



# Requêtes SQL complexes pour le schéma PEP-3

## Analyses centrées sur les patients

sql

Copy

```
-- Objectif : Lister Le nombre d'évaluations et La moyenne des scores PEP-3 p
SELECT P.id_patient, P.nom, P.prenom,
       COUNT(E.id_evaluation) AS nb_evaluations,
       ROUND(AVG(R.score_competence),2) AS moyenne_score
FROM Patient P
LEFT JOIN Evaluation E ON P.id_patient = E.id_patient
LEFT JOIN Resultat R ON E.id_evaluation = R.id_evaluation
GROUP BY P.id_patient, P.nom, P.prenom
ORDER BY nb_evaluations DESC, moyenne_score DESC;
```

sql

Copy

```
-- Objectif : Trouver Les patients ayant réalisé plus de 5 évaluations
SELECT P.id_patient, P.nom, P.prenom, COUNT(E.id_evaluation) AS nb_evaluation
FROM Patient P
JOIN Evaluation E ON P.id_patient = E.id_patient
GROUP BY P.id_patient, P.nom, P.prenom
HAVING COUNT(E.id_evaluation) > 5
ORDER BY nb_evaluations DESC;
```

sql

Copy

```
-- Objectif : Rapport mensuel du nombre de nouvelles inscriptions de patients
SELECT TO_CHAR(P.date_inscription,'YYYY-MM') AS mois,
       COUNT(*) AS nb_inscriptions
FROM Patient P
```

```
GROUP BY TO_CHAR(P.date_inscription, 'YYYY-MM')
ORDER BY mois;
```

sql

 Copy

```
-- Objectif : Identifier les patients sans évaluation depuis plus d'un an (an
SELECT P.id_patient, P.nom, P.prenom, MAX(E.date) AS date_derniere_eval
FROM Patient P
LEFT JOIN Evaluation E ON P.id_patient = E.id_patient
GROUP BY P.id_patient, P.nom, P.prenom
HAVING MAX(E.date) < ADD_MONTHS(SYSDATE, -12) OR MAX(E.date) IS NULL;
```

sql

 Copy

```
-- Objectif : Récapitulatif des bilans annuels pour chaque patient
SELECT P.id_patient, P.nom, P.prenom, B.annee, B.synthese, B.progres_observes
FROM Patient P
LEFT JOIN Bilan_Annuel B ON P.id_patient = B.id_patient
ORDER BY P.id_patient, B.annee;
```

sql

 Copy

```
-- Objectif : Obtenir le dernier bilan annuel de chaque patient (sous-requête
SELECT P.id_patient, P.nom, P.prenom, B.annee, B.synthese, B.progres_observes
FROM Patient P
LEFT JOIN Bilan_Annuel B ON P.id_patient = B.id_patient
WHERE B.annee = (
    SELECT MAX(B2.annee)
    FROM Bilan_Annuel B2
    WHERE B2.id_patient = P.id_patient
);
```

sql

 Copy

```
-- Objectif : Patients n'ayant pas de bilan annuel pour l'année en cours (anc
SELECT P.id_patient, P.nom, P.prenom
FROM Patient P
WHERE NOT EXISTS (
    SELECT 1 FROM Bilan_Annuel B
```

```
WHERE B.id_patient = P.id_patient
AND B.annee = EXTRACT(YEAR FROM SYSDATE)
);
```

sql

Copy

```
-- Objectif : Lister les parents associés à chaque patient
SELECT P.id_patient, P.nom AS nom_patient, P.prenom AS prenom_patient,
       PR.numCIN_parent, PR.nom AS nom_parent, PR.prenom AS prenom_parent,
       PR.relation_avec_enfant
FROM Patient P
LEFT JOIN Parent_Enfant PE ON P.id_patient = PE.id_patient
LEFT JOIN Parent PR ON PE.numCIN_parent = PR.numCIN_parent
ORDER BY P.id_patient;
```

