

Base de Données & Langage SQL

Correction TP : Interface Access 2016 et requêtes SQL

Exercice : fleuves du monde

L'objectif de cet exercice est de créer une Base de Données contenant plusieurs tables sous MS Access et d'interroger cette BD avec des requêtes simples en utilisant le langage SQL.

Dans la base de données FleuvesMonde, on considère le schéma relationnel suivant (**en gras les clés primaires**) :

1. Continent (**num_continent**, nomC, superficieC, populationC)
 2. Pays (**num_pays**, nomP, superficieP, populationP)
 3. Fleuve (**num_fleuve**, nomF, longueur)
 4. Appartient (**num_pays, num_continent**, pourcent), où num_pays est une clé étrangère qui référence la relation "pays" et num_continent la relation "continent". L'attribut pourcent indique la proportion de la surface du pays appartenant au continent (ex. la Turquie est à 10% en Europe et à 90% en Asie).
 5. Traverse (**num_fleuve, num_pays**) Où num_fleuve est une clé étrangère qui référence la relation "fleuve" et num_pays la relation "pays".
- Dans la base de données **FleuvesMonde**, créer les tables correspondantes au schéma relationnel.

```
CREATE TABLE Continent (  
num_continent AUTOINCREMENT,  
nomC VARCHAR(25) NOT NULL,  
superficieC INT,  
populationC INT,  
PRIMARY KEY (num_continent));  
  
CREATE TABLE Pays (  
num_pays AUTOINCREMENT,  
nomP VARCHAR(15) NOT NULL,  
superficieP INT,  
populationP INT,  
PRIMARY KEY (num_pays));  
  
CREATE TABLE Fleuve (  
num_fleuve AUTOINCREMENT,  
nomF VARCHAR(15) NOT NULL,  
longueurF INT,  
PRIMARY KEY (num_fleuve));  
  
CREATE TABLE Appartient (  
num_pays INT,  
num_continent INT,  
pourcentage INT,  
PRIMARY KEY (num_pays, num_continent));  
  
CREATE TABLE Traverse (  
num_fleuve INT,  
num_pays INT,  
PRIMARY KEY (num_fleuve, num_pays));  
  
CREATE TABLE Appartient (  
num_pays INT,  
num_continent INT,  
pourcentage INT,  
FOREIGN KEY(num_pays)REFERENCES pays(num_pays),
```

```
FOREIGN KEY(num_continent) REFERENCES continent(num_continent),
PRIMARY KEY (num_pays, num_continent));
```

```
CREATE TABLE Traverse (
num_fleuve INT,
num_pays INT,
FOREIGN KEY(num_fleuve)REFERENCES fleuve(num_fleuve),
FOREIGN KEY(num_pays) REFERENCES pays(num_pays),
PRIMARY KEY (num_fleuve, num_pays));
```

