

Objectifs

Réaliser des requêtes classiques et des requêtes avec différentes formes de sous-requêtes SQL sous Postgres.

préambule

La connexion au SGBD PostgreSQL en mode interactif se fait par la commande :

```
psql -h psqlserv da2i -U monlogin
```

L'exécution d'un fichier de requêtes se fait avec la commande `\i nomfichier`

Durant ce TP nous allons créer une nouvelle base de gestion des vols d'un aéroport. Pour cela nous utiliserons les tables suivantes :

avion(ano, type, places, compagnie)
pilote(pno, nom, prenom, adresse)
ligne(lno, depart, arrivee)
vol(ano, pno, lno, hdep, harr)

Récupérez les scripts `tp02_tables.sql` et `tp02_data.sql` sur Moodle et exécutez les.

1 Requêtes SQL simples

1. Lister la table des pilotes

.....
.....

```
select * from pilote;
```

2. Lister les noms de tous pilotes

.....
.....

```
select noms from pilote;
```

3. Lister les différentes compagnies référencées

.....
.....

```
select distinct compagnie form avion;
```

4. Lister les noms, prénoms et noms de login des pilotes (constitué des 7 premiers caractères du nom ainsi que l'initiale du prenom, le tout en majuscule)¹

.....
.....

```
select nom, prenom, upper(substring(nom,1,7) || substring(prenom,1,1) as login from pil
```

1. Postgres possède la fonction `substring(string [from int] [for int])` pour extraire une sous-chaine, la fonction `upper` pour mettre en majuscule, et `||` pour concaténer deux chaînes.

5. Lister les avions triés par ordre descendant du nombre de places

.....
.....

```
select * from avion order by place desc;
```

6. Lister les vols triés par heure de départ croissante

.....
.....

```
select * from vol order by hdep;
```

7. Liste des lignes au départ de lille

.....
.....

```
select * from ligne where depart='lille';
```

8. liste des avions qui ont entre 100 et 200 places

.....
.....

```
select * from avion where places between 100 and 200;
```

9. Liste les pilotes dont le nom commence par 'H'

.....
.....

```
select * from pilote where nom like 'H%';
```

10. Liste des vols de moins de 45mn

.....
.....

```
select * from vol where harr-hdep <= '00:45:00';
```

11. Lister les vols triés par durée de vol décroissante

.....
.....

```
select * from vol order by harr-hdep desc;
```

12. Lister les numéros d'avions qui ont des vols à partir de Paris

.....
.....

```
select ano from vol v, ligne l
where v.lno = l.lno and depart='Paris';
```

13. De l'importance de mettre les bonnes tables dans la clause from !

Créer une table bidon (nb int) . Ajouter 5 lignes dedans.

— Combien de lignes donne select * from pilote;

22

— Combien de lignes donne select * from pilote, bidon;

110

— Combien de lignes donne `select * from pilote as l1, pilote as l2, pilote as l3;`

```
223 = 10648
```

14. Liste des noms de pilotes dont la totalité d'un vol est entre 12h et 14h00 inclus

.....
.....

```
select * from pilote p , vol v
where p.pno=v.pno
and v.hdep>='12:00:00' and v.harr<='14:00:00'
```

15. Lister les numéros, types et places d'avions qui ont des vols à partir de Lille, Nice ou Marseille

.....
.....

```
select ano, type, place
from avion a, ligne l, vol v
where a.ano=v.ano and ligne.lno=v.lno
depart in ('Lille','Nice','Marseille');
```

16. Liste des vols au départ de lille

.....
.....

```
select * from vol v, ligne l
where v.lno = l.lno and depart='Lille';
```

17. Lister les compagnies qui partent de Lille

.....
.....

```
select distinct compagnie from avion a, ligne l , vol v
where a.ano=v.ano and l.lno=v.ano and depart=lille;
```

18. Lister les compagnies qui desservent Lille (au départ comme à l'arrivée)

.....
.....

```
select distinct compagnie from avion a, ligne l , vol v
where a.ano=v.ano and l.lno=v.lno and (depart='Lille' or arrivee='Lille');
```

