

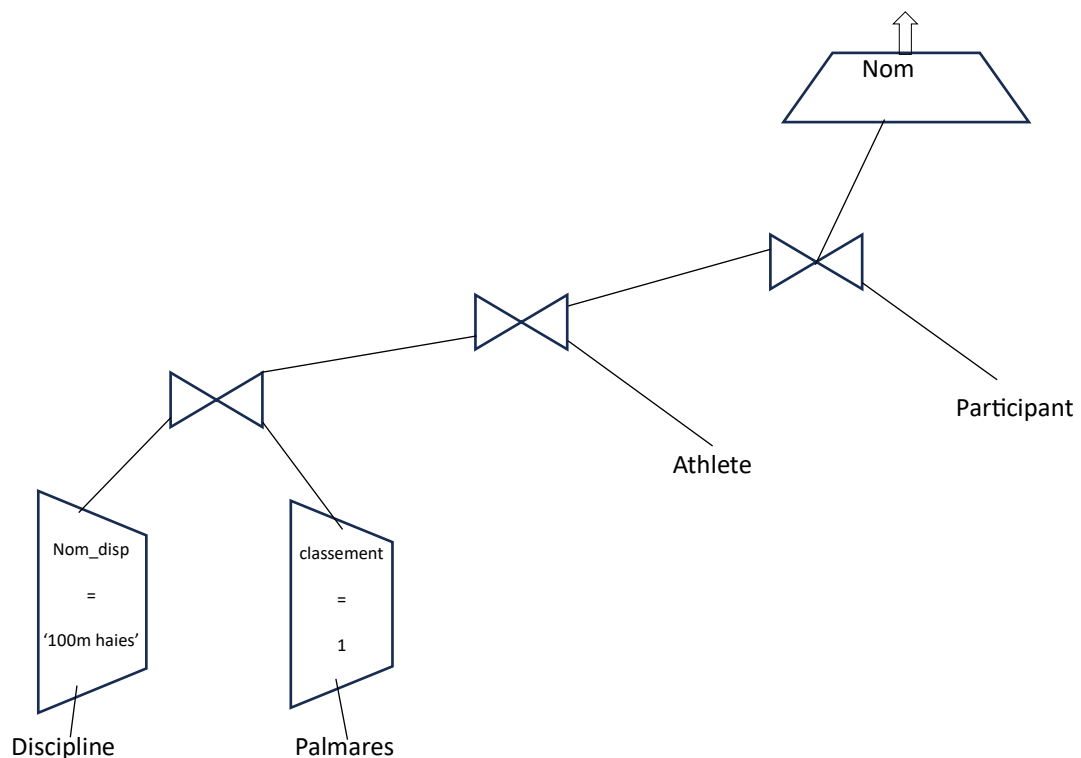
## Réponse sous forme SQL et algébrique (si possible) aux questions proposées sur le sujet

QUESTION 1 : Quelles sont les athlètes qui ont remporté la médaille d'or du 100m papillon ?

1. Forme SQL:

```
select pt.nom
from palmares p, athlete a ,DISCIPLINE d ,participant pt
where pt.id_part = a.id_part and
a.id_athlete = p.id_athlete and
d.id_discp = p.id_discp and
d.nom_discp = '100m papillon' and
p.classement = 1;
```

2. Forme algébrique :



$$\Pi_{nom} (Participant \bowtie (Athlete \bowtie (\sigma_{\substack{nom_{discp} \\ = \\ '100\ haies'}}} (Disipline)) \bowtie \sigma_{\substack{classement \\ = \\ 1}} (Palmares))$$

QUESTION 2 : Quelle nationalité a remporté le plus de médailles ?

