TD — Interroger une base de données en SQL

Informatique tronc commun – **Corrigé**

La base de données

Une classe préparatoire enregistre les notes de ses élèves dans une base de données dont voici le **schéma relationnel** :

```
Eleve(id : INTEGER, nom : TEXT, prenom : TEXT, classe : TEXT)
Matiere(id : INTEGER, intitule : TEXT, coef : INTEGER)
Devoir(id : INTEGER, id_matiere : INTEGER, semaine : INTEGER, duree : INTEGER)
Note(id_eleve : INTEGER, id_devoir : INTEGER, valeur : REAL)
```

— Chaque attribut est suivi de son domaine (types SQLite) : `INTEGER` (entier), `REAL` (flottant), `TEXT` (chaîne de caractères).

— Les attributs **soulignés** forment la clé primaire de chaque table ; les attributs en **orange** sont des clés étrangères.

— `Devoir.id_matiere` fait référence à `Matiere.id` ; `Note.id_eleve` fait référence à `Eleve.id` ; `Note.id_devoir` fait référence à `Devoir.id`.

— La clé primaire de `Note` est le *couple* `(id_eleve, id_devoir)` : un élève a au plus une note par devoir.

— `semaine` est le numéro de la semaine de l'année où le devoir a eu lieu ; `duree` est la durée du devoir en heures ; `valeur` est une note sur 20 ; `classe` vaut `'MP'` ou `'PC'`.

Extraits des tables (les tables réelles contiennent beaucoup plus de lignes) :

Eleve

id	nom	prenom	classe
1	Dupont	Léa	MP
2	Martin	Hugo	MP
3	Bernard	Chloé	PC
4	Petit	Nathan	PC
5	Durand	Sami	MP
6	Martin	Inès	PC
	...		

Matiere

id	intitule	coef
1	Maths	3
2	Physique	2
3	Info	1
	...	

Devoir

id	id_matiere	semaine	duree
1	1	38	4
2	2	39	2
3	3	40	2
4	1	41	4
5	2	42	3
	...		

Note

id_eleve	id_devoir	valeur
1	1	14.5
2	1	17
3	1	11
4	1	8
1	2	12
3	2	15.5
5	4	13
	...	

Questions

Écrire une requête SQL répondant à chacune des questions suivantes. Les chaînes de caractères s'écrivent entre apostrophes : `'MP'`.

Niveau 1 — Une seule table

Q1. Donner l'intitulé et le coefficient de chaque matière.

Q2. Donner le nom et le prénom des élèves de la classe de MP.

Q3. Donner la liste des classes représentées dans la base, chaque classe n'apparaissant qu'une fois.

Q4. Donner l'identifiant des devoirs durant au moins 3 heures, du plus récent au plus ancien.

Q5. Donner les trois meilleures notes obtenues au devoir d'identifiant 4.

Q6. Donner les notes classées de la 4^{e} à la 6^{e} place à ce même devoir.

Q7. Donner l'identifiant des élèves qui ont une note au devoir 1 mais pas de note au devoir 2.

Niveau 2 — Jointures

Q8. Donner le nom, le prénom et la note de chaque élève ayant composé au devoir 3.

Q9. Pour chaque note obtenue dans un devoir de maths, donner le nom de l'élève, la semaine du devoir et la note.

Niveau 3 — Agrégats

Q10. Combien d'élèves compte la base ? Combien d'élèves sont en PC ?

Q11. Donner la moyenne, la note minimale et la note maximale du devoir 1.

Q12. Donner l'effectif de chaque classe, du plus grand effectif au plus petit.

Q13. Donner le nom, le prénom et la moyenne en physique des élèves dont la moyenne en physique est strictement inférieure à 10.

Q14. Donner le nom et le prénom du ou des élèves ayant obtenu la meilleure note au devoir 1. *(On pourra utiliser une requête imbriquée.)*

Niveau 4 — Autojointures

Q15. a) Donner les paires d'élèves distincts portant le même nom, chaque paire n'apparaissant qu'une fois.

b) Donner l'identifiant des devoirs où au moins un élève a obtenu une note strictement supérieure à celle de Léa Dupont, avec le nombre de ces élèves.

