|  |  |
| --- | --- |
| Accueil - ScoDoc  **Partie 2 : Insertion des données, requêtes SQL**  **SAÉ 2.04 – Exploitation d’une base de données** | **RÉsumÉ**  Dans ce rapport vous allez pouvoir observer les différents réponses SQL pour une question donnée, avec une image du résultat obtenue.  **Groupe :**   * **42**   **Auteurs**:   * **HÂTET Lucas** * **TASDEMIR Sefer** * **TINASTEPE Feyza** |

**SOMMAIRE**

[1. Intitulé et suivi des requêtes 2](#_Toc162956669)

1. Intitulé et résultats des requêtes

1. Identifier les étudiants sans note pour une évaluation spécifique (en précisant le libellé de l'évaluation et le code du module).

SELECT E.NOM, E.PRENOM, E.NUMETUDIANT, EV.LIBELLEEVAL, M.CODEMODULE

FROM ETUDIANT E

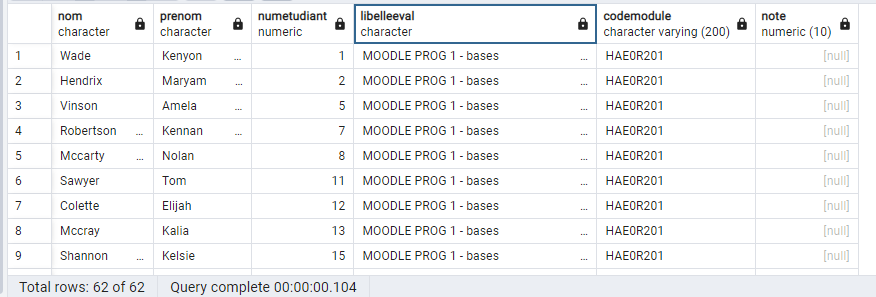
CROSS JOIN EVALUATION EV

JOIN MODULE M ON EV.NUMMODULE = M.NUMMODULE

LEFT JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT AND EV.NUMEVAL = NS.NUMEVAL

WHERE EV.LIBELLEEVAL = 'MOODLE PROG 1 - bases'

AND M.CODEMODULE = 'HAE0R201' AND NS.NOTE IS NULL;



2. Repérer les étudiants sans note pour des évaluations dans un module donné, en affichant les évaluations manquantes (avec le code et le libellé du module).

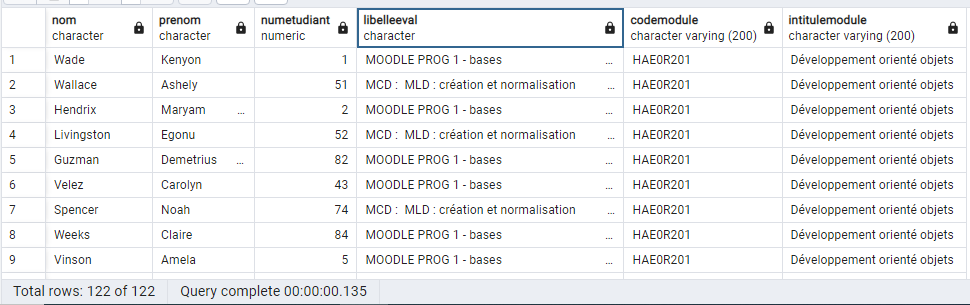
SELECT E.NOM,E.PRENOM,E.NUMETUDIANT,EV.LIBELLEEVAL,M.CODEMODULE,M.INTITULEMODULE FROM ETUDIANT E

CROSS JOIN EVALUATION EV

JOIN MODULE M ON EV.NUMMODULE = M.NUMMODULE

LEFT JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT AND EV.NUMEVAL = NS.NUMEVAL

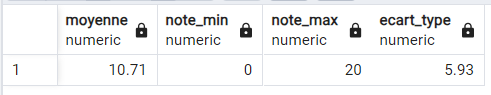
WHERE M.CODEMODULE = 'HAE0R201' AND NS.NOTE IS NULL;