- Peupler les tables avec les données suivantes :

```
INSERT INTO Continent (num_continent, nomC, superficieC, populationC)
VALUES(1,'Afrique', '29600000', '720000000');
INSERT INTO Continent (num_continent, nomC, superficieC, populationC) VALUES(2,
'Amerique du Nord', '18400000', '296000000');
INSERT INTO Continent (num_continent, nomC, superficieC, populationC) VALUES(3,
'Amerique du Sud', '20000000', '480000000');
INSERT INTO Continent (num_continent, nomC, superficieC, populationC) VALUES(4,
'Europe', '10000000', '620000000');
INSERT INTO Continent (num_continent, nomC, superficieC, populationC) VALUES(5,
'Asie', '42000000', '350000000');
INSERT INTO Continent (num_continent, nomC, superficieC, populationC) VALUES(6,
'Océanie', '9000000', '30000000');
```

```
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('USA', '9631418', '295734134');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Perou', '1285220', '27925628');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Bresil', '8511965', '186112794');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Australie', '7686850', '20090437');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Egypte', '1001450', '77505756');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Inde', '3287590', '1080264388');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Allemagne', '357021', '82431390');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Soudan', '2505810', '40187486');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Hongrie', '93030', '10006835');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Autriche', '83870', '8184691');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Ouganda', '236040', '27269482');
INSERT INTO Pays (nomP, superficieP, populationP)
VALUES('Kenya', '582650', '33829590');
```

```
INSERT INTO Fleuve (nomF, longueurF) VALUES('Nil', '6671');
INSERT INTO Fleuve (nomF, longueurF) VALUES('Le Gange', '2700');
INSERT INTO Fleuve (nomF, longueurF) VALUES('Danube', '2500');
INSERT INTO Fleuve (nomF, longueurF) VALUES('Murray', '2574');
INSERT INTO Fleuve (nomF, longueurF) VALUES('Mississipi', '6260');
INSERT INTO Fleuve (nomF, longueurF) VALUES('Amazone', '6400');
```

```
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES('1', '2', '100');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES('2', '3', '100');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES('3', '3', '100');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES('4', '6', '100');
```

```

INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES ('5', '1', '90');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES ('6', '5', '100');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES ('7', '4', '100');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES ('8', '1', '100');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES ('9', '4', '100');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES ('10', '4', '100');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES ('11', '1', '100');
INSERT INTO Appartient (num_pays, num_continent, pourcentage)
VALUES ('12', '1', '100');

```

```

INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('1', '5');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('1', '8');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('1', '11');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('1', '12');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('2', '6');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('3', '7');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('3', '9');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('3', '10');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('4', '4');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('5', '1');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('6', '2');
INSERT INTO Traverse (num_fleuve, num_pays) VALUES ('6', '3');

```

Si les clés étrangères n'ont pas été définies lors de la création des tables, créer et exécuter une seule fois les requêtes suivantes :

```

-- Num_fleuve est clé étrangère dans la table Traverse
ALTER TABLE Traverse
ADD FOREIGN KEY (num_fleuve) REFERENCES Fleuve(num_fleuve);

-- Num_pays est clé étrangère dans la table Traverse
ALTER TABLE Traverse
ADD FOREIGN KEY (num_pays) REFERENCES Pays(num_pays);

-- Num_pays est clé étrangère dans la table Appartient
ALTER TABLE Appartient
ADD FOREIGN KEY (num_pays) REFERENCES Pays(num_pays);

-- Num_continent est clé étrangère dans la table Appartient
ALTER TABLE Appartient
ADD FOREIGN KEY (num_continent) REFERENCES Continent(num_continent);

```

Requêtes

A partir du menu Créer, Création de requêtes, Affichage SQL, écrire en SQL les requêtes suivantes (sauvegarder les requêtes avec un numéro et un nom significatif).

- 1) Afficher les fleuves de plus de 3000km de long.

```

-- 01 Afficher les fleuves de plus de 3000km de long.
SELECT nomF
FROM Fleuve
WHERE longueurF > 3000 ;

```

- 2) Afficher les pays dont la population dépasse les 10 millions d'habitants.

```

-- 02 Afficher les pays dont la population dépasse les 10 millions d'habitants.

```

```
SELECT *
FROM Pays
WHERE populationP>10000000;
```

- 3) Calculer et afficher la densité de la population des pays affichés par ordre alphabétique.

```
-- 03 Calculer et afficher la densité de la population des pays affcihés par
ordre alphabétique.
SELECT nomP, (populationP/superficieP) AS densiteP
FROM Pays
ORDER BY nomP;
```

- 4) Afficher les fleuves traversant des pays de plus de 100 mille km2 par ordre décroissant de la superficie

```
-- 04 Afficher les fleuves traversant des pays de plus de 100 mille km2 par
ordre décroissant de la superficie
SELECT nomF, nomP, superficieP
FROM (Pays INNER JOIN Traverse ON Pays.num_pays=Traverse.num_pays )
INNER JOIN Fleuve ON Fleuve.num_fleuve=Traverse.num_fleuve
WHERE superficieP>100000
ORDER BY superficieP DESC;
```

