Final NF18/AI23 P2021

- 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 1h30 (2h pour les tiers-temps).
- **Répondez sur deux copies séparées : copie 1 pour les exercices 1 et 2, copie 2 pour les exercices 3 et 4 !** Vous pourrez être pénalisés si vous ne respectez pas cette consigne.

Barème :

- Ex 1 - Base de données des patients d'un hôpital [30min]
  Question 1 : 3.5 pts
  Question 2 : 3 pts
- Ex 2 - Requêtes de base de données relationnelles [15min]
  Question 1 : 1.5 pt
  Question 2 : 2 pt
- Ex 3 - Base de données non-relationnelles avec mongo [20min]
  Question 1 : 3 pts
  Question 2 : 2 pts
- Ex4 - Normalisation fonctionnelle [25mn]
  Question 1 : 2 pts
  Question 2 : 1.5 pts
  Question 3 : 1.5 pt

A. Base de données des patients d'un hôpital

Contexte

Un hôpital veut gérer les données de ses patients en rapport avec l'épidémie du virus Bovid-19.

Un patient peut être identifié par son nom, son prénom et sa date de naissance, ou par son numéro de sécurité sociale. L'hôpital veut pouvoir renseigner les liens de cas contacts unissant deux patients à une date et une heure de contact afin de faciliter la détection du virus.

Des données médicales des patients sont régulièrement enregistrées : chaque enregistrement est identifié pour un patient à un temps (date et heure) donné, et comporte les informations de pouls, pression sanguine et température du patient.

L'hôpital veut savoir si un patient est vacciné ou non. S'il est vacciné, on stockera la date de vaccination et le type de vaccin donné.

Les patients séjournent dans des chambres, qui ont un numéro et un service. Chaque chambre peut accueillir un nombre maximum de patients. L'hôpital veut garder en mémoire quel patient a séjourné dans quel chambre et quand. Un patient peut avoir séjourné dans plusieurs chambres à l'hôpital mais aussi plusieurs fois dans la même chambre.

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 utilisez, le cas échéant.

B. Requêtes de base de données relationnelles

Contexte

Le diabolique directeur de l'agence du mal F.A.N.T.Ô.M.E. a enregistré la liste de ses armes de destruction dans une base de données SQL suivant le schéma relationnel ci-dessous.

Notez que tous les attributs seront considérés NOT NULL par défaut, et qu'un véhicule peut être équipé de différentes armes :

Munitions (#designation : string, type : enum{laser, balle, radiation}, force : integer) avec {force >0}

Armes (#nom : string, tirs_par_seconde : float, munition => Munitions.designation) avec {tirs par seconde >0}

Véhicules (#nom : string, #armes => Armes.nom, nombre : integer) avec (nombre >0)

Question 1

Donnez en algèbre relationnel et en SQL une requête permettant de connaître les véhicules qui ne sont pas équipés d'armes ayant des tirs de type 'radiation'.

Question 2

Proposer un programme python permettant de se connecter à une base de données, puis d'afficher pour chaque véhicule qui a au moins trois armes son nombre total de tirs par seconde.

Indice :

Le programme pourra avoir la forme globale suivante. Vous pouvez juste écrire les 4 blocs A, B, C et D sur votre copie sans réécrire le reste.

```
#!/usr/bin/python3
import psycopg2
# Connection à la base de données
HOST = "examen.nf18.fr"
USER = "moi"
PASSWORD = "secret"
DATABASE = "mydb"
```

A : ...

```
#Exécution de la requête
B : ...
#Traitement du résultat
C : ...
# Fermeture de la connexion
D : ...
```

C. Base de données non-relationnelles avec mongo

Contexte

Une agence de jeux de société veut enregistrer ses produits dans une base de données. Un jeu de société a un nom, un descriptif, une année d'édition, un nombre minimum et maximum de joueurs, un âge minimum, un type (puzzle, enquête, aventure, développement) et des éléments de contenu, qui peuvent être des cartes, des éléments de plateau et/ou des pièces.

Pour les cartes, on veut stocker pour chaque carte son nom, le nombre de cartes identiques dans le jeu et une description.

Pour les éléments de plateau, on veut stocker pour chaque élément son nom, sa description et le nombre d'éléments similaire dans le jeu.

Pour les pièces, on veut stocker pour chacune son nom et le nombre de pièce similaire dans le jeu.

Chaque jeu peut avoir des extensions, qui ont toutes un nom, un descriptif, une année d'édition et des éléments de contenus similaires aux jeux de base.

Question 1

Proposez une implémentation sous MongoDB, pour laquelle vous insérerez dans une collection 'BoardGames' les deux jeux suivants :

- Un jeu de Go, pour 2 joueurs à partir de 8 ans, de type stratégie, avec les éléments de contenu suivants : aucune carte, un unique plateau principal, 90 pions blancs et 90 pions noirs. Sans extension.
- Le jeu Aliens vs Space Marines, édité en 2000, pour 2 à 4 joueurs à partir de 12 ans, de type stratégie, avec les éléments de contenu suivants : aucune carte, trois uniques plateaux de jeu (vaisseau spatial, ruche alien, colonie abandonnée), 1 pion capitaine Space Marine, 2 pions porteurs de Lance-flamme Space Marine, 4 pions soldats Space Marine, 20 pions aliens, 2 dés à six faces.

Vous pouvez inventer les données qui ne sont pas précisées.

Question 2

Proposez une réponse en MongoDB aux requêtes suivantes :

- retournez le nom des jeux de plateau de type stratégie qui peuvent être joué à 4 joueurs
- retournez le nom et l'année d'édition des jeux de plateau qui ont une extension sortie après l'année 2020.

Indice :

> *On pourra utiliser l'objet {$gt:value} à la place de la valeur de l'attribut à tester pour 'strictement plus grand que' ($gt signifie 'greater than') et l'objet {$lt:value} à la place de la valeur de l'attribut à tester pour 'strictement plus petit que' ($lt signifie 'lesser than').*

D. Normalisation fonctionnelle

Contexte

Un département veut mettre en place une base de données sur les festivals qui ont lieu dans ses villes. Il a identifié les attributs suivants : le nom du festival, la date du festival, la ville où a lieu le festival, une description du festival, un type de manifestation, l'identifiant du responsable auprès de la ville pour l'organisation du festival (un responsable pouvant couvrir plusieurs villes), un correspondant de communication dépendant du type de festival (un correspondant ne couvrant qu'un seul type de festival).

A partir de ces attributs, les dépendances fonctionnelles suivantes ont été identifiées :

- nom → description, type, correspondant
- ville →responsable
- nom, date → ville
- ville, date → nom, description
- type → correspondant
- correspondant → type

Question 1

Donnez la fermeture transitive, les couvertures minimales et les clés de la relation contenant l'ensemble des attributs identifiés. Justifiez vos réponses.

Question 2

Quelle est la forme normale de cette relation ? Justifiez.

Question 3

Proposez un découpage en 3NF sans perte d'informations.