A partir du script SQL Gaulois fourni par votre formateur, écrivez et exécutez les requêtes SQL suivantes :

1. Nom des lieux qui finissent par 'um' (trié par ordre alphabétique).

```
SELECT l.nom_lieu
FROM lieu l
WHERE LOWER(l.nom_lieu) LIKE "%um"
```

2. Nombre de personnages par lieu (trié par nombre de personnages décroissant).

```
SELECT l.nom_lieu AS lieu, COUNT(p.id_personnage) AS nb_habitants
FROM personnage p, lieu l
WHERE l.id_lieu = p.id_lieu
GROUP BY l.id_lieu
ORDER BY nb_habitants DESC
```

3. Nom des personnages + spécialité + adresse et lieu d'habitation, triés par lieu puis par nom de personnage.

```
SELECT p.nom_personnage, l.nom_lieu, s.nom_specialite
FROM personnage p, lieu l, specialite s
WHERE l.id_lieu = p.id_lieu
AND s.id_specialite = p.id_specialite
ORDER BY l.nom_lieu ASC, p.nom_personnage ASC
```

4. Nom des spécialités avec nombre de personnages par spécialité (trié par nombre de personnages décroissant).

```
SELECT s.nom_specialite, COUNT(p.id_personnage) AS nb_personnages
FROM specialite s
LEFT JOIN personnage p
ON s.id_specialite = p.id_specialite
GROUP BY s.id_specialite
ORDER BY nb_personnages DESC
```

Note : LEFT JOIN vivement conseillé, sinon la spécialité "Agriculteur" n'apparaitra pas dans le jeu de résultat.

5. Nom, date et lieu des batailles, de la plus récente à la plus ancienne (dates affichées au format jj/mm/aaaa).

```
SELECT b.nom_bataille, DATE_FORMAT(b.date_bataille, '%d %M -%y'),
l.nom_lieu
FROM bataille b, lieu l
WHERE b.id_lieu = l.id_lieu
ORDER BY YEAR(b.date_bataille) ASC, MONTH(b.date_bataille) DESC,
DAY(b.date_bataille) DESC
```

Note : les dates ont lieu avant Jésus-Christ, le tri doit être croissant sur les années et décroissant sur les mois et les jours (afficher le '-' est un bonus).

ELAN
14 rue du Rhône
67100 STRASBOURG
TEL : 03.88.30.78.30
www.elan-formation.fr

6. Nom des potions + coût de réalisation de la potion (trié par coût décroissant).

```
SELECT p.nom_potion, SUM(i.cout_ingredient*c.qte) AS cout_potion
FROM potion p
LEFT JOIN composer c
ON c.id_potion = p.id_potion
LEFT JOIN ingredient i
ON c.id_ingredient = i.id_ingredient
GROUP BY p.id_potion
ORDER BY cout_potion DESC
```

Note : LEFT JOIN permet de constater que la potion 'Miniaturisation' ne coûte rien.

7. Nom des ingrédients + coût + quantité de chaque ingrédient qui composent la potion 'Santé'.

```
SELECT i.nom_ingredient, SUM(i.cout_ingredient*c.qte) AS cout_ingredient
FROM potion p, composer c, ingredient i
WHERE p.id_potion = c.id_potion
AND c.id_ingredient = i.id_ingredient
AND p.nom_potion = 'Santé'
GROUP BY i.id_ingredient
```

8. Nom du ou des personnages qui ont pris le plus de casques dans la bataille 'Bataille du village gaulois'.

```
SELECT p.nom_personnage, SUM(pc.qte) AS nb_casques
FROM personnage p, bataille b, prendre_casque pc
WHERE p.id_personnage = pc.id_personnage
AND pc.id_bataille = b.id_bataille
AND b.nom_bataille = 'Bataille du village gaulois'
GROUP BY p.id_personnage
HAVING nb_casques >= ALL(
            SELECT SUM(pc.qte)
            FROM prendre_casque pc, bataille b
            WHERE b.id_bataille = pc.id_bataille
            AND b.nom_bataille = 'Bataille du village gaulois'
            GROUP BY pc.id_personnage
)
```

