exercices sql

Table des matières

EX1....3
1 - Affichez le nom de la région dont dépend Toulouse en plus des informations ci dessus.....3
2 - Lister les villes dont le nom commence par TOU dans les départements comprenant le nom Garonne (haute).....3
3 - Compter le nombre de lignes que vous retourne la recherche de la liste des départements dans la région Centre.....3
4 - Lister les villes qui ont un 'u' en 3ème position et 'r' en 4ême position dans leur nom et afficher leur régions.....3
5 - Compléter cette requête pour trouver les villes en Indre et Loire.....4
6 - Lister les villes et leur région d'appartenance, traversées par le 45ème parallèle (45.00).....4
7 - Pont-de-l'Isère ne figurant pas dans la liste, après recherche des informations nécessaire sur Wikipédia, insérez cette ville dans la base de données.....4
Ex2....5
1 - Reconstruire la base de données avec le moteur InnoDB, permettant ainsi de gérer les relations et les transactions dans la base. Vous aurez à donc à implémenter les contraintes liées aux relations des différentes tables existantes.....5
2 - Répondre au besoin d'actualiser cette base en ajoutant les nouvelles régions.....6

EX1

1 - Affichez le nom de la région dont dépend Toulouse en plus des informations ci dessus.

```
SELECT
nom AS 'Nom du bled',
cp AS 'Code postal',
nom_departement AS 'Département' ,
nom_region AS 'Région'
FROM maps_ville V
INNER JOIN departement D ON V.id_departement = D.id_departement
INNER JOIN region R ON D.id_region = R.id_region
WHERE nom LIKE 'TOULOUSE';
```

2 - Lister les villes dont le nom commence par TOU dans les départements comprenant le nom Garonne (haute).

```
SELECT
nom AS 'Nom du bled',
cp AS 'Code postal',
nom_departement AS 'Département',
nom_region AS 'Région'
FROM maps_ville V
INNER JOIN departement D ON V.id_departement = D.id_departement
INNER JOIN region R ON D.id_region = R.id_region
WHERE nom LIKE 'Tou%'
AND nom_departement LIKE 'GARONNE (HAUTE)';
```

3 - Compter le nombre de lignes que vous retourne la recherche de la liste des départements dans la région Centre.

```
SELECT COUNT(*)
FROM departement D
INNER JOIN region R ON D.id_region = R.id_region
WHERE nom_region LIKE 'Centre';
```

4 - Lister les villes qui ont un 'u' en 3ème position et 'r' en 4ême position dans leur nom et afficher leur régions.

```
SELECT
nom AS 'Nom du bled',
nom_region AS 'Région'
```

```
FROM maps_ville V
INNER JOIN departement D ON V.id_departement=D.id_departement
INNER JOIN region R ON D.id_region=R.id_region
WHERE nom LIKE '__UR%';
```

5 - Compléter cette requête pour trouver les villes en Indre et Loire.

```
SELECT
nom AS 'Nom du patelin',
nom_region AS 'Région'
FROM maps_ville V
INNER JOIN departement D ON V.id_departement = D.id_departement
INNER JOIN region R ON D.id_region = R.id_region
WHERE nom LIKE '__UR%'
AND nom_departement LIKE 'Indre et Loire';
```

6 - Lister les villes et leur région d'appartenance, traversées par le 45ème parallèle (45.00).

```
SELECT
nom AS 'Nom du bled',
nom_region AS 'Région'
FROM maps_ville V
INNER JOIN departement D ON V.id_departement = D.id_departement
INNER JOIN region R ON D.id_region = R.id_region
WHERE lat LIKE '45.00%';
```

7 - Pont-de-l'Isère ne figurant pas dans la liste, après recherche des informations nécessaire sur Wikipédia, insérez cette ville dans la base de données.

```
INSERT INTO maps_ville (id_departement, nom, cp, lat, lon)
VALUE (24, 'Pont-de-l\'Isère', 26600, 45.0020, 45.203);
```

Ex2

1 - Reconstruire la base de données avec le moteur InnoDB, permettant ainsi de gérer les relations et les transactions dans la base. Vous aurez à donc à implémenter les contraintes liées aux relations des différentes tables existantes.

```
CREATE TABLE IF NOT EXISTS `test`.`nouvelles_regions` (
`id_nouvelle_region` INT NOT NULL AUTO_INCREMENT,
`nom_nouvelle_region` VARCHAR(45) NULL,
PRIMARY KEY (`id_nouvelle_region`))
ENGINE = InnoDB
DEFAULT CHARACTER SET = utf8
COLLATE = utf8_general_ci;

INSERT INTO nouvelles_regions (nom_nouvelles_regions)
VALUES ('Auvergne-Rhône-Alpes'),
('Bourgogne-Franche-Comté'),
('Bretagne'),
('Centre-Val de Loire'),
('Corse'),
('Grand Est'),
('Hauts-de-France'),
('Île-de-France'),
('Normandie'),
('Nouvelle-Aquitaine'),
('Occitanie'),
('Pays de la Loire'),
('Provence-Alpes-Côte d\'Azur'),
('Guadeloupe'),
('Martinique'),
('Guyane'),
('La Réunion'),
('Mayotte');
```

2 - Répondre au besoin d'actualiser cette base en ajoutant les nouvelles régions.

```
UPDATE region
SET id_nouvelle_region = 1
WHERE id_region = 2
OR id_region = 22;

UPDATE region
SET id_nouvelle_region = 2
WHERE id_region = 3
OR id_region = 9;

UPDATE region
SET id_nouvelle_region = 3
WHERE id_region = 4;

UPDATE region
SET id_nouvelle_region = 4
WHERE id_region = 5;

UPDATE region
SET id_nouvelle_region = 5
WHERE id_region = 7;

UPDATE region
SET id_nouvelle_region = 6
WHERE id_region = 6
OR id_region = 13
OR id_region = 23;

UPDATE region
SET id_nouvelle_region = 7
WHERE id_region = 15
OR id_region = 19;

UPDATE region
SET id_nouvelle_region = 8
WHERE id_region = 10;

UPDATE region
SET id_nouvelle_region = 9
WHERE id_region=17
OR id_region=24;

UPDATE region
SET id_nouvelle_region = 10
WHERE id_region = 1
```

```
OR id_region = 12
OR id_region = 20;


UPDATE region
SET id_nouvelle_region = 11
WHERE id_region = 14
OR id_region = 11;
UPDATE region SET id_nouvelle_region = 12
WHERE id_region = 18;
UPDATE region
SET id_nouvelle_region = 13
WHERE id_region = 21;
UPDATE region
SET id_nouvelle_region = 14
WHERE id_region=8;
```

3 – test des transactions

Au début du script :

```
-- on désactive l'autocommit
SET AUTOCOMMIT = 0;
-- on commence la transaction
START TRANSACTION ;
```

pour annuler la transaction :

```
ROLLBACK ;
```