❺ Midi'Home Camping™ - Divisions

R70 : le nom des bungalows qui proposent tous les services (c'est-à-dire tous les services qui sont dans la table Services).

```
SELECT nomBungalow
FROM Bungalows b
     JOIN Proposer p ON b.idBungalow = p.idBungalow
GROUP BY b.idBungalow, nomBungalow
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM Services);
Ou
SELECT nomBungalow
FROM Bungalows b
WHERE NOT EXISTS (SELECT idService
                  FROM Services
                 MINUS
                  SELECT idService
                  FROM Proposer p
                  WHERE p.idBungalow = b.idBungalow);
```

R71 : le nom des bungalows qui proposent tous les services de la catégorie 'Luxe'.

```
SELECT nomBungalow
FROM Bungalows b
     JOIN Proposer p ON b.idBungalow = p.idBungalow
     JOIN Services s ON p.idService = s.idService
WHERE categorieService = 'Luxe'
GROUP BY b.idBungalow, nomBungalow
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM Services
                   WHERE categorieService = 'Luxe');
Ou
SELECT nomBungalow
FROM Bungalows b
WHERE NOT EXISTS (SELECT idService
                  FROM Services
                  WHERE categorieService = 'Luxe'
                 MINUS
                  SELECT idService
                  FROM Proposer p
                  WHERE p.idBungalow = b.idBungalow);
```

R72 : le nom des bungalows qui proposent tous les services proposés par le bungalow qui se nomme 'La Poubelle'.

```
SELECT nomBungalow
FROM Bungalows b
     JOIN Proposer p ON b.idBungalow = p.idBungalow
WHERE idService IN (SELECT idService
                    FROM Bungalows b
                         JOIN Proposer p ON b.idBungalow = p.idBungalow
                    WHERE nomBungalow = 'La Poubelle')
GROUP BY b.idBungalow, nomBungalow
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM Bungalows b
                        JOIN Proposer p ON b.idBungalow = p.idBungalow
                   WHERE nomBungalow = 'La Poubelle');
Ou
SELECT nomBungalow
FROM Bungalows b
WHERE NOT EXISTS (SELECT idService
                  FROM Bungalows b
                       JOIN Proposer p ON b.idBungalow = p.idBungalow
                  WHERE nomBungalow = 'La Poubelle'
                 MINUS
                  SELECT idService
                  FROM Proposer p
                  WHERE p.idBungalow = b.idBungalow);
```

R73 : le nom des clients qui ont réalisé au moins une location dans toutes les villes pour lesquelles il y a des campings.

```
SELECT nomClient
FROM Clients cl
     JOIN Locations l ON cl.idClient = l.idClient
     JOIN Bungalows b ON l.idBungalow = b.idBungalow
     JOIN Campings ca ON b.idCamping = ca.idCamping
GROUP BY cl.idClient, nomClient
HAVING COUNT(DISTINCT villeCamping) = (SELECT COUNT(DISTINCT villeCamping)
                                       FROM Campings);
Ou
SELECT nomClient
FROM Clients c
WHERE NOT EXISTS (SELECT villeCamping
                  FROM Campings
                 MINUS
                  SELECT villeCamping
                  FROM Locations l
                       JOIN Bungalows b ON l.idBungalow = b.idBungalow
                       JOIN Campings camp ON b.idCamping = camp.idCamping
                  WHERE l.idClient = c.idClient);
```

R74 : le nom des clients qui ont loué tous les bungalows loués par la cliente Agathe Zeblouse.

```
SELECT nomClient
FROM Clients c
     JOIN Locations l ON c.idClient = l.idClient
WHERE idBungalow IN (SELECT idBungalow
                     FROM Locations l
                          JOIN Clients c ON l.idClient = c.idClient
                     WHERE nomClient = 'Zeblouse'
                     AND prenomClient = 'Agathe')
GROUP BY c.idClient, nomClient
HAVING COUNT(DISTINCT idBungalow) = (SELECT COUNT(DISTINCT idBungalow)
                                     FROM Locations l
                                          JOIN Clients c ON l.idClient = c.idClient
                                     WHERE nomClient = 'Zeblouse'
                                     AND prenomClient = 'Agathe');
Ou
SELECT nomClient
FROM Clients c
WHERE NOT EXISTS (SELECT idBungalow
                  FROM Locations l
                       JOIN Clients c ON l.idClient = c.idClient
                  WHERE nomClient = 'Zeblouse'
                  AND prenomClient = 'Agathe'
                 MINUS
                  SELECT idBungalow
                  FROM Locations l
                  WHERE l.idClient = c.idClient);
```

