1. Nom des lieux qui finissent par 'um'

**SELECT** nom\_lieu

**FROM** lieu

**WHERE** nom\_lieu **LIKE** "%um"

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

**SELECT** nom\_lieu, **COUNT**(nom\_personnage) **AS** nbPerso

**FROM** personnage p, lieu l

**WHERE** p.id\_lieu = l.id\_lieu

**GROUP** **BY** nom\_lieu

**ORDER** **BY** nbPerso **DESC**

Autre version

**SELECT** l.nom\_lieu, **COUNT**(p.id\_personnage) **AS** nbHabitants

**FROM** personnage p

**INNER** **JOIN** lieu l **ON** p.id\_lieu = l.id\_lieu

**GROUP** **BY** l.nom\_lieu

**ORDER** **BY** nbHabitants **DESC**

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

**SELECT** p.nom\_personnage, s.nom\_specialite, p.adresse\_personnage, l.nom\_lieu

**FROM** personnage p, lieu l, specialite s

**WHERE** p.id\_lieu = l.id\_lieu

**AND** p.id\_specialite = s.id\_specialite

**ORDER** **BY** l.nom\_lieu, p.nom\_personnage

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

**SELECT** s.nom\_specialite, **COUNT**(p.nom\_personnage) **AS** nbPerso

**FROM** personnage p, specialite s

**WHERE** s.id\_specialite = p.id\_specialite

**GROUP** **BY** s.nom\_specialite

**ORDER** **BY** nbPerso **DESC**

Autre version pour afficher les zero (LEFT JOIN ou RIGHT JOIN a la place de INNER JOIN)

**SELECT** s.nom\_specialite, **COUNT**(p.nom\_personnage) **AS** nbPerso

**FROM** specialite s

**LEFT** **JOIN** personnage p **ON** s.id\_specialite = p.id\_specialite

**GROUP** **BY** s.nom\_specialite

**ORDER** **BY** nbPerso **DESC**

5. Nom, date et lieu des batailles, classées 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") **AS** dateBataille, l.nom\_lieu

**FROM** bataille b, lieu l

**WHERE** b.id\_lieu = l.id\_lieu

**ORDER** **BY** b.date\_bataille **DESC**

(Avec les personnages qui ont combattu)

**SELECT** p.nom\_personnage, b.nom\_bataille, **DATE\_FORMAT**(b.date\_bataille, "%d %M %Y") **AS** dateBataille, l.nom\_lieu

**FROM** personnage p, bataille b, lieu l

**WHERE** p.id\_lieu = l.id\_lieu

**AND** b.id\_lieu = l.id\_lieu

**ORDER** **BY** b.date\_bataille **DESC**

Autre version de date (pour montrer que ça s’est passé avant JC)

**SELECT** b.nom\_bataille, **DATE\_FORMAT**(b.date\_bataille, "%d %M -%y") **AS** dateBataille, 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**

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

**SELECT** p.nom\_potion, **SUM**(i.cout\_ingredient\*s.qte) **AS** coutTotal

**FROM** potion p, ingredient i, composer s

**WHERE** p.id\_potion = s.id\_potion

**AND** i.id\_ingredient = s.id\_ingredient

**GROUP** **BY** p.nom\_potion

**ORDER** **BY** coutTotal **DESC**

Avec affichage 10,00 euro :

**SELECT** p.nom\_potion, **ROUND**(**SUM**(i.cout\_ingredient\*s.qte), 2) **AS** coutTotal

**FROM** potion p, ingredient i, composer s

**WHERE** p.id\_potion = s.id\_potion

**AND** i.id\_ingredient = s.id\_ingredient

**GROUP** **BY** p.nom\_potion

**ORDER** **BY** coutTotal **DESC**

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

**SELECT** i.nom\_ingredient, s.qte, i.cout\_ingredient

**FROM** ingredient i, composer s, potion p

**WHERE** i.id\_ingredient = s.id\_ingredient

**AND** s.id\_potion = p.id\_potion

**AND** p.nom\_potion = "Santé"

(avec le nom de la potion)

**SELECT** p.nom\_potion ,i.nom\_ingredient, s.qte, i.cout\_ingredient

**FROM** ingredient i, composer s, potion p

**WHERE** i.id\_ingredient = s.id\_ingredient

**AND** s.id\_potion = p.id\_potion

**AND** p.nom\_potion = "Santé"

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

**SELECT** p.nom\_personnage, pc.qte

**FROM** personnage p, bataille b, prendre\_casque pc

**WHERE** pc.id\_bataille = b.id\_bataille

**AND** pc.id\_personnage = p.id\_personnage

**AND** b.nom\_bataille = "Bataille du village gaulois"

**HAVING** pc.qte >= **ALL** (

**SELECT** pc.qte

**FROM** bataille b, prendre\_casque pc

**WHERE** pc.id\_bataille = b.id\_bataille

**AND** b.nom\_bataille = "Bataille du village gaulois"

)