19. Lister les couples de lignes permettant d'aller de Lille à Marseille avec une seule escale

.....
.....

```
select l1.lno, l2.lno
from ligne l1, ligne l2
where l1.depart='Lille' and l1.arrivee=l2.depart
and l2.arrivee='Marseille';
```

20. Lister les compagnies qui permettent d'aller de Lille à Marseille en une escale sans changer d'avion

.....
.....

```
select a.compagnie,l1.lno, l2.lno
from ligne l1, vol v1, ligne l2, vol v2, avion a
where l1.depart='Lille' and l1.arrivee=l2.depart
and l2.arrivee='Marseille'
and v1.lno=l1.lno and v2.lno=l2.lno
and a.ano = v1.ano
and v1.ano=v2.ano ;
```

21. Lister le nombre de vols dans la base de données

.....
.....

```
select count(*) from vol ;
```

22. Lister le nombre d'occurrences de chaque prénom de pilote trié par nombre décroissant

.....
.....

```
select prenom, count(*)  
from pilote  
group by prenom  
order by count(*) desc ;
```

23. Lister le nombre de vols par numero d'avion

.....
.....

```
select ano, count(*)  
from vol  
group by ano ;
```

24. Lister la moyenne des places disponibles par compagnie

.....
.....

```
select compagnie, avg(places)  
from avion  
group by compagnie ;
```

25. Lister le cumul des places disponibles par compagnie

.....
.....

```
select compagnie, sum(places)  
from avion  
group by compagnie ;
```

26. Lister la moyenne des durées de vols par numéro de pilote

.....
.....

```
select pno, avg(hdep-harr)  
from vol  
group by pno ;
```

27. Lister les initiales de chaque pilote avec leurs occurrences listées par ordre décroissant des occurrences

.....
.....

```
select substring(nom,1,1)||substring(prenom,1,1) as initiales , count(*) as compteur  
from pilote  
group by substring(nom,1,1)||substring(prenom,1,1)  
order by count(*) desc ;
```

28. Lister le nombre de vols par nom de pilote décroissant

.....
.....

```
select nom, count(*)
from vol v, pilote p
where v.pno = p.pno
group by nom
order by nom desc ;
```

29. Lister le nombre de vols accessible au départ de chaque ville

.....

.....

```
select depart, count(*)
from vol v, ligne l
where v.lno = l.lno;
group by depart ;
```

30. Lister la dernière heure de départ pour chaque nom de pilote

.....

.....

```
select nom, max(hdep)
from vol v, pilote p
where v.pno = p.pno
group by nom ;
```

31. Lister les compagnies qui ont plus de 2000 places au total

.....

.....

```
select compagnie
from avions
group by compagnie
having sum(places) > 2000 ;
```

32. Lister les numeros de pilotes qui effectuent plus de 3 vols

.....

.....

```
select pno from vol
group by pno
having count(*) > 3 ;
```

33. Lister les numeros de pilotes qui volent entre 4 et 8 heures au total

.....

.....

```
select pno from vol
group by pno
having sum(harr-hdep) between '04:00:00' and '8:00:00' ;
```

34. Lister les villes d'où partent au moins 2 vols

.....

.....

```
select depart
from ligne l , vol v
where v.lno = l.lno
group by depart
having count(*) >= 3 ;
```

35. Lister les avions qui volent plus de 10 heures au total au départ de Lille

.....
.....

```
select a.ano
from avion a , vol v, ligne l
where v.ano=a.ano and v.lno=l.lno and depart="Lille"
group by a.ano
having sum(harr-hdep) > '10:00:00';
```

36. Lister les lignes sur lesquelles il y a plus de 400 places au total

.....
.....

```
select l.lno
from ligne l, vol v, avion a
where v.lno = l.lno and v.ano=a.ano
group by lno
having sum(places) > 1000 ;
```

2 Requêtes SQL évoluées

1. Lister les pilotes qui habitent la même ville que "Wabinski"

(a) Solution avec jointures

.....
.....

```
select p1.nom
from pilote p1, pilote p2
where p1.ville = p2.ville
and p2.nom='Wabinski';
```

