

MPSI 1 : D.S. 3 d'Informatique Commune SQL

- 1) Une clé d'une table est un attribut ou un ensemble d'attributs K tel que si on projette la table sur cet ensemble d'attribut K (autrement dit qu'on ne garde que les colonnes correspondant à K) toutes les lignes de la table obtenue restent distinctes. Autrement dit : deux lignes de la table initiale sont distinctes si, (et seulement si), leurs valeurs sur K sont distinctes.

Une clé primaire est une clé qu'on a choisi pour cette table et déclaré comme telle, ce qui donnera des contraintes aux utilisateurs remplissant la table par exemple.

- 2) Pour identifier un match, puisque les match ne sont pas numérotés, on se dit qu'on a besoin de tout : son année, son tournoi et les deux joueuses du match ! Donc une clé primaire naturelle est formée des quatre attributs.

En fait, comme dans un tournoi donné d'une année donnée, chaque joueuse ne peut perdre qu'un match (et que dans chaque match il y a une perdante) , on peut enlever un attribut et identifier le match par tournoi, année et le numéro de la perdante.

Remarque pour la suite : lorsqu'on voudra utiliser ces attributs comme clés étrangères pour **gain** on utilisera **tournoi**, **annee** et l'égalité de la valeur **num** avec la valeur de *l'un* des deux attributs **numg** ou **nump**. Quelque chose comme **gain.num=match.numg and gain.num=match.nump** donnera une réponse vide !

- 3) En fait l'énoncé aurait dû être sympa et demander seulement le nom et le prénom de toutes les joueuses françaises ce qui aurait donné la gentille requête suivante :

```
SELECT nom, prenom
FROM joueuse
WHERE nationalite='française'
```

Hélas, il y a eu un vilain « ayant participé à Roland Garros » qui n'a pas été enlevé au montage... ce qui donne, à froid, une requête avec jointure c'est méchant, mais après tout cela arrive dans les concours les sujets pas progressifs... notez qu'ensuite les questions sont plus progressives. Ici la réponse était donc :

```
SELECT nom, prenom FROM joueuse j JOIN gain g ON g.num=j.num
WHERE j.nationalite="française" AND g.tournoi="Roland Garros"
```

Remarque : si on utilise la table **match** plutôt que la table **gain** c'est plus compliqué car la jointure doit être faite entre (**joueuse.num** et **match.numg**) OU (**joueuse.num** et **match.nump**) ce qui donne :

```
SELECT nom, prenom FROM joueuse j JOIN match ON match.numg=j.num OR match.nump=j.num
WHERE j.nationalite="française" AND g.tournoi="Roland Garros"
```

Ou bien encore on fait une double jointure avec **match** :

```
SELECT nom, prénom FROM joueuse j
JOIN match m ON m.numg=j.num
JOIN match m ON m.nump=j.num
WHERE nationalite="française " AND tournoi=" Roland Garros "
```

- 4) Comme on ordonne année par année, il est naturel de faire apparaître l'attribut **annee** dans le résultat.

```
SELECT DISTINCT tournoi, annee
FROM gain
WHERE annee >= 2017
ORDER BY annee
```

- Attention à ne pas confondre **ORDER BY** et **GROUP BY**. Ce dernier ne s'utilise qu'avec les opérateurs d'agrégations **MAX**, **MIN**, **SUM**, **AVG**, **COUNT** donc pas ici.
- On peut aussi remplacer ici la table **gain** par la table **match** qui contient aussi ces deux attributs **tournoi**, **annee**.

- 5)

```
SELECT DISTINCT num, tournoi, annee
FROM gain
WHERE sponsor='Lacoste' AND annee >= 2000 AND annee <= 2017
```

- 6)

```
SELECT COUNT(*) AS nb
FROM match
WHERE tournoi='Roland Garros' AND annee=2017
```
- 7)

```
SELECT SUM(prime) AS Total
FROM gain
WHERE tournoi='Roland Garros' AND annee=2017
```
- 8)

```
SELECT annee, AVG(prime) AS MoyPrime
FROM gain GROUP BY annee
```
- 9)

```
SELECT DISTINCT num
FROM gain
WHERE prime=(SELECT MAX(prime) FROM gain WHERE annee=2017)
AND annee=2017
```

Ici, ne pas oublier que la condition `annee=2017` doit figurer *à la fois* dans la sous-requête et dans la requête principale.

- 10)

```
SELECT j.nom, SUM(g.prime) AS pactole
FROM joueuse j JOIN gain g ON j.num=g.num
WHERE g.annee=2017
GROUP BY j.num
ORDER BY pactole DESC
```

- 11) Attention la requête suivante ne marche pas :

```
SELECT nom, nationalite
FROM joueuse j
JOIN gain g ON j.num=g.num AND j.annee=g.annee
WHERE g.annee=2017 AND g.tournoi='Roland Garros' AND g.tournoi='Wimbledon'
```

Pourquoi? Parce que dans chaque ligne l'attribut `g.tournoi` ne peut pas valoir simultanément 'Roland Garros' et 'Wimbledon' donc le booléen `g.tournoi='Roland Garros' AND g.tournoi='Wimbledon'` vaudra toujours faux.

