

EFREI Ing-1-NEW

Mini Projet Base de données

Partie 2 : Insertion & Interrogation des données

Grégoire BELLEPERCHE & Thomas QUERREC
20/10/2025

Table des matières

1) Contexte métier minimal et rôle utilisateur	2
2) Questions métier → types de requêtes (ce qu'on cherche et pourquoi)	2
Projections / Sélections (Q1 → Q5 + liste DISTINCT)	2
Agrégations (Q6 → Q10 + guildes/classes)	3
Jointures (Q11 → Q15 + variantes)	3
Sous-requêtes (Q16 → fin) : EXISTS, NOT EXISTS, ANY, ALL, comparaisons.....	3
Résumé :.....	4
Ce qu'on a fait en conséquence	5
3) Respect des consignes (Scénario).....	5

Après création et insertion des données dans la base de donnée. On s'est demandé comment relier nos tables à un véritable besoin. Plutôt que d'enchaîner des requêtes "gratuites", on a posé un rôle (utilisateur) et des questions métier. L'idée : si on comprend *pourquoi* on interroge, on écrit des requêtes plus simples, plus propres, et surtout alignées avec le cahier des charges (projections/sélections, agrégations, jointures variées, sous-requêtes).

1) Contexte métier minimal et rôle utilisateur

- Contexte narratif : le monde de Solarys (nos référentiels : royaumes, espèces, lignées, ressources, guildes, classes, etc. déjà modélisés/peuplés).
- Rôle utilisateur : l'Archiviste royal conseille le Roi. Il doit extraire des informations pour orienter trois décisions : (I) fiscalité/ressources, (II) capacité militaire/titres, (III) santé publique/expéditions.
- Principe : chaque requête répond à un *ordre du Roi* clairement énoncé en commentaire dans 4_interrogation.sql. Nous avons gardé un style "RP" pour la vidéo mais une structure 100% SQL et vérifiable.

2) Questions métier → types de requêtes (ce qu'on cherche et pourquoi)

Projections / Sélections (Q1 → Q5 + liste DISTINCT)

Objectif : obtenir vite des listes "actionnables" pour l'intendance et la régulation.

- Q1 (royaumes pacifiques, triés par richesse, LIMIT 10) : repérer les contributeurs fiscaux "prioritaires".
- Q2 (métiers contenant "mage", LIKE) : balisage des professions à surveiller (magie).
- Q3 (espèces sapientes) : clarifie "qui est gouvernable" au sens des espèces intelligentes.
- Q4 (ressources à "aura magique", LIKE) : matière première pour taxes/contrôles.
- Q5 (royaumes de richesse moyenne via BETWEEN 50000 AND 80000) : on a introduit BETWEEN ici pour montrer la palette d'opérateurs ; ça reste lisible et pertinent.
- DISTINCT (rôles de combat) : une vue synthétique de la typologie de classes.

Agrégations (Q6 → Q10 + guildes/classes)

Objectif : fournir des indicateurs au Roi, pas juste des listes.

- Q6 (moyenne des richesses) : thermomètre macro des domaines.
- Q7 (compte des espèces sages) : ordre de grandeur de la population “pensante”.
- Q8 (ressources par initiale, GROUP BY + HAVING > 3) : petit contrôle de volumétrie (utile pour montrer HAVING).
- Q9 (“de la” dans les lignées) : exemple d’agrégat sémantique simple (filtre lexical).
- Q10 (somme d’or des royaumes “paix”) : estimation du trésor “stabilisé”.
- Guildes ≥ 3 membres / Classes par rôle : *santé* des organisations et équilibre des profils (DPS/Tank/Soin/Support).

Jointures (Q11 → Q15 + variantes)

Objectif : naviguer proprement entre référentiels et n–n, en diversifiant les types de jointures.

- Q11 INNER JOIN (Personnage ↔ Royaume) : registre d’origine.
- Q12 LEFT JOIN (Personnage ↔ Lignée) : on voit ceux sans lignée (utile en data quality).
- Q13 JOIN (RoyaumeRessource ↔ Ressource ↔ Royaume) : vue standard “ressource par royaume”.
- Variante RIGHT JOIN (même tableau “par la droite”) : on montre une autre manière d’arriver au même résultat, pour satisfaire “joins variés”.
- Q14 (PersonnageClasse avec niveaux) : rattache les compétences combat à des individus.
- Q15 (Guildes ↔ Membres) : le “who’s who” des organisations.
- CROSS JOIN (rôles × types de métiers) : matrice conceptuelle pour les planifs de formation/affectation.

Sous-requêtes (Q16 → fin) : EXISTS, NOT EXISTS, ANY, ALL, comparaisons

Objectif : exprimer des règles “métiers” naturelles (exister / ne pas exister, être au-dessus d’une moyenne locale).