- 5) Afficher les fleuves qui traversent des pays d'Asie

```
-- 05 Afficher les fleuves qui traversent des pays d'Asie
SELECT nomF, nomP
FROM (((Pays INNER JOIN Traverse ON Pays.num_pays=Traverse.num_pays)
INNER JOIN Fleuve ON Traverse.num_fleuve =Fleuve.num_fleuve)
INNER JOIN Appartient ON Pays.num_pays=Appartient.num_pays)
INNER JOIN Continent ON Appartient.num_continent=Continent.num_continent
WHERE nomC="Asie";
```

- 6) Afficher les pays et leurs fleuves respectifs par ordre alphabétique.

```
-- 06 Afficher les pays et leurs fleuves respectifs par ordre alphabétique.
SELECT nomP, nomF
FROM (Pays INNER JOIN Traverse ON Pays.num_pays=Traverse.num_pays )
INNER JOIN Fleuve ON Fleuve.num_fleuve=Traverse.num_fleuve
ORDER BY nomP, nomF;
```

- 7) Afficher la superficie des continents dont un pays est traversé par le Nil.

```
-- 07 Afficher la superficie des continents dont un pays est traversés par le
Nil.
SELECT DISTINCT nomC, superficieC
FROM (((Pays INNER JOIN Traverse ON Pays.num_pays=Traverse.num_pays)
INNER JOIN Fleuve ON Traverse.num_fleuve =Fleuve.num_fleuve)
INNER JOIN Appartient ON
Pays.num_pays=Appartient.num_pays)
INNER JOIN Continent ON
Appartient.num_continent=Continent.num_continent
WHERE nomF="Nil";
```

- 8) Lister les pays et leur capitale et les ranger par continent et par ordre alphabétique

```
-- 08 Lister les pays et leur capitale et les ranger par continent et par ordre
alphabétique
SELECT nomC, nomP, capitale
FROM (Pays INNER JOIN Appartient ON Pays.num_pays=Appartient.num_pays)
INNER JOIN Continent ON Appartient.num_continent=Continent.num_continent
ORDER BY nomC, nomP;
```

9) Ajouter le champ « capitale » dans la table pays.

```
-- 09 Ajouter le champ « capitale » dans la table pays.  
ALTER TABLE Pays  
ADD capitale VARCHAR(30);
```

10) Insérer des données dans ce nouveau champ et pour des tuples déjà présents.

```
-- 10 Insérer des données dans ce nouveau champ et pour des tuples déjà  
présents.  
UPDATE Pays SET capitale='Washington' WHERE nomP='USA';  
UPDATE Pays SET capitale='Lima' WHERE nomP='Perou';  
UPDATE Pays SET capitale='Rio de Janeiro' WHERE nomP='Bresil';  
UPDATE Pays SET capitale='Camberra' WHERE nomP='Australie';  
UPDATE Pays SET capitale='Le Caire' WHERE nomP='Egypte';  
UPDATE Pays SET capitale='New Delhi' WHERE nomP='Inde';  
UPDATE Pays SET capitale='Berlin' WHERE nomP='Allemagne';  
UPDATE Pays SET capitale='Khartoun' WHERE nomP='Soudan';  
UPDATE Pays SET capitale='Budapest' WHERE nomP='Hongrie';  
UPDATE Pays SET capitale='Vienne' WHERE nomP='Autriche';  
UPDATE Pays SET capitale='Kampala' WHERE nomP='Ouganda';  
UPDATE Pays SET capitale='Nairobi' WHERE nomP='Kenya';
```

11) Afficher les continents et leurs pays qui ont une densité supérieure à 100

```
-- 11 Afficher les continents et leurs pays qui ont une densité supérieure à  
100  
SELECT nomC, nomP, (populationP/superficieP) AS densite  
FROM ((Pays INNER JOIN Appartient ON Pays.num_pays=Appartient.num_pays)  
      INNER JOIN Continent ON  
      appartient.num_continent=Continent.num_continent)  
WHERE (populationP/superficieP)>100;
```

