

Révisions sur les bases de données

La base de données (simplifiée) d'un établissement scolaire contient :

- La table *eleves* donnant l'identifiant, le nom, le prénom et la classe de l'élève.
- La table *notes* donnant la valeur de la note, l'identifiant de l'élève l'ayant obtenue, la matière et la date d'obtention.
- La table *professeurs* donnant l'identifiant, le nom et le prénom du professeur.
- La table *enseigne* donnant l'identifiant d'un professeur le nom d'une classe dans laquelle il enseigne et la matière enseignée.

id	nom	prenom	classe
integer	text	text	text

La table eleves

valeur	id_e	matiere	date
decimal	integer	text	date

La table notes

id	nom	prenom
integer	integer	integer

La table professeurs

id_p	classe	matiere
integer	text	text

La table enseigne

Exercice 1. Décrire ce que donnent les requêtes suivantes :

QA : `SELECT * FROM eleves WHERE classe = 'Pcsi3' AND prenom = 'Jean'` (**sélection a priori**)

QB : `SELECT nom, prenom FROM eleves` (**projection**)

QC : `SELECT P.nom FROM professeurs AS P JOIN enseigne AS E ON P.id = E.id_p WHERE E.classe = 'Pcsi3'` (**jointure, renommage de table**)

QD : `SELECT COUNT(*) FROM professeurs` (**fonction d'agrégation**)
Rappeler les différentes **fonctions d'agrégation** et leur écriture en SQL

QE : `SELECT matiere, MIN(valeur) FROM notes GROUP BY matiere` (**fonction d'agrégation appliquée à des groupements**)
Combien de lignes contient cette table ?

QF : `SELECT id_p FROM enseigne WHERE matiere = 'mathématiques'`
`INTERSECT`
`SELECT id_p FROM enseigne WHERE matiere = 'informatique'` (**opération ensembliste**)
Pourquoi l'utilisation du connecteur logique AND n'est pas pertinente pour cette question ?
Rappeler les différentes **opérations ensemblistes** et leur écriture en SQL.

QG : `SELECT matiere, COUNT(*) AS NB FROM notes GROUP BY matiere HAVING NB >= 1000` (**sélection a posteriori, renommage d'attribut**)

QH : `SELECT matiere, COUNT(*) AS NB FROM notes WHERE valeur = 20 GROUP BY matiere HAVING NB >= 10` (**sélection a priori et a posteriori**)
Notez l'ordre dans lequel apparaissent WHERE, GROUP BY et HAVING.

QI : `SELECT MAX (NB) FROM (SELECT COUNT(*) AS NB FROM eleves GROUP BY nom)` (**requêtes imbriquées**)

Exercice 2. *Écrire une requête SQL dont le résultat donne...*

Q1 : le prénom des élèves dont le nom de famille est Germain ou Martin.

Q2 : le nom et prénom des élèves de la Pcsi3.

Q3 : l'effectif total de l'établissement (nombre total d'élèves).

Q4 : le nom et prénom des élèves ayant une note strictement inférieure à 3.

Q5 : le nom et prénom des élèves ayant une note strictement inférieure à 3 et une note strictement supérieure à 18.

Q6 : le nom et l'effectif des classes.

Exercice 3. *Écrire une requête SQL dont le résultat donne...*

Q7 : le nom du professeur de physique de Jean Germain.

Q8 : le nombre de professeurs de Jean Germain.

Q9 : le nom, prénom des élèves et leur moyenne de physique.

Q10 : la plus haute moyenne de physique d'un élève de l'établissement. (Quelle table faut-il interroger pour obtenir cette réponse ?)

Pourquoi `SELECT MAX(AVG(valeur))...` ne peut pas être une requête correcte ?

La question 10 illustre le fait qu'**avant d'utiliser une fonction d'agrégation, il faut souvent construire une table dans laquelle un nouvel attribut est créé auquel on applique ensuite la fonction d'agrégation.**

Exercice 4. *Écrire une requête SQL dont le résultat donne...*

Q11 : le nom des élèves ayant une moyenne de physique supérieure ou égale à 10. (Quelle table faut-il interroger pour obtenir cette réponse ?)

