

Corrigé : ED SQL

Exercice 1

R1: Nom et prime des joueurs sponsorisés par Peugeot entre 1985 et 1990.

```
select distinct NomJoueur, Prime
from GAIN
where NomSponsor = 'Peugeot'
and Année between 1985 and 1990;
```

La clause Distinct demande l'élimination des doubles après projection sur NomJoueur et Prime. Le prédicat between est équivalent à l'expression suivante: $1985 \leq \text{Année} \leq 1990$.

R2: Nom et âge des joueurs ayant participé au tournoi de Roland Garros de 1989.

```
select J.Nom, J.Age
from JOUEUR J, GAIN G
where G.LieuTournoi = 'RolandGarros'
and G.Année = 1989
and J.Nom = G.NomJoueur;
```

R3: Nom et nationalité des joueurs sponsorisés par Peugeot et ayant gagné à Roland Garros.

Première interprétation: le sponsoring de Peugeot et le gain d'une rencontre à Roland Garros ne coïncident pas nécessairement (sponsoring pour ce tournoi, cette année la).

```
select J.Nom, J.Nationalité
from JOUEUR J, RENCONTRE R, GAIN G
where R.LieuTournoi = 'Roland Garros'
and G.Sponsor = 'Peugeot'
and J.Nom = R.NomGagnant
and J.Nom = G.NomJoueur;
```

Deuxième interprétation: les joueurs ayant gagné une rencontre à Roland Garros au moment où ils sont sponsorisés par Peugeot.

```
select J.Nom, J.Nationalité
from JOUEUR J, RENCONTRE R, GAIN G
where R.LieuTournoi='Roland Garros'
and G.Sponsor='Peugeot'
and J.Nom = R.NomGagnant
and J.Nom = G.NomJoueur
and G.Année=R.Année
and G.LieuTournois=R.LieuTournois;
```

R4: Nom et nationalité des joueurs ayant participé à la fois au tournoi de Roland Garros et à celui de Wimbledon, en 1985.

```
select J.Nom, J.Nationalité
from JOUEUR J, GAIN G
where G.LieuTournoi='RolandGarros' and G.Année=1985
and J.Nom = G.NomJoueur
Intersect
select J.Nom, J.Nationalité
from JOUEUR J, GAIN G
where G.LieuTournoi='Wimbledon' and G.Année=1985
and J.Nom = G.NomJoueur;
```

Si la clause Intersect n'est pas implantée dans le système utilisé, on peut exprimer la même requête sans l'opérateur d'intersection. Il suffit d'exprimer le fait qu'un élément appartient à l'intersection de X et Y s'il appartient simultanément à l'un et à l'autre des deux ensembles.

```
select Nom, Nationalité
from JOUEUR
where Nom In (select distinct NomJoueur
              from GAIN
              where LieuTournoi='RolandGarros'
                and Année=1985)
and Nom In (select Distinct NomJoueur
            from GAIN
            where LieuTournoi='Wimbledon'
              and Année=1985);
```

La clause **in** teste l'appartenance d'une valeur de Nom à un ensemble de NomJoueur calculé par la sous-requête dans la parenthèse suivante.

On peut aussi exprimer cette requête en définissant deux variables sur la relation GAIN. Tout se passe alors comme si cette relation a été dupliquée pour vérifier le critère 'RolandGarros' dans l'une, et le critère 'Wimbledon' dans l'autre.

```
select J.Nom, J.Nationalité
from JOUEUR J, GAIN G1, GAIN G2
where J.Nom=G1.NomJoueur
and G1.LieuTournoi='RolandGarros'
and G1.Année=1985
and J.Nom=G2.NomJoueur
and G2.LieuTournoi='Wimbledon' and G2.Année=1985;
```

R5: Nom des joueurs ayant toutes leurs primes des tournois de Roland Garros supérieures à 1 MF.