C'est donc un bon exemple où on a besoin d'un `INTERSECT` au lieu du `AND`

- (M1) Voici une solution avec `INTERSECT` :

```
SELECT nom, nationalite FROM joueuse WHERE num IN (
(SELECT num FROM joueuse j JOIN gain g ON j.num=g.num
WHERE g.annee=2017 AND g.tournoi="Roland Garros")
INTERSECT
(SELECT num FROM joueuse j JOIN gain g ON j.num=g.num
WHERE g.annee=2017 AND g.tournoi="Winbledon"))
```

En fait il ne faut pas mettre de parenthèses pour le `INTERSECT` mais je les ai mises pour la lisibilité.

- (M2) En utilisant *deux fois* la table `gain` :

```
SELECT nom, prénom FROM joueuse j
JOIN gain g1 ON g1.num=j.num
JOIN gain g2 ON g2.num=j.num
WHERE nationalite=« française » AND tournoi=« Roland Garros »
```

- 12) Là il fallait faire la jointure entre `joueuse` et `match`, et l'attribut `num` car on veut les joueuses ayant perdu un match ;

```
SELECT j.nom, j.prenom
FROM joueuse j
JOIN match m ON j.num=m.num
WHERE m.tournoi='Roland Garros' AND g.annee=2017
```

- 13) Ici, la sous-requête `SELECT AVG(prime) FROM gain WHERE annee=2017` s'impose, reste le problème qu'il faut comparer le MIN des primes au résultat de cette sous-requête.

Reste le problème du MIN(prime) qu'on ne peut pas mettre dans le WHERE comme on aurait envie :
(M1) on peut néanmoins réaliser cette envie avec un HAVING.

```
SELECT tournoi
FROM gain
WHERE annee=2017
GROUP BY tournoi
HAVING MIN(prime) > (SELECT AVG(prime) FROM gain WHERE annee=2017)
```

Noter la syntaxe du HAVING avec le GROUP BY placé avant le HAVING autrement dit

```
SELECT column-names
FROM table-name
WHERE condition
GROUP BY column-names
HAVING condition
```

(M2) Sans HAVING avec un EXCEPT : il suffit d'enlever les tournois où au moins une prime est inférieure (disons strictement) à la moyenne :

```
SELECT tournoi
FROM gain
WHERE annee=2017
EXCEPT SELECT tournoi FROM gain
WHERE annee=2017
AND prime < SELECT AVG(prime) FROM gain WHERE annee=2017
```

- 14) Attention, on ne peut pas emboîter MAX(SUM(...)) et donc on n'échappe pas à une sous-requête :

(M1) Sans HAVING

```
SELECT sponsor, MAX(dotation)
FROM (SELECT sponsor, SUM(prime) AS dotation
      FROM gain
      WHERE tournoi='Roland Garros' AND annee=2017
      GROUP BY sponsor)
```

(M2) Avec une double sous-requête et même un HAVING (un peu plus compliqué ici) !

```
SELECT sponsor, SUM(prime) as S FROM gain
WHERE annee=2017 and tournoi="Roland Garros"
HAVING S =(SELECT MAX(Somme) FROM (SELECT SUM(prime) AS Somme FROM gain
WHERE annee=2017 AND tournoi="Roland Garros" GROUP BY sponsor))
```

- 15)

```
SELECT nom FROM joueuse
EXCEPT
SELECT j.nom
FROM joueuse j JOIN gain g ON j.num=g.num
WHERE g.tournoi='Roland Garros'
```

ou

```
SELECT nom, prenom FROM joueuse
WHERE num NOT IN (SELECT num FROM gain WHERE tournoi='Roland Garros')
```

- 16)

```
SELECT DISTINCT j.nom, j.prenom
FROM joueuse j
JOIN match m ON j.num=m.numg
WHERE tournoi='Roland Garros' AND annee=2017
```

La requête renvoie les nom et prénom des joueuses ayant gagné au moins un match à Roland Garros en 2017.

- 17)
- ```
SELECT DISTINCT nom
FROM joueuse j JOIN gain g ON g.num=j.num
WHERE tournoi='Roland Garros'
GROUP BY j.num
HAVING SUM(prime) > 100000
```

Comment on peut éviter le **HAVING** avec une sous-requête ? On crée une table contenant le numéro des joueuse et la somme de leur primes à roland garros, on appelle **g** cette table, et on fait une jointure de cette table avec la table joueuse et la condition de sélection sur la somme des primes :

```
SELECT nom FROM joueuse j JOIN
(SELECT num, SUM(prime) as s FROM gain
WHERE tournoi="Roland Garros"
GROUP BY num) AS g
ON j.num=g.num
WHERE s>100 000
```

- 18) Par exemple, l'identifiant de la gagnante figure dans la table gain (elle a donc participé) mais pas dans la colonne **num** du même tournoi la même année.
- 19) Dans la sous-requête, on fait la différence entre toutes les participantes – et toutes les perdantes. Il ne reste donc que la gagnante.

```
SELECT nom, prenom
FROM joueuse
WHERE num=(SELECT DISTINCT num
 FROM gain WHERE annee=2017 AND tournoi='Roland Garros'
 EXCEPT
 SELECT DISTINCT num
 FROM match WHERE annee=2017 AND tournoi='Roland Garros')
```