R75 : le nom et le prénom des clients qui ont sont allés exactement dans les mêmes campings que la cliente Agathe Zeblouse.

```
SELECT nomClient, prenomClient
FROM Clients c1
WHERE NOT EXISTS (SELECT idCamping
                  FROM Bungalows b
                       JOIN Locations l ON l.idBungalow = b.idBungalow
                       JOIN Clients c ON c.idClient = l.idClient
                  WHERE nomClient = 'Zeblouse' AND prenomClient = 'Agathe'
                 MINUS
                  SELECT idCamping
                  FROM Bungalows b
                       JOIN Locations l ON l.idBungalow = b.idBungalow
                  WHERE l.idClient = c1.idClient)
AND NOT EXISTS (SELECT idCamping
                FROM Bungalows b
                     JOIN Locations l ON l.idBungalow = b.idBungalow
                WHERE l.idClient = c1.idClient
               MINUS
                SELECT idCamping
                FROM Bungalows b
                     JOIN Locations l ON l.idBungalow = b.idBungalow
                     JOIN Clients c ON c.idClient = l.idClient
                WHERE nomClient = 'Zeblouse' AND prenomClient = 'Agathe');
```

R80 : le numéro, le nom et le prénom des clients qui n'ont jamais fait de location dans un camping de Palavas. Les clients doivent être classés par ordre lexicographique de leur nom.

```
 SELECT idClient, nomClient, prenomClient
 FROM Clients c
MINUS
 SELECT cl.idClient, nomClient, prenomClient
 FROM Clients cl
      JOIN Locations l ON cl.idClient = l.idClient
      JOIN Bungalows b ON l.idBungalow = b.idBungalow
      JOIN Campings ca ON b.idCamping = ca.idCamping
 WHERE villeCamping = 'Palavas'
ORDER BY nomClient;
```

R81 : le nom des services qui sont proposés dans tous les bungalows dont la superficie est supérieure à 60 m².

```
SELECT nomService
FROM Services s
JOIN Proposer p ON s.idService = p.idService
JOIN Bungalows b ON b.idBungalow = p.idBungalow
WHERE superficieBungalow > 60
GROUP BY nomService, s.idService
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM Bungalows b
                   WHERE superficieBungalow > 60);

Ou

SELECT nomService
FROM Services s
WHERE NOT EXISTS (SELECT idBungalow
                  FROM Bungalows
                  WHERE superficieBungalow > 60
                 MINUS
                  SELECT idBungalow
                  FROM Proposer p
                  WHERE p.idService = s.idService);
```

R82 : le nom des services proposés par le bungalow le plus grand du camping 'Les Flots Bleus'.

```
SELECT nomService
FROM Services s
     JOIN Proposer p ON p.idService = s.idService
WHERE idBungalow IN (SELECT idBungalow
                     FROM Bungalows b
                          JOIN Campings c ON b.idCamping = c.idCamping
                     WHERE nomCamping = 'Les Flots Bleus'
                     AND superficieBungalow = (SELECT MAX(superficieBungalow)
                                               FROM Bungalows b
                                                    JOIN Campings c
                                                         ON b.idCamping = c.idCamping
                                               WHERE nomCamping = 'Les Flots Bleus'));
```

R83 : pour chacun des employés du camping 'La décharge Monochrome', le nom et le prénom de l'employé ainsi que le nombre de subordonnés qu'il possède.

```
SELECT e.nomEmploye, e.prenomEmploye, COUNT(sub.idEmploye) AS "NB SUBORDONNES"
FROM Employes e
     JOIN Campings c ON e.idCamping = c.idCamping
     LEFT OUTER JOIN Employes sub ON sub.idEmployeChef = e.idEmploye
WHERE nomCamping = 'La Décharge Monochrome'
GROUP BY e.idEmploye, e.nomEmploye, e.prenomEmploye;
```

R84 : le nom des campings où tous les bungalow ont une superficie supérieure à 50 m².

```
SELECT nomCamping
FROM Campings c
WHERE NOT EXISTS (SELECT *
                  FROM Bungalows b
                  WHERE superficieBungalow <= 50
                  AND b.idCamping = c.idCamping);
```

R85 : le nom des clients qui ont réalisé le même nombre de locations que le client Agathe Zeblouse (il n'y a qu'un seul client qui se nomme Agathe Zeblouse).

```
SELECT nomClient
FROM Clients c
     JOIN Locations l ON c.idClient = l.idClient
GROUP BY c.idClient, nomClient
HAVING COUNT(*) = (SELECT COUNT(*)
                   FROM Locations l
                        JOIN Clients c ON l.idClient = c.idClient
                   WHERE nomClient = 'Zeblouse'
                   AND prenomClient = 'Agathe');
```

R86 : le nom des services qui sont proposés dans moins de cinq bungalows.

```
SELECT nomService
FROM Services s
     LEFT OUTER JOIN Proposer p ON p.idService = s.idService
GROUP BY s.idService, nomService
HAVING COUNT(idBungalow) < 5;
```

R87 : le nom du camping qui a le plus d'employés.