Première interprétation : un joueur n'ayant pas participé à tous les tournois de Roland Garros est néanmoins pris en compte dans le résultat.

```
select Distinct NomJoueur
from GAIN
where LieuTournoi='RolandGarros'
Except
select Distinct NomJoueur from GAIN
where LieuTournoi='RolandGarros' and Prime<=1MF;
```

Cette requête peut aussi s'exprimer de la façon suivante, si la clause **except** n'existe pas:

```
select NomJoueur
from GAIN
where LieuTournoi='RolandGarros'
and NomJoueur Not In
      (select Distinct NomJoueur
       from GAIN
       where LieuTournoi='RolandGarros'
         and Prime<=1MF);
```

On peut aussi l'exprimer en imposant que la plus petite prime est supérieure à 1 MF. La requête suivante donne la nouvelle expression.

```

select NomJoueur
from GAIN
where LieuTournoi='RolandGarros'
group by NomJoueur
having Min(Prime)>1MF;

```

Deuxième interprétation : un joueur n'ayant pas participé à tous les tournois de Roland Garros n'est pas pris en compte dans le résultat.

```

select Distinct NomJoueur
from GAIN G
where G.LieuTournoi = 'RolandGarros' and G.Prime>1MF
and G.NomJoueur not in(select Distinct NomJoueur
from GAIN
where LieuTournoi='RolandGarros'
and Prime<=1MF)
Having Count(G.Année)=(select Count(Distinct R.Année)
from RENCONTRE R
where G.NomJoueur=R.NomPerdant
or G.NomJoueur=R.NomGagnant);

```

Cette interprétation est excessive si on considère que la relation RENCONTRE répertorie les tournois depuis leur création ; ce qui représente plusieurs dizaines d'années pour certains. Ceci va au-delà de la carrière d'un joueur. La première formulation est donc la plus proche de la réalité, mais elle ne permet pas d'éliminer les joueurs ayant sauté un tournoi à Roland Garros durant leur carrière.

R6: Nom, prénom, âge et nationalité des joueurs ayant participé à tous les Rolland Garros.

```

select J.Nom, J.Prénom, J.Age, J.Nationalité
from JOUEUR J, GAIN G
where G.LieuTournoi='RolandGarros'
and J.Nom=G.NomJoueur
group by G.NomJoueur
Having Count(Distinct G.Année)=(select Count(Distinct R.Année)
from GAIN
where LieuTournoi='RolandGarros');

```

R7: Somme des primes gagnées pour chaque édition de Roland Garros ayant accueilli plus de 100 joueurs.

```

select Sum(Prime)
from GAIN
where LieuTournoi='RolandGarros'
group by Année
Having Count(Distinct NomJoueur)= 100;

```

R8 : La moyenne des primes gagnées par année

```

select Année, avg(all Prime)
from GAIN
group by Année

```

La fonction **avg** calcule la moyenne des primes pour chaque partition de la relation générée par la clause **group by**.

Exercise 2

a) Quels aliments a mangé Félix le 31 décembre 1999 ?

```
Select n°aliment
From Animaux, PlatMenu,
Where animaux.nom = 'Felix'
And Animaux.race = PlatMenu.race
And jour = '31-12-99';
```

b) Combien Dolly a t-elle eu d'enfants ?

```
Select count(N° animal enfant)
From filiation, Animaux
Where filiation.n°animal mère = Animaux.n°animal
And Animaux.nom animal = 'Dolly';
```

c) Quels aliments interdits a mangé Postit ?

```
Select PlatMenu.n°aliment
From Aliments_interdits, animaux, PlatMenu
Where animaux.nom = 'Postit'
And animaux.race = aliments_interdits.race
And aliment_interdits.n°aliment = PlatMenu.n°aliment
And animaux.race = PlatMenu.race;
```

d) Quelle race d'animaux n'a aucun aliment interdit ?

```
Select race
From animaux
Minus
Select race
From aliments_interdits;
```

e) Sachant que le 25 décembre 2000 les chiens n'ont eu qu'un seul repas où ils n'ont mangé qu'un seul aliment (du chocolat), quelle quantité de glucides ont-ils absorbé chacun ce jour là ?

```
Select quantité * glucide
From platmenu, aliment
Where jour = '25-12-2000'
And race = 'chien'
And PlatMenu.n°aliment = Aliment.n°aliment;
```

f) Ecrire la requête SQL permettant de voir s'il n'y a pas eu d'erreur avec l'insertion d'un animal femelle comme "père" dans la relation "filiation", ou d'un animal mâle comme mère dans la relation filiation.

```
Select animaux.n° animal
From animaux, filiation
Where animaux.n°animal = filiation.n°animal pere
And animaux.sexe = 'femelle'
UNION
Select animaux.n° animal
From animaux, filiation
Where animaux.n°animal = filiation.n°animal mere
```

```
And animaux.sexe = 'mâle';
```