

SQL – CAS TOUR DE FRANCE

EQUIPE (CodeEquipe, NomEquipe, DirecteurSportif)
COUREUR (NuméroCoureur, NomCoureur, CodeEquipe*, CodePays*)
PAYS (CodePays, NomPays)
TYPE_ETAPE (CodeType, LibelléType)
ETAPE (NuméroEtape, DateEtape, VilleDép, VilleArr, NbKm, CodeType*)
PARTICIPER (NuméroCoureur*, NuméroEtape*, TempsRéalisé)
ATTRIBUER_BONIFICATION (NuméroEtape*, km, Rang, NbSecondes, NuméroCoureur*)

1 - Quelle est la composition de l'équipe FESTINA (Numéro, nom et pays des coureurs) ?

```
R1=SELECTION(EQUIPE, NomEquipe="FESTINA")
R2=JOINTURE(R1, COUREUR, R1.CodeEquipe=COUREUR.CodeEquipe)
R3=JOINTURE(R2, PAYS, R2.CodePays=PAYS.CodePays)
R4=PROJECTION(R3, NuméroCoureur, NomCoureur, NomPays)
Et en langage SQL...
SELECT NuméroCoureur, NomCoureur, NomPays
FROM EQUIPE A INNER JOIN COUREUR B ON A.CodeEquipe=B.CodeEquipe
INNER JOIN PAYS C ON B.CodePays=C.CodePays
WHERE NomEquipe="FESTINA" ;
```

2 - Quel est le nombre de kilomètres total du Tour de France 97 ?

```
N=CALCULER(ETAPE, SOMME(NbKm))
Et en langage SQL...
SELECT SUM(Nbkm) FROM ETAPE ;
```

3 - Quel est le nombre de kilomètres total des étapes de type HAUTE MONTAGNE ?

```
R1=SELECTION(TYPE_ETAPE, LibelléType="HAUTE MONTAGNE")
R2=JOINTURE(R1, ETAPE, R1.CodeType=ETAPE.CodeType)
N=CALCULER(R2, SOMME(NbKm))
Et en langage SQL...
SELECT SUM(Nbkm)
FROM ETAPE A INNER JOIN TYPE_ETAPE B ON A.CodeType=B.CodeType
WHERE LibelléType="HAUTE MONTAGNE" ;
```

4 - Quels sont les noms des coureurs qui n'ont pas obtenu de bonifications ?

```
R1=PROJECTION(COUREUR, NuméroCoureur)
R2=PROJECTION(ATTRIBUER_BONIFICATION, NuméroCoureur)
R3=DIFFERENCE(R1,R2)
R4=JOINTURE(R3, COUREUR, R3.NuméroCoureur=COUREUR.NuméroCoureur)
R5=PROJECTION(R4, NomCoureur)
Et en langage SQL...
SELECT NomCoureur
FROM COUREUR
WHERE NuméroCoureur NOT IN (SELECT NuméroCoureur FROM ATTRIBUER_BONIFICATION) ;
ou
SELECT NomCoureur
FROM COUREUR C LEFT JOIN ATTRIBUER_BONIFICATION A ON
C.NuméroCoureur=A.NuméroCoureur
WHERE A.NuméroCoureur IS NULL ;
```

5 - Quels sont les noms des coureurs qui ont participé à toutes les étapes ?

R1=PROJECTION(PARTICIPER, NuméroCoureur, NuméroEtape)

R2=PROJECTION(ETAPE, NuméroEtape)

R3=DIVISION(R1, R2)

R4=JOINTURE(R3, COUREUR, R3.NuméroCoureur=COUREUR.NuméroCoureur)

R5=PROJECTION(R4, NomCoureur)

ou

N=CALCULER(ETAPE, Comptage())

R1=REGROUPER_ET_CALCULER(PARTICIPER, NuméroCoureur, Nb:Comptage())

R2=SELECTION(R1, Nb=N)

R3=JOINTURE(R2, COUREUR, R2.NuméroCoureur=COUREUR.NuméroCoureur)

R4=PROJECTION(R3, NomCoureur)

Et en langage SQL...

SELECT NomCoureur

FROM PARTICIPER A INNER JOIN COUREUR B ON A.NuméroCoureur=B.NuméroCoureur

GROUP BY NuméroCoureur, NomCoureur

HAVING COUNT(*)=(SELECT COUNT(*) FROM ETAPE) ;

6 - Quel est le classement général des coureurs (nom, code équipe, code pays et temps des coureurs) à l'issue des 13 premières étapes sachant que les bonifications ont été intégrées dans les temps réalisés à chaque étape ?

R1=SELECTION(PARTICIPER, NuméroEtape=13)

R2=PROJECTION(R1, NuméroCoureur)

-> R2 représente l'ensemble des coureurs présents jusqu'à la 13ème étape (ce qui va permettre de ne pas prendre en compte dans le classement ceux qui ont abandonné avant !)

R3=JOINTURE(R2, PARTICIPER, R2.NuméroCoureur=PARTICIPER.NuméroCoureur)

R4=SELECTION(R3, NuméroEtape<=13)

R5=REGROUPER_ET_CALCULER(R4, NuméroCoureur, Total:Somme(TempsRéalisé))

R6=JOINTURE(R5, COUREUR, R5.NuméroCoureur=COUREUR.NuméroCoureur)

R7=PROJECTION(R6, NomCoureur, CodeEquipe, CodePays, Total)

R8=TRI(R7, Totalé)

Et en langage SQL...

SELECT NomCoureur, CodeEquipe, CodePays, SUM(TempsRéalisé) AS Total

FROM PARTICIPER A INNER JOIN COUREUR B ON A.NuméroCoureur=B.NuméroCoureur

WHERE NuméroEtape<=13 AND A.NuméroCoureur IN

(SELECT NuméroCoureur FROM PARTICIPER WHERE NuméroEtape=13)

GROUP BY A.NuméroCoureur, NomCoureur, CodeEquipe, CodePays ORDER BY 4 ;