

IR IG1- 2016-17– Base de données – TD#3

1. Récupérer la base de données 'mondial' et exécuter le script sur votre compte sous postgresql.
2. Rédiger les requêtes suivantes:
 - a. Donner le nom, le code et la capital des pays (respectivement name, code et capital de la relation country) dont le nom (name) débute par la lettre U.
 - b. Donner le nom, le code et la population des pays dont la population est de plus de 200 million d'habitants.
 - c. Donner le nom des villes (name de city) dont le pays est USA (country dans city)
 - d. Donner le nom des villes (name de city) dont le nom du pays est 'United States' (name de country).
 - e. Pour chaque pays, fournir le code du pays et son nombre de villes.
 - f. Pour chaque pays, fournir le code du pays et son nombre de villes mais uniquement pour les pays avec plus de 20 villes.
 - g. Idem question f mais on veut afficher le nom du pays à la place de son code (soit United States au lieu de USA)
 - h. Afficher le nom des rivières et leur longueur pour les rivières françaises.
 - i. A partir de la table 'is_member', donner pour chaque organisation, le nombre de pays impliqués, dans l'ordre décroissant du nombre de pays impliqué.
 - j. Donner le nom des pays qui sont impliqués dans l'organisation mondiale de la santé (WHO).
 - k. Donner la liste des pays (les noms) qui sont dans l'organisation WHO et dans l'organisation NATO (OTAN)
 - l. Idem requête 'i' mais uniquement pour les organizations ayant au moins 100 pays impliqués.
 - m. Rédiger la contrainte d'intégrité du type foreign key entre is_member et organization
 - n. Idem requête 'l' mais en affichant le libellé des organizations
 - o. Afficher les pays qui sont impliqués dans toutes les organizations de la requête l.
 - p. Trouver une requête permettant de vérifier si votre requête 'o' donne de bons résultats.
 - q. Afficher le nom des pays et le nombre de mers de ceux-ci pour les pays ayant au moins 3 mers. Affichage dans l'ordre décroissant du nombre de mers. On pourra créer une vue à l'aide de l'instruction 'CREATE VIEW nom_vue AS une requête select'.
 - r. A l'aide de la table 'economy', afficher toutes les informations économiques des pays ayant plus de 2 mers.
 - s. Donner la moyenne (avg), le minimum (min) et le maximum (max) de la colonne industry des pays concernés par la requête 'r'.
 - t. Afficher le nom des pays et le pourcentage de la population habitant sur les provinces des côtes (des mers) pour les pays européen. Vous pouvez créer une view. Voyez vous une erreur dans les résultats.
Pour faire une division avec un resultat réel:
`cast(a as real) / cast( b as real) as real)`

Correction

2.a

`select * from country where name ilike 'U%';`

(7 réponses)

2.b.

`select name, code, population from country where population > 200000000 ;`

(4 réponses)

2.c.

`select name from city where country ilike 'usa';`

(27 réponses)

2.d.

`select ci.name from city ci, country co where co.name ilike 'United States' and ci.country=co.code;`

(27 réponses)

2.e.

`select country, count(*) from city group by country;`

(169 réponses)

2.f.

`select country, count(*) from city group by country HAVING count(*)>20;`

(3 réponses)

2.g.

`select co.name, count(*) from city ci, country co where ci.country=co.code group by co.name having count(*)>20;`

`select co.name, count(*) from city ci JOIN country co ON ci.country=co.code GROUP BY co.name having count(*)>20;`

(3 réponses)

2.h.

`select distinct gr.river, length from geo_river gr, river r where r.name= gr.river and country ilike 'F%';`

(5 rows)

2.i. `select organization, count(*) from is_member group by organization order by count(*) desc;`

(152 rows)

2.j. `select co.name from is_member i, country co where organization ilike 'WHO' and co.code=i.country;`

(186 rows)

2.k.

`select co.name from is_member i, country co, is_member i2 where i.organization ilike 'WHO' and i2.organization ilike 'NATO' and co.code=i.country and co.code=i2.country;`

(16 rows)

2.l.

`select organization, count(*) from is_member group by organization having count(*)>100 order by count(*) desc;`

(30 rows)

2.m.

`ALTER TABLE is_member ADD CONSTRAINT "is_member_organization_fkey" FOREIGN KEY (organization) REFERENCES organization(abbreviation) ;`

2.n.

`select name, count(*) from is_member, organization where abbreviation=organization group by`

name having count(*)>100 order by count(*) desc;
(30 rows)

2.o.

select code from country where not exists ((select organization from is_member group by organization having count(*)>100) except (select organization from is_member where country=code));

(19 rows)

2.p.

select organization from is_member group by organization having count(*)>100 except select organization from is_member where country ilike 'MS';

2.q.

create view seaco as select distinct country, sea from geo_sea;

select co.name, count(*) from country co, seaco s where s.country=code group by name having count(*)>=3 order by count(*) desc;

(13 rows) russia 7

2.r.

select * from economy e where country in (select country from seaco s group by country having count(*)>=3 order by count(*) desc);

2.s.

select min(industry), max(industry), avg(industry) from economy where country in (select country from seaco s group by country having count(*)>=3 order by count(*) desc);

t.

SELECT p.country, sum(p.population) AS sum

FROM province p, geo_sea s

WHERE s.country::text = p.country::text AND s.province::text = p.name::text

GROUP BY p.country;

select code, sum , population, cast(cast(sum as real)/cast(population as real) as real) as t from country c, costpop p where p.country=code and code in (select country from is_member where organization ilike 'eu');

Problème : sum de dk et irl car 1 province pour 2 mers dont compte double. Et taux =2 ?