

Correction du DS3

Exercice 1 :

1. |SELECT * FROM Bouee WHERE localisation = 'Mediterranee' AND frequence = 2

2. |SELECT COUNT(*) FROM Bouee WHERE nomSite = 'Porquerolles'

3. |SELECT idTempete FROM Tempete AS t JOIN Bouee AS b ON b.idBouee=t.idBouee
| WHERE localisation = 'Mediterranee' AND finTempete < '2023'

4. Il faut les trois tables

```
|SELECT b.idBouee ,idCampagne ,idTempete FROM Bouee AS b JOIN Campagne AS c JOIN  
| Tempete AS t ON b.idBouee=c.idBouee AND b.idBouee=t.idBouee WHERE  
| localisation = 'Mediterranee'
```

5. Les sites sans tempête en 2023 n'apparaissent pas dans la table **Tempete** donc inutile de vérifier si le nombre de tempêtes est ≥ 1 .

```
|SELECT nomSite FROM Bouee AS b JOIN Tempete AS t ON b.idBouee=t.idBouee WHERE  
| debutTempete>='2023' AND debutTempete<'2024'
```

6. |SELECT idBouee ,MAX(Hmax) FROM Tempete GROUP BY idBouee

7. |SELECT idBouee FROM Tempete WHERE debutTempete>='2023' AND finTempete<'2024'
| GROUP BY idBouee HAVING COUNT(*)>4

8. On commence par calculer la hauteur de houle maximale puis on utilise une sous-requête

```
|SELECT idTempete FROM Tempete WHERE Hmax = (SELECT MAX(Hmax) FROM Tempete)
```

9. |SELECT idCampagne ,c.idBouee ,COUNT(*) FROM Campagne AS c JOIN Tempete AS t ON
| c.idBouee = t.idBouee WHERE debutCampagne>='2023' AND finCampagne<'2024'
| AND debutTempete>=debutCampagne AND finTempete<=finCampagne GROUP BY
| idCampagne

10. Ce sont les identifiants des bouées qui n'apparaissent pas dans la table **Tempete** en 2023 ; en supposant que toutes les bouées ont été utilisées dans au moins une campagne en 2023 (sinon elles ne risquent pas de trouver une tempête)

```
|SELECT (DISTINCT) idBouee FROM Campagne WHERE debutCampagne>='2023' AND  
| finCampagne<'2024' EXCEPT (SELECT IdBouee FROM Tempete WHERE debutTempete  
| >='2023' AND finTempete<'2024')
```

11. Il faut « dupliquer la table » **Tempete** (autojointure) en faisant attention de ne pas garder les couples dont les deux identifiants sont identiques

```
|SELECT a.idTempete ,b.idTempete FROM Tempete AS a JOIN Tempete AS b ON a.  
| debutTempete=b.debutTempete AND a.finTempete=b.finTempete WHERE a.  
| idTempete<b.idTempete
```

Exercice 2 :

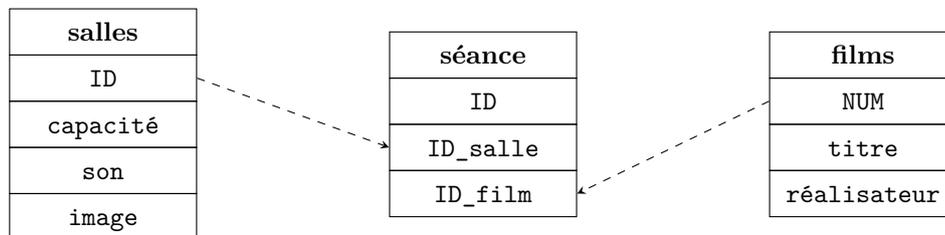
1. a) Un film peut être projeté dans différentes salles mais chaque salle ne propose qu'une seule séance donc

`salles - films est de type 1 - *`

b) Le modèle 2 oblige à recopier les données de chaque film qui est diffusé dans plusieurs salles alors que le modèle 1 ne demande que de remplir la « case » `ID_film` dans la table `salles` : il vaut donc mieux utiliser le modèle 1.

2. On a cette fois une association * - * qui demanderait de recopier des lignes presque identiques avec le modèle 1 comme avec le modèle 2.

On sépare cette association en deux associations 1 - * en introduisant une table `séance` pour obtenir le schéma relationnel suivant :



L'attribut ID de la table **séance** est une clef primaire permettant de distinguer les éventuels doublons (si un film est diffusé deux fois dans la même salle); les attributs ID_salle et ID_film sont des clefs étrangères qui font référence respectivement aux clefs primaires ID et NUM des tables **salles** et **films** respectivement.

- Si on rajoute un attribut pour identifier chaque bobine dans la table **films** (qui viendrait à remplacer NUM en tant que clef primaire), on va devoir recopier tous les autres attributs de chaque film, donc multiplier les lignes de cette table. Il vaut donc mieux introduire une autre table pour faire cette association :