1. Forme SQL:

```
select p.NATIONALITE
from Participant p , obtient_resultats_athlete ora ,athlete a
where a.id_part = p.id_part and ora.id_athlete = a.id_athlete and
classement_athlete IN(1,2,3)
group by NATIONALITE
having count(*) = (select max(count(*) )
                    from Participant p , obtient_resultats_athlete ora ,athlete a
                    where a.id_part = p.id_part and ora.id_athlete = a.id_athlete and
                    classement_athlete IN(1,2,3)
                    group by NATIONALITE );
```

QUESTION 3 : liste triée des pays en fonction du palmarès (nombre de médailles)

1. Forme SQL:

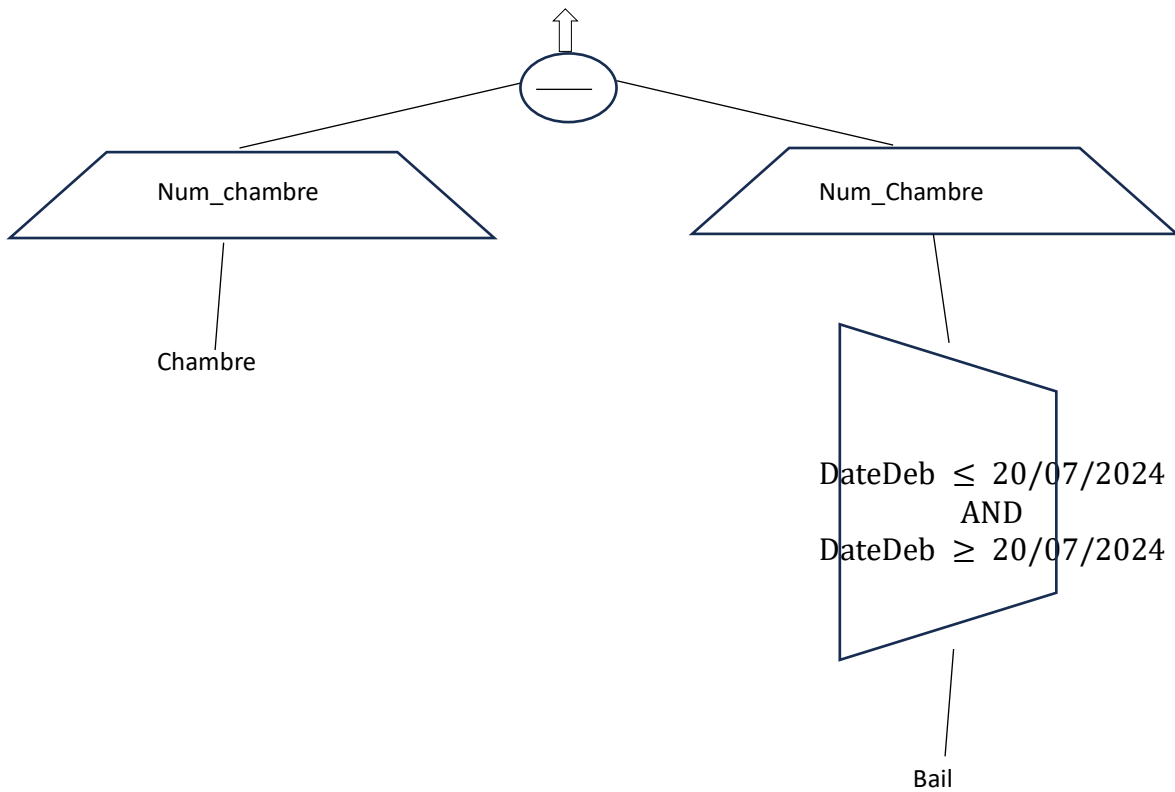
```
select NATIONALITE , count(*) as NBR_MEDAILLES
from obtient_resultats_athlete o , athlete a , participant pt
where pt.id_part = a.id_part and
o.id_athlete = a.id_athlete
group by NATIONALITE
ORDER BY NBR_MEDAILLES ;
```

QUESTION 4 : quelles sont les chambres qui sont dispo le 20/07/2024 :

1. Forme SQL:

```
select Num_chambre
from Chambre
minus
select Num_chambre
from Bail
where DateDeb <= TO_DATE('20/07/2024','DD/MM/YYYY') and
DateFin >= TO_DATE('20/07/2024','DD/MM/YYYY');
```

2. Forme algébrique :



$$\Pi_{num\_chambre} (chambre) - \Pi_{num\_chambre} (\sigma_{\substack{DateDeb \leq 20/07/2024 \\ \text{AND} \\ DateDeb \geq 20/07/2024}} (bail))$$

QUESTION 5 : quels sont les arbitres qui ont arbitré tous les quarts de la natation synchronisée :

