

# Bases de données - Langage SQL - TP n°2 - Énoncé

## Base de données

On utilise dans ce TP la base de données "world" (accessible sur le web).

La base de données contient de très nombreuses tables, trois documents présentent les caractéristiques de la base de données :

- Diagramme E-R (Entity-Relation) : tables (rectangles roses), attributs (ovales bleus) et tables de liaisons (losanges verts) qui permettent de relier plusieurs tables entre elles.
- RS (Relational schema) : définition des attributs des tables.
- Referential Dependencies (graphe des clés).

## Vocabulaire

### *Types de dépendances entre tables*

- **Relation un-à-un (one-to-one) notée 1-1** : un enregistrement d'une table A est relié à un unique enregistrement d'une autre table B via un couple clé primaire A / clé étrangère unique dans B (une clé étrangère unique est une clé qui ne peut apparaître que dans un seul enregistrement de la table B).  
Exemple : une personne (table A) ne peut avoir qu'une seule carte d'identité (table B).  
Réalisation pratique en SQL : `table_A AS A JOIN table_B AS B ON A.cléPrimaire=B.cléEtrangère` (celle-ci étant unique).
- **Relation un-à-plusieurs (one-to-many) notée 1-\*** : un enregistrement d'une table A est relié à un ou plusieurs enregistrements d'une autre table B via un couple clé primaire A / clé étrangère non unique dans B (la clé étrangère peut apparaître dans plusieurs enregistrements de la table B).  
Exemple : un auteur de roman (table A) peut écrire plusieurs livres (table B) mais un livre n'est associé qu'à un seul auteur.  
Réalisation pratique en SQL : `table_A AS A JOIN table_B AS B ON A.cléPrimaire=B.cléEtrangère` (celle-ci n'étant pas nécessairement unique).
- **Relation plusieurs-à-plusieurs (many-to-many) notée \*-\*** : deux tables A et B sont reliées via une troisième table appelée table de liaison ou d'association contenant deux clés étrangères reliées respectivement à une clé primaire dans chacune des tables A et B.  
Exemple : un rédacteur de presse (table A) peut écrire plusieurs articles (table B) et par ailleurs un même article (table B) peut avoir plusieurs rédacteurs (table A).  
Dans le TP n°1, la table Emprunts avec ses clés Qui et Quoi était une table de liaison entre la table Emprunteurs et la table Livres.  
Dans ce TP, la table encompasses est également une table de liaison.  
Ces tables sont les losanges verts dans le schéma E-R ci-dessus.  
Réalisation pratique en SQL : **double jointure** `table_A JOIN table_liaison ON ... JOIN table_B ON`  
....

# 1 Opérateur ensembliste UNION

 Requête n°1

Afficher la table *Continent* triée par ordre alphabétique.

 Requête n°1 - suite

Quelle requête doit-on écrire si souhaite n'afficher que les n-uplets concernant l'Afrique et l'Asie ?

## UNION

```
SELECT attributs FROM nom_table_1 WHERE formule_1
UNION
SELECT attributs FROM nom_table_2 WHERE formule_2
```

Remarque : UNION est plus générale que OR car elle permet de rassembler deux requêtes distinctes (qui doivent donc renvoyer le même nombre de colonnes avec les mêmes types d'attributs et dans le même ordre).

 Vérifier que la commande suivante donne bien les mêmes résultats précédents.

```
SELECT * FROM Continent WHERE Name='Africa'
UNION
SELECT * FROM Continent WHERE Name='Asia'
```

# 2 Fonctions d'agrégation derrière SELECT

## 2.1 Moyenne d'un attribut

 Requête n°2

Quelle est la moyenne des surfaces des continents ?

## 2.2 Recherche de renseignements associés à un extremum

La recherche du maximum d'un attribut est facile.

  Par exemple, pour déterminer la surface du continent le plus vaste :

```
SELECT MAX(Area)
FROM Continent
```

  Trouver le continent le plus vaste est également aisé : il suffit d'ajouter l'attribut correspondant dans les projections.

```
SELECT name, MAX(area)
FROM continent
```

 La recherche de renseignements associée à la fonction d'agrégation est parfois plus compliquée. La méthode la plus générale est illustrée avec ce même exemple (continent le plus vaste).

### Erreur fréquente

**ATTENTION : la requête "intuitive" suivante est incorrecte**

```
SELECT attribut_1 FROM table WHERE attribut_2 = MAX(attribut_2)
```

⇒ message d'erreur : **Error : misuse of aggregate function MAX().**

**La fonction d'agrégation doit obligatoirement se situer après l'instruction SELECT (avant FROM) ou bien après HAVING.**

 Requête n°3

Quel est le continent le plus vaste ?

