

## Exercice 1 (6 points)

Cet exercice porte sur les bases de données relationnelles, les requêtes SQL, la programmation en Python et la manipulation de listes.

Dans cet exercice, on pourra utiliser les clauses du langage SQL pour :

- construire des requêtes d'interrogation à l'aide de `SELECT`, `FROM`, `WHERE` (avec les opérateurs logiques `AND` et `OR`) et `JOIN ... ON` ;
- construire des requêtes d'insertion et de mise à jour à l'aide de `UPDATE`, `INSERT` et `DELETE` ;
- affiner les recherches à l'aide de `DISTINCT` et `ORDER BY`.

Le but de cet exercice est d'établir une prédiction de la météo du jour en utilisant les observations du jour précédent de plusieurs stations météorologiques voisines.

### Partie A

Une version simplifiée des observations peut être représentée sous forme de tables dont la description est donnée ci-dessous. Les clés primaires ont été soulignées et les clés étrangères sont indiquées par un # :

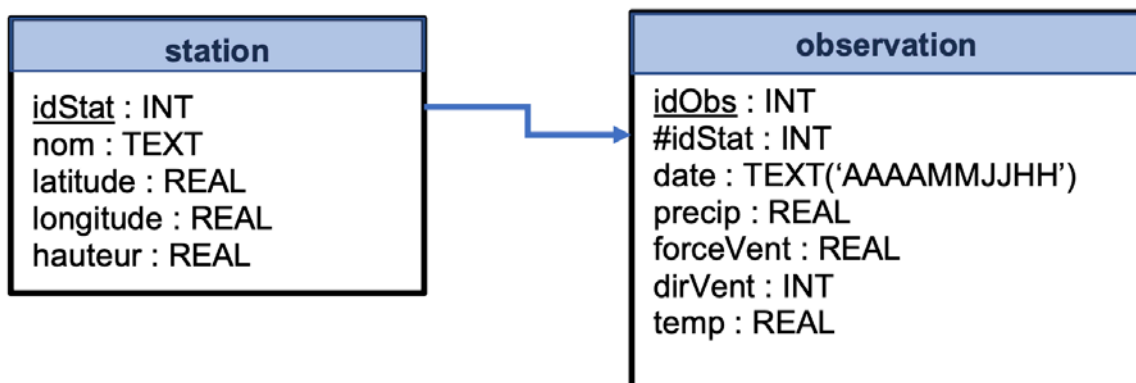


Figure 1. Tables

Dans cette partie, on considère les observations météorologiques de la Nouvelle Calédonie.

La table `station` contient l'identifiant `idStat`, le nom `nom` et les coordonnées géographiques de toutes les stations météorologiques.

La table `observation` contient l'identifiant `idStat` de l'observation, la date de l'observation `date`, la hauteur de précipitation `precip`, la force du vent `forceVent`, la direction du vent `dirVent` et la température `temp` heure par heure de toutes les stations.

Extrait table station				
idStat	nom	latitude	longitude	hauteur
...	...	...	...	...
98818001	NOUMEA	-22.276000	166.452833	69
98818002	MAGENTA	-22.260333	166.473667	3
...	...	...	...	...

Extrait de la table observation						
idObs	idStat	date	precip	forceVent	dirVent	temp
...	...	...	...	...	...	...
123456	9881800 1	202312312 1	0.0	5.7	260	24.4
123457	9881800 1	202312312 2	0.0	5.5	260	24.4
123458	9881800 1	202312312 3	0.2	5.5	250	24.1
123459	9881800 2	202301010 0	0.0	4.7	260	24.1
123460	9881800 2	202301010 1	1.4	3.5	80	23.5
123461	9881800 2	202301010 2	0.4	2.1	190	23.4
123462	9881800 2	202301010 3	0.2	1.7	330	23.4
123463	9881800 2	202312312 2	0.1	1.8	310	22.7
...	...	...	...	...	...	...

1. Donner le résultat de la requête ci-dessous en considérant les extraits de table fournis.

```
SELECT nom
FROM station
WHERE latitude = -22.276000 AND longitude = 166.452833
```

2. Écrire une requête permettant d'obtenir le nom de toutes les stations météorologiques triées par ordre alphabétique.