## Suivi des compétences évaluées et testées

sql

Copy

```
-- Objectif : Obtenir la moyenne des scores par niveau de difficulté des tests
SELECT T.niveau_difficulte,
       ROUND(AVG(RT.score_brut),2) AS score_moyen,
       COUNT(*) AS nb_tests
FROM Test T
JOIN Resultat_test RT ON T.id_test = RT.id_test
GROUP BY T.niveau_difficulte
ORDER BY T.niveau_difficulte;
```

sql

Copy

```
-- Objectif : Liste des compétences ciblées par chaque plan d'intervention
SELECT PI.id_plan, PI.objectifs, C.id_competence, C.nom_capacite
FROM Plan_Intervention PI
JOIN Plan_Competence PC ON PI.id_plan = PC.id_plan
JOIN Competence C ON PC.id_competence = C.id_competence
ORDER BY PI.id_plan;
```

sql

Copy

```
-- Objectif : Calculer le score moyen par compétence pour un patient donné (1)
SELECT C.id_compétence, C.nom_capacité, ROUND(AVG(R.score_compétence),2) AS score_moyen
FROM Compétence C
JOIN Resultat R ON C.id_compétence = R.id_compétence
JOIN Evaluation E ON R.id_evaluation = E.id_evaluation
WHERE E.id_patient = :id_patient
GROUP BY C.id_compétence, C.nom_capacité
ORDER BY score_moyen DESC;
```

sql

Copy

```
-- Objectif : Compétences peu évaluées (moins de 10 résultats enregistrés)
SELECT C.id_compétence, C.nom_capacité, COUNT(R.id_resultat) AS occurrences
FROM Compétence C
LEFT JOIN Resultat R ON C.id_compétence = R.id_compétence
GROUP BY C.id_compétence, C.nom_capacité
HAVING COUNT(R.id_resultat) < 10
ORDER BY occurrences;
```

sql

Copy

```
-- Objectif : Détourner les résultats de test supérieurs à la moyenne générale
SELECT RT.id_resTest, RT.score_brut, RT.date_test
FROM Resultat_test RT
WHERE RT.score_brut > (SELECT AVG(score_brut) FROM Resultat_test)
ORDER BY RT.score_brut DESC;
```

sql

Copy

```
-- Objectif : Détails des tests : nom, difficulté, score brut et observation
SELECT T.nom_test, T.niveau_difficulté, RT.score_brut, RT.observation
FROM Resultat_test RT
JOIN Test T ON RT.id_test = T.id_test
ORDER BY RT.score_brut DESC, T.niveau_difficulté;
```

sql

Copy

```
-- Objectif : Nombre de compétences associées à chaque plan d'intervention
SELECT PI.id_plan, COUNT(PC.id_competence) AS nb_competences
FROM Plan_Intervention PI
LEFT JOIN Plan_Competence PC ON PI.id_plan = PC.id_plan
GROUP BY PI.id_plan
HAVING COUNT(PC.id_competence) > 0
ORDER BY nb_competences DESC;
```

sql

Copy

```
-- Objectif : Répartition des compétences ciblées par les plans d'intervention
SELECT C.id_competence, C.nom_capacite, COUNT(DISTINCT PC.id_plan) AS nb_plans
FROM Competence C
JOIN Plan_Competence PC ON C.id_competence = PC.id_competence
JOIN Plan_Intervention PI ON PC.id_plan = PI.id_plan
WHERE PI.statut = 'en cours'
GROUP BY C.id_competence, C.nom_capacite
ORDER BY nb_plans DESC;
```

## Activité des praticiens et des centres

sql

Copy

```
-- Objectif : Nombre d'évaluations réalisées par chaque praticien
SELECT Pr.id_praticien, Pr.nom, Pr.prenom, COUNT(E.id_evaluation) AS nb_evaluations
FROM Praticien Pr
LEFT JOIN Evaluation E ON Pr.id_praticien = E.id_praticien
GROUP BY Pr.id_praticien, Pr.nom, Pr.prenom
ORDER BY nb_evaluations DESC;
```

sql

Copy

```
-- Objectif : Nombre de rendez-vous pris par chaque praticien au cours du dernier mois
SELECT Pr.id_praticien, Pr.nom, Pr.prenom, COUNT(R.id_rdv) AS nb_rendezvous
FROM Praticien Pr
LEFT JOIN RendezVous R ON Pr.id_praticien = R.id_praticien
AND R.date_heure >= ADD_MONTHS(TRUNC(SYSDATE, 'MM'), -1)
AND R.date_heure < TRUNC(SYSDATE, 'MM');
```

```
GROUP BY Pr.id_praticien, Pr.nom, Pr.prenom
ORDER BY nb_rendezvous DESC;
```

sql

 Copy

```
-- Objectif : Nombre de sessions de formation animées par centre
SELECT C.id_centre, C.nom, COUNT(SF.id_session) AS nb_sessions
FROM Centre C
LEFT JOIN Session_Formation SF ON C.id_centre = SF.id_centre
GROUP BY C.id_centre, C.nom
ORDER BY nb_sessions DESC;
```