Alternative :

```
SELECT p.nom_personnage, SUM(pc.qte) AS nb_casques
FROM personnage p, bataille b, prendre_casque pc
WHERE p.id_personnage = pc.id_personnage
AND pc.id_bataille = b.id_bataille
AND b.nom_bataille = 'Bataille du village gaulois'
GROUP BY p.id_personnage
HAVING nb_casques = (
            SELECT SUM(pc.qte) AS nb_casques
            FROM prendre_casque pc, bataille b
            WHERE b.id_bataille = pc.id_bataille
            AND b.nom_bataille = 'Bataille du village gaulois'
            GROUP BY pc.id_personnage
            ORDER BY nb_casques DESC
            LIMIT 1
)
```


ELAN
14 rue du Rhône
67100 STRASBOURG
TEL : 03.88.30.78.30
www.elan-formation.fr

9. Nom des personnages et, en distinguant chaque potion, la quantité de potion bue. Les classer du plus grand buveur au plus modeste.

```
SELECT p.nom_personnage, pt.nom_potion, SUM(b.dose_boire) AS qte_bue
FROM personnage p, boire b, potion pt
WHERE p.id_personnage = b.id_personnage
AND b.id_potion = pt.id_potion
GROUP BY p.id_personnage, pt.id_potion
ORDER BY qte_bue DESC
```

10. Nom de la bataille où le nombre de casques pris a été le plus important.

```
SELECT b.nom_bataille, SUM(pc.qte) AS nb_casques
FROM bataille b, prendre_casque pc
WHERE b.id_bataille = pc.id_bataille
GROUP BY b.id_bataille
HAVING nb_casques >= ALL(
            SELECT SUM(pc.qte)
            FROM bataille b, prendre_casque pc
            WHERE b.ID_BATAILLE = pc.ID_BATAILLE
            GROUP BY b.id_bataille

)
```

Note : similaire à la requête 7, la variante avec LIMIT fonctionne également ici.

11. Combien existe-t-il de casques de chaque type et quel est leur montant total ? (classés par nombre décroissant)

```
SELECT COUNT(c.id_casque) AS nb_casques, tc.nom_type_casque,
SUM(c.cout_casque) AS total
FROM type_casque tc
LEFT JOIN casque c
ON tc.id_type_casque = c.id_type_casque
GROUP BY tc.id_type_casque
ORDER BY nb_casques DESC
```

Note : LEFT JOIN est préférable dans le cas où un type de casque ne soit représenté par aucun enregistrement de la table 'casque', afin que celui-ci apparaisse tout de même dans les résultats

12. Nom des potions dont la recette comporte du poisson.

```
SELECT p.nom_potion
FROM potion p, ingredient i, composer c
WHERE p.id_potion = c.id_potion
AND c.id_ingredient = i.id_ingredient
AND LOWER(i.nom_ingredient) LIKE "%poisson%"
```

Note : il y a deux ingrédients dont le nom contient "poisson" !

ELAN
14 rue du Rhône
67100 STRASBOURG
TEL : 03.88.30.78.30
www.elan-formation.fr

13. Nom du / des lieu(x) possédant le plus d'habitants, en dehors du village gaulois.

```
SELECT l.nom_lieu, COUNT(p.id_personnage) AS nb
FROM personnage p, lieu l
WHERE p.id_lieu = l.id_lieu
AND l.nom_lieu != 'Village gaulois'
GROUP BY l.id_lieu
HAVING nb >= ALL (
	SELECT COUNT(p.id_personnage)
	FROM personnage p, lieu l
	WHERE l.id_lieu = p.id_lieu
	AND l.nom_lieu != 'Village gaulois'
	GROUP BY l.id_lieu
)
```

Note : attention à exclure le village gaulois des deux requêtes ! (Une variante avec LIMIT dans la sous-requête fonctionnerait également ici).

