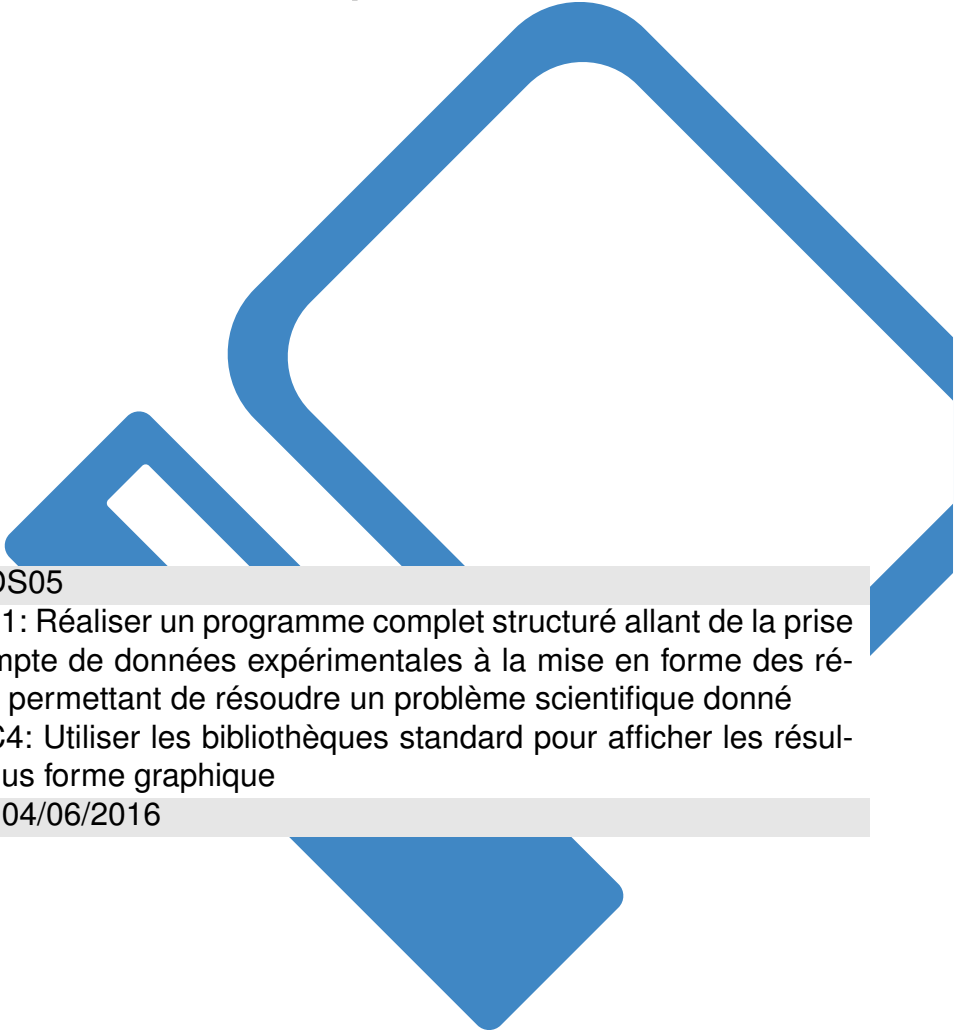


DS Informatique



Référence	S04- DS05
Compétences	Ing - C1: Réaliser un programme complet structuré allant de la prise en compte de données expérimentales à la mise en forme des résultats permettant de résoudre un problème scientifique donné Ing - C4: Utiliser les bibliothèques standard pour afficher les résultats sous forme graphique
Description	Fait le 04/06/2016

Les 3 parties sont à rendre sur des copies séparées.

1 Application nutritionnelle

1.1 Description de la base

L'objectif de cet exercice est de déterminer les requêtes SQL nécessaires à la réalisation d'une application permettant de déterminer les apports nutritionnels de recettes.

Pour expliquer la signification des nutriments dans les aliments, nous avons pris l'exemple suivant. La suite présente les quantités des principaux nutriments présents dans 100g de pomme (attention aux unités) :

- Protéines (en g) : 0.28,
- Lipides (en g) : 0.42,
- Glucides (en g) : 14.41,
- Calories (en kcal) : 56,
- Eau (en g) : 85.63,
- Sucres totaux (en g) : 10.39.

Pour cela nous allons utiliser la base appelée `food`.

