

Bases de Données - TD6

Licence Informatique 2006-2007

Sous-requêtes et requêtes d'aggrégation

◇ **Exercice 1. Tour de France**

Soit la schéma relationnel suivant :

EQUIPE(codeEquipe, nomEquipe, directeurSportif)

COUREUR(numero, nomCoureur, cEquipe, cPays)

PAYS(codePays, nomPays)

ETAPE(numero, dateEtape, villeDepart, villeArrive, nbKm)

TEMPS(numeroCoureur, numeroEtape, tempsRealis)

Donnez les requêtes SQL correspondant aux expressions suivantes :

1. Il existe deux types d'étapes : les contre la montre où les coureurs sont seuls et qui font moins de 60 km et les étapes en ligne où tous les coureurs partent en même temps et qui font plus de 100 km. Quelles sont les étapes en ligne ?

```
SELECT * FROM ETAPES WHERE NbKms >= 100;
```

2. Quelles sont les étapes de contre la montre ?

```
SELECT * FROM ETAPES WHERE NbKms <= 60;
```

3. Quels sont, pour chaque étape, les coureurs (numéros) ayant participé à l'étape.

```
select NumeroCoureur, NumeroEtape from ETAPES;
```

Fonctions d'agregation

COUNT

4. Le nombre total de coureurs

AVG et SUM

5. Le temps moyen et le nombre total de kilomètres parcourus par le coureur numéro 32 sur l'ensemble du tour

MAX

6. L'étape comportant le plus grand nombre de kilomètres

MIN

7. Le nom des coureurs ayant réalisé le meilleur temps réalisé pour l'étape numéro 10

AVG

8. La moyenne kilométrique des coureurs pour l'étape numéro 10

Clause GROUP BY

9. Le nombre de coureurs par équipe
10. Le meilleur temps réalisé par équipe pour l'étape numéro 10
11. Le meilleur temps réalisé par équipe par étape

CLAUSE HAVING

12. Le nombre de coureurs des équipes ayant au moins 10 coureurs
13. Donner la liste des coureurs (numéros) ayant participé à toutes les étapes.

```
select NumeroCoureur from temps
group by numeroCoureurs
having count(*) = (select count(*) from etapes);
```

14. La moyenne kilométrique sur l'ensemble du tour par coureur des coureurs ayant participé à au moins 10 étapes

```
select NumeroCoureur, AVG(tempsRealis) from temps
group by numeroCoureurs
having count(*) > 10
```

15. A la fin de chaque étape, le coureur dont le temps total est le plus faible obtient le maillot jaune. Quel coureur est maillot jaune à la fin de l'étape k ? (Numéro du coureur).
16. Quel coureur a gagné le tour de France ? (Numéro du coureur).

```
select NumeroCoureur from temps
group by numeroCoureurs
having count(*) = (select count(*) from etapes)
and sum(tempsRealis) <= all (select sum(tempsRealis)
                             from temps
                             group by numeroCoureurs
                             having count(*) = (select count(*) from etapes) );
```

17. Donner la liste des coureurs (noms et numéros) ayant gagné au moins une étape.

```
A = SELECT MIN(TempsRealise) AS TPS, NumeroEtape FROM TEMPS
GROUP BY NumeroEtape;
B = SELECT DISTINCT NumeroCoureur from A, TEMPS
WHERE (A.NumeroEtape = TEMPS.NumeroEtape) AND (A.TPS = TEMPS.TempsRealise);
```

18. Donner la liste des coureurs (noms et numéros) ayant gagné au moins une étape et ayant participé à toutes les étapes.

```

select nomcoureur, numerocoureur
from coureurs
where numerocoureur in (select NumeroCoureur
                        from temps t1
                        where tempsRealis = (select min(tempsRealis)
                        from temps t2
                        where t2.etape = t1.etape);)
and numerocoureur in ( select NumeroCoureur from temps
                      group by numeroCoureur
                      having count(*) = (select count(*) from etapes));

```

19. Pour chaque équipe dont au moins un des coureurs a terminé le tour de France, donner le numéro du coureur le mieux classé.

```

select cequipe, numeroCoureur
from coureurs c
where numerocoureur in ( select NumeroCoureur from
                        group by numeroCoureur
                        having count(*) = (select count(*) from etapes))
and numerocoureur in (select NumeroCoureur
from temps, coureurs
where temps.numerocoureur = coureurs.numerocoureur
and coureurs.cequipe = c.cequipe
                        group by numeroCoureur
                        having count(*) = (select count(*) from etapes)
and sum(tempsRealis) <= all (select sum(tempsRealis)
from temps, coureurs
where temps.numerocoureur = coureurs.numerocoureur
and coureurs.cequipe = c.cequipe
                        group by numeroCoureur
                        having count(*) = (select count(*) from etapes) ));

```

20. Quelle est la vitesse moyenne en contre la montre pour chaque coureur ?

```

A = SELECT * FROM ETAPES WHERE NbKms <= 60;
B = SELECT TEMPS.NumeroCoureur,TEMPS.NumeroEtape,TEMPS.TempsRealise,A.NbKm FROM A,TEMPS
WHERE (TEMPS.NumeroEtape = A.NumeroEtape);
C = SELECT SUM(TempsRealise)/SUM(NbKm) FROM B
GROUP BY NumeroCoureur;

```

21. Quelle est la vitesse moyenne sur les étapes en ligne pour chaque coureur ?

```

A = SELECT * FROM ETAPES WHERE NbKms >= 100;
B = SELECT TEMPS.NumeroCoureur,TEMPS.NumeroEtape, TEMPS.TempsRealise,A.NbKm FROM A,TEMPS
WHERE (TEMPS.NumeroEtape = A.NumeroEtape);
C = SELECT SUM(TempsRealise)/SUM(NbKm) FROM B
GROUP BY NumeroCoureur;

```

22. Chaque équipe comporte 10 coureurs au départ. Quelles équipes ont fini le tour de France au complet ?

```

A = SELECT Max(NumeroEtape) AS MaxEtape FROM ETAPES;
B = SELECT Coureur.NumeroCoureur,Coureur.CodeEquipe
      FROM A,COUREURS,TEMPS WHERE (A.MaxEtape = Temps.NumeroEtape)
      AND (TEMPS.NumeroCoureur = Coureurs.NumeroCoureur);
C = SELECT COUNT(NumeroCoureur) AS NCoureur, CodeEquipe FROM B
      GROUP BY CodeEquipe;
D = SELECT * FROM C WHERE NCoureur = 10;
E = SELECT NomEquipe FROM Equipe,D WHERE D.CodeEquipe = Equipe.CodeEquipe;

```

23. Qui a fini dernier du tour de France ?

```

A = Table Tps de ceux qui ont fini;
B = SELECT SUM(TempsRealise),NumeroCoureur AS TPS FROM A
      GROUP BY NumeroCoureur;
C = SELECT NomCoureur FROM B,Coureur
      WHERE Coureur.NumeroCoureur = B.NumeroCoureur;

```

24. Quels sont les français ayant terminé le tour de france ?

```

F = SELECT * FROM Coureurs WHERE CodePays = 'FRA';
B = Table de ceux qui ont fini
C = SELECT NomCoureur FROM F,B WHERE F.CodeCoureur = B.CodeCoureur;

```