

Révisions sur les bases de données

Base de données Mondial

Voici un schéma relationnel extrait de la base de données Mondial <http://www.semwebtech.org/sqlfrontend/> sous la forme d'une liste de tables, chacune suivie par ses attributs.

Les attributs soulignés sont la clef primaire de la table et ceux en italique renvoient vers la clef primaire d'une autre table (qui porte en général le même nom, avec une majuscule). Les autres attributs sont de type chaîne de caractères ou nombre.

Pays, villes, continents et pays qui en font partie, frontières :

- **Country** : name, *code*, capital , area, population ; (code = 'F' pour la France) ;
- **City** : name, *country*, population ;
- **Continent** : name, area ;
- **Encompasses** : *country*, *continent* ;
- **Borders** : *country1*, *country2*, length ; (country1 < country2 pour l'ordre lexicographique).

Données politiques (régimes, organisations internationales et leurs membres) et économiques :

- **Politics** : *country*, government ;
- **Organization** : abbreviation, name, *city*, *country*, established ; (established est la date de fondation)
- **IsMember** : *country*, *organization* ;
- **Economy** : *country*, GDP, agriculture, service, industry, inflation ; (GDP est le PIB ; agriculture, service, industry sont des pourcentages du PIB ; le pays est dit "majoritairement agricole" quand agriculture \geq 50 ; idem pour service ou industry)

Montagnes, mers, rivières, lacs et les pays qui les bordent :

- **Mountain** : name, elevation, type ;
- **GeoMountain** : *mountain*, *country* ;
- **Sea** : name, depth ;
- **GeoSea** : *sea*, *country* ;
- **Lake** : name, area, depth, elevation, type ;
- **GeoLake** : *lake*, *country* ;
- **River** : name, river, lake, sea, length ; (chaque rivière se jette dans une rivière, un lac ou la mer ; deux de ces trois attributs ont donc la valeur NULL)
- **GeoRiver** : *river*, *country*.

Attention : *country* renvoie à l'attribut code de la table Country (et non à son attribut name).

Exercices

Dans chacune des questions suivantes on demande de rédiger une requête SQL pour obtenir la liste complète de ce qui est indiqué.

EXERCICE 1. Requêtes de base

- Q1 les continents et leurs superficies, triés par superficie croissante ;
- Q2 les capitales mondiales, triées par ordre alphabétique ;
- Q3 le nom et la ville de fondation des organisations mondiales triées de la plus récemment créée à la plus ancienne ;
- Q4 le nom des volcans (montagne de type 'volcano') de plus de 6000 m ;

EXERCICE 2. Jointures

- Q5 les noms des pays membres des Nations Unies (organization = 'UN') ;
- Q6 les noms de pays qui sont des républiques (gouvernement = 'republic'), triés par ordre alphabétique ;
- Q7 le nom des montagnes de France de plus de 4000 m ;
- Q8 le nom des montagnes africaines (Encompasses.continent = 'Africa') triées de la plus haute à la moins haute ;
- Q9 les fleuves (c'est-à-dire les rivières qui se jettent dans la mer) de France — sans répétition.
- Q10 les pays frontaliers de la France, triés par population décroissante ;

EXERCICE 3. Fonctions d'agrégation

- Q11 la longueur moyenne des fleuves qui se jettent dans la Mer Noire ('Black Sea') ;
- Q12 le nombre de rivières françaises présentes dans la base de données ;
- Q13 le nombre total de pays traversés par un fleuve qui se jette dans la mer Méditerranée ('Mediterranean Sea') ;
- Q14 la liste des pays des Amériques ('America') avec leur plus haute montagne et son altitude.
- Q15 la longueur terrestre de la frontière française ;

EXERCICE 4. Opérations ensemblistes

- Q16 le nom des pays non membres des Nations Unies ;
- Q17 les affluents directs du Nil ('Nile') ainsi que les affluents de ses affluents ;
- Q18 les paires de pays qui ont une frontière montagneuse ;

EXERCICE 5. Sélection a posteriori, requêtes imbriquées

- Q19 organisations regroupant plus de 100 pays, avec la population totale des pays qui forment ces organisations ;
- Q20 les pays majoritairement agricoles (d'après leur PIB) dont le taux d'inflation est inférieur à la moyenne mondiale ;
- Q21 le pays de chaque continent dont le taux d'inflation est le plus faible parmi les pays industriels.

Corrigé

EXERCICE 1. Requêtes de base

- Q1 les continents et leurs superficies, triés par superficie croissante ;
SELECT name, area FROM Continent ORDER BY area ASC
- Q2 les capitales mondiales, triées par ordre alphabétique ;
SELECT capital FROM Country ORDER BY capital ASC
- Q3 le nom et la ville de fondation des organisations mondiales triées de la plus récemment créée à la plus ancienne ;
**SELECT name, city, established FROM Organization
ORDER BY established DESC**
- Q4 le nom des volcans (montagne de type 'volcano') de plus de 6000 m ;
**SELECT name FROM Mountain
WHERE type = 'volcano' AND elevation >= 6000**

