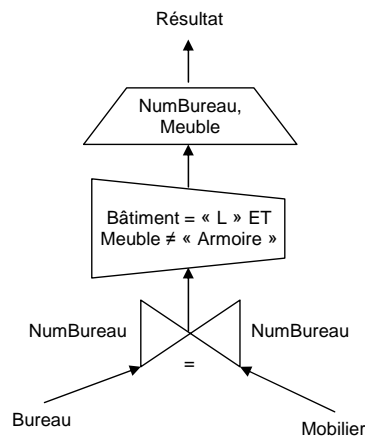


Documents autorisés – Durée : 2 h – Barème fourni à titre indicatif – Questions indépendantes

Exercice 1 : Algèbre relationnelle (2 points)

Reformulez la requête suivante de manière à ce que son exécution soit la plus rapide possible.



Exercice 2 : Requêtes SQL (10 points)

Soit la base de données « coupe du monde de rugby » dont le schéma relationnel est donné ci-dessous.

EQUIPE (CodeEquipe, Pays, CouleurMaillot, Entraîneur)

POSTE_SUR_LE_TERRAIN (Numéro, Libellé)

JOUEUR (NumJoueur, Nom, Numéro#, CodeEquipe#)

STADE (NumStade, Nom, Capacité)

ARBITRE (NumArbitre, Nom)

MATCH (NumMatch, DateMatch, NumStade#, NbSpectateurs, Equipe_Locaux#, Score_Locaux, NbEssais_Locaux, Equipe_Visiteurs#, Score_Visiteurs, NbEssais_Visiteurs)

ARBITRER (NumMatch#, NumArbitre#)

JOUER (NumMatch#, NumJoueur#, Titulaire, Temps_de_Jeu, NbPointsMarqués, NbEssaisMarqués)

Formuler en SQL les requêtes suivantes.

- Liste composée des noms d'équipe (pays), des joueurs de chaque équipe (noms des joueurs) et des postes (numéros et libellés) de ces joueurs.
- Liste des matchs en précisant pour chacun la date, les noms des équipes participantes (pays) et du stade où se déroule le match.

- Détection des erreurs : numéro des matches dont le nombre de spectateurs est supérieur à la capacité du stade dans lequel ils se déroulent.
- Vainqueur de chaque match (au format PAYS1, PAYS2, VAINQUEUR). Ignorer les matches nuls éventuels.
- Pays vainqueur du tournoi (un match nul est impossible en finale).
- Noms des joueurs qui n'ont joué aucun match.
- Liste des joueurs (noms) triée par total de points marqués décroissant.
- Nombre moyen d'essais et de points marqués par match.
- Libellés des postes pour lesquels les joueurs sont restés le plus de temps sur le terrain.
- Totaux du temps de jeu et du nombre d'essais marqués comme titulaire, d'une part, et comme remplaçant, d'autre part, pour un joueur dont le numéro est saisi au clavier.

Exercice 3 : Administration-Sécurité (2 points)

Vous devez donner accès en lecture à un grand nombre de vos tables (T1, T2, ... TN) à plusieurs utilisateurs (U1, U2...). Le nombre de ces utilisateurs est susceptible d'augmenter dans le futur. Concevoir une stratégie efficace de transmission de droits d'accès et l'exprimer en SQL.

Exercice 4 : Requêtes XQuery (6 points)

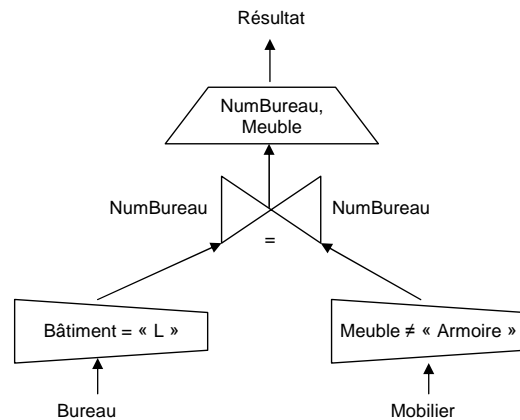
Soit le document XML `dimension.xml`, qui représente une hiérarchie de dimension dans un entrepôt de données et dont l'arborescence d'éléments et d'attributs (préfixés par @) est représentée ci-dessous. Le caractère * indique que l'élément concerné peut apparaître plusieurs fois dans le document.

```
Dimension
...@dim_id
...Level *
... ..@lvl_id
... .. Instance *
... .. ..@inst_id
... .. ..@rollup
... .. ..@drilldown
... .. .. Member *
... .. .. ..@mb_id
... .. .. ..@value
```

Formuler à l'aide du langage XQuery les requêtes suivantes (utiliser, de la syntaxe XPath ou FLWOR, la plus appropriée).

- Nombre de niveaux hiérarchiques (Level).
- Valeurs (value) distinctes du membre (Member) d'identifiant « p_name ».
- Instances qui n'ont pas d'attribut drilldown.
- Instance dont l'identifiant est égal à l'attribut rollup de l'instance d'identifiant 18.
- Valeur (value) moyenne des prix (identifiant de membre « p_retailprice ») pour le niveau d'identifiant « Part ».
- Nombre d'instances par niveau hiérarchique.

