

Médian AI23 P2023

- Tous les documents papier sont autorisés.
- Aucun appareil connecté n'est autorisé (ordinateur, téléphone, montre connectée, etc.).
- Les réponses peuvent être données en anglais pour les étudiants étrangers (le préciser sur la copie).
- Le nom doit être écrit lisiblement sur chaque copie.
- La durée totale de l'examen est de 2h.

Barème :

- Ex 1 - Base de données d'une entreprise de construction [50min]
Q1 : 4 pts, Q2 : 4 pts
- Ex 2 - Base de données d'associations étudiantes [70min]
Q1 : 1,5 pt, Q2 : 2pt, Q3 : 2pt, Q4 : 2pt, Q5 : 2pt, Q6 : 1pt, Q7 : 1,5

1. Base de données de magasins de meubles

L'entreprise de meubles AKEI veut enregistrer ses produits, ses magasins et ses ventes dans une base de données relationnelle.

Un modèle de meuble AKEI peut être une étagère, un sommier, un fauteuil ou un bureau. Tous les modèles de meubles ont un nom unique, un prix, et trois dimensions (longueur, largeur et hauteur en mètre). Les modèles de sommiers ont également un nombre de place, les modèles de fauteuil peuvent être fixes ou à roulette, et les modèles d'étagères ont un nombre de casiers. Certains modèles d'étagères peuvent être montés au-dessus d'autres modèles d'étagères pour permettre d'avoir un meuble personnalisé ; on veut garder cette information dans la base de données.

Chaque modèle de meuble est monté par l'acheteur avec plusieurs composants réutilisés dans différents modèles. Un composant peut être une planche, un composant visserie, ou un composant divers. Chaque composant a un code et un numéro, les deux ensemble étant unique pour chaque composant, et un nom. Les planches ont des dimensions en mètre (longueur, largeur, hauteur) et un matériau (bois, contreplaqué, plastique ou verre). Les composants visserie ont une longueur et un matériau (métal, plastique ou bois). Les composants divers ont une description textuelle. On veut bien sûr savoir dans la base de données quels sont les composants de chaque modèle de meuble, en sachant qu'un modèle de meuble a au moins une planche ; attention, un meuble peut avoir plusieurs composants de même type (par exemple, le meuble 'bureau confort' a deux composants visserie de code 'WDPLG' et de numéro 432).

L'entreprise AKEI a des magasins, chacun ayant une adresse unique, un numéro de téléphone unique, un nom et une superficie. On veut enregistrer les achats réalisés dans chaque magasin : les modèles de meuble achetés et en quelle quantité, la date et l'heure d'achat, le mode de paiement (espèce, chèque ou carte bancaire), et optionnellement un identifiant de chèque cadeau si le client en a utilisé un. On veut également enregistrer les informations des clients qui ont acheté des meubles : leur nom, prénom, adresse, numéro de téléphone et mail (ces deux dernières informations étant facultatives), et bien sûr quels achats ils ont réalisés, un client ayant obligatoirement réalisé au moins un achat.

Enfin, on veut enregistrer les employés de chaque magasin. Un employé a un identifiant unique, un nom, un prénom, une adresse et un numéro de téléphone unique. On veut savoir quel est le poste des employés et dans quel magasin, en sachant qu'un employé donné a un poste dans un magasin de l'entreprise.

Question 1

Proposez un MCD en UML pour modéliser la base de données à développer. Pensez à bien représenter toutes les contraintes existantes. Posez des hypothèses si des informations semblent vous manquer dans le sujet.

Question 2

Proposez une transformation de votre MCD en MLD relationnel, sans perte d'informations. Justifiez les transformations d'héritage que vous avez utilisées, s'il y en a.

2. Bases de données de guilde d'aventurier

La guilde d'aventurier de Minas Tirith veut faire une base de données relationnelles des quêtes proposées à ses membres. Elle considère le schéma relationnel suivant, dont tous les attributs sont NOT NULL :

Types_monstres (#type : string)

Monstres (#nom : string, difficulté : enum{'A', 'B', 'C', 'D'}, experience : int, type =>types_monstres.type)

Ingrédients (#nom : string, type : enum{'plante', 'mineral'})

Quêtes (#client : str, #date : timestamp, durée : int, récompense : int, difficulté: enum{'A', 'B', 'C', 'D'}) avec {recompense >=0 and durée > 0}

Quêtes_monstres (#client =>Quêtes.client, #date=>Quetes.date, monstre => #Monstres.nom, nombre : int) avec {nombre > 0}

Quêtes_ingrédients (#client =>Quêtes.client, #date=>Quetes.date, #ingrédient => Ingrédients.nom, nombre : int) avec {nombre > 0}

Question 1

Rétro-concevez un modèle conceptuel possible ayant donné ce modèle relationnel.

Question 2

Donnez en SQL les instructions nécessaires pour créer les tables des trois relations Quêtes, Ingrédients et Quêtes_ingrédients.

Question 3

Donnez **en algèbre relationnel et en SQL** le client et la date de création des quêtes qui demande de tuer au moins 5 monstres goblins.

Question 4

Donnez **en algèbre relationnel et en SQL** une requête permettant de connaître sans doublons les clients des quêtes qui demandent à la fois de tuer des créatures de type 'mort-vivant' et de récolter des herbes de soin.

Question 5

Donnez **en algèbre relationnel et en SQL** une requête permettant de connaître les clients et les récompenses des quêtes qui ne demandent pas de récolter des minéraux.

Question 6

Donnez **en algèbre relationnel seulement** une requête permettant de connaître les types de monstre qu'on demande de tuer dans toutes les quêtes dont le client est Elrond.

Question 7

Ecrivez un script python permettant d'afficher toutes les quêtes demandées par un client. Le nom du client sera entré par l'utilisateur.