- Q16 (plus riches que la moyenne + ORDER BY) : priorisation fiscale.
- Q17 (lignées sans descendants) : repérer les “branches mortes”.
- Q18 (métiers non attribués) : trous de couverture dans les filières.
- Q19 (royaumes sans ressource “principale”) et Q20 (royaumes avec ressource “rare”) : double filtre NOT IN / EXISTS pour illustrer deux styles.
- Mana (≥ moyenne d’espèce, ALL) : vrais “héros” à repérer.
- Titres actifs (EXISTS) : statut nobiliaire “en cours”.

- Santé (maladies) : suivi sanitaire élémentaire.
- Guildes sans bataille (NOT EXISTS) : inactivité opérationnelle.
- Armées > moyenne du royaume (ANY + sous-requête corrélée) : focus sur “sur-capacités”.
- Expéditions sans membres (NOT EXISTS) : housekeeping (éviter les coquilles vides).
- **Notes : Q17 et Q18** utilisent NOT IN. Si une sous-requête peut renvoyer des **NULL**, NOT IN peut conduire à “aucun résultat” (3-valued logic). Dans nos données d’insertion, lignee_id est géré proprement ; sinon, on passerait en **NOT EXISTS** :
Exemple (Q17 robuste) :
WHERE NOT EXISTS (SELECT 1 FROM Personnage p WHERE p.lignee_id = Lignee.id_lignee)
Idem pour Q18 avec PersonnageMetier. On garde NOT IN ici pour varier les patterns, mais on connaît la limite.

Résumé :

L’Archiviste royal travaille sur trois grands axes de décision :

1. **La richesse et les ressources du royaume**
connaître la distribution de l’or, les royaumes prospères, les ressources rares et les secteurs en déclin.
Ces informations sont essentielles pour orienter la fiscalité et les alliances économiques.
2. **La population et la structure sociale**
recenser les peuples, les lignées nobles, les métiers et les classes de combat.
L’objectif est de mieux organiser la société, repérer les manques de formation, et suivre la noblesse active.
3. **La sécurité et la santé du continent**
analyser les guildes, les armées, les expéditions et les maladies.
Ces données servent à évaluer la force militaire, la cohésion des organisations, et les risques sanitaires.

En résumé, le Roi a besoin d’une vue complète de son royaume : richesse, population, forces, faiblesses.

Et l’Archiviste, donc nous dans le scénario, doit être capable de répondre rapidement à toutes ces questions grâce au SQL.

Ce qu'on a fait en conséquence

Pour répondre à ces besoins, on a structuré nos requêtes en quatre grands ensembles, chacun répondant à une utilité bien précise.

1. Projections et sélections : “voir et filtrer”

On a commencé par extraire les informations de base : les royaumes les plus riches, les métiers liés à la magie, les espèces intelligentes, ou encore les ressources particulières.

Ces requêtes servent à identifier et comparer rapidement les entités du monde.

Elles posent la base du rapport : *qui possède quoi, et où ?*

2. Fonctions d'agrégation : “mesurer et évaluer”

Ensuite, on a calculé des moyennes, des sommes et des comptages.

Cela donne une vision plus globale du royaume : combien d'or circule, combien d'espèces existent, combien de guildes sont réellement actives.

Ces indicateurs permettent de prendre des décisions macroéconomiques et d'avoir un tableau de bord clair.

3. Jointures : “lier et croiser les informations”

On a relié les tables pour établir des correspondances concrètes : quel personnage appartient à quel royaume, quelles ressources sont liées à quelle région, quelles guildes regroupent quels membres.

L'idée ici était d'obtenir une vue relationnelle complète du monde — indispensable pour gérer les interactions entre royaumes, métiers, lignées et guildes.

On a volontairement utilisé plusieurs types de jointures (INNER, LEFT, RIGHT, CROSS, FULL OUTER via UNION) pour montrer la variété et répondre au cahier des charges.

4. Sous-requêtes : “analyser et décider”

Enfin, on a créé des requêtes plus fines, qui comparent, filtrent ou détectent des cas particuliers.

On s'en est servi pour repérer les royaumes plus riches que la moyenne, les lignées sans descendants, les métiers sans apprentis, ou les expéditions sans membre.

Ces requêtes simulent les décisions intelligentes de l'Archiviste : elles permettent d'agir (par exemple : mobiliser une guilde inactive, dissoudre une expédition vide, ou taxer un royaume prospère).

3) Respect des consignes (Scénario)

- Il relie chaque requête à un besoin utilisateur explicite (le Roi/gouvernance), ce que le sujet demande : un scénario d'utilisation avec liste de données à extraire.
- Il couvre toutes les catégories imposées (≥5 chacune) et diversifie les jointures (INNER/LEFT/RIGHT/CROSS et FULL OUTER émulé).
- Il montre les bonnes pratiques SQL : DISTINCT à bon escient, LIKE pour masques, GROUP BY/HAVING, EXISTS/NOT EXISTS, comparaison à la moyenne, ALL/ANY pour

exprimer des contraintes relatives, et tri quand l'usage décisionnel l'exige (ex. priorisation fiscale).