3. Calculer la moyenne, la note minimale, la note maximale et l'écart-type pour une évaluation particulière.

SELECT ROUND(AVG(NOTE),2) AS MOYENNE, MIN(NOTE) AS NOTE\_MIN, MAX(NOTE) AS NOTE\_MAX, ROUND(STDDEV(NOTE),2) AS ECART\_TYPE

FROM NOTE\_SUR

WHERE NUMEVAL = '1'

4. Calculer la moyenne et l'écart-type pour un module donné.

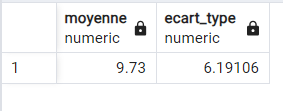
SELECT ROUND(AVG(NS.NOTE),2) AS Moyenne,ROUND(STDDEV(NS.NOTE),5) AS Ecart\_Type

FROM NOTE\_SUR NS

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

JOIN MODULE M ON EV.NUMMODULE = M.NUMMODULE

WHERE M.CODEMODULE = 'HAE0R201';



5. Trouver l'étudiant ayant obtenu la note la plus basse dans une évaluation spécifique.

SELECT E.NOM, E.PRENOM, E.NUMETUDIANT, EV.LIBELLEEVAL, NS.NOTE AS Note\_Minimale

FROM ETUDIANT E

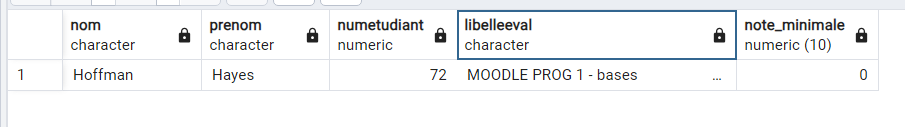
JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

WHERE EV.LIBELLEEVAL = 'MOODLE PROG 1 - bases'

ORDER BY NS.NOTE ASC

LIMIT 1;



6. Identifier l'étudiant ayant obtenu la moyenne la plus basse dans un module donné.

SELECT E.NOM, E.PRENOM, E.NUMETUDIANT,M.CODEMODULE,ROUND(AVG(NS.NOTE),0) AS Moyenne

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

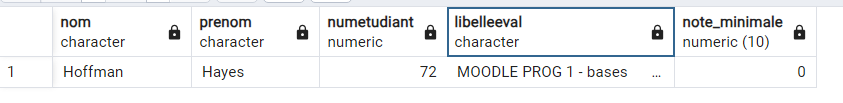
JOIN MODULE M ON EV.NUMMODULE = M.NUMMODULE

WHERE M.CODEMODULE = 'HAE0R201'

GROUP BY E.NOM, E.PRENOM, E.NUMETUDIANT, M.CODEMODULE

ORDER BY Moyenne ASC

LIMIT 1;



7. Trouver l'étudiant ayant obtenu la note la plus haute dans une évaluation spécifique.

SELECT E.NUMETUDIANT, E.NOM, E.PRENOM, EV.LIBELLEEVAL, NS.NOTE AS Note\_Maximale

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

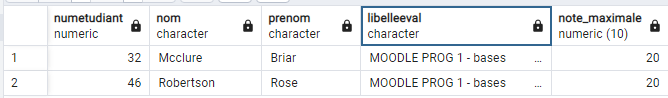
WHERE EV.LIBELLEEVAL = 'MOODLE PROG 1 - bases'

AND NS.NOTE =

(SELECT MAX(NOTE)

FROM NOTE\_SUR

WHERE EV.LIBELLEEVAL = 'MOODLE PROG 1 - bases' );



8. Identifier l'étudiant ayant obtenu la moyenne la plus haute dans un module donné.

SELECT E.NOM, E.PRENOM, E.NUMETUDIANT, M.CODEMODULE, ROUND(AVG(NS.NOTE),2) AS Moyenne

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

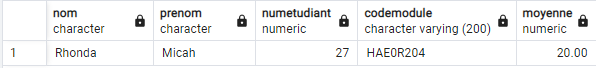
JOIN MODULE M ON EV.NUMMODULE = M.NUMMODULE

