

Ing1 - Examen de Base de Données – EISTI 2010-2011

CORRIGE

1^{ère} Partie – 8 points

Réponse 1 : 1 point

- 1) nom -> métier
- 2) nom -> sport
- 3) nom -> nationalité
- 4) nom -> compétition
- 5) nom -> année
- 6) sport -> métier
- 7) sport -> compétition
- 8) sport -> année
- 9) compétition -> année
- 10) compétition -> métier

Réponse 2 : 1 point

Toutes les dépendances dans la question précédente sont élémentaires

Réponse 3 : 1 point

Le modèle n'est pas en 3FN car les dépendances nom -> métier, nom -> compétition, nom -> année, sport -> année ne sont pas directe (i.e : nom -> sport et sport -> métier). On n'a pas d'autres dépendances de type attribut_non_clé -> clé, donc le modèle suivant sera en BCNF :

Sportif(nom, sport, nationalité)

Sport(sport, compétition)

Compétition(compétition, année, métier)

Réponse 4 : 1 point

0,1 : un produit ou n'est pas dangereux, ou corresponde au maximum à un danger

1,n : un danger corresponde à un ou plusieurs produits

Réponse 5 : 1 point

La taille de la table résultat de la 1^{ère} jointure est 100.000 d'enregistrements, celle de la 2^{ème} est seulement 10.000.

Réponse 6 : 1 point

La séquence ne se finit pas, car après le 5^{ème} ordre (S2 : UPDATE T SET b = 0 WHERE b > 2;), la ligne où a = 3 est verrouillée, la transaction S1 sera mise en attente à cause de cet ordre (S1 : DELETE FROM T WHERE a = 3;) => on ne peut pas en suite faire un COMMIT dans S1.

Réponse 7 : 1 point

La liste des index à créer :

- index sur la colonne ville
- index sur la colonne age

Réponse 8 : 1 point

Il n'y aura aucune modification, car la vue ne contient pas l'enregistrement d'id 10.

2^{ème} Partie – Exercice – 12 points**Réponse 1 : 2 points**

```
a) SELECT titre
FROM spectacle
WHERE duree > (SELECT AVG(duree)
FROM spectacle);
```

```
b) SELECT c.nom, COUNT(a.id)
FROM compagnie c, artiste a
WHERE c.id = a.idCompagnie
GROUP BY c.id, c.nom
HAVING COUNT(a.id) > 10;
```

Ou

```
SELECT c.nom
FROM compagnie c
WHERE 10 < ( SELECT COUNT(*)
FROM artiste
WHERE idCompagnie = c.id);
```

Réponse 2 : 4 points

```
a) R1 = RESTRICT(Derouler, to_char(ddate, 'MM/YYYY') = '01/2011')
R2 = RESTRICT (Spectacle, titre = 'Au fil de l'eau')
R3 = JOIN(R1,R2)
R4 = JOIN(R3, Salle)
```

R = PROJECT(R4, ddate, heure, nom)

Ou

$\Pi_{ddate, heure, nom}(\sigma_{titre='Au\ fil\ de\ l'eau' \wedge to_char(ddate, 'MM/YYYY')='01/2011'}(Salle \bowtie_{Salle.id=idSalle} Derouler \bowtie_{idSpectacle=id} Spectacle))$

```
SELECT ddate, heure, nom
FROM Salle s, Derouler d, Spectacle sp
WHERE s.id = d.idSalle
AND d.idSpectacle = sp.id
AND titre = 'Au fil de l'eau'
AND to_char(ddate, 'MM/YYYY') = '01/2011';
```

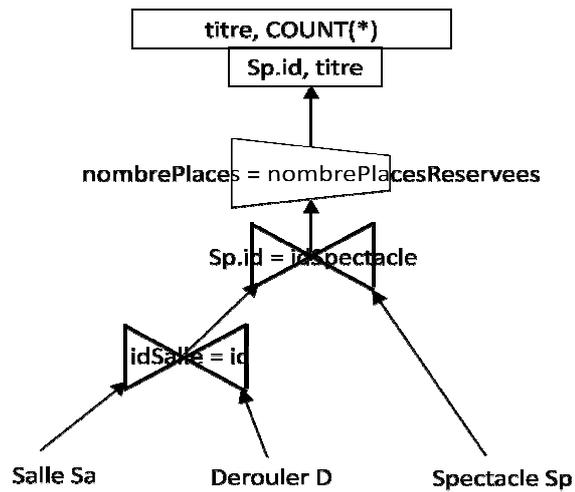
b) Division :

$\rho_{nom, prenom, id}(\Pi_{nom, prenom, idSpectacle} (Artiste \bowtie_{id=idArtiste} Participer)) \div (\Pi_{id} (\sigma_{idCompagnie = Artiste.idCompagnie} Spectacle))$