(b) Solution avec sous requête

.....
.....

```
select *
from pilote
where adresse = (select adresse
                  from pilote
                  where nom = 'Wabinski') ;
```

2. Lister les avions qui ont plus de place que le plus gros porteur de chez Air France.

.....
.....

```
select *
from avion
where place > (select max(place)
               from avions
               where compagnie = 'Air France') ;
```

3. Afficher le nombre d'avions qui possèdent le maximum de places dans notre base.

.....
.....

```
select count(*)
from avions
where place = (select max(place)
               from avions) ;
```

4. Lister les vols menés par des pilotes qui démarrent parfois de Lille (avec au moins un vol au départ de Lille)

(a) Solution avec jointures

.....
.....

```
select vl.*  
from vol as v1, vol as v2, ligne as l  
where v1.pno = v2.pno  
and v2.lno = l.lno  
and l.depart='Lille' ;
```

(b) Solution avec sous requête

.....
.....

Non corrélative :

```
select * from vol  
where pno in  
  (select pno  
   from ligne as l, vol as v  
   where l.lno = v.lno  
   and depart = 'Lille') ;
```

non correlative, 2 niveaux de sous-requetes :

```
select * from vol where pno in  
(select pno from vol where lno=  
(select lno from ligne where depart='Lille')) ;
```

5. Lister les numéros des avions qui sont au nombre de places minimum de leur compagnie (corrélative)

.....
.....

```
select *  
from avion a1 where  
place in  
  (select min(place)  
   from avion a2  
   where a2.compagnie=a1.compagnie ;
```

aucun Group by n'est nécessaire dans la sous-requete puisqu'elle est corrélative.

6. Lister les avions pilotés toujours par le même pilote (corrélative)

.....
.....

```
-- pour lesquels il n'existe pas d'autre pilote qui le piloterait  
select ano  
from vol v1  
where not exists (select * from  
                  vol v2  
                  where v2.pno<>v1.pno and v1.ano=v2.ano) ;
```

ou sinon

```
select ano from vol  
group by ano  
having count(distinct pno)=1 ;
```

7. Lister les villes d'arrivée à plus de 2h00 de vol de toutes les autres villes. (corrélative)

.....
.....

```
-- Pour lesquelles il n'y a aucune ville à moins de 2h00
select arr
from ligne l1
where not exists (select * from vol v2, ligne l2
                  where l2.lno = v2.lno
                  and l2.arr = l1.arr
                  and harr - interval '2 hours' <= hdep ;
```

8. Lister la ville d'arrivée la plus desservie pour chaque compagnie (corrélative)

.....

.....

```
select arr, compagnie
from ligne l1, vol v1, avion a1
where v1.lno=l1.lno and v1.ano=a1.ano
group by arr, compagnie
having count(*) >= all (select count(*)
                       from ligne l2, vol v2, avion a2
                       where v2.lno=l2.lno and v2.ano=a2.ano
                       and a2.compagnie = a1.compagnie      -- CORRELATION
                       group by arr, compagnie ;
```

9. Lister les repos (cumul des temps entre deux vols) de chaque pilote

.....

.....

```
SELECT vl.pno, SUM(v2.hdep-v1.harr)
FROM vol v1, vol v2
WHERE v1.pno = v2.pno AND v1.hdep < v2.hdep
AND v2.hdep = (SELECT MIN(v3.hdep) FROM vol v3
              WHERE v3.pno = v1.pno AND v1.harr < v3.hdep)
GROUP BY vl.pno
ORDER BY vl.pno;
```

10. lister les couples de compagnies ayant le même nombre de vols

.....

.....

```
SELECT T1.compagnie, T2.compagnie, T1.nb_vol
FROM
(SELECT compagnie, COUNT(*) AS nb_vol
 FROM vol vl, avion al
 WHERE vl.ano = al.ano
 GROUP BY compagnie) T1
INNER JOIN
(SELECT compagnie, COUNT(*) AS nb_vol
 FROM vol vl, avion al
 WHERE vl.ano = al.ano
 GROUP BY compagnie) T2
ON (T1.nb_vol = T2.nb_vol)
WHERE T1.compagnie < T2.compagnie;
```