14. Nom des personnages qui n'ont jamais bu aucune potion.

```
SELECT p.nom_personnage
FROM personnage p
LEFT JOIN boire b
ON p.id_personnage = b.id_personnage
WHERE b.id_personnage IS NULL
GROUP BY p.id_personnage
```

Alternative :

```
SELECT p.nom_personnage
FROM personnage p
WHERE p.id_personnage NOT IN (
	SELECT p.id_personnage
	FROM personnage p, boire b
	WHERE p.id_personnage = b.id_personnage
)
```

15. Nom du / des personnages qui n'ont pas le droit de boire de la potion 'Magique'.

```
SELECT p.nom_personnage
FROM personnage p
WHERE p.id_personnage NOT IN (
	SELECT id_personnage
	FROM autoriser_boire a, potion pt
	WHERE pt.id_potion = a.id_potion
	AND pt.nom_potion = 'Magique'
)
```

Note : bien entendu, Obélix est dans la liste !

ELAN
14 rue du Rhône
67100 STRASBOURG
TEL : 03.88.30.78.30
www.elan-formation.fr

En écrivant toujours des requêtes SQL, modifiez la base de données comme suit :

Note : il est préférable de se baser uniquement sur les informations données : ne pas saisir directement les identifiants correspondants aux critères mais les obtenir systématiquement grâce à un SELECT depuis leurs noms.

A. **Ajoutez le personnage** suivant : Champdeblix, agriculteur résidant à la ferme Hantassion de Rotomagus.

```
INSERT INTO personnage (nom_personnage, adresse_personnage, id_lieu,
id_specialite)
VALUES (
	'Champdeblix',
	'Ferme Hantassion',
	(SELECT id_lieu FROM lieu WHERE nom_lieu = 'Rotomagus'),
	(SELECT id_specialite FROM specialite WHERE nom_specialite =
	'Agriculteur')
)
```

B. **Autorisez** Bonemine à boire de la potion magique, elle est jalouse d'Iélosubmarine...

```
INSERT INTO autoriser_boire (id_potion, id_personnage)
VALUES (
	(SELECT id_potion FROM potion WHERE nom_potion = 'Magique'),
	(SELECT id_personnage FROM personnage WHERE nom_personnage =
	'Bonemine')
)
```

C. Supprimez **les casques grecs** qui n'ont **jamais été pris** lors d'une bataille.

```
DELETE FROM casque
WHERE id_type_casque = (
	SELECT id_type_casque
	FROM type_casque
	WHERE nom_type_casque = 'Grec'
)
AND id_casque NOT IN (
	SELECT pc.id_casque
	FROM prendre_casque pc
)
```

D. Modifiez l'adresse de Zérozérosix : il a été **mis en prison à Condate**.

```
UPDATE personnage
SET adresse_personnage = 'Prison',
    id_lieu = (SELECT id_lieu FROM lieu WHERE nom_lieu = 'Condate')
WHERE nom_personnage = 'Zérozérosix'
```


ELAN
14 rue du Rhône
67100 STRASBOURG
TEL : 03.88.30.78.30
www.elan-formation.fr

E. La potion 'Soupe' ne **doit plus contenir de persil**.

```
DELETE FROM composer
WHERE id_potion = (
	SELECT id_potion
	FROM potion
	WHERE nom_potion = 'Soupe'
)
AND id_ingredient = (
	SELECT id_ingredient
	FROM ingredient
	WHERE nom_ingredient = 'Persil'
)
```

F. Obélix s'est trompé : ce sont **42** casques **Weisenau**, et non Ostrogoths, qu'il a pris lors de la bataille 'Attaque de la banque postale'. Corrigez son erreur !

```
UPDATE prendre_casque
SET id_casque = (
	SELECT id_casque FROM casque WHERE nom_casque = 'Weisenau'
)
WHERE id_personnage = (
	SELECT id_personnage FROM personnage WHERE nom_personnage = 'Obélix'
)
AND id_bataille = (
	SELECT id_bataille FROM bataille WHERE nom_bataille = 'Attaque de la
	banque postale'
)
AND id_casque = (
	SELECT id_casque FROM casque WHERE nom_casque = 'Ostrogoth'
)
```


ELAN
14 rue du Rhône
67100 STRASBOURG
TEL : 03.88.30.78.30
www.elan-formation.fr