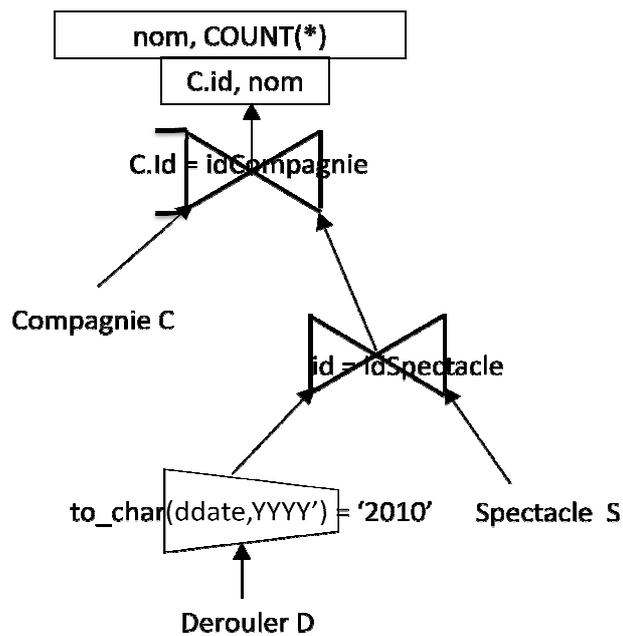
```
SELECT nom, prenom
FROM artiste a, participer p
WHERE a.id = p.idArtiste
GROUP BY a.id, nom, prenom, a.idCompagnie
HAVING COUNT(idSpectacle) = (SELECT COUNT(id)
FROM Spectacle s
WHERE s.idCompagnie = a.idCompagnie) ;
```

Réponse 3 : 4 points

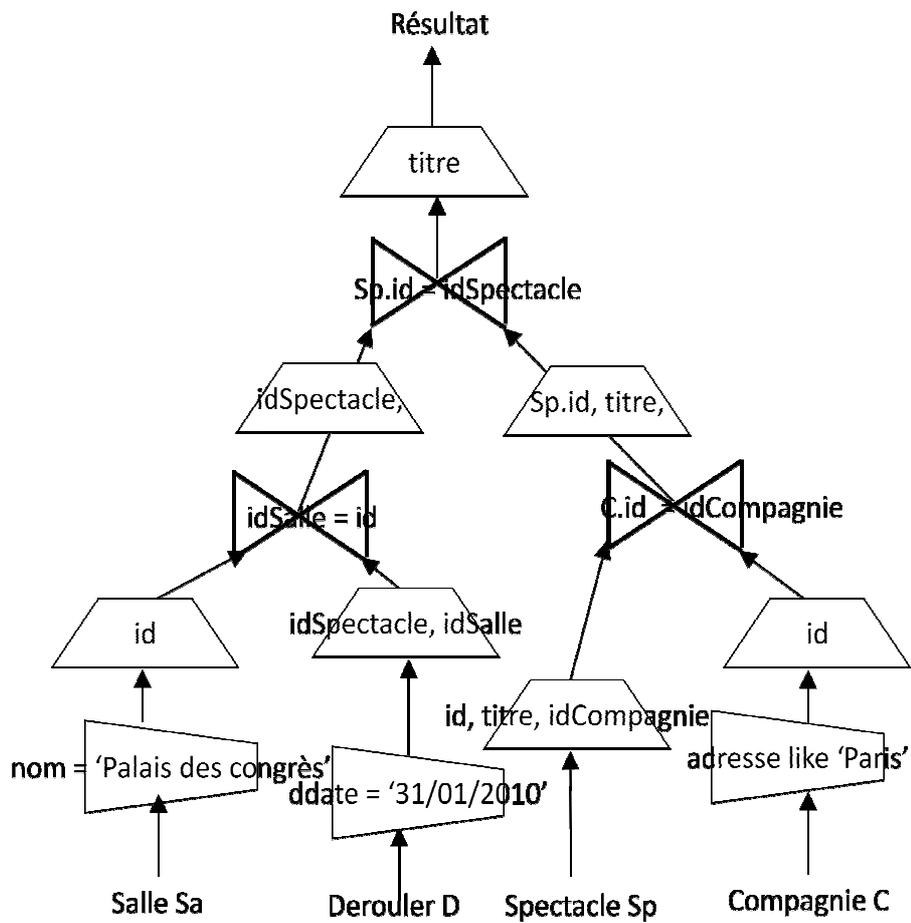
```
a) SELECT titre, COUNT(*)
FROM Spectacle sp, Salle s, Derouler d
WHERE sp.id = d.idSpectacle
AND s.id = d.idSalle
AND nombrePlaces = nombrePlacesReservees
GROUP BY sp.id, titre;
```



b) `SELECT nom, COUNT(idSpectacle)`
`FROM compagnie c LEFT JOIN (Spectacle s JOIN (SELECT * FROM derouler WHERE`
`to_char(ddate, 'YYYY') = '2010') ON s.id = d.idSpectacle) ON c.id = idCompagnie`
`GROUP BY c.id, nom;`



Réponse 4 : 2 points



Nota : sans projections intermédiaires => 1,5 points