Exoplus n°6

Objectifs
Fonction d'agrégations, Group By et having
Section
M2106: IUT Campus 3 - Département « Informatique »
Auteur
E.Porcq
Références
1-Pres_TDF.pdf
2-cours_sql.pdf.
3-sql_tdf_exercices.pdf

Tester et commenter les requêtes suivantes. Certaines requêtes sont fausses. Les commenter.

1) Utilisation des fonctions d'agrégation et de group by

```
Select * from vt_etape;
Select n_etape from vt_etape;
Select moyenne from vt_etape;

Select count(*) from vt_etape;
Select count(*) as nb from vt_etape;
Select count(n_etape) as nb from vt_etape;
Select count (moyenne) as nb from vt_etape;

Select max(total_seconde) as maxi, min(total_seconde) as mini,
round(avg(total_seconde),4) as moyenne from vt_temps;

Select sum(total_seconde) as somme,count(total_seconde) as nombre from vt_temps;

Select sum(total_seconde) as somme,count(*) as nombre from vt_temps;

select n_coureur from VT_PARTI_COUREUR order by n_coureur;

select count(*) from VT_PARTI_COUREUR ;

select count(*),n_coureur from VT_PARTI_COUREUR order by n_coureur;

select n_coureur, count(*) as nb from VT_PARTI_COUREUR
group by n_coureur
order by nb desc;

select n_coureur,n_equipe,count(*) as nb
from VT_PARTI_COUREUR
group by n_coureur
order by nb,n_equipe desc;

select n_coureur,n_equipe,count(*) as nb_tours from VT_PARTI_COUREUR
group by n_equipe ,n_coureur
order by 1,2 desc;
```

2) Différence entre where et having

```
select n_coureur from vt_temps order by n_coureur desc;
select distinct n_coureur from vt_temps  order by n_coureur desc;
select count(*) from vt_temps ;

select n_coureur from vt_temps where n_coureur > 1760;
select n_coureur from vt_temps having n_coureur > 1760;

select n_coureur from vt_temps where n_coureur > 1760  group by n_coureur ;
select n_coureur from vt_temps having n_coureur > 1760  group by n_coureur ;

select count(*) from vt_temps where n_coureur > 1760  ;
select count(*) from vt_temps having n_coureur > 1760  ;

select count(*) from vt_temps where n_coureur > 1760  group by n_coureur ;
select count(*) from vt_temps having n_coureur > 1760 group by n_coureur ;
select count(*) as nb from vt_temps where count(*) > 300  group by n_coureur ;

select count(*) from vt_temps having count(*) > 300 group by n_coureur ;

select n_coureur from vt_temps having count(*) > 300 group by n_coureur ;

select sum(total_seconde),avg(total_seconde) from vt_temps having max(total_seconde) > 31000;
```

3) Utilisation de having

```
select n_coureur,n_equipe,count(*) as nb_par from VT_PARTI_COUREUR
group by n_coureur,n_equipe
where  count(*) > 12
order by n_coureur;

select n_coureur,n_equipe,count(*) as nb_par from VT_PARTI_COUREUR
group by n_coureur,n_equipe
where nb_par > 12
order by n_coureur;

select n_coureur from VT_PARTI_COUREUR
group by n_coureur
having count(*) >12
order by n_coureur;

select n_coureur from VT_PARTI_COUREUR
group by n_coureur
having count(*) = 18;

select n_coureur from VT_PARTI_COUREUR
group by n_coureur
having count(*) =
(
select max(count(n_coureur)) from VT_PARTI_COUREUR
group by n_coureur
);
```

4) Utilisation de rollup et cube

```
select annee, cat_code,  count(*) as nb_km from vt_etape
group by rollup (annee,cat_code)
order by annee desc,nb_km desc;

select annee, cat_code,  count(*) as nb_km from vt_etape
group by cube (annee,cat_code)
order by annee desc,nb_km desc;
```