Correction Exercise 1



Correction Exercise 2

```
--1
SELECT Pays, Nom, J.Numéro, Libellé
FROM EQUIPE E, JOUEUR J, POSTE_SUR_LE_TERRAIN P
WHERE E.CodeEquipe = J.CodeEquipe AND J.Numéro = P.Numéro;

--2
SELECT DateMatch, E1.Pays, E2.Pays, Nom
FROM MATCH M, EQUIPE E1, EQUIPE E2, STADE S
WHERE Equipe_Locaux = E1.CodeEquipe AND Equipe_Visiteurs = E2.CodeEquipe AND
M.NumStade = S.NumStade;

--3
SELECT NumMatch
FROM MATCH M, STADE S
WHERE M.NumStade = S.NumStade AND NbSpectateurs > Capacité;

--4
SELECT E1.Pays PAYS1, E2.Pays PAYS2, E1.Pays VAINQUEUR
FROM MATCH M, EQUIPE E1, EQUIPE E2
WHERE Equipe_Locaux = E1.CodeEquipe AND Equipe_Visiteurs = E2.CodeEquipe
AND Score_Locaux > Score_Visiteurs
UNION
SELECT E1.Pays PAYS1, E2.Pays PAYS2, E2.Pays VAINQUEUR
FROM MATCH M, EQUIPE E1, EQUIPE E2
WHERE Equipe_Locaux = E1.CodeEquipe AND Equipe_Visiteurs = E2.CodeEquipe
AND Score_Locaux < Score_Visiteurs;

--5
SELECT Pays
FROM MATCH, EQUIPE E
WHERE Equipe_Locaux = E.CodeEquipe AND Score_Locaux > Score_Visiteurs
AND DateMatch = (SELECT MAX(DateMatch) FROM MATCH)
UNION
SELECT Pays
FROM MATCH, EQUIPE E
WHERE Equipe_Visiteurs = E.CodeEquipe AND Score_Locaux < Score_Visiteurs
```

```
AND DateMatch = (SELECT MAX(DateMatch) FROM MATCH);
```

```
--6
SELECT Nom
FROM JOUEUR
WHERE NumJoueur NOT IN
    (SELECT NumJoueur FROM JOUEUR);

--7
SELECT Nom, SUM(NbPointsMarqués) TotalPoints
FROM JOUEUR J1, JOUEUR J2
WHERE J1.NumJoueur = J2.NumJoueur
GROUP BY Nom
ORDER BY TotalPoints DESC;

--8
SELECT NumMatch, AVG(Score_Locaux+Score_Visiteurs),
    AVG(NbEssais_Locaux+NbEssais_Visiteurs)
FROM MATCH
GROUP BY NumMatch;

--9
SELECT Libellé
FROM POSTE_SUR_LE_TERRAIN P, JOUEUR J1, JOUEUR J2
WHERE P.Numéro = J1.Numéro AND J1.NumJoueur = J2.NumJoueur
GROUP BY Libellé
HAVING SUM(Temps_de_Jeu) = (
    SELECT MAX(SUM(Temps_de_Jeu))
    FROM POSTE_SUR_LE_TERRAIN P, JOUEUR J1, JOUEUR J2
    WHERE P.Numéro = J1.Numéro AND J1.NumJoueur = J2.NumJoueur
    GROUP BY Libellé);

--10
SELECT SUM(Temps_de_Jeu), SUM(NbEssaisMarqués)
FROM JOUEUR
WHERE Titulaire = 'O' AND NumJoueur = &n
GROUP BY NumJoueur
UNION
SELECT SUM(Temps_de_Jeu), SUM(NbEssaisMarqués)
FROM JOUEUR
WHERE Titulaire = 'N' AND NumJoueur = &n
GROUP BY NumJoueur;
```

Correction Exercise 3

```
CREATE ROLE R1;
GRANT SELECT ON T1 TO R1;
GRANT SELECT ON T2 TO R1;
-- ...
GRANT SELECT ON TN TO R1;
-- La création du rôle R1 évite de devoir passer N requêtes GRANT pour chaque
-- utilisateur.
GRANT R1 TO U1;
GRANT R1 TO U2;
-- ...
```

Correction Exercice 4

```
<!-- 1 -->
count(//Level)

<!-- 2 -->
distinct-values(//Member[@mb_id="p_name"]/@value)

<!-- 3 -->
//Instance[not(@drilldown)]

<!-- 4 -->
//Instance[@inst_id=//Instance[@inst_id=18]/@rollup]

<!-- 5 -->
avg(//Level[@lvl_id="Part"]/Instance/Member[@mb_id="p_retailprice"]/@value)

<!-- 6 -->
for $l in distinct-values(//Level/@lvl_id)
let $c:=count(//Level[@lvl_id=$l]/Instance)
return <Level lvl_id="{data($l)}" nb_instances="{ $c }"/>
```

Barème détaillé

Exercice 1 (2 points)

- 1 point par restriction bien (dé)placée

Exercice 2 (10 points)

- 1 point par requête (-0,5 point par problème)

Exercice 3 (2 points)

- 1 point pour l'idée de créer un rôle
- 1 point pour la syntaxe (-0,5 point par problème)

Exercice 4 (6 points)

- 1 point par requête (-0,5 point par problème)