WHERE M.CODEMODULE = 'HAE0R204' AND NS.NOTE IS NOT NULL

GROUP BY E.NOM, E.PRENOM, E.NUMETUDIANT, M.CODEMODULE

ORDER BY Moyenne DESC

LIMIT 1;



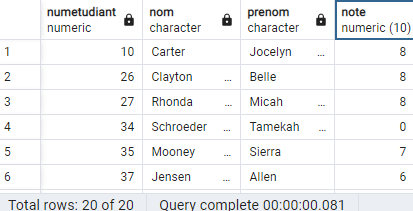
9. Liste des noms et numéros des étudiants ayant obtenu moins de 10 à une évaluation spécifique, pour planifier des soutiens.

SELECT E.NUMETUDIANT, E.NOM, E.PRENOM , NS.NOTE

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

WHERE NUMEVAL = '6' and NS.NOTE < 10



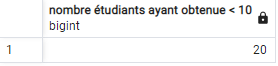
10. Déterminer le nombre d'étudiants ayant obtenu moins de 10 à une évaluation spécifique, pour planifier le nombre de participants aux soutiens.

SELECT COUNT(\*) "nombre étudiants ayant obtenue < 10"

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

WHERE NS.NUMEVAL ='6' AND NS.NOTE < 10



11. Liste des noms et numéros des étudiants ayant obtenu moins de 10 à une évaluation spécifique, avec une indication supplémentaire : '<8' ou 'Entre 8 et 10'.

SELECT E.NOM, E.NUMETUDIANT,

CASE

WHEN NS.NOTE < 8 THEN '<8'

WHEN NS.NOTE BETWEEN 8 AND 10 THEN 'Entre 8 et 10'

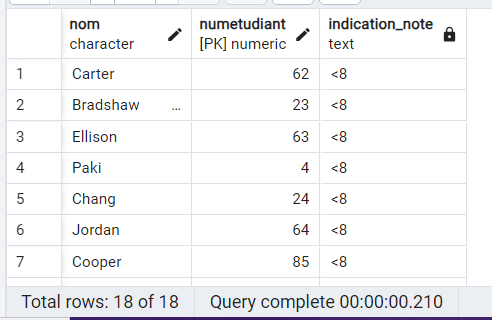
END AS Indication\_Note

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

WHERE EV.LIBELLEEVAL = 'MOODLE PROG 1 - bases' AND NS.NOTE < 10;



12. Identifier les étudiants n'ayant pas validé un module donné (moyenne du module < 10), en ajoutant une information supplémentaire : '<8' ou 'Entre 8 et 10'.

SELECT E.NOM, E.NUMETUDIANT, M.CODEMODULE,

CASE

WHEN AVG(NS.NOTE) < 8 THEN '<8'

WHEN AVG(NS.NOTE) BETWEEN 8 AND 10 THEN 'Entre 8 et 10'

END AS Indication\_Moy\_Module

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

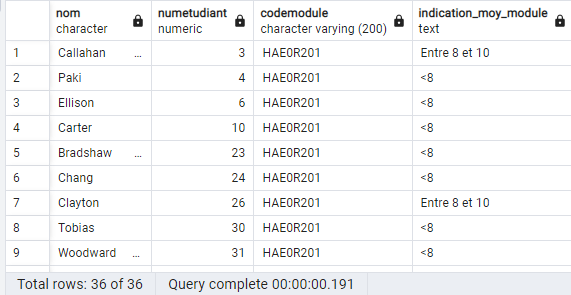
JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

JOIN MODULE M ON EV.NUMMODULE = M.NUMMODULE

WHERE M.CODEMODULE = 'HAE0R201'

GROUP BY E.NOM, E.NUMETUDIANT, M.CODEMODULE

HAVING AVG(NS.NOTE) < 10;



13. Déterminer le nombre d'étudiants n'ayant pas validé un module donné (code du module) en fonction des critères suivants : moyenne entre 8 et 10 et moyenne < 8.