1. Forme SQL:

```
select distinct nom
from participant p , personnel pr
where pr.id_part = p.id_part and pr.type_pers = 'arbitre'
and NOT EXISTS (
    select num_epreuve
    from Epreuve e , discipline d ,competition c
    where
        d.nom_discp = 'Natation synchronisée' and
        d.id_cat = 1 and
        d.id_discp = c.id_discp and
        c.id_compet = e.id_compet and
        e.type_epreuve = 'Quart de finale'
    and NOT EXISTS (
        select *
        from participe pt
        where pt.id_part = p.id_part and
        pt.TYPE_EPREUVE = e.TYPE_EPREUVE and
        pt.num_epreuve = e.num_epreuve and
        pt.id_compet = e.id_compet));
```

2. forme logique :

$a \in \ll \text{arbitre} \gg \text{Personnel}$  ,  $\forall e \in \text{Epreuve} \ll \text{Quart de finale} \gg \times [ \text{competition} \times \text{discipline}(100\text{m papillon}) ]$  ,  $(a, e) \in \text{participe}$  .

### 3. SQL avec count

```

select nom
from participant p , personnel pr , discipline d , competition c , participe pt
where
-- c un arbitre.
p.id_part = pr.id_part and pr.type_pers = 'arbitre' and
-- c la discipline Natation synchronisée masculine.
c.id_discp = d.id_discp and d.nom_discp = 'Natation synchronisée' and d.id_cat = 1 and
-- ce sont des quarts de finale
c.id_compet = pt.id_compet and pt.TYPE_EPREUVE ='Quart de finale' and
-- pour lesquels il a participé.
p.id_part = pt.id_part
group by nom
having count(*) = (select count(*)
                    from epreuve e , competition c , discipline d
                    where e.id_compet = c.id_compet and c.id_discp = d.id_discp
                    and d.nom_discp = 'Natation synchronisée' and d.id_cat = 1 and e.type_epreuve =
                    'Quart de finale' );

```

### 4. Forme Algébrique :

$$R = \frac{S}{T} = \frac{\prod_{num\_preuve, id\_part} (\sigma_{type\_epreuve = 'quart de finale'} (Participe) \bowtie \prod_{id\_part} (\sigma_{type\_pers = 'arbitre'} (Personnel)))}{\prod_{num\_epreuve} (\sigma_{type\_epreuve = 'quart de finale'} (Epreuve) \bowtie (Competition \bowtie \sigma_{nom\_disc = '100m haies'} (Discipline)))}$$

5. SQL avec minus :

```
select nom
```

```
from participant p , personnel pr
```

```
where pr.type_pers = 'arbitre' and pr.id_part = p.id_part
```

```
and not exists (
```

```
    select num_epreuve
```

```
    from epreuve e , competition c , discipline d
```

```
    where e.id_compet = c.id_compet and c.id_discp = d.id_discp
```

```
    and d.nom_discp = 'Natation synchronisée' and d.id_cat = 1 and e.type_epreuve = 'Quart de finale'
```

```
    minus
```

```
    select num_epreuve
```

```
    from participe pt , discipline d1 , competition c1
```

```
    where pt.id_part = p.id_part and pt.type_epreuve = 'Quart de finale'
```

```
    and d1.id_discp = c1.id_discp and d1.nom_discp = 'Natation synchronisée'
```

```
    and d1.id_cat = 1 and c1.id_compet = pt.id_compet );
```

