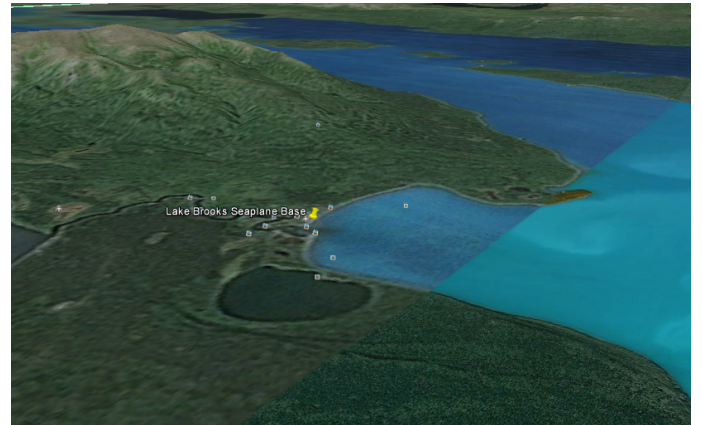


# CI 4 : BASES DE DONNÉES

## TP BASE DE DONNÉES DES AÉROPORTS DU MONDE

### Présentation

Le site internet <http://ourairports.com/> recueille des informations sur les aéroports du monde. Ces informations sont disponibles sous la forme d'une base de donnée SQLite.



Vu Google Earth de la base d'hydravion de Lake Brooks – Alaska

**Copier le dossier *Informatique/TP\_11\_BDD/* dans votre répertoire personnel.**

L'utilitaire SQLite Database Browser permet de visualiser le contenu d'une base de données au format SQLite. Pour consulter la base :

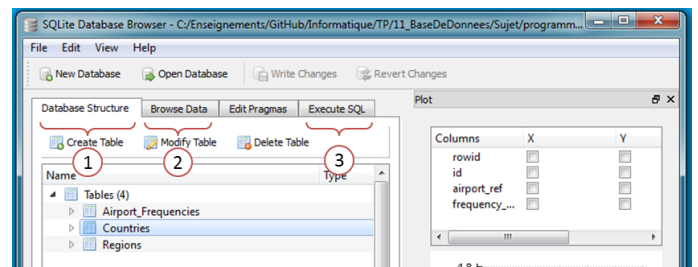
- ouvrir l'utilitaire SQLite Database Browser ;
- ouvrir le fichier Aeroports.sql3.

En l'état, la base de données est constituée de 4 relations :

- Airport\_Frequencies recense les fréquences radio sur lesquelles les aéroports émettent ;
- Countries recense la liste des pays ;
- Airports recense la liste des aéroports ;
- Regions recense une liste des régions.

La relation Airports est constituée des attributs suivants :

- id : un identifiant
- type : un type (heliport, small\_airport, seaplane\_base ...)
- name : un nom
- des coordonnées géographiques (latitude\_deg, longitude\_deg, elevation\_ft, à savoir les latitudes et longitudes en degrés ainsi que l'altitude en pieds) ;
- ...



- ① Structure de la table  
⇒ Possibilité de modifier les attributs et leur type
- ② Contenu de la table  
⇒ Possibilité de modifier les valeurs
- ③ Exécution de requêtes  
⇒ Test de requêtes SQL

### Analyse de la base de données

#### Question 1

Lors de la création de la base de données, le concepteur n'a pas pris garde au type des attributs. Proposer un type pour les attributs latitude\_deg et municipality. Dans la mesure du possible, modifier ces champs en éditant la table correspondante. Quel type de champ proposer pour l'attribut id ? D'après vous quelle précaution est-il nécessaire de prendre pour ce champ ?

#### Question 2

Quels sont les attributs des relations Countries, Regions et Airport\_Frequencies ?

Dans le but d'optimiser la base de données (à savoir diminuer le temps d'accès aux données, diminuer l'espace mémoire utilisé et éviter la redondance d'informations) on souhaite disposer de la relation Airport\_Type. Cette relation aura pour

attribut un identifiant, un attribut de type texte correspondant au type d'aéroport en anglais, un attribut de type texte correspondant au type d'aéroport en français.