SELECT Indication\_Moy\_Module, COUNT(\*) AS Nombre\_Etudiants

FROM (

SELECT

CASE

WHEN AVG(NS.NOTE) BETWEEN 8 AND 10 THEN 'Entre 8 et 10'

WHEN AVG(NS.NOTE) < 8 THEN '<8'

ELSE '>10' END AS Indication\_Moy\_Module,

E.NUMETUDIANT

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

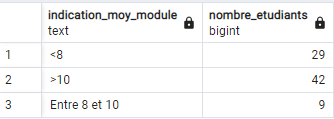
JOIN MODULE M ON EV.NUMMODULE = M.NUMMODULE

WHERE M.CODEMODULE = 'HAE0R201'

GROUP BY E.NUMETUDIANT

) AS Subquery

GROUP BY Indication\_Moy\_Module;



14. Établir le classement des étudiants pour une évaluation spécifique.

SELECT E.NUMETUDIANT, E.NOM, E.PRENOM,NS.NOTE

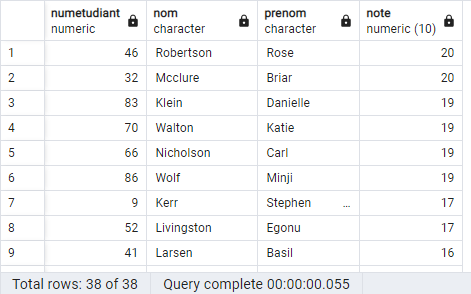
FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

WHERE NS.NUMEVAL = '1'

AND NS.NOTE IS NOT NULL

ORDER BY NS.NOTE desc



15. Établir le classement des étudiants dans un module donné.

SELECT E.NOM, E.PRENOM, E.NUMETUDIANT, M.CODEMODULE,ROUND( AVG(NS.NOTE),2 )AS Moyenne\_Module

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

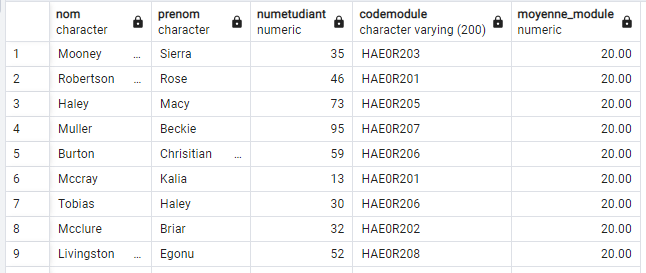
JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

JOIN MODULE M ON EV.NUMMODULE = M.NUMMODULE

WHERE NS.NOTE IS NOT NULL

GROUP BY E.NOM, E.PRENOM, E.NUMETUDIANT, M.CODEMODULE

ORDER BY Moyenne\_Module DESC;



La directrice des études souhaite savoir :

16. Le nombre d'évaluations par module. En fonction de ce nombre, afficher xxxun message indiquant s'il est considéré comme important :

- Si le nombre est = 1, afficher "Peu d'évaluations".

- S'il est compris entre 2 et 4, afficher "Nombre correct d'évaluations".

- Sinon, afficher "Beaucoup d'évaluations".

SELECT M.CODEMODULE, COUNT(\*) "Nombre évaluations",

CASE

WHEN COUNT(\*) = 1 THEN 'Peu d''évaluations'

WHEN COUNT(\*) BETWEEN 2 AND 4 THEN 'Nombre correct d''évaluations'

