

SQL aux concours

I. TPC

I.1. 2024

On dispose d'une base de données composée de quatre tables (**annexe 1**) :

- La table **Continents** constituée des champs suivants :
 - *nom* : nom du continent (chaîne de caractères) ;
 - *surface* : surface du continent en kilomètres carrés (entier).
- La table **Pays** constituée des champs suivants :
 - *nom* : nom du pays (chaîne de caractères) ;
 - *code_pays* : identifiant unique du pays (chaîne de caractères) ;
 - *capitale* : capitale administrative du pays (chaîne de caractères) ;
 - *population* : nombre d'habitants du pays (entier).
- La table **Inclusion** constituée des champs suivants :
 - *code_pays* : identifiant unique du pays (chaîne de caractères) ;
 - *continent* : nom du continent auquel appartient le pays (chaîne de caractères).
- La table **Frontieres** constituée des champs suivants :
 - *code_pays1* : identifiant unique du premier pays (chaîne de caractères) ;
 - *code_pays2* : identifiant unique du second pays (chaîne de caractères) ;
 - *longueur* : longueur en kilomètres de la frontière entre *pays1* et *pays2* (nombre flottant strictement positif).

On a toujours $code_pays1 < code_pays2$ pour l'ordre lexicographique, ce qui assure que chaque frontière n'apparaît qu'une fois dans la table **Frontieres**.

1. Écrire une requête SQL permettant de récupérer le code du pays nommé 'France'.
2. Écrire une requête SQL permettant d'obtenir les noms des pays appartenant au continent 'Europe'.

Dans les deux dernières questions, on admet que le code du pays nommé 'France' est 'F' et on peut utiliser librement cette information.

3. Écrire une requête SQL permettant de récupérer la longueur totale de la frontière du pays nommé 'France'.
4. Écrire une requête SQL permettant de récupérer les noms des pays frontaliers du pays nommé 'France'.

ANNEXE 1 - Base de données géographiques

Table **Continents**

nom	surface
'Asie'	44579000
'Europe'	9938000
...	...

Table Pays

nom	code_pays	capitale	population
'Albanie'	'AL'	'Tirana'	3088385
'Algerie'	'DZ'	'Alger'	44487616
...

Table Inclusion

code_pays	continent
'AL'	'Europe'
'DZ'	'Afrique'
...	...

Table Frontieres

code_pays1	code_pays2	longueur
'AL'	'GR'	282.8
'AL'	'MK'	151.5
...

I.2. 2023

L'entreprise possède une base de données nommée *Gestion_Entreprise* constituée de trois tables : clients, produits et ventes. Les contenus de ces tables se trouvent en **annexe 2**.

La **table 'clients'** est constituée de 5 champs :

- *id* : de type INTEGER – clé primaire auto-incrémentée ;
- *num_secu* : de type INTEGER - entier de 15 chiffres ;
- *nom* : de type TEXT ;
- *prenom* : de type TEXT ;
- *num_CB* : de type INTEGER.

La **table 'produits'** est constituée de 5 champs :

- *id* : de type INTEGER – clé primaire auto-incrémentée ;
- *ref_produit* : de type INTEGER ;
- *nom_produit* : de type TEXT ;
- *qrcode* : de type TEXT ;
- *prix* : de type DECIMAL.

La **table 'ventes'** est constituée de 3 champs :

- *date* : de type TEXT ;
- *ref_produit* : de type INTEGER ; – clé étrangère, pointe vers la clé primaire *id* de la table *produits* ;
- *num_clients* : de type INTEGER ; – clé étrangère, pointe vers la clé primaire *id* de la table *clients*.

Les dates dans cette table sont définies par une chaîne de 10 caractères suivant le format *année-mois-jour*. Exemples de dates : '2019-06-01', '2022-12-30'.