En SQL, la fonction d'agrégation COUNT permet de compter le nombre d'enregistrements dans une table.

Pour connaître le nombre de lignes totales dans une colonne, la syntaxe est la suivante :

```
SELECT COUNT(nom_colonne)
FROM table
```

Par exemple pour compter le nombre de stations météorologiques de la Nouvelle Calédonie, la requête est la suivante :

```
SELECT COUNT(idStat)
FROM station
```

Dans la table `observation`, les relevés météorologiques sont effectués au même moment pour toutes les stations (date identique). Ainsi, chaque station a le même nombre de relevés.

3. Écrire une requête permettant d'obtenir la force et la direction du vent à BOURAKE le 2 janvier 2023 à 14h.
4. Écrire une requête permettant d'obtenir le nombre total de relevés en Nouvelle Calédonie.

On souhaite regrouper toutes les informations dans une seule table `meteo`.

5. Écrire le schéma relationnel de la table `meteo` en supprimant les données `hauteur`, `precip`, `forceVent` et `dirVent`.

## Partie B

Les données collectées sont stockées dans un unique fichier texte au format csv (*Comma Separated Values*, valeurs séparées par des virgules). Le module Python `csv` implémente des classes pour lire et écrire des données tabulaires au format csv.

On fournit ci-dessous un extrait du fichier `observations.csv` qui donne heure par heure les précipitations en millimètre, la force du vent en mètre par seconde et la direction du vent en degré (de 0 à 360 degrés) ainsi que la température en degré Celsius de la journée du 01/01/2024 pour toutes les stations météorologiques de Nouvelle Calédonie :

```
observations.csv
Fichier  Modifier  Affichage

ID_STATION,NOM_STATION,LATITUDE, LONGITUDE, ALTITUDE, AAAAMJJHH, PRECIPITATION, FORCE_VENT, DIR_VENT, TEMPERATURE
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010100,0.0,5.0,80,25.7
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010101,0.0,4.5,80,25.9
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010102,0.0,4.8,80,26.1
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010103,0.0,5.2,90,26.4
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010104,0.0,5.5,90,26.4
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010105,0.0,5.1,80,26.3
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010106,0.0,4.8,80,26.6
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010107,0.0,4.9,90,27.3
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010108,0.0,5.4,90,29.5
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010109,0.0,6.2,80,30.2
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010110,0.0,7.3,80,29.5
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010111,0.6,7.0,90,28.3
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010112,0.0,6.9,90,30.7
98801001,BELEP AEROD.,-19.719833,163.661000,88,2024010113,0.0,6.5,90,31.0

Ln 27, Col 69 | 83 607 caractères 100% Windows (CRLF) UTF-8
```

Figure 2. Extrait fichier observations.csv

Source : d'après [meteo.data.gouv](http://meteo.data.gouv)

Pour la suite de l'exercice, on dispose du code Python donné en annexe ainsi que de la documentation suivante :

#### DOCUMENTATION :

- `with open('mon_fichier.csv', 'r') as csvfile`, ouvre le fichier `mon_fichier.csv` en mode lecture (`r`) ;
- `csv.reader(csvfile, delimiter=',')`, renvoie un objet lecteur, qui itérera sur les lignes de l'objet `csvfile` donné. Chaque ligne lue depuis le fichier csv est renvoyée comme une liste de chaînes de caractères.

Dans la console, on saisit la suite d'instructions suivante :

```
>>>liste_obs = creation_liste_obs('observations.csv')
>>>liste_obs = supp_champs(liste_obs)
>>>transtype(liste_obs)
>>>liste_obs[0]
[98801001, 'BELEP AEROD.', -19.719833, 163.661, 88, 2024010100,
0.0, 5.0, 80, 25.7]
```

6. Expliquer cette liste de commande et le résultat obtenu.

Dans la suite de l'exercice, la variable `liste_obs` est initialisée avec les valeurs du fichier `observations.csv`.