## ⚠ *Méthode générale*

### Etape 1

Ecrire la requête recherchant l'aire maximum (sans autre renseignement).

### Etape 2

Il suffit de demander le nom dans la table Continent pour lequel Area est égale au résultat de la requête précédente (qui devient donc une sous-requête derrière WHERE)

Ecrire la requête complète.

### Conclusion

Dans le cas de l'égalité entre un attribut et sa valeur maximum, cette syntaxe est inutile comme précisé ci-dessus mais elle devient indispensable dans le cas d'une inégalité par exemple.

La requête suivante repose sur ce schéma :

```
SELECT ... FROM ... WHERE ... > (SELECT AVG(...) FROM ...)
```

etc...

### ✏ Requête n°4

Quels sont les continents dont la surface est supérieure à la surface moyenne des continents?

## 3 Filtrage du nombre de résultats - LIMIT

### 📖 *LIMIT*

```
SELECT ... FROM ... LIMIT entier_1, entier_2
```

### ✏ Requête n°5

- Afficher la table *Country* et constater qu'elle est longue...

- Tester la requête : `SELECT * FROM Country LIMIT 10`

### ✏ Requête n°5 bis

Que fait la requête `SELECT * FROM Country LIMIT 2, 5`?

Cette requête peut aussi s'écrire (mais pas dans ce notebook) : `SELECT * FROM Country OFFSET 2 LIMIT 5`

## 4 Calculs mathématiques sur les données

### 📖 *Attribut sous forme de formule*

Il est possible de créer un nouvel attribut en utilisant une formule à partir de plusieurs attributs d'une table.

```
SELECT formule entre attributs FROM ...
```

La formule est construite à partir des opérateurs `+` `-` `*` `/` et des fonctions mathématiques usuelles (`POWER(x, n)`, `ROUND(x, n)`, ...).

### ✏ Requête n°6

Afficher la liste des cinq pays d'Afrique les plus peuplés ainsi que leur population classés par ordre décroissant de la population.

### ✏ Requête n°7

Afficher la liste des pays d'Afrique ainsi que leur **densité de population** (habitants /  $km^2$ ) classés par ordre décroissant de la densité de population, en se limitant à une liste de cinq pays (on pourra utiliser un alias pour la colonne des calculs).

### ✏ Requête n°8

Quel est le pays d'Afrique le plus densément peuplé et quelle est sa densité de population?

### ✏ Requête n°8 bis

Quels sont les pays d'Afrique dont la densité de population est supérieure à la densité moyenne de ce continent?

## 5 Fonctions d'agrégation dans les regroupements derrière HAVING

Il est courant d'effectuer des calculs (somme, moyenne, max...) sur des regroupements d'enregistrements (ci-dessous, par continent par exemple).

Dans un tel cas, la fonction d'agrégation doit être placée dans la clause HAVING suivant l'instruction de regroupement GROUP BY.

 Requête n°9

Quels sont les continents dont la densité moyenne de population est supérieure à 50 hab/km<sup>2</sup>? Afficher également la densité.

Utiliser la fonction **ROUND**(*x*, *n*) afin d'obtenir des valeurs entières.

## 6 Exercices

 Exercice n°1

Dans combien de pays la langue française est-elle parlée?

 Exercice n°2

Quelles sont les langues parlées par moins de 10 000 personnes dans le monde (en tenant compte des pourcentages)? Afficher également le nombre de locuteurs.

 Exercice n°3

Afficher le nombre de pays dans le monde ainsi que la population mondiale.

 Exercice n°4

Combien y a-t-il de villes traversées par une rivière dans chaque pays? Classer sur le nombre.

 Exercice n°5

Quels sont les pays possédant plus de 20 villes traversées par une rivière? Classer par ordre alphabétique.

### *Chaînes et dates*

Il existe des fonctions SQL dédiées au traitement de dates mais elles ne figurent pas au programme de CPGE et elles ne sont pas implémentées dans ce notebook.

Il est néanmoins possible d'effectuer des opérations simples car les dates de cette base de données sont de type chaîne de caractères.

SUBSTRING (*chaîne*, *a*, *b*) permet d'extraire la sous-chaîne débutant à l'indice *a* (la numérotation débute à 1) et terminant à l'indice *b* inclus.

CAST (*chaîne* AS UNSIGNED) permet de convertir une chaîne en entier positif.

 Exercice n°6

Déterminer les organisations internationales (table "organization") créées après 2010, ainsi que leur date de création.

 Exercice n°7

Quelle est l'année de création de l'OTAN (NATO en anglais)?

 Exercice n°8

Combien d'organisations comportent 'Europe' dans leur nom?

 Exercice n°9

Lister le nombre de créations d'organisations internationales par année.