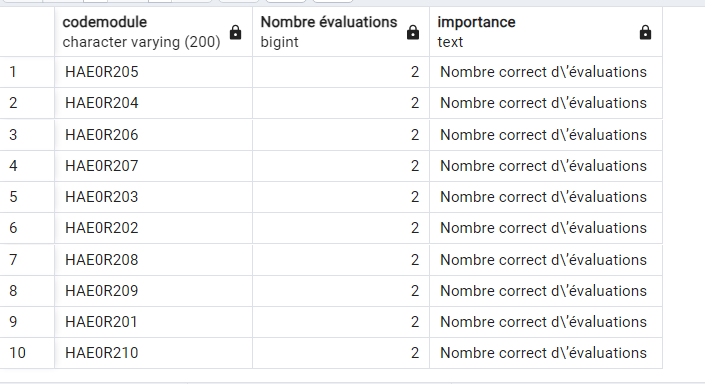
ELSE 'Beaucoup d''évaluations'

END AS Importance

FROM EVALUATION EV

JOIN MODULE M ON EV.NUMMODULE = M.NUMMODULE

GROUP BY M.CODEMODULE



17. Calculer la moyenne des étudiants par unité d'enseignement (UE), en prenant en compte les coefficients.

SELECT E.NUMETUDIANT, E.NOM, E.PRENOM, UE.CODEUE, UE.LIBELLEUE, ROUND(SUM(NS.NOTE \* ELA.COEFF) / SUM(ELA.COEFF),2) AS "MOYENNE UE"

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

JOIN EST\_LIE\_A ELA ON EV.NUMMODULE = ELA.NUMMODULE

JOIN UE ON ELA.NUMUE = UE.NUMUE

GROUP BY E.NUMETUDIANT, E.NOM, E.PRENOM, UE.CODEUE, UE.LIBELLEUE;



18. Identifier les noms des étudiants n'ayant pas validé une UE donnée (moyenne de l'UE < 10).

SELECT E.NOM, E.PRENOM, UE.CODEUE

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

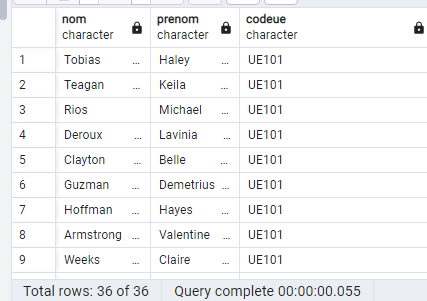
JOIN EST\_LIE\_A ELA ON EV.NUMMODULE = ELA.NUMMODULE

JOIN UE ON ELA.NUMUE = UE.NUMUE

WHERE UE.CODEUE ='UE101'

GROUP BY E.NUMETUDIANT, E.NOM, E.PRENOM , UE.CODEUE

HAVING AVG(NS.NOTE) < 10;



19. Déterminer le nombre d'étudiants n'ayant pas validé une UE donnée (moyenne de l'UE < 10).

SELECT COUNT(DISTINCT E.NUMETUDIANT) AS Nombre\_Etudiants\_Non\_Valides

FROM ETUDIANT E

WHERE E.NUMETUDIANT NOT IN (

SELECT E.NUMETUDIANT

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

JOIN EVALUATION EV ON NS.NUMEVAL = EV.NUMEVAL

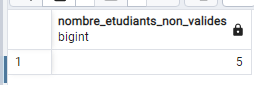
JOIN EST\_LIE\_A ELA ON EV.NUMMODULE = ELA.NUMMODULE

JOIN UE ON ELA.NUMUE = UE.NUMUE

GROUP BY E.NUMETUDIANT ,NS.NOTE, ELA.COEFF

HAVING (NS.NOTE \* ELA.COEFF)/SUM(ELA.COEFF)>= 10

);



20. Calculer la moyenne générale de chaque étudiant.

SELECT E.NUMETUDIANT, E.NOM, E.PRENOM, ROUND(AVG(NS.NOTE),2) AS MOYENNE\_GENERALE

FROM ETUDIANT E

JOIN NOTE\_SUR NS ON E.NUMETUDIANT = NS.NUMETUDIANT

GROUP BY E.NUMETUDIANT, E.NOM, E.PRENOM;

