

## CI 4 : BASES DE DONNÉES

### TP BASE DE DONNÉES DES AÉROPORTS DU MONDE

#### Présentation

#### 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?

Corrigé

Pour les attributs `longitude_deg` et `latitude_deg`, il serait préférable d'utiliser le type `REAL`.  
Pour l'attribut `municipality`, on peut laisser le type `TEXTE`.  
Pour l'attribut `id`, il serait préférable d'utiliser un entier (`INT`) en imposant une valeur non nulle (`NOT NULL`) et éventuellement un incrément automatique. . On peut envisager de déclarer cet attribut en clef primaire.

##### Question 2

Quels sont les attributs des relations `Countries`, `Regions` et `Airport_Frequencies`?

Corrigé

- `Countries` :
  - `id`;
  - `code`;
  - `name`;
  - `continent`;
  - `wikipedia_link`;
  - `keywords`.
- `Regions` :
  - `id`;
  - `code`;
  - `local_code`;
  - `name`;
  - `continent`;
  - `iso_country`;
  - `wikipedia_link`;
  - `keywords`.
- `Regions` :
  - `id`;
  - `code`;
  - `local_code`;
  - `name`;
  - `continent`;
  - `iso_country`;
  - `wikipedia_link`;
  - `keywords`.

##### 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?

Corrigé

Manipulation à faire avec le logiciel.

## 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 ?

Corrigé

Il s'agit d'une projection permettant de sélectionner le nom de tous les pays de la table Countries.  
Elle se traduit par :

$$\pi_{\text{name}}(\text{Countries})$$

### 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".

Corrigé

SQL

```
SELECT name FROM Countries WHERE continent="EU";
```

$$\pi_{\text{name}}(\sigma_{\text{continent}="EU"}(\text{Countries}))$$

### Question 6

On donne la requête suivante :

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

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

Corrigé

SQL

Le but de cette requête est de déterminer la liste des régions de France.

```
SELECT name FROM Regions WHERE iso_country="FR";
```

## Consultation de la base de données en Python

### Question 7

Remplir les champs nécessaires dans le fichier BDD\_eleve.py. Exécuter le code et expliquer les résultats.

Corrigé

Il suffit de donner le nom du fichier correspondant à la base.  
Les lignes permettent de donner tous les attributs de 3 bases d'hydravions (au «hasard»).

### Question 8

Que permettent les lignes de code suivantes :

python

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

Corrigé

Ces lignes permettent de donner le nombre d'aéroports présents dans la base.

### 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 ?

Corrigé

Voir Fichier.py.

SQL

```
SELECT COUNT(*) FROM airports WHERE type='seaplane_base'
```

Il y a 899 bases.

### 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.

Corrigé

SQL

```
SELECT municipality FROM airports WHERE type='seaplane_base' AND iso_country='FR'
```

$$\pi_{\text{municipality}}(\sigma_{\text{iso\_country}='FR' \wedge \text{type}='seaplane\_base'}(\text{airports}))$$

La seule infrastructure présente dans la base est celle de Marseille (Marignanne).

### 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.

SQL

```
SELECT name,municipality FROM airports WHERE type='seaplane_base' AND continent='EU'
```

Corrigé

$$\pi_{\text{municipality}}(\sigma_{\text{type}='seaplane\_base' \wedge \text{continent}='EU'}(\text{airports}))$$

### Question 12

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

SQL

```
SELECT name FROM
  (SELECT name,iso_country FROM airports WHERE type='seaplane_base') as aeroports
JOIN
  (SELECT Countries.code FROM Countries WHERE name='United States') as pays
ON pays.code = aeroports.iso_country
```

Corrigé

$$\pi_{\text{name}} \left( \pi_{\text{name,iso\_country}} (\sigma_{\text{type}="seaplane\_base"} (\text{airports})) \bowtie_{\text{airports.iso\_country}=\text{Countries.code}} \pi_{\text{code}} (\sigma_{\text{name}="United States"} (\text{Countries})) \right)$$

## 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 &.

Corrigé

Exemple de ligne :

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

Exemple