SQL la division telle qu'elle est exprimée en algèbre relationnel.

Aide : vous pouvez utiliser des vues pour éviter d'avoir une requête trop complexe.

#### **Exercice 4 : Trigger (2 points)**

Considérer le schéma relationnel simplifié suivant :

Equipes (**idEqu**, Nom)

Athletes (**IdAth**, Nom, Pays)

Membres (**#IdEqu**, **#IdAth**)

Les clés primaires sont en gras souligné et les clés étrangères sont précédées du caractère #.





Considérer également le trigger suivant :

```
CREATE OR REPLACE TRIGGER NOM_A_TROUVER
BEFORE INSERT ON Membres
FOR EACH ROW
DECLARE
    message EXCEPTION;
    I_count INTEGER;
BEGIN
    SELECT COUNT (*) INTO I_count
    FROM MEMBRES, ATHLETES A1, ATHLETES A2
    WHERE MEMBRES.IdEqu=:NEW.IdEqu
        AND MEMBRES.IdAth = A1.IdAth
        AND :NEW.IdAth = A2.IdAth
        AND A1.Pays <> A2.pays;
    IF I_count <> 0 THEN RAISE message;
END IF;
EXCEPTION
WHEN message THEN RAISE_APPLICATION_ERROR(-20324,' MESSAGE A TROUVER');
END;
/
```

**Question :** Expliquer le rôle et le fonctionnement de ce trigger. Proposer un nom de trigger à la place de NOM\_A\_TROUVER ainsi que le message à afficher à la place de 'MESSAGE A TROUVER' qui soit significatif du rôle du trigger.