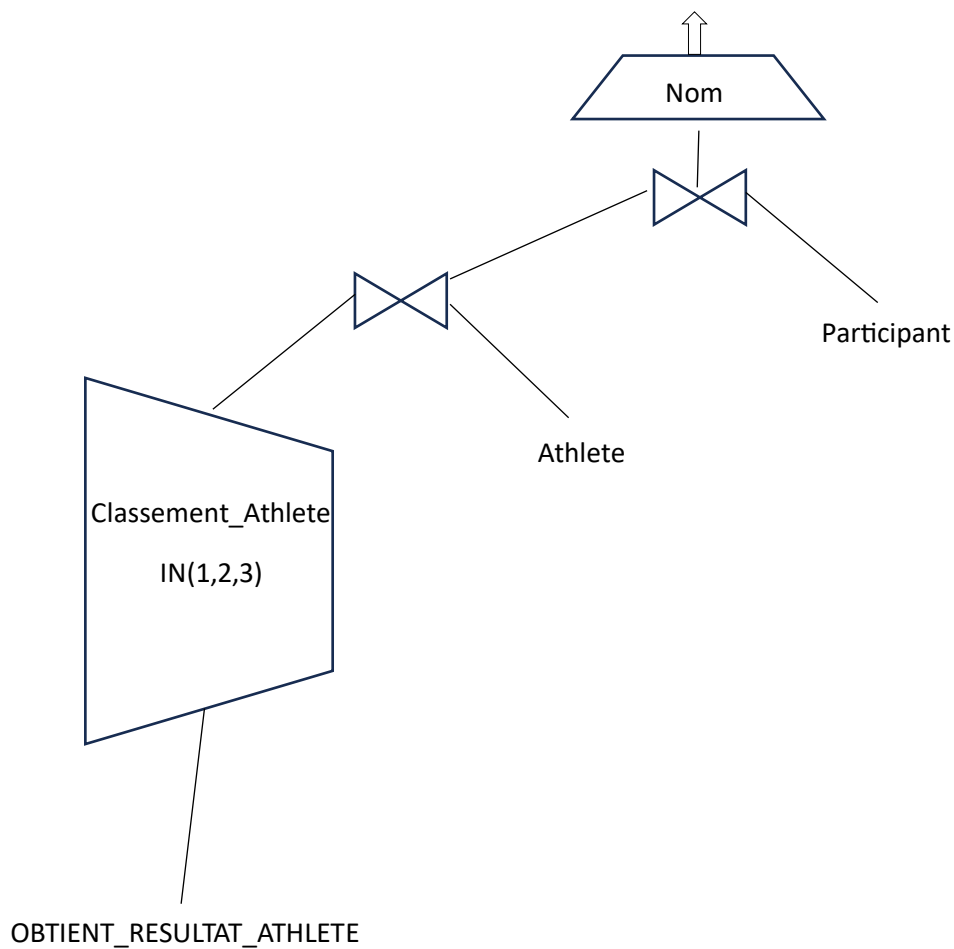
QUESTION 6 : Liste des athlètes qui ont amélioré leur palmarès par rapport au jeux précédents

Précision : tout athlète qui obtient une nouvelle médaille (OR | | ARGENT | | BRONZE) est considéré comme un athlète qui a amélioré son palmarès.

1. Forme SQL:

```
select DISTINCT nom
from OBTIENT_RESULTATS_ATHLETE o, PARTICIPANT p , ATHLETE a
where o.id_athlete = a.id_athlete and a.id_part = p.id_part
and o.classement_athlete IN(1,2,3) ;
```

2. Forme algébrique :



QUESTION 7 : quelles sont les disciplines ou aucun record n'a été battu pendant ces jeux ?

Précision : cette réponse n'est pas entièrement correcte, en effet dans certaines disciplines celui qui fait moins de temps est le gagnant.