Corrigé

Plusieurs requêtes correctes sont souvent possibles ; on donne ici une solution.

Q1.

```sql
SELECT intitule, coef FROM Matiere;
```

Q2.

```sql
SELECT nom, prenom FROM Eleve WHERE classe = 'MP';
```

Q3.

```sql
SELECT DISTINCT classe FROM Eleve;
```

Q4.

```sql
SELECT id FROM Devoir WHERE duree >= 3 ORDER BY semaine DESC;
```

Q5.

```sql
SELECT valeur FROM Note WHERE id_devoir = 4 ORDER BY valeur DESC LIMIT 3;
```

Q6.

```sql
SELECT valeur FROM Note WHERE id_devoir = 4 ORDER BY valeur DESC LIMIT 3 OFFSET 3;
```

Q7. Les deux requêtes doivent renvoyer les mêmes colonnes.

```sql
SELECT id_eleve FROM Note WHERE id_devoir = 1
EXCEPT
SELECT id_eleve FROM Note WHERE id_devoir = 2;
```

Q8.

```sql
SELECT e.nom, e.prenom, n.valeur
FROM Eleve AS e JOIN Note AS n ON e.id = n.id_eleve
WHERE n.id_devoir = 3;
```

Q9. Il faut traverser les quatre tables.

```sql
SELECT e.nom, d.semaine, n.valeur
FROM Note AS n JOIN Eleve   AS e ON n.id_eleve   = e.id
               JOIN Devoir  AS d ON n.id_devoir  = d.id
               JOIN Matiere AS m ON d.id_matiere = m.id
WHERE m.intitule = 'Maths';
```

Q10.

```sql
SELECT COUNT(*) FROM Eleve;
SELECT COUNT(*) FROM Eleve WHERE classe = 'PC';
```

Q11.

```sql
SELECT AVG(valeur), MIN(valeur), MAX(valeur) FROM Note WHERE id_devoir = 1;
```

Q12.

```sql
SELECT classe, COUNT(*) AS effectif
FROM Eleve
GROUP BY classe
ORDER BY effectif DESC;
```

Q13. Le `WHERE` filtre les notes (lignes), le `HAVING` filtre les élèves (groupes). On groupe par `e.id` pour ne pas confondre deux homonymes.

```sql
SELECT e.nom, e.prenom, AVG(n.valeur) AS moy
FROM Eleve AS e JOIN Note    AS n ON e.id = n.id_eleve
                JOIN Devoir  AS d ON n.id_devoir  = d.id
```

```
                JOIN Matiere AS m ON d.id_matiere = m.id
WHERE m.intitule = 'Physique'
GROUP BY e.id, e.nom, e.prenom
HAVING AVG(n.valeur) < 10;
```

Q14. La sous-requête renvoie une seule valeur ; en cas d'ex æquo, tous les élèves concernés apparaissent (ce que `ORDER BY ... LIMIT 1` ne garantirait pas).

```
SELECT e.nom, e.prenom
FROM Eleve AS e JOIN Note AS n ON e.id = n.id_eleve
WHERE n.id_devoir = 1
  AND n.valeur = (SELECT MAX(valeur) FROM Note WHERE id_devoir = 1);
```

Q15 a). La condition `e1.id < e2.id` élimine les paires d'un élève avec lui-même et évite d'avoir chaque paire deux fois.

```
SELECT e1.nom, e1.prenom, e2.prenom
FROM Eleve AS e1 JOIN Eleve AS e2 ON e1.nom = e2.nom
WHERE e1.id < e2.id;
```

Q15 b). `n1` est la note de Léa Dupont, `n2` une note d'un autre élève au même devoir.

```
SELECT n1.id_devoir, COUNT(*) AS nb_meilleurs
FROM Note AS n1 JOIN Eleve AS e  ON n1.id_eleve  = e.id
                JOIN Note  AS n2 ON n1.id_devoir = n2.id_devoir
WHERE e.nom = 'Dupont' AND e.prenom = 'Léa' AND n2.valeur > n1.valeur
GROUP BY n1.id_devoir;
```