12) Afficher le nombre de pays qui ont plus de 200 millions d'habitants et leur continent associé.

```
-- 12 Afficher le nombre de pays qui ont plus de 200 millions d'habitants et  
leur continent associé.  
SELECT nomC, COUNT(populationP) AS population_importante  
FROM (Pays  
      INNER JOIN Appartient ON  
      Pays.num_pays=Appartient.num_pays)  
      INNER JOIN Continent ON  
      Appartient.num_continent=Continent.num_continent  
WHERE (populationP)>200000000  
GROUP BY nomC;
```

13) Calculer et afficher la superficie et la population mondiale.

```
-- 13 Calculer et afficher la superficie et la population mondiale.  
SELECT SUM(superficieC) AS SuperficieM, SUM(populationC) AS PpopulationM  
FROM Continent;
```

14) Longueur moyenne de tous les fleuves.

```
-- 14 Longueur moyenne de tous les fleuves.  
SELECT AVG (longueurF) AS longueur_moyenne_des_fleuves  
FROM Fleuve;
```

15) Les pays dont le nom commence par un A

```
-- 15 Les pays dont le nom commence par un A
```

```

SELECT NomP
FROM Pays
WHERE NomP LIKE 'A*';

```

16) Les continents et les fleuves qui les traversent

```

-- 16 Continents et fleuves les traversent
SELECT DISTINCT nomC, nomF
FROM (((Pays INNER JOIN Traverse ON Pays.num_pays=Traverse.num_pays)
      INNER JOIN Fleuve ON Traverse.num_fleuve =Fleuve.num_fleuve)
      INNER JOIN Appartient ON
Pays.num_pays=Appartient.num_pays)
      INNER JOIN Continent ON
Appartient.num_continent=Continent.num_continent
ORDER BY nomC;

```

17) Le nom des fleuves qui le traversent l'Europe

```

-- 17 Le nom des fleuves qui traversent l'Europe
SELECT DISTINCT nomC, nomF
FROM (((Pays INNER JOIN Traverse ON Pays.num_pays=Traverse.num_pays)
      INNER JOIN Fleuve ON Traverse.num_fleuve =Fleuve.num_fleuve)
      INNER JOIN Appartient ON
Pays.num_pays=Appartient.num_pays)
      INNER JOIN Continent ON
Appartient.num_continent=Continent.num_continent
WHERE Continent.nomC = "Europe"

```

18) Les fleuves de plus de 1500 km traversant un pays de plus de 15 millions d'habitants

```

-- 18 Les fleuves de plus de 1500 km traversant un pays de plus de 15 millions
d'habitants
SELECT nomF, nomP, populationP, longueurF
((Pays INNER JOIN Traverse ON Pays.num_pays=Traverse.num_pays)
  INNER JOIN Fleuve ON Traverse.num_fleuve =Fleuve.num_fleuve)
WHERE longueurF >1500
AND populationP >15000000

```

19) Trouver et afficher le nom du fleuve le plus long.

```

-- 19 Trouver et afficher le nom du fleuve le plus long.
SELECT NomF AS Fleuve_Le_Plus_Long, longueurF AS Sa_Longueur
FROM Fleuve
WHERE longueurF =
(SELECT Max(longueurF) FROM Fleuve);

```

20) Calculer et afficher le nombre de fleuves par continent.

```

-- 20 Calculer et afficher le nombre de fleuves par continent.
SELECT nomC, COUNT(Fleuve.num_fleuve) AS nbfleuves
FROM ( SELECT DISTINCT nomC, Fleuve.num_fleuve
      FROM (((Pays INNER JOIN Traverse ON
Pays.num_pays=Traverse.num_pays)
      INNER JOIN Fleuve ON Traverse.num_fleuve =Fleuve.num_fleuve)
      INNER JOIN Appartient ON
Pays.num_pays=Appartient.num_pays)
      INNER JOIN Continent ON
Appartient.num_continent=Continent.num_continent)
GROUP BY Continent.nomC;

```