1. Écrire, en SQL, la requête (1) qui permet d'obtenir le numéro de la carte de crédit de toutes les personnes référencées dans la base de données de l'entreprise dont le numéro de sécurité sociale commence par 2. On utilisera le caractère '_' comme séparateur des milliers. Par exemple 10000000 sera réécrit comme 10_000_000.

2. Écrire, en SQL, la requête (**2**) permettant d'obtenir le nom et le prénom de toutes les personnes ayant effectué un achat avec un résultat sans doublon.
3. Écrire, en SQL, la requête (**3**) qui permet d'obtenir les produits associés à chaque numéro de carte de crédit du client et qui ont été vendus entre le 1 juin 2020 et le 30 juillet 2020. On rappelle que SQL compare les variables de type **TEXT** grâce à l'ordre lexicographique.
Par exemple '1989-06-13 < 1999-07-13' est vrai.

Annexe 2

Base de donnée 'Gestion_Enterprise'

Le contenu des tables *clients*, *produits* et *ventes* de la base de données *Gestion_Enterprise* est donné ci-après.

Tables clients

id	num_secu	nom	prenom	num_CB
1	286128817863441	Eldyn	Sophie	6767342589219928
2	298082934500890	Gomez	Maria	2324563490665454
3	298082934500896	Ruiza	Flor	9889454573204522
4	109086723487917	Kovitz	Boris	6789543778653678
5	175105642102321	Mottreff	Erwan	4745342178563217
6	189027511732543	Settin	Michel	7856432167453492
7	191017511318196	Valérie	Georges	8787564521392354
...

Table produits

id	ref_produit	nom_produit	qrcode	prix
1	27	Buffet chêne	27.jpg	320
2	102	Chaise rustique	102.jpg	65
3	453	Table ronde	453.jpg	75
4	756	Table ovale	756.jpg	120
5	921	Coffret Bali	921.jpg	170
...

Table ventes

date	ref_produit	num_client
2020-05-15	2	2
2020-06-17	3	1
2020-06-21	3	7
2020-07-19	4	5
2020-08-19	5	5
2020-09-05	4	6

II. TSI

II.1. 2022

SQLite permet de créer et de gérer des bases de données. Le module Python `pysqlite3` permet d'interagir avec des bases de données SQLite.

Les tables utiles pour répondre aux questions sont en **annexe 3**. On remarquera que dans la table **Photos**, les latitudes et longitudes sont des entiers : une division par 10000 permet d'avoir la latitude et la longitude réelle.

On notera que dans la table **Photos**, **Date** correspond à la date de création (année/mois/jour) de photos présentes dans le dossier (colonne nommée Dossier). On précise qu'à un nom de photo correspond une unique photo et que chaque dossier ne contient que des photos.

1. Écrire la requête SQL qui permet d'avoir le nom, la latitude et la longitude des photos faites le 2020/06/16.
2. Écrire la requête SQL qui permet d'avoir le nom des photos faites le 2020/06/16 et entre les latitudes 35935 et 35940.
3. Écrire la requête SQL qui permet de compter le nombre de photos situées dans le Dossier C:\Images\LasVegas\B
4. Écrire la requête SQL qui permet de trouver le nom des photos prises le 2020/06/17 et situées dans le dossier C:\Images\LasVegas\Bellagio

ANNEXE 3 - Tables SQL

Table Photos

Nom	Date	Latitude	Longitude	id
Photo10	2020/06/11	35937	-1149686	1
Photo11	2020/06/22	35937	-1149685	2
Photo24	2020/06/17	35937	-1149686	3
Photo15	2020/06/17	35938	-1149686	4
Photo33	2020/06/16	35939	-1149688	5
⋮	⋮	⋮	⋮	⋮

Table Dir

id	Dossier	NomPhoto
1	C:\Images\LasVegas\Bellagio	Photo10
2	C:\Images\LasVegas\Palazzo	Photo11
3	C:\Images\LasVegas\Flamingo	Photo24
⋮	⋮	⋮
10	C:\Images\LasVegas\Flamingo	Photo40
11	C:\Images\LasVegas\Palazzo	Photo30
12	C:\Images\LasVegas\Flamingo	Photo15
⋮	⋮	⋮

