

DS 1

Informatique de tronc commun, classe de PC

Julien REICHERT

Ce devoir consiste en une partie de programmation et deux parties sur des bases de données totalement indépendantes. Les fonctions et requêtes à écrire ne sont pas forcément de difficulté croissante.

Partie 1 : Programmation

Toutes les questions de cette section sont à traiter en Python et sont indépendantes sauf **1.2** et **1.3**.

Exercice 1.1 : Écrire une fonction prenant en argument une liste et envoyant le premier élément à la fin, en décalant tous les autres. Il s'agit effectivement de muter la liste en argument.

Exercice 1.2 : Écrire une fonction prenant en argument une liste croissante d'entiers (de taille supposée au moins deux) et renvoyant la plus grande différence entre deux éléments consécutifs de la liste. Il n'est pas nécessaire de vérifier que la liste respecte les propriétés énoncées (si la liste n'est pas croissante ou pas de taille au moins deux, le comportement est au choix, en pratique).

Exercice 1.3 : Même exercice mais il faut renvoyer la plus petite différence parmi les différences non nulles. Si la liste est constante, on interrompra le programme avec `assert False`.

Exercice 1.4 : Écrire une fonction prenant en argument une liste de couples (vues comme des listes étant toutes de taille deux) et déterminant si à chaque indice noté i et supérieur ou égal à 1 la condition suivante s'applique : aucun couple à un indice strictement inférieur à i n'a tous ses deux éléments supérieurs ou égaux aux éléments de même indice dans le couple à l'indice i .

Exercice 1.5 : Écrire une fonction prenant en argument un entier et renvoyant son nombre de chiffres. Toute méthode qui fonctionne est autorisée, à condition de respecter la syntaxe et le typage de Python.

Partie 2 : Base de données « e-commerce »

On considère la base de données suivante :

- **Clients**, avec comme attributs `id_client` (entier et clé primaire), `identite` (chaîne de caractères), `email` (chaîne de caractères), `adresse` (chaîne de caractères) et `telephone` (chaîne de caractères).
- **Produits**, avec comme attributs `id_produit` (entier et clé primaire), `nom_produit` (chaîne de caractères), `categorie` (chaîne de caractères), `stock` (entier) et `description` (texte potentiellement long).
- **Commandes**, avec comme attributs `id_commande` (entier et clé primaire), `id_client` (entier), `id_produit` (entier) et `quantite` (entier).

Exercice 2.1 : Préciser les clés étrangères.

Exercice 2.2 : Justifier par le modèle entité-relation le schéma retenu pour la base de données.

Exercice 2.3 : Écrire une requête permettant de déterminer tous les noms de produits dont le nombre d'exemplaires en stock est inférieur ou égal à cinq.

Exercice 2.4 : Écrire une requête permettant de déterminer toutes les informations sur les commandes passées par un client dont l'identité est précisée (on écrira `id` sans guillemets pour l'identité, comme si c'était une variable).

Exercice 2.5 : Écrire une requête permettant de déterminer toutes les informations sur les commandes passées pour un produit en commandant une quantité déjà strictement supérieure au stock.

Exercice 2.6 : Écrire une requête permettant de déterminer, pour la catégorie de produits dont le stock total est maximal, quel est ce stock maximal. Inutile de faire apparaître la catégorie.

Exercice 2.7 : Écrire une requête permettant de déterminer tous les identifiants de produits dont la quantité totale des commandes est strictement supérieure au stock.

Exercice 2.8 : Écrire une requête permettant de déterminer tous les identifiants de produits que tous les clients ont commandés (il ne s'agit en particulier pas de déterminer l'intégralité des produits qui ont été commandés au moins une fois).

Partie 3 : Base de données « recettes »

On considère la base de données suivante, permettant de partager des recettes entre internautes. Les tables sont les suivantes, avec des attributs intuitifs :

- UTILISATEURS, avec les attributs `idutil` (entier, clé primaire) et `identite` (chaîne de caractères) ;
- INGREDIENTS, avec les attributs `idingr` (entier, clé primaire), `ingredient` (chaîne de caractères) et `categorie` (chaîne de caractères) ;
- ALLERGENES, avec les attributs `idallerg` (entier, clé primaire) et `allergene` (chaîne de caractères) ;
- INGRALL, avec les attributs `ingredient` (entier) et `allergene` (entier) ;
- RECETTES, avec les attributs `idrec` (entier, clé primaire), `nom` (chaîne de caractères), `contributeur` (entier), `plat` (chaîne de caractères) et `nbpers` (entier) ;
- INGREC, avec les attributs `recette` (entier), `ingredient` (entier) et `quantite` (flottant) ;
- EVALUATIONS, avec les attributs `utilisateur` (entier), `recette` (entier) et `note` (entier) ;

Quelques explications :

- L'attribut `plat` de la table RECETTES précise si la recette est recommandée pour une entrée, un plat principal, un dessert, etc. (le nom du plat est l'attribut `nom`).
- L'attribut `nbpers` de la table RECETTES précise le nombre de personnes pour les quantités précisées dans la table INGREC.
- Des plats de mêmes noms mais d'identifiants différents seront séparés dans tous les exercices.
- Les quantités sont exprimées en grammes pour le poids, en litres pour le volume, en unités sinon.

Exercice 3.1 : Préciser les clés étrangères.

Exercice 3.2 : Justifier par le modèle entité-relation le schéma retenu pour la base de données en précisant la nature des relations. On pourra en profiter pour justifier l'absence d'une table UTILREC par exemple.

Exercice 3.3 : Écrire une requête permettant de déterminer la liste des noms de plats dont un utilisateur dont on fournit l'identité a publié la recette (on écrira `id` sans guillemets pour l'identité, comme si c'était une variable).

Exercice 3.4 : Écrire une requête permettant de déterminer la liste des noms d'allergènes utilisés au moins une fois par la recette du cheesecake au citron.

Exercice 3.5 : Écrire une requête permettant de déterminer la liste des noms de plats qui utilisent œufs et beurre.

Exercice 3.6 : Écrire une requête permettant de déterminer le nom de plat dont la recette a obtenu la meilleure note moyenne (en cas d'égalité, on accepte de ne faire figurer qu'un résultat).

Exercice 3.7 : Écrire une requête permettant de déterminer l'identité des trois internautes ayant noté le plus de recettes (en cas d'égalité, on ne fera figurer que trois résultats sans essayer d'en faire figurer plus).

Exercice 3.8 : Écrire une requête permettant de déterminer la liste des recettes qu'une personne allergique aux fruits à coques et aux crustacés peut manger.

Exercice 3.9 : Écrire une requête permettant de déterminer le nombre de personnes qu'on peut inviter à manger des crêpes si on dispose d'un kilogramme de farine et des autres ingrédients en quantité largement suffisante.