Bonne version :

**SELECT** p.nom\_personnage, **SUM**(pc.qte) **AS** qttCasque

**FROM** prendre\_casque pc

**INNER** **JOIN** bataille b **ON** b.id\_bataille = pc.id\_bataille

**INNER** **JOIN** personnage p **ON** p.id\_personnage = pc.id\_personnage

**WHERE** b.nom\_bataille = "Bataille du village gaulois"

**GROUP** **BY** p.nom\_personnage

**HAVING** qttCasque >= **ALL** ( **SELECT** **SUM**(pc.qte)

**FROM** prendre\_casque pc

**INNER** **JOIN** bataille b **ON** b.id\_bataille = pc.id\_bataille

**INNER** **JOIN** personnage p **ON** p.id\_personnage = pc.id\_personnage

**WHERE** b.nom\_bataille = "Bataille du village gaulois"

**GROUP** **BY** p.nom\_personnage )

9. Nom des personnages et leur quantité de potion bue (en les classant du plus grand buveur au plus petit)

**SELECT** p.nom\_personnage, **SUM**(b.dose\_boire) **AS** qtt

**FROM** personnage p, boire b

**WHERE** p.id\_personnage = b.id\_personnage

**GROUP** **BY** p.nom\_personnage

**ORDER** **BY** qtt **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** total

**FROM** bataille b, prendre\_casque pc

**WHERE** pc.id\_bataille = b.id\_bataille

**GROUP** **BY** b.nom\_bataille

**HAVING** total >= **ALL**(

**SELECT** **SUM**(pc.qte)

**FROM** bataille b, prendre\_casque pc

**WHERE** pc.id\_bataille = b.id\_bataille

**GROUP** **BY** b.nom\_bataille

)

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

**SELECT** tc.nom\_type\_casque, **COUNT**(c.nom\_casque), **SUM**(c.cout\_casque) **AS** total

**FROM** casque c, type\_casque tc

**WHERE** c.id\_type\_casque = tc.id\_type\_casque

**GROUP** **BY** tc.nom\_type\_casque

**ORDER** **BY** total **DESC**

12. Nom des potions dont un des ingrédients est le poisson frais

**SELECT** p.nom\_potion

**FROM** potion p, ingredient i, composer c

**WHERE** p.id\_potion = c.id\_potion

**AND** i.id\_ingredient = c.id\_ingredient

**AND** i.nom\_ingredient = "Poisson frais"

On peut mettre LOWER pour ne pas avoir a mettre de majuscules

**SELECT** p.nom\_potion

**FROM** potion p, ingredient i, composer c

**WHERE** p.id\_potion = c.id\_potion

**AND** i.id\_ingredient = c.id\_ingredient

**AND** **LOWER**(i.nom\_ingredient) = "poisson frais"

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** nbPerso

**FROM** personnage p, lieu l

**WHERE** p.id\_lieu = l.id\_lieu

**AND** l.nom\_lieu != "Village gaulois"

**GROUP** **BY** l.nom\_lieu

**HAVING** nbPerso >= **ALL** (

**SELECT** **COUNT**(p.id\_personnage)

**FROM** personnage p, lieu l

**WHERE** p.id\_lieu = l.id\_lieu

**AND** l.nom\_lieu != "Village gaulois"

**GROUP** **BY** l.nom\_lieu

)

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

**SELECT** p.nom\_personnage

**FROM** personnage p

**WHERE** p.id\_personnage

**NOT** **IN** (

**SELECT** b.id\_personnage

**FROM** boire b

)

Ou

**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.nom\_personnage

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

Personnage pas autorisé a boire de la potion magique

**SELECT** p.nom\_personnage

**FROM** personnage p

**WHERE** p.id\_personnage

**NOT** **IN** (

**SELECT** ab.id\_personnage

**FROM** personnage p, autoriser\_boire ab, potion po

**WHERE** p.id\_personnage = ab.id\_personnage

**AND** ab.id\_potion = po.id\_potion

**AND** po.nom\_potion = "Magique"

)

Même version que précédemment mais raccourci car le Personnage était inutile dans le NOT IN

**SELECT** p.nom\_personnage

**FROM** personnage p

**WHERE** p.id\_personnage

**NOT** **IN** (

**SELECT** ab.id\_personnage

**FROM** autoriser\_boire ab, potion po

**WHERE** ab.id\_potion = po.id\_potion

**AND** po.nom\_potion = "Magique"

)