Type des données par Table

Table	Champ	Type
Photos	Nom	Chaîne de caractères
Photos	Date	Chaîne de caractères
Photos	Latitude	Entier
Photos	Longitude	Entier
Photos	id	Entier
Dir	id	Entier
Dir	NomPhoto	Chaîne de caractères
Dir	Dossier	Chaîne de caractères

II.2. 2021

À chaque livraison, l'entreprise stocke des données relatives à celle-ci. L'entreprise dispose de 20 locaux, numérotés de 1 à 20, disposant chacun d'un certain nombre de camions. Pour faciliter ses livraisons, l'entreprise découpe la France en 30 zones et associe à chaque local, trois zones possibles de livraisons.

Ces données sont enregistrées dans une base de données contenant trois tables :

- La table **livraison** constituée des champs suivants :
 - × *date* : date de la livraison au format 'jj-mm-aaaa' (chaîne de caractères) ;
 - × *heure* : heure de la livraison au format 'hh-mm-ss' (chaîne de caractères) ;
 - × *id_client* : identifiant du client recevant la livraison (entier) ;
 - × *id_local* : identifiant du local de l'entreprise (entier compris entre 1 et 20).
- La table **client** constituée des champs suivants :
 - × *id* : identifiant du client (entier) ;
 - × *zone* : entier compris entre 1 et 30.
- La table **local** constituée des champs suivants :
 - × *id* : identifiant du local (entier compris entre 1 et 20) ;
 - × *zone1* : entier ;
 - × *zone2* : entier ;
 - × *zone3* : entier.

1. Donner une clé primaire pour la table **livraison**.
2. Écrire une requête SQL permettant d'obtenir les identifiants des clients livrés le 10 janvier 2021.
3. Écrire une requête SQL permettant de récupérer les dates et les heures de toutes les livraisons ayant eu lieu dans la zone 5 le 2 mars 2021.
4. Écrire une requête SQL permettant de compter le nombre de livraisons effectuées le 3 février 2021 par des camions dont les locaux ne livrent que dans des zones possibles inférieures ou égales à dix.

II.3. 2020

La Collection Française de Bactéries Phytopathogènes (CFBP) possède deux bases de données :

- l'une, appelée Echantillon, qui permet de stocker les différents échantillons d'ADN ;
- l'autre, appelée Sequence, qui permet de mémoriser quelle personne est responsable de l'obtention de la séquence (*i.e.* de l'extraction d'un gène particulier dans les différents échantillons d'ADN).

Des extraits des tables Echantillon et Sequence sont données par les **tableaux 1 et 2**.

Echantillon			
ADN	Genre	Espèce	Sous-espèce
309	Pseudomonas	syringae	morsprunorum
...
3589	Pseudomonas	syringae	vignae
...

TAB. 1 Table recensant toutes les séquences d'ADN dont dispose la CFBP

Sequence					
Code	Date	ADN	Gène	Protocole	Employé
A	'01-03-2018'	309	gyrB	Spilker	Dupont
B	'01-03-2018'	2028	recA	Cesbron and Manceau	Martin
...
AGZ	'10-03-2018'	2028	leuS	Deletoile	Martin
...

TAB. 2 Table recensant tous les travaux réalisés à la CFBP en mars 2018

1. Définir le but et le résultat de la requête(1), écrite en SQL :

```
SELECT count(*) FROM Sequence WHERE Date = '01 - 03 - 2018'
```

2. Écrire en SQL la requête(2) qui permet d'obtenir la liste des échantillons d'ADN contenant le gène `leuS`.
3. Écrire en SQL la requête(3) qui permet d'obtenir la liste des espèces étudiées par M. Martin le 10 mars 2018.
4. Écrire en SQL la requête(4) permettant d'obtenir le nombre d'échantillons prélevés par chaque employé.