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

R88 : les bungalows qui proposent des services de toutes les catégories.

```
SELECT nomBungalow
FROM Bungalows b
     JOIN Proposer p ON b.idBungalow = p.idBungalow
     JOIN Services s ON p.idService = s.idService
GROUP BY b.idBungalow, nomBungalow
HAVING COUNT(DISTINCT categorieService) = (SELECT COUNT(DISTINCT categorieService)
                                           FROM Services);
Ou
SELECT nomBungalow
FROM Bungalows b
WHERE NOT EXISTS (SELECT categorieService
                  FROM Services
                 MINUS
                  SELECT categorieService
                  FROM Services s
                       JOIN Proposer p ON s.idService = p.idService
                  WHERE p.idBungalow = b.idBungalow);
```

R89 : le nom des bungalows qui proposent exactement les mêmes services que le bungalow 'La Suite Régalienne' – DIVISION EXACTE

```
SELECT nomBungalow
FROM Bungalows b1
WHERE NOT EXISTS (SELECT idService
                  FROM Bungalows b
                       JOIN Proposer p ON b.idBungalow = p.idBungalow
                  WHERE nomBungalow = 'La Suite Régalienne'
                 MINUS
                  SELECT idService
                  FROM Proposer p
                  WHERE p.idBungalow = b1.idBungalow)
AND NOT EXISTS (SELECT idService
                FROM Proposer p
                WHERE p.idBungalow = b1.idBungalow
               MINUS
                SELECT idService
                FROM Bungalows b
                     JOIN Proposer p ON b.idBungalow = p.idBungalow
                WHERE nomBungalow = 'La Suite Régalienne');
```

R90 : le nom du plus petit des bungalows qui n'a pas eu de location.

```
SELECT nomBungalow
FROM Bungalows b
WHERE NOT EXISTS (SELECT *
                  FROM Locations l
                  WHERE l.idBungalow = b.idBungalow)
AND superficieBungalow = (SELECT MIN(superficieBungalow)
                          FROM Bungalows b
                          WHERE NOT EXISTS (SELECT *
                                            FROM Locations l
                                            WHERE l.idBungalow = b.idBungalow));
```

R91 : le nom des bungalows dont la superficie est supérieure à la superficie moyenne des bungalows de son camping.

```
SELECT nomBungalow
FROM Bungalows b1
WHERE superficieBungalow > (SELECT AVG(superficieBungalow)
                            FROM Bungalows b2
                            WHERE b2.idCamping = b1.idCamping);
```

R92 : 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);
```

R93 : pour chaque bungalow, indiquer le nombre de locations qui ont eu lieu en juin 2021.

```
 SELECT nomBungalow, COUNT(*) AS "nb locations"
 FROM Bungalows b
      JOIN Locations l ON l.idBungalow = b.idBungalow
 WHERE dateDebut <= '30/06/2021'
 AND dateFin >= '01/06/2021'
 GROUP BY b.idBungalow, nomBungalow
UNION
 SELECT nomBungalow, 0 AS "nb locations"
 FROM Bungalows b
 WHERE NOT EXISTS (SELECT *
                   FROM Locations l
                   WHERE dateDebut <= '30/06/2021'
                   AND dateFin >= '01/06/2021'
                   AND l.idBungalow = b.idBungalow)
ORDER BY "nb locations" DESC;
```

Ou

```
SELECT nomBungalow, COUNT(l.idLocation) AS "nb locations"
FROM Bungalows b
     LEFT OUTER JOIN Locations l ON l.idBungalow = b.idBungalow
     AND dateDebut <= '30/06/2021'
     AND dateFin >= '01/06/2021'
GROUP BY b.idBungalow, nomBungalow
ORDER BY "nb locations" DESC;
```

R94 : 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;
```

R95 : pour chacun des services, le nombre de bungalows qui proposent uniquement ce service (et pas un autre service).

```
SELECT nomService, COUNT(b.idBungalow) AS "nb bungalows"
FROM Services s
     LEFT OUTER JOIN Proposer p ON s.idService = p.idService
     LEFT OUTER JOIN (SELECT idBungalow
                      FROM Proposer p
                      GROUP BY idBungalow
                      HAVING COUNT(*) = 1) b ON p.idBungalow = b.idBungalow
GROUP BY s.idService, nomService;
```

R96 : le nom et le prénom des trois salariés les mieux payés.

```
SELECT e1.nomEmploye, e1.prenomEmploye
FROM Employes e1
JOIN Employes e2 ON e1.salaireEmploye >= e2.salaireEmploye
GROUP BY e1.idEmploye, e1.nomEmploye, e1.prenomEmploye, e1.salaireEmploye
HAVING COUNT(*) > (SELECT COUNT(*) - 3
                   FROM Employes)
ORDER BY e1.salaireEmploye DESC;
```

Ou

```
SELECT e1.nomEmploye, e1.prenomEmploye, e1.salaireEmploye
FROM Employes e1
LEFT JOIN Employes e2 ON e1.salaireEmploye < e2.salaireEmploye
GROUP BY e1.idEmploye, e1.nomEmploye, e1.prenomEmploye, e1.salaireEmploye
HAVING COUNT(e1.idEmploye) + 1 <= 3
ORDER BY e1.salaireEmploye DESC;
```