sql

 Copy

```
-- Objectif : Identifier Le praticien ayant animé Le plus de sessions de form
SELECT Pr.id_praticien, Pr.nom, Pr.prenom, COUNT(SF.id_session) AS nb_session
FROM Praticien Pr
LEFT JOIN Session_Formation SF ON Pr.id_praticien = SF.id_praticien
GROUP BY Pr.id_praticien, Pr.nom, Pr.prenom
ORDER BY nb_sessions DESC
FETCH FIRST 1 ROWS ONLY;
```

sql

 Copy

```
-- Objectif : Activité mensuelle des centres : évaluations et rendez-vous
SELECT C.id_centre, C.nom AS centre, TO_CHAR(E.date, 'YYYY-MM') AS mois,
       COUNT(DISTINCT E.id_evaluation) AS nb_evaluations,
       COUNT(R.id_rdv) AS nb_rendezvous
FROM Centre C
LEFT JOIN Evaluation E ON C.id_centre = E.id_centre
LEFT JOIN RendezVous R ON C.id_centre = R.id_centre
GROUP BY C.id_centre, C.nom, TO_CHAR(E.date, 'YYYY-MM')
ORDER BY C.id_centre, mois;
```

sql

 Copy

```
-- Objectif : Centres ayant organisé plus de 20 rendez-vous au cours du dern
SELECT C.id_centre, C.nom, COUNT(R.id_rdv) AS nb_rendezvous
FROM Centre C
JOIN RendezVous R ON C.id_centre = R.id_centre
```

```
WHERE R.date_heure >= ADD_MONTHS(TRUNC(SYSDATE, 'MM'), -3)
      AND R.date_heure < TRUNC(SYSDATE, 'MM')
GROUP BY C.id_centre, C.nom
HAVING COUNT(R.id_rdv) > 20;
```

## Gestion des plans d'intervention et leur impact

sql

 Copy

```
-- Objectif : Nombre de plans d'intervention créés par chaque praticien
SELECT Pr.id_praticien, Pr.nom, Pr.prenom, COUNT(PI.id_plan) AS nb_plans
FROM Praticien Pr
LEFT JOIN Plan_Intervention PI ON Pr.id_praticien = PI.id_praticien
GROUP BY Pr.id_praticien, Pr.nom, Pr.prenom
ORDER BY nb_plans DESC;
```

sql

 Copy

```
-- Objectif : Objectifs et statut des plans d'intervention créés dans la dernière année
SELECT id_plan, objectifs, statut, date_creation
FROM Plan_Intervention
WHERE date_creation > ADD_MONTHS(SYSDATE, -12)
ORDER BY date_creation;
```

sql

 Copy

```
-- Objectif : Plans d'intervention en cours sans révision récente (depuis 6 mois)
SELECT id_plan, date_creation, date_revision, statut
FROM Plan_Intervention
WHERE statut = 'en cours'
      AND (date_revision IS NULL OR date_revision < ADD_MONTHS(SYSDATE, -6));
```

sql

 Copy

```
-- Objectif : Nombre de patients ciblés par chaque plan d'intervention
SELECT PI.id_plan, COUNT(PP.id_patient) AS nb_patients
FROM Plan_Intervention PI
```

```
LEFT JOIN Plan_Patient PP ON PI.id_plan = PP.id_plan
GROUP BY PI.id_plan;
```

sql

 Copy

```
-- Objectif : Répartition des plans d'intervention par année de création
SELECT EXTRACT(YEAR FROM date_creation) AS annee, COUNT(*) AS nb_plans
FROM Plan_Intervention
GROUP BY EXTRACT(YEAR FROM date_creation)
ORDER BY annee;
```

sql

 Copy

```
-- Objectif : Plans d'intervention dont les objectifs contiennent le mot 'communication'
SELECT id_plan, objectifs, statut
FROM Plan_Intervention
WHERE LOWER(objectifs) LIKE '%communication%';
```

sql

 Copy

```
-- Objectif : Nombre de compétences visées par chaque plan (sous-requête corrélée)
SELECT PI.id_plan, PI.objectifs,
       (SELECT COUNT(*) FROM Plan_Compétence PC WHERE PC.id_plan = PI.id_plan) AS nb_comp
FROM Plan_Intervention PI;
```

sql

 Copy

```
-- Objectif : Délai moyen entre création et révision des plans d'intervention
SELECT ROUND(AVG(date_revision - date_creation), 2) AS delai_moyen_jours
FROM Plan_Intervention
WHERE date_revision IS NOT NULL;
```

## All Sources