La fonction `distance` renvoie la distance entre deux points définis par leur latitude et leur longitude. Cette fonction utilise des fonctions du module Python `math`.

7. Donner la ligne de commande nécessaire à l'utilisation du module Python `math`.

On rappelle les informations relatives à une observations sont données dans l'ordre suivant :

ID\_STATION, NOM\_STATION, LATITUDE, LONGITUDE, ALTITUDE, AAAAMMJJHH, P  
RECIPITATION, FORCE\_VENT, DIR\_VENT, TEMPERATURE

8. Compléter les lignes 40 et 41 de la fonction `coord`, qui prend en paramètres une liste d'observations `l_obs` et un nom de station `stat_ref`, et qui renvoie un tuple composé de sa latitude et sa longitude.

On considère la fonction `liste_stations` qui prend en paramètres une liste d'observations `l_obs`, un nom de station `stat_ref` et un flottant `dist` et qui renvoie la liste des identifiants `ID_STATION` des stations données dans la liste `l_obs` situées à une distance inférieure à `dist` de la station de référence `stat_ref`.

9. Écrire un algorithme en pseudo-code de la fonction `liste_stations`.
10. Écrire une fonction `nettoyage` qui prend en paramètres une liste d'observations `l_obs` et station de référence `stat_ref` (nom de la station), et qui renvoie la liste des températures des stations données dans la liste d'observations `l_obs` situées à une distance inférieure à 2000 unités de la station de référence `stat_ref`.
11. Écrire la fonction `moyenne` qui calcule et renvoie la moyenne de toutes les valeurs de type `float` contenues dans la liste passée en paramètre.

On considère maintenant le fichier `observations2.csv` donnant heure par heure les observations de la journée du 01/01/2024 pour toutes les stations météorologiques de France.

12. Donner les commandes permettant d'obtenir la moyenne des températures des stations situées à moins de 2000 unités de la station Paris\_11 le 1<sup>er</sup> janvier 2024.

## ANNEXE

```
1 import csv
2
3 def creation_liste_obs(fichier) :
4     liste_obs=[]
5     with open(fichier,'r') as csvfile:
6         fic=csv.reader(csvfile,delimiter=',')
7         for ligne in fic:
8             liste_obs.append(ligne)
9     return liste_obs
10
11 def supp_champs(L) :
12     res = []
13     for i in range(1,len(L)):
14         res.append(L[i])
15     return res
16
17 def transtype(L):
```

```

18     i=0
19     while i < len(L):
20         L[i] = [int(L[i][0]),
21             L[i][1],
22             float(L[i][2]),
23             float(L[i][3]),
24             int(L[i][4]),
25             int(L[i][5]),
26             float(L[i][6]),
27             float(L[i][7]),
28             int(L[i][8]),
29             float(L[i][9])]
30         i = i + 1
31
32     def distance(p1, p2):
33         """Renvoie la distance entre deux points définis par leur
34         latitude et leur longitude. p1 et p2 sont des tuples
35         (latitude,longitude)"""
36         # Cette fonction n'est pas à compléter
37
38     def coord(l_obs , stat_ref):
39         """Renvoie la latitude et la longitude données dans la
40         liste d'observation l_obs de la station stat_ref"""
41         # Cette fonction est à compléter à la question 8.
42         for obs in l_obs :
43             if ... :
44                 return ....,....
45
46     def liste_stations(l_obs, stat_ref, dist):
47         """Renvoie la liste des identifiants ID_STATION des
48         stations données dans la liste l_obs situées à une distance
49         inférieure à dist de la station de référence stat_ref"""
50         # Cette fonction n'est pas à compléter
51
52     def nettoyage(l_obs, stat_ref):
53         """Renvoie la liste des températures des stations données
54         dans la liste d'observations l_obs situées à une distance
55         inférieure à 2000 unités de la station de référence
56         stat_ref."""
57         # Cette fonction est à compléter à la question 10.
58
59     def moyenne(L):
60         """Calcule et renvoie la moyenne de tous les nombres
61         contenus dans la liste passée en paramètre. L est une liste de
62         flottants."""
63         # Cette fonction est à compléter à la question 11.

```