Cette base contient 4 tables :

- **foodcats** : contient des catégories d'aliments,
 - **fdgrp_cd** : code de la catégorie,
 - **fdgrp_desc** : nom de la catégorie,
 - **id** : id de la catégorie.
- **fooddescs** : contient les descriptions des aliments,
 - **ndb_no** : code de l'aliment,
 - **fdgrp_cd** : code de sa catégorie,
 - **long_desc** : nom de l'aliment,
 - **id** : id de l'aliment.
- **nutrientdefs** : contient des informations sur les nutriments,
 - **nutr_no** : code du nutriment,
 - **units** : unité du nutriment,
 - **nutr_desc** : nom du nutriment,
 - **id** : id du nutriment.
- **nutrientdata** : indique quels nutriments (et les quantités) sont présents dans les aliments.
 - **ndb_no** : code de l'aliment,
 - **nutr_no** : code du nutriment,
 - **nutr_val** : quantité de nutriment présent dans l'aliment en unité du nutriment pour 100g d'aliment,
 - **id** : id du couple nutriment/aliment.

Question 1 : Proposer un schéma de cette base de données.

Question 2 : Écrire la requête qui permet de lister l'ensemble du contenu de la table `foodcats`.

Le résultat de la requête est le suivant.

```
(0100, 'Dairy and Egg Products', 1)
(0200, 'Spices and Herbs', 2)
(0300, 'Baby Foods', 3)
(0400, 'Fats and Oils', 4)
(0500, 'Poultry Products', 5)
(0600, 'Soups, Sauces, and Gravies', 6)
(0700, 'Sausages and Luncheon Meats', 7)
(0800, 'Breakfast Cereals', 8)
(0900, 'Fruits and Fruit Juices', 9)
(1000, 'Pork Products', 10)
(1100, 'Vegetables and Vegetable Products', 11)
(1200, 'Nut and Seed Products', 12)
(1300, 'Beef Products', 13)
(1400, 'Beverages', 14)
(1500, 'Finfish and Shellfish Products', 15)
(1600, 'Legumes and Legume Products', 16)
(1700, 'Lamb, Veal, and Game Products', 17)
(1800, 'Baked Products', 18)
(1900, 'Sweets', 19)
(2000, 'Cereal Grains and Pasta', 20)
(2100, 'Fast Foods', 21)
(2200, 'Meals, Entrees, and Sidedishes', 22)
(2500, 'Snacks', 23)
(3500, 'Ethnic Foods', 24)
```

Question 3 : Écrire la requête qui permet de lister l'ensemble du contenu de la table `fooddescs` pour les aliments appartenant à la catégorie des fruits ou des jus de fruits et dont le code `ndb_no` est strictement inférieur à 09016.

Le résultat de la requête est le suivant.

```
(09001, 0900, 'Acerola, (west indian cherry), raw', 2067)
(09002, 0900, 'Acerola juice, raw', 2068)
(09003, 0900, 'Apples, raw, with skin', 2069)
(09004, 0900, 'Apples, raw, without skin', 2070)
(09005, 0900, 'Apples, raw, without skin, cooked, boiled', 2071)
(09006, 0900, 'Apples, raw, without skin, cooked, microwave', 2072)
(09007, 0900, 'Apples, canned, sweetened, sliced, drained, unheated', 2073)
(09008, 0900, 'Apples, canned, sweetened, sliced, drained, heated', 2074)
(09009, 0900, 'Apples, dehydrated (low moisture), sulfured, uncooked', 2075)
(09010, 0900, 'Apples, dehydrated (low moisture), sulfured, stewed', 2076)
(09011, 0900, 'Apples, dried, sulfured, uncooked', 2077)
(09012, 0900, 'Apples, dried, sulfured, stewed, without added sugar', 2078)
(09013, 0900, 'Apples, dried, sulfured, stewed, with added sugar', 2079)
(09014, 0900, 'Apples, frozen, unsweetened, unheated', 2080)
(09015, 0900, 'Apples, frozen, unsweetened, heated', 2081)
```

Question 4 : Écrire la requête qui permet de lister le contenu des colonnes `nutr_no`, `units`, `nutrdesc` de la table `nutrientdefs` pour les nutriments ayant un `id` est strictement inférieur à 7.