On écrira une version avec `WHERE` (**sélection a priori**) et une version plus simple utilisant une **sélection a posteriori** (`HAVING`)

Q12 : le nom des élèves ayant une moyenne de physique supérieure ou égale à 10 sans avoir obtenu une note de physique inférieure ou égale à 5.

Q13 : le nom et prénom des élèves ayant eu la plus haute note de physique de l'établissement.

La question 13 est un exemple de recherche des **"points" où le maximum est atteint.**

Exercice 5. *Écrire une requête SQL dont le résultat donne...*

Q14 : le nom et l'effectif des classes.

Q15 : l'effectif maximum d'une classe.

Q16 : le nom de la ou les classes d'effectif maximum. (**point où le maximum par groupe est atteint**)

Exercice 6. *Écrire une requête SQL dont le résultat donne...*

Q17 : le nombre de matières enseignées dans une classe et le nom de cette classe.

Q18 : le nom des professeurs ayant donné (au moins) une note de valeur 0.

Q19 : le nom des professeurs ayant donné une note de valeur 0 et aucune note de valeur 20.

Exercice 7. *Écrire une requête SQL dont le résultat donne...*

Q20 : les noms et prénoms des élèves ayant un professeur de même nom qu'eux.

Q21 : le nom des professeurs enseignant deux matières différentes.

Corrigé

- Q1 : Le prénom des élèves dont le nom de famille est Germain ou Martin.
SELECT prenom FROM eleves WHERE nom = 'Germain' OR nom = 'Martin'
ou
SELECT prenom FROM eleves WHERE nom = 'Germain'
UNION
SELECT prenom FROM eleves WHERE nom = 'Martin'
- Q2 : Le nom et prénom des élèves de la Pcsi3.
SELECT nom, prenom FROM eleves WHERE classe = 'Pcsi3'
- Q3 : L'effectif total de l'établissement
SELECT COUNT(*) FROM eleves
- Q4 : Le nom et prénom des élèves ayant une note inférieure à 3.
SELECT DISTINCT nom, prenom FROM eleves JOIN notes ON id = id_e
WHERE valeur < 3
- Q5 : Le nom et prénom des élèves ayant une note inférieure à 3 et une note supérieure à 18.
SELECT DISTINCT nom, prenom FROM eleves JOIN notes ON id = id_e
WHERE valeur < 3
INTERSECT
SELECT DISTINCT nom, prenom FROM eleves JOIN notes ON id = id_e
WHERE valeur > 18
- Q6 : Le nom et l'effectif des classes
SELECT classe, COUNT(*) FROM eleves GROUP BY classe
- Q7 : Le nom du professeur de physique de Jean Germain.
SELECT prof.nom FROM professeurs AS prof
JOIN enseigne ON id = id_p
JOIN eleves ON enseigne.classe = eleves.classe
WHERE matiere = 'physique' AND eleves.nom = 'Germain' AND eleves.prenom = 'Jean'
ou
SELECT nom FROM professeurs JOIN enseigne ON id = id_p
WHERE matiere='physique'
AND classe = (SELECT classe FROM eleves WHERE nom = 'Germain' AND prenom = 'Jean')
- Q8 : Le nombre de professeurs de Jean Germain.
SELECT COUNT(*) FROM eleves AS E JOIN enseigne AS EN
ON E.classe = EN.classe
WHERE E.prenom = 'Jean' AND E.nom = 'Germain'
ou
SELECT COUNT(id) FROM enseigne
WHERE classe = (SELECT classe FROM eleves WHERE nom = 'Germain' AND prenom = 'Jean')
- Q9 : Le nom, prénom des élèves et leur moyenne de physique.
SELECT nom, prenom, AVG(valeur) FROM eleves JOIN notes ON id = id_e
WHERE matiere = 'Physique' GROUP BY id_e (ou id)
- Q10 : La plus haute moyenne de physique d'un élève de l'établissement.
SELECT MAX(moyenne) FROM
(SELECT AVG(valeur) AS moyenne FROM notes
WHERE matiere = 'Physique' GROUP BY id_e)

