

4 MIDI'HOME CAMPING™ - GROUP BY - HAVING

R50 : pour chacun des clients qui a réalisé au moins une location, indiquer son nom, son prénom et le nombre de ses locations (les clients doivent être classés par ordre décroissant du nombre de locations).

```
SELECT nomClient, prenomClient, COUNT(*) AS "nb locations"
FROM Clients c
JOIN Locations l ON c.idClient = l.idClient
GROUP BY c.idClient, nomClient, prenomClient
ORDER BY "nb locations" DESC ;
```

R51 : pour chacun des campings pour lesquels on a au moins un employé, indiquer le salaire moyen des employés qui travaillent dans le camping.

```
SELECT nomCamping, AVG(salaireEmploye) as "salaire moyen"
FROM Campings c
JOIN Employes e ON c.idCamping = e.idCamping
GROUP BY c.idCamping, nomCamping;
```

R52 : le nom des campings qui possèdent plus de 3 employés.

```
SELECT nomCamping
FROM Campings c
JOIN Employes e ON c.idCamping = e.idCamping
GROUP BY c.idCamping, nomCamping
HAVING COUNT(*) > 3;
```

R53 : le nom des campings dont le salaire moyen des employés est supérieur à 1 400€.

```
SELECT nomCamping
FROM Campings c
JOIN Employes e ON c.idCamping = e.idCamping
GROUP BY c.idCamping, nomCamping
HAVING AVG(salaireEmploye) > 1400;
```

R54 : la liste des campings, classés par rapport au nombre de bungalows de moins de 65 m² qu'ils possèdent (de celui qui en possède le moins à celui qui en possède le plus).

```
SELECT nomCamping
FROM Campings c
JOIN Bungalows b ON b.idCamping = c.idCamping
WHERE superficieBungalow < 65
GROUP BY c.idCamping, nomCamping
ORDER BY COUNT(idBungalow);
```

R55 : le nom des campings où tous les employés ont un salaire supérieur ou égal à 1 000 €.

```
SELECT nomCamping
FROM Campings c
JOIN Employes e ON c.idCamping = e.idCamping
GROUP BY c.idCamping, nomCamping
HAVING MIN(salaireEmploye) >= 1000;
```

R56 : le nom des bungalows qui proposent le même nombre de services que le bungalow qui se nomme 'Le Royal'.

```
SELECT nomBungalow
FROM Bungalows b
JOIN Proposer p ON b.idBungalow = p.idBungalow
GROUP BY b.idBungalow, nomBungalow
HAVING COUNT(*) = (SELECT COUNT(*)
FROM Bungalows b
JOIN Proposer p ON b.idBungalow = p.idBungalow
WHERE nomBungalow = 'Le Royal');
```

R57 : le nombre de services proposés par chacun des bungalows ; on veut également afficher les bungalows qui ne proposent pas de service. Les bungalows doivent être classés par rapport au nombre de services proposés.

```
SELECT nomBungalow, COUNT(idService) AS "nb services"
FROM Bungalows b
LEFT OUTER JOIN Proposer p ON b.idBungalow = p.idBungalow
GROUP BY b.idBungalow, nomBungalow
ORDER BY "nb services" DESC;
```

ou

```
SELECT nomBungalow, COUNT(*) AS "nb services"
FROM Bungalows b
JOIN Proposer p ON b.idBungalow = p.idBungalow
GROUP BY b.idBungalow, nomBungalow
UNION
SELECT nomBungalow, 0 AS "nb services"
FROM Bungalows b
WHERE NOT EXISTS (SELECT *
FROM Proposer p
WHERE p.idBungalow = b.idBungalow)
ORDER BY "nb services" DESC;
```

R58 : le nom de bungalow – avec le nom du camping dans lequel il se trouve – de tous les bungalows qui ont été loués plus de deux fois en juin 2017.

```
SELECT nomBungalow, nomCamping
FROM Campings c
JOIN Bungalows b ON c.idCamping = b.idCamping
JOIN Locations l ON l.idBungalow = b.idBungalow
WHERE dateDebut <= '30/06/2017'
AND dateFin >= '01/06/2017'
GROUP BY b.idBungalow, nomBungalow, nomCamping
HAVING COUNT(*) > 2;
```

R59 : le bungalow qui propose le plus de services.

```
SELECT nomBungalow
FROM Bungalows b
JOIN Proposer p ON b.idBungalow = p.idBungalow
GROUP BY b.idBungalow, nomBungalow
HAVING COUNT(*) = (SELECT MAX(COUNT(*))
FROM Proposer p
GROUP BY idBungalow);
```

R60 : le nom des bungalows qui n'ont pas été loués en juin 2017.

```
SELECT nomBungalow
FROM Bungalows b
WHERE NOT EXISTS (SELECT *
FROM Locations l
WHERE dateDebut <= '31/08/2017'
AND dateFin >= '01/08/2017'
AND l.idBungalow = b.idBungalow)
```

R61 : pour chacun des campings où il y a des employés, indiquer le nom et le prénom de l'employé le mieux payé.

```
SELECT nomCamping, nomEmploye, prenomEmploye
FROM Campings c
JOIN Employes e ON c.idCamping = e.idCamping
WHERE (salaireEmploye, c.idCamping) IN (SELECT MAX(salaireEmploye), idCamping
FROM Employes
GROUP BY idCamping);
```

ou

```
SELECT nomCamping, nomEmploye, prenomEmploye
FROM Campings c1
JOIN Employes e1 ON c1.idCamping = e1.idCamping
WHERE salaireEmploye = (SELECT MAX(salaireEmploye)
FROM Employes e2
WHERE e2.idCamping = c1.idCamping);
```

R62 : pour chacun des bungalows (où il y a eu des locations), à chaque date de début d'une nouvelle location, indiquer le chiffre d'affaires cumulé du bungalow (c'est-à-dire le cumul de tous les montants des locations jusqu'à la date présente).

```
SELECT nomBungalow, lNew.dateDebut, SUM(lOld.montantLocation) AS CA
FROM Bungalows b
JOIN Locations lNew ON lNew.idBungalow = b.idBungalow
JOIN Locations lOld ON lOld.idBungalow = b.idBungalow
AND lNew.dateDebut >= lOld.dateDebut
GROUP BY lNew.idBungalow, nomBungalow, lNew.dateDebut
ORDER BY lNew.idBungalow, lNew.dateDebut;
```