Le résultat de la requête est le suivant.

```
(203, 'g', 'Protein')
(204, 'g', 'Total lipid (fat)')
(205, 'g', 'Carbohydrate, by difference')
(207, 'g', 'Ash')
(208, 'kcal', 'Energy')
(209, 'g', 'Starch')
```

Question 5 : Écrire la requête qui permet de lister le contenu des colonnes `ndb_no`, `nutr_no`, `nutr_val` de la table `nutrientdata` pour les pommes crues avec ou sans peau (au cas où cela serait nécessaire (on sait jamais) il peut être nécessaire de rappeler que *crue* se dit *raw* et *peau* se dit *skin*). On se limitera aux pommes 09003 et 09004. Il faudra aussi se limiter aux nutriments dont l'`id` est strictement inférieur à 7 (il sera possible d'utiliser le résultat de la requête précédente afin de filtrer en utilisant le code du nutriment).

Le résultat de la requête est le suivant.

```
(09003, 203, 0.26)
(09003, 204, 0.17)
(09003, 205, 13.81)
(09003, 207, 0.19)
(09003, 208, 52.0)
(09003, 209, 0.05)
(09004, 203, 0.27)
(09004, 204, 0.13)
(09004, 205, 12.76)
(09004, 207, 0.17)
(09004, 208, 48.0)
```

1.2 Requêtes sur un aliment

Question 6 : Écrire la requête qui permet de lister le contenu des colonnes `ndb_no`, `fdgrp_cd`, `long_desc` de la table `fooddescs` pour toutes les pommes (vous utiliserez la fonction `LIKE` (vue en TP) qui permet de rechercher une chaîne de caractère dans un attribut).

Le résultat de la requête est le suivant.

```
(09008, 0900, 'Apples, canned, sweetened, sliced, drained, heated')
(09007, 0900, 'Apples, canned, sweetened, sliced, drained, unheated')
(09010, 0900, 'Apples, dehydrated (low moisture), sulfured, stewed')
(09009, 0900, 'Apples, dehydrated (low moisture), sulfured, uncooked')
(09013, 0900, 'Apples, dried, sulfured, stewed, with added sugar')
(09012, 0900, 'Apples, dried, sulfured, stewed, without added sugar')
(09011, 0900, 'Apples, dried, sulfured, uncooked')
(09015, 0900, 'Apples, frozen, unsweetened, heated')
(09014, 0900, 'Apples, frozen, unsweetened, unheated')
(09501, 0900, 'Apples, raw, golden delicious, with skin')
(09003, 0900, 'Apples, raw, with skin')
(09004, 0900, 'Apples, raw, without skin')
(09005, 0900, 'Apples, raw, without skin, cooked, boiled')
(09006, 0900, 'Apples, raw, without skin, cooked, microwave')
```

Question 7 : Écrire la requête qui permet d'afficher 'Protéine', `nutr_val`, `long_desc` pour 100g de pommes sous toutes ses formes. Il faudra se limiter au nutriment **protéine**. Les résultats devront être classés par ordre décroissant de `nutr_val`.

Question 8 : Écrire la requête qui permet de déterminer `long_desc`, `ndb_no`, `nutr_val` pour la valeur maximale de la teneur en protéine pour 100g de pommes sous toutes ses formes.

Question 9 : Écrire la requête qui permet d'afficher pour les pêches (*Peaches*), les pommes (*Apples*) et les bananes (*Bananas*) la moyenne respective de l'apport en énergie en kcal. Il est possible ici d'effectuer cette recherche à partir de trois requêtes différentes.

1.3 Informations globales sur le site

Afin de mettre en avant le contenu de la base de données sur la première page de l'application, nous souhaitons afficher la phrase suivante : *l'application compte aujourd'hui XXXX aliments*, ou XXXX sera le nombre d'aliment de la base. Ce nombre devra être modifié dynamiquement lorsque des aliments sont ajoutés.

Question 10 : Écrire la requête qui permet de déterminer le nombre d'aliment dans la base de données afin de l'écrire sur la première page.

Question 11 : Écrire la requête qui permet de déterminer le nombre d'aliment dans la base de données par catégorie d'aliment. Vous afficherez les résultats comme suit `c.fmgrp_desc` suivi du nombre d'aliment de la catégorie.

1.4 Calculs pour une recette

Dans l'application, un utilisateur peut ajouter une recette. Pour cela il remplit un formulaire internet PHP. L'application va ensuite enregistrer les résultats du formulaire dans des listes Python.