### Question 3

Créer la relation précédente dans la base de données. Quelles seraient les modifications à réaliser sur les données pour prendre en compte cette nouvelle relation ?

## Consultation de la base de données en utilisant SQLite Database Browser

Nous allons travailler dans l'onglet Execute SQL.

### Question 4

Que permet la requête suivante :

SQL

```
SELECT name FROM Countries;
```

Quelle est sa traduction en algèbre relationnelle ?

### Question 5

On souhaite sélectionner tous les noms de pays européens de la relation Countries. Exprimer la requête dans l'algèbre relationnelle puis en langage SQL. Tester la requête. L'attribut des pays européens est désigné par "EU".

### Question 6

On donne la requête suivante :

$$\pi_{name}(\sigma_{iso\_country="FR"}(Regions))$$

Que signifie-t-elle ? La traduire en langage SQL puis tester le résultat.

## Consultation de la base de données en Python

Remarque

Ouvrir Spyder et le fichier BDD\_eleve.py.

Il est possible d'interroger la base de données en utilisant Python. Pour cela, on utilise le formalisme suivant :

python

```
import sqlite3 # Import des commandes permettant de manipuler la base de données
basesql = u"airports.db3" # Base de données initiale

cnx = sqlite3.connect(basesql)
curseur = cnx.cursor()

requete = "SELECT * FROM airports"
curseur.execute(requete)
```

Le préfixe u permet d'importer les fichiers encodés en UTF-8. Dans un certaine mesure, les caractères spéciaux sont alors pris en compte.

python

Le curseur est un objet contenant le résultat de la requête. Pour visualiser la première entité de la requête, la syntaxe est la suivante :

```
data = curseur.fetchone()
print(data)
```

Attention, à chaque utilisation de `fetchone()`, l'entité est supprimée du curseur. Pour parcourir chacune des entités, on peut utiliser la syntaxe suivante :



```
for cur in curseur:
    print(cur)
```

Attention, cette opération peut s'avérer maladroite si la requête a un grand nombre d'entités.

### Question 7

Remplir les champs nécessaires dans le fichier `BDD_eleve.py`. Exécuter le code et expliquer les résultats.

### Question 8

Que permettent les lignes de code suivantes :



```
requete = "SELECT * FROM airports"
curseur.execute(requete)
res = []
for cur in curseur:
    res.append(cur)
print(len(res))
```

### Question 9

En utilisant les possibilités de Python, donner le nombre de bases d'hydravion existantes. En utilisant la documentation ou le cours, comment utiliser la fonction d'agrégation `COUNT` pour obtenir un résultat équivalent ?

### Question 10

Donner la liste des villes françaises (`iso_country='FR'`) hébergeant de telles bases. Vous donnerez la requête SQL ainsi que son expression en algèbre relationnelle.

### Question 11

Donner la liste des villes européennes (`continent='EU'`) hébergeant de telles bases ainsi que leur nom et leur pays. Vous donnerez la requête SQL ainsi que son expression en algèbre relationnelle.

### Question 12

En utilisant une jointure entre les relations `Countries` et `airports`, donner la liste des bases d'hydravion américains (United States).

## Affichage des bases d'hydravion sur une carte

On souhaite afficher toutes les bases d'hydravion européennes sur une carte GoogleEarth.

### Question 13

A partir des résultats d'une requête, mettre dans un tableau les lignes constituées du nom de l'aéroport, de sa longitude, de sa latitude et de son type. Une ligne constituera une seule chaîne de caractère, chaque champ étant séparé par une virgule. La fin de la ligne sera terminée par un `\n`. De plus, les chaînes de caractères ne devront pas contenir le caractère `&`.

Exemple

Exemple de ligne :

```
'Blinn Lake Seaplane Base,-162.753005981445,55.2515983581543,seaplane_base\n'
```