1. Forme SQL:

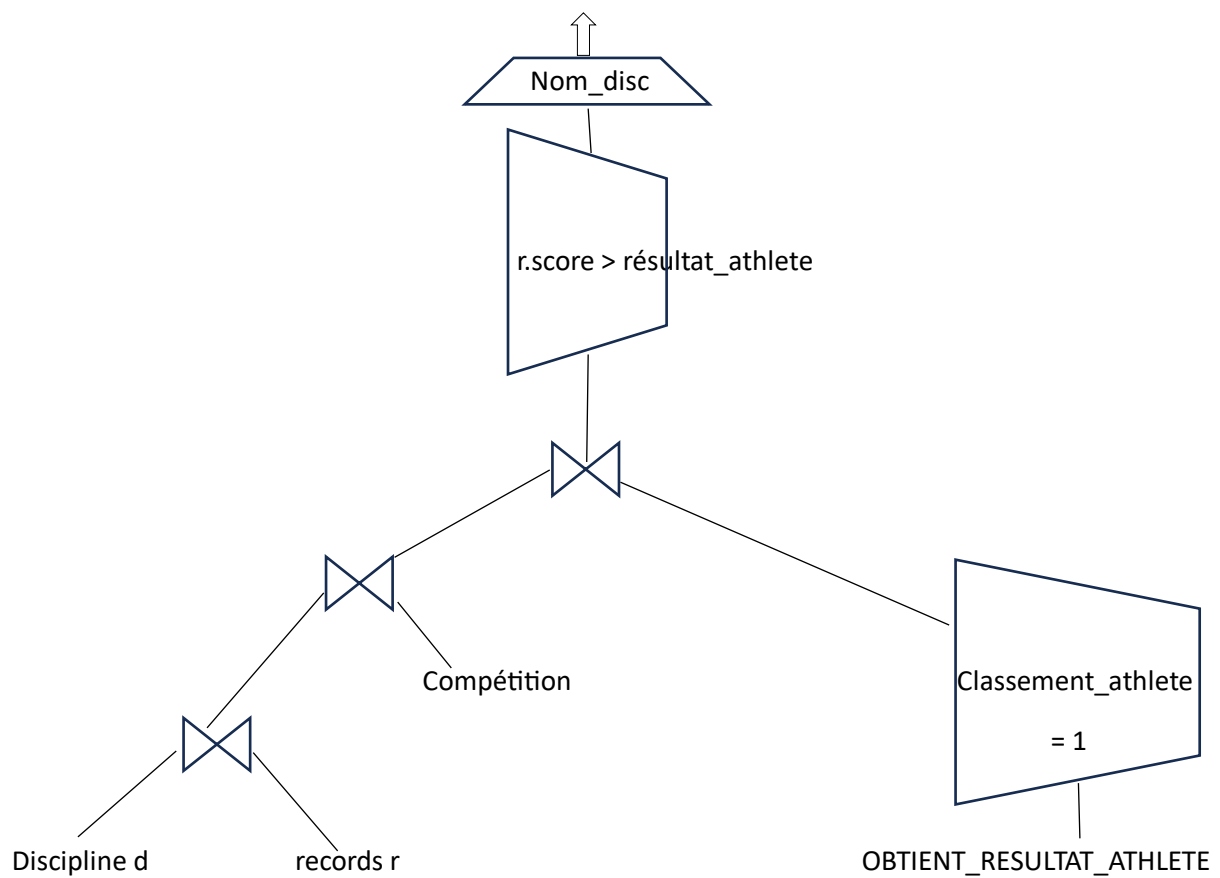
```
select nom_discp
from discipline d1 , records r
where d.id_discp = r.id_discp
and NOT EXISTS (
    select *
    from OBTIENT_RESULTATS_ATHLETE o , competition c
    where c.id_discp = d1.id_discp and
    o.id_compet = c.id_compet and
    o.classement_athlete = 1 and
    o.resultat_athlete > r.score )

UNION

select nom_discp
from discipline d1 , records r
where d.id_discp = r.id_discp
and NOT EXISTS (
    select *
    from OBTIENT_RESULTATS_EQUIPE o , competition c
    where c.id_discp = d1.id_discp and
    o.id_compet = c.id_compet and
    o.classement_athlete = 1 and
    o.resultat_athlete > r.score );
```

2. Forme Algébrique :





QUESTION 8 : Liste des participant sans médailles mais ayant participé à au moins 3 finales.

1. Forme SQL:

```
select nom ,p.id_part
from participant p , OBTIENT_RESULTATS_ATHLETE o , athlete a
where a.id_athlete = o.id_athlete and
p.id_part = a.id_part and
o.classement_athlete NOT IN (1,2,3) and
p.id_part IN ( select id_part
               from participe p
               where p.type_epreuve = 'Finale'
               group by id_part
               having count(*) >= 3 );
```

QUESTION 9 : Liste des disciplines ou une nation a remporté toutes les médailles (or, argent, bronze).

1. Forme SQL:

```
select nom_discp
from discipline dX
where EXISTS (
    select NATIONALITE
    from participant p , OBTIENT_RESULTATS_ATHLETE o , athlete a , competition c
    where
        a.id_athlete = o.id_athlete and
        a.id_part = p.id_part and
        c.id_discp = dX.id_discp and
        c.id_compet = o.id_compet and
        o.classement_athlete IN (1,2,3)
    group by NATIONALITE
    having count(*) = 3 ) ;
```

QUESTION 10 : Quels participants logent dans la même chambre et participent à la même épreuve ?