Il est donc important de pouvoir effectuer les requêtes SQL dans un programme python afin de récupérer les données de la base et pour les intégrer à nos calculs.

C'est ce que permet le code suivant, il doit être intégré au programme python à réaliser.

```
import MySQLdb # Bibliotheque python pour MySQL

conn = MySQLdb.connect (host = 'www.costadoat.fr',
                        user = 'dorian',
                        passwd = 'dorian',
                        db = 'food')
cursor = conn.cursor () # Connexion à la base

cursor.execute ("SELECT * FROM foodcats") # Requête SQL
rows = cursor.fetchall ()

for row in rows: #rows est le tableau résultat de la requête
    print row #row correspond à une ligne du tableau
```

Il est possible d'insérer des variables à la requête comme suit.

```
a=0400
b=04016
cursor.execute("SELECT * FROM fooddescs WHERE fdgrp_cd={0} AND ndb_no<{1}".format(a,b))
```

Revient à écrire.

```
cursor.execute("SELECT * FROM fooddescs WHERE fdgrp_cd=0400 AND ndb_no<04016")
```

Les paramètres {0} et {1} sont remplacés par python par les éléments dans la fonction `format()`, il faut les écrire dans l'ordre et commencer par {0} pour que cela fonctionne.

Une des liste générée à partir du remplissage du formulaire est la liste de ingrédients. Son format est le suivant :

```
['nom de l'ingrédient', 'ndb_no', 'masse en grammes'].
```

Par exemple, la liste suivante présente les ingrédients nécessaires à la fabrication d'un croque monsieur.

```
recette=[['jambon', 07029, 30], ['beurre', 01001, 5],
          ['pain cereales', 18035, 80], ['cheddar', 01009, 20]]
```

Question 12 : Proposer le code python nécessaire afin de déterminer la valeur énergétique d'un croque monsieur en kcal.

```
Valeur energetique croque monsieur: 377.35 kcal
```

1.5 Éliminer

Une fonctionnalité supplémentaire de l'application est de déterminer un « équivalent jogging » permettant d'éliminer l'équivalent énergétique de la consommation de la recette.

Ainsi, l'utilisateur indique :

- sa masse (en kg),
- la distance de son circuit habituel (en km).

La formule est la suivante :

$$K_{conso} = m \times d \times F_v, \text{ avec } F_v = a.v^2 + b.v + c$$

Question 13 : Déterminer l'expression littérale de la vitesse du coureur qui lui permettra de consommer l'équivalent énergétique de deux croques monsieur.

2 Exercice de course

Dans la suite, l'objectif est de calculer la distance parcourue lors d'un jogging en fractionné progressif.

Ce type d'exercice consiste à découper la course comme suit :

1. $[t_0, t_1[$: Vitesse constante (90 secondes) : $v = v_0$,
2. $[t_1, t_2[$: Accélération continue (90 secondes) : $v = v_0 + a.(t - t_1)$,
3. $[t_2, t_3[$: Vitesse constante (90 secondes) : $v = v_0$,
4. $[t_3, t_4[$: Accélération continue (90 secondes) : $v = v_0 + a.(t - t_3)$,
5.

Avec dans le cas de notre exercice :

- $v_0 = 9km.h^{-1} = 2.5m.s^{-1}$,
- $a = 0.02m.s^{-2}$

Une période étant constituée d'une phase de vitesse constante et d'une phase d'accélération, l'exercice sera composé de **5 périodes**.

Question 14 : Donner le code Python permettant de générer la liste `temps` qui contient toutes les secondes $[1, 2, \dots]$ jusqu'à la fin de l'exercice. Générer la liste `vitesse` qui contient toutes les vitesses en $km.h^{-1}$ pour chaque seconde $[9, 9, \dots]$ jusqu'à la fin de l'exercice.

Pour la suite, on souhaite obtenir une fonction $F(t)$ qui pour un instant t en seconde renvoie la vitesse à cet instant.

Question 15 : Donner le code Python permettant de créer la fonction $F(t)$ à partir des listes générées à la question 14.

Question 16 : Donner le code Python permettant d'utiliser la méthode d'Euler afin de déterminer une liste présentant la distance parcourue à chaque instant.

Question 17 : Donner le code Python permettant de déterminer la distance totale parcourue durant l'exercice.

Question 18 : Donner le code Python permettant de tracer la vitesse en fonction du temps.

Question 19 : Donner le code Python permettant de tracer la position en fonction du temps.