EXERCICE 2. Jointures

- Q5 les noms des pays membres des Nations Unies (organization = 'UN') ;
**SELECT Country.name FROM Country
JOIN IsMember ON Country.code = IsMember.country
WHERE organization = 'UN'**
- Q6 les noms de pays qui sont des républiques (gouvernement = 'republic'), triés par ordre alphabétique ;
**SELECT name FROM Country
JOIN Politics ON Country.code = Politics.country
WHERE government = 'republic'
ORDER BY name ASC**
- Q7 le nom des montagnes de France de plus de 4000 m ;
**SELECT name FROM Mountain
JOIN GeoMountain ON Mountain.name = GeoMountain.mountain
WHERE country = 'F' AND elevation >= 4000**
- Q8 le nom des montagnes africaines (Encompasses.continent='Africa') triées de la plus haute à la moins haute ;
**SELECT name, elevation FROM Mountain
JOIN GeoMountain ON Mountain.name = GeoMountain.mountain
JOIN Encompasses ON GeoMountain.country = Encompasses.country
WHERE continent = 'Africa'
ORDER BY elevation DESC**
- Q9 les fleuves (c'est-à-dire les rivières qui se jettent dans la mer) de France — sans répétition.
**SELECT DISTINCT name FROM River
JOIN GeoRiver ON River.name = GeoRiver.river
WHERE sea IS NOT NULL AND country = 'F'**
- Q10 les pays frontaliers de la France, triés par population décroissante ;
**SELECT Country.name, population FROM Country
JOIN Borders ON Country.code = Borders.Country2
WHERE Country1 = 'F'
UNION
SELECT Country.name, population FROM Country
JOIN Borders ON Country.code = Borders.Country1
WHERE Country2 = 'F'
ORDER BY population DESC**

EXERCICE 3. Fonctions d'agrégation

- Q11** la longueur moyenne des fleuves qui se jettent dans la Mer Noire ('Black Sea');
SELECT AVG(length) FROM River WHERE sea = 'Black Sea'
- Q12** le nombre de rivières françaises présentes dans la base de données;
SELECT COUNT(DISTINCT river) FROM GeoRiver WHERE country = 'F'
- Q13** le nombre total de pays traversés par un fleuve qui se jette dans la mer Méditerranée ('Mediterranean Sea');
SELECT Georiver.river, COUNT(country)
FROM GeoRiver
JOIN River ON GeoRiver.river = River.name
WHERE sea = 'Mediterranean Sea'
GROUP BY Georiver.river
- Q14** la liste des pays des Amériques ('America') avec leur plus haute montagne et son altitude.
 Une sous-requête détermine une liste de codes pays avec l'altitude maximale des montagnes de ces pays. Une requête principale donne une liste de pays et de montagnes avec leur altitude et vérifie si le code du pays ET l'altitude de chaque montagne se trouvent dans la liste de la sous-requête (mot clé IN). Si tel est le cas, il conserve la montagne.
SELECT Country.name, Mountain.name, Mountain.elevation
FROM GeoMountain
JOIN Country ON GeoMountain.country = Country.code
JOIN Mountain ON GeoMountain.mountain = Mountain.name
JOIN Encompasses ON GeoMountain.country = Encompasses.country
WHERE continent = 'America'
WHERE Country.code, Mountain.elevation
IN
(SELECT Country.code, MAX(Mountain.elevation)
FROM GeoMountain
JOIN Mountain ON GeoMountain.mountain = Mountain.name
JOIN Encompasses ON GeoMountain.country = Encompasses.country
WHERE continent = 'America'
GROUP BY GeoMountain.country)
- Q15** la longueur terrestre de la frontière française;
SELECT SUM(length) FROM Borders WHERE country1 = 'F' OR country2 = 'F'

EXERCICE 4. Opérations ensemblistes

- Q16** le nom des pays non membres des Nations Unies;
SELECT name FROM Country
EXCEPT
SELECT name FROM Country
JOIN IsMember ON Ismember.country = Country.code
WHERE organization = 'UN'
- Q17** les affluents directs du Nil ainsi que les affluents de ses affluents;
SELECT name FROM River WHERE river = 'Nile'
UNION
SELECT R2.name FROM River AS R2
JOIN River AS R1 ON R2.river = R1.name
WHERE R1.river = 'Nile'

Q18 les paires de pays qui ont une frontière montagneuse ;
SELECT country1, country2 **FROM** Borders
JOIN GeoMountain **ON** Borders.country1 = Geomontains.country
INTERSECT
SELECT country1, country2 **FROM** Borders
JOIN GeoMountain **ON** Borders.country2 = Geomontains.country

EXERCICE 5. Sélection a posteriori, requêtes imbriquées

Q19 organisations regroupant plus de 100 pays, avec la population totale des pays qui forment ces organisations ;
SELECT Organization.name, SUM(population)
FROM Country
JOIN IsMember **ON** Country.code = IsMember.country
JOIN Organization **ON** IsMember.organization = Organization.abbreviation
GROUP BY Organization.abbreviation
HAVING COUNT(Country) >= 100

Q20 les pays majoritairement agricoles (d'après leur PIB) dont le taux d'inflation est inférieur à la moyenne mondiale ;
SELECT Country.name
FROM Country
JOIN Economy **ON** Country.code = Economy.country
WHERE agriculture >= 50
AND inflation <= (SELECT AVG(inflation) **FROM** Economy)

Q21 le pays de chaque continent dont le taux d'inflation est le plus faible parmi les pays industriels.
SELECT Country.name
FROM Country
JOIN Economy **ON** Country.code = Economy.country
JOIN Encompasses **ON** Country.code = Encompasses.country
WHERE inflation =
(SELECT MIN(inflation) **FROM** Economy **WHERE** industry >= 50)
GROUP BY Encompasses.continent