Personnage qui n’ont jamais bu de potion magique

**SELECT** p.nom\_personnage

**FROM** personnage p

**WHERE** p.id\_personnage

**NOT** **IN** (

**SELECT** b.id\_personnage

**FROM** boire b, personnage p, potion po

**WHERE** p.id\_personnage = b.id\_personnage

**AND** b.id\_potion = po.id\_potion

**AND** po.nom\_potion = "Magique"

)

PARTIE 2 :

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

**INSERT** **INTO** personnage (id\_personnage, nom\_personnage, adresse\_personnage, id\_lieu, id\_specialite)

**VALUES** (100, "Champdeblix", "ferme Hantassion", 6, 12)

On peut aussi enlever l’ID pour le laisser s’incrémenter tout seul :

**INSERT** **INTO** personnage (nom\_personnage, adresse\_personnage, id\_lieu, id\_specialite)

**VALUES** ("Champdeblix", "ferme Hantassion", 6, 12)

Ou

**INSERT** **INTO** personnage (nom\_personnage, adresse\_personnage, id\_lieu, id\_specialite)

**VALUES** (

"Champdeblix",

"ferme Hantassion",

(**SELECT** l.id\_lieu **FROM** lieu l **WHERE** l.nom\_lieu = "Rotomagus"),

(**SELECT** s.id\_specialite **FROM** specialite s **WHERE** s.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** (1, 12)

Ou

**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 = 2

**AND** id\_casque **NOT** **IN** (

**SELECT** id\_casque

**FROM** prendre\_casque

)

Ou

**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** id\_casque

**FROM** prendre\_casque

)

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

**UPDATE** personnage

**SET** adresse\_personnage = "prison", id\_lieu = 9

**WHERE** nom\_personnage = "Zérozérosix"

Ou

**UPDATE** personnage

**SET** adresse\_personnage = "prison",

id\_lieu = (

**SELECT** id\_lieu

**FROM** lieu

**WHERE** nom\_lieu = "Condate"

)

**WHERE** nom\_personnage = "Zérozérosix"

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

**DELETE** **FROM** composer

**WHERE** id\_potion = 9

**AND** id\_ingredient = 19

Ou

**DELETE** **FROM** composer

**WHERE** id\_potion = (

**SELECT** p.id\_potion

**FROM** potion p

**WHERE** p.nom\_potion = "Soupe"

)

**AND** id\_ingredient = (

**SELECT** i.id\_ingredient

**FROM** ingredient i

**WHERE** i.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 = 10

**WHERE** id\_personnage = 5

**AND** id\_bataille = 9

Ou

**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"

)

Ou

**UPDATE** prendre\_casque

**SET** id\_casque = (**SELECT** id\_casque **FROM** casque **WHERE** nom\_casque = "Weisenau"),

qte = 42

**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")

Test

**SELECT** p.nom\_personnage, **SUM**(pc.qte) **AS** quantite

**FROM** personnage p

**LEFT** **JOIN** prendre\_casque pc **ON** p.id\_personnage = pc.id\_personnage

**WHERE** p.nom\_personnage **LIKE** "%ix"

**GROUP** **BY** p.nom\_personnage

**ORDER** **BY** quantite **DESC**

BONUS

1) Adresse mail pour chaque gaulois

Premières lettres du nom du gaulois + un point + 4 premières lettres du lieu d’habitation + “gaulois.fr :

**SELECT** **LOWER**(**CONCAT**(**LEFT**(p.nom\_personnage, 5), ".",**LEFT**(p.adresse\_personnage, 4), "@gaulois.fr"))

**FROM** personnage p

2) Nombre de jours écoulés entre 2 batailles de votre choix :

**SELECT** "Siège d'Alesia", "Bataille des cinq armées", **DATEDIFF**(

(**SELECT** b.date\_bataille

**FROM** bataille b

**WHERE** b.nom\_bataille = "Siège d'Alesia"),

(**SELECT** b.date\_bataille

**FROM** bataille b

**WHERE** b.nom\_bataille = "Bataille des cinq armées")

)

**AS** nbJour

3) Nom des types de casque + nom des casques associés :

**SELECT** tc.nom\_type\_casque, **GROUP\_CONCAT**(" ", c.nom\_casque)

**FROM** type\_casque tc, casque c

**WHERE** tc.id\_type\_casque = c.id\_type\_casque

**GROUP** **BY** tc.nom\_type\_casque

Ou

**SELECT** tc.nom\_type\_casque **AS** "Type casque",

**GROUP\_CONCAT**(**DISTINCT**(c.nom\_casque) **SEPARATOR** ", ") **AS** "Nom casque"

**FROM** casque c

**INNER** **JOIN** type\_casque tc **ON** tc.id\_type\_casque = c.id\_type\_casque

**GROUP** **BY** tc.nom\_type\_casque