1. Frome SQL :

```
SELECT DISTINCT SB1.ID_PART AS Participant1, SB2.ID_PART AS Participant2,
B.NUM_CHAMBRE AS num_Chambre, PA1.ID_COMPET AS Competition, PA1.TYPE_EPREUVE
AS Type_epreuve , PA1.NUM_EPREUVE AS Num_Epreuve
FROM PARTICIPE PA1
JOIN SIGNE_BAIL SB1 ON PA1.ID_PART = SB1.ID_PART
JOIN PARTICIPE PA2 ON PA1.ID_COMPET = PA2.ID_COMPET AND PA1.NUM_EPREUVE =
PA2.NUM_EPREUVE AND PA1.TYPE_EPREUVE = PA2.TYPE_EPREUVE
JOIN SIGNE_BAIL SB2 ON PA2.ID_PART = SB2.ID_PART
JOIN BAIL B ON B.ID_BAIL = SB1.ID_BAIL
WHERE SB1.ID_BAIL = SB2.ID_BAIL and SB1.ID_PART < SB2.ID_PART;
```

QUESTION 11 : Quels sont les athlètes qui ont remporté 2 médailles dans 2 disciplines différentes ?

1. Frome SQL :

```
select id_athletE
from obtient_resultats_athlete o
where classement_athlete IN(1,2,3)
group by id_athlete
having COUNT (id_compet ) = 2;
```

QUESTION 12 : Liste des athlètes ayant obtenu une médaille dans chaque compétition dans laquelle ils étaient inscrits.

Précision : En sachant que la table obtient\_resultats\_athlete est considérée comme une table combinant à la fois les inscriptions et les résultats, on a jugé qu'il était inutile de créer une table 'inscrit' et une table 'résultat'. Si un athlète est disqualifié, on insère un zéro dans la table résultat.

1. Forme logique :

$a \in \text{OBTIENT\_RESULTATS\_ATHLETE}(0)$  ,  $\forall c \in \text{OBTIENT\_RESULTATS\_ATHLETE}(1) \ll \text{avec } O1.\text{id\_athlete} = a.\text{id\_athlete} \gg$  ,  $(a, c) \in \text{OBTIENT\_RESULTATS\_ATHLETE}(2)$ .

un athlète, pour toute compétition pour laquelle il a participé, avec classement\_athlete IN (1,2,3)

2. Frome SQL :

```
select DISTINCT o.id_athlete
from OBTIENT_RESULTATS_ATHLETE o
where not exists (
    select id_compet
    from obtient_resultats_athlete o1
    where o1.id_athlete = o.id_athlete
    and not exists (
        select *
        from obtient_resultats_athlete o2
        where o.id_athlete = o2.id_athlete and
        o1.id_compet = o2.id_compet and
        o2.classement_athlete IN (1,2,3) ));
```

QUESTION 13 : Identifiez les participants ayant atteint la finale dans toutes les disciplines où ils étaient inscrits, mais n'ayant pas remporté de médaille.

Un athlète dans ora o , pour toutes compétition dans ora o1 du dernier athlète , (a, c) dans participe 'finale'

1. Forme SQL :

```
select id_athlete
from obtient_resultats_athlete o
where not exists (
    select id_compet
    from obtient_resultats_athlete o1
    where o1.id_athlete = o.id_athlete
    and not exists (
        select *
        from participe p ,athlete a
        where
            p.id_compet = o1.id_compet and
            a.id_athlete = o1.id_athlete and
            p.id_part = a.id_part and
            p.type_epreuve = 'Finale'
    )
)
INTERSECT
select id_athlete
from obtient_resultats_athlete
minus
select id_athlete
from obtient_resultats_athlete
where classement_athlete IN (1,2,3);
```

QUESTION 14 : Quelle est la moyenne d'âge des athlètes participant aux compétitions de natation synchronisée ?

1. Forme SQL :

```
SELECT AVG(EXTRACT(YEAR FROM DATE '2024-01-01') - EXTRACT(YEAR FROM p.DATENAISS))  
AS moyenne_dage  
  
FROM obtient_resultats_athlete o, athlete a, participant p, discipline d, competition c  
  
WHERE o.id_athlete = a.id_athlete  
  
AND p.id_part = a.id_part  
  
AND c.id_discp = d.id_discp  
  
AND d.nom_discp = 'Natation synchronisée'  
  
AND c.id_compet = o.id_compet ;
```