- Q11 : Le nom des élèves ayant une moyenne de physique supérieure ou égale à 10.
 Sélection a priori (WHERE) :
SELECT nom FROM
(SELECT nom, AVG(valeur) AS moyenne
FROM eleves JOIN notes ON id = id_e
WHERE matiere = 'Physique' GROUP BY id)
WHERE moyenne >= 10
- Sélection a posteriori (HAVING) :
SELECT nom FROM eleves JOIN notes ON id = id_e
WHERE matiere = 'Physique' GROUP BY id
HAVING AVG(valeur) >= 10
- Q12 : Le nom des élèves ayant une moyenne de physique supérieure ou égale à 10 sans avoir obtenu une note de physique inférieure ou égale à 5.
SELECT nom FROM eleves JOIN notes ON id = id_e
WHERE matiere = 'Physique' GROUP BY id HAVING AVG(valeur) >= 10
EXCEPT
SELECT nom FROM eleves JOIN notes ON id = id_e
WHERE matiere = 'Physique' AND valeur <= 5.
- Q13 : Le nom et prénom des élèves ayant eu la plus haute note de physique de l'établissement.
SELECT nom FROM eleves JOIN notes ON id = id_e
WHERE matiere = 'Physique'
AND valeur = (SELECT MAX(valeur) FROM notes
WHERE matiere = 'Physique')
- Q14 : Le nom et l'effectif des classes
SELECT classe,COUNT(id) FROM eleves GROUP BY classe
- Q15 : L'effectif maximum d'une classe.
SELECT MAX(effectif) FROM
(SELECT classe, COUNT(id) AS effectif FROM eleves GROUP BY classe)
- Q16 : Le nom de la ou les classes d'effectif maximum.
SELECT classe FROM
(SELECT classe, COUNT(id) AS effectif FROM eleves GROUP BY classe
HAVING effectif =
(SELECT MAX(effectif) FROM (SELECT classe, COUNT(id) AS effectif
FROM eleves GROUP BY classe)))
- Q17 : Le nombre de matières enseignées dans une classe et le nom de cette classe.
SELECT COUNT(*), classe FROM enseigne GROUP BY classe
- Q18 : Le nom des professeurs ayant donné (au moins) une note de valeur 0.
 Jointures via la classe :
SELECT P.nom FROM professeurs AS P
JOIN enseigne AS EN ON P.id = id_p
JOIN eleves AS E ON E.classe = EN.classe
JOIN notes AS N ON N.id_e = E.id WHERE valeur = 0
- ou jointures via la matière :
SELECT nom FROM professeurs
JOIN enseigne AS EN ON id = id_p
JOIN notes AS N ON N.matiere = EN.matiere WHERE valeur=0

Q19 : Le nom des professeurs ayant donné une note de valeur 0 et aucune note de valeur 20.

Jointures via la classe :

```
SELECT P.nom FROM professeurs AS P
JOIN enseigne AS EN ON P.id = id_p
JOIN eleves AS E ON E.classe = EN.classe
JOIN notes AS N ON N.id_e = E.id WHERE valeur = 0
EXCEPT
SELECT P.nom FROM Professeur AS P
JOIN enseigne AS EN ON P.id = id_p
JOIN eleves AS E ON E.classe = EN.classe
JOIN notes AS N ON N.id_e = E.id WHERE valeur = 20
```

ou jointures via la matière :

```
SELECT nom FROM professeurs
JOIN enseigne AS EN ON id = id_p
JOIN notes AS N ON N.matiere = EN.matiere WHERE valeur = 0
EXCEPT
SELECT nom FROM professeurs
JOIN enseigne AS EN ON id = id_p
JOIN notes AS N ON N.matiere = EN.matiere WHERE valeur=20
```

Q20 : Les noms et prénoms des élèves ayant un professeur de même nom qu'eux.

```
SELECT E.nom, prenom FROM eleves AS E
JOIN enseigne AS EN ON E.classe = EN.classe
JOIN professeurs AS P ON P.id = id_p WHERE E.nom = P.nom
```

Q21 : Le nom des professeurs enseignant deux matières différentes.

```
SELECT DISTINCT nom FROM professeurs
JOIN enseigne AS E1 ON id = E1.id_p
JOIN enseigne AS E2 ON E1.id_p = E2.id_p
WHERE E1.matiere <> E2.matiere
```