

CHAPITRE 3. Les Bases de Données Relationnelles

Correction Devoir de maison 1

Base de données universitaire

Une université possède la base de données (simplifiée) relatives à ses étudiants suivantes :

ETUDIANTS(ne, nom, prenom, id_formation, date_naissance, adresse, salarie)

FORMATION(id_formation, libelle_f, type)

MODULE(id_module, id_formation, libelle_m)

RESULTAT(ne, id_module, note, annee)

Indications sur les attributs :

ne : numéro étudiant, type : INT

nom : nom de l'étudiant, type : VARCHAR

prenom : prenom de l'étudiant, type : VARCHAR

date_naissance : date de naissance de l'étudiant, type : DATE

adresse : adresse de l'étudiant, type : VARCHAR

id_module : référence du module, type : INT

annee : annee d'obtention du module, type : INT

id_formation : référence de la formation, type : INT

libelle_m : libellé du module , type : VARCHAR

libelle_f : libellé de la formation , type : VARCHAR

type : type de formation , type : VARCHAR

salarie : indique si l'étudiant est salarié (=1) ou non (=0), type : INT

note : note de l'étudiant à l'examen relatif au module, type : INT

Autres Indications :

- Pour simplifier, tous les modules d'une formation ont même coefficient.
- La référence d'une formation est unique.
- Les types de formation sont « L1 », « L2 », « L3 », « M1 », « M2 ».
- Une formation dure un semestre.
- Une formation comprend plusieurs modules.
- Pour simplifier, un module ne peut pas être partagé par plusieurs formations.
- La formation est validée si la moyenne des modules relatifs à cette formation est supérieure ou égal à 10.
- Un module est validée si la note obtenu à l'examen est supérieur ou égale à 10, même si la formation dans son ensemble n'est pas validée.
- On peut donc valider sa formation sans valider chacun des modules.
- Seuls les modules non validés sont à repasser l'année suivante.
- Les notes sont entières et comprises entre 0 et 20.
- L'adresse d'un étudiant est donnée uniquement par le nom de sa ville de résidence.
- Pour simplifier, on suppose que chaque étudiant ne suit qu'une seule formation par année.

T.A.F :

Les requêtes seront exprimés en langage SQL.

1. Repérer et donner une clé primaire pour chaque relation de la base de données.

Corrigé : Une clé primaire est soulignée

ETUDIANTS(ne, nom, prenom, id_formation, date_naissance, adresse, salarie)
FORMATION(id_formation, libelle_f, type)
MODULE(id_module, id_formation, libelle_m)
RESULTAT(ne, id_module, note, annee)

2. Repérer et donner la ou les clés étrangères pour chaque relation de la base de données.

Corrigé : Une clé étrangère est repérée par : '#'

ETUDIANTS(ne, nom, prenom, # id_formation, date_naissance, adresse, salarie)
FORMATION(id_formation, libelle_f, type)
MODULE(id_module, # id_formation, libelle_m)
RESULTAT(# ne, # id_module, note, annee)

3. Requête affichant les noms et prénoms des étudiants salariés.

Corrigé :

```
1 SELECT nom, prenom
2 FROM etudiants
3 WHERE salaire = 1 ;
```

4. Requête affichant la note de chacun des étudiants ayant suivi le module 1402 en 2016. Les étudiants seront identifiés par leurs numéros d'étudiant.

Corrigé :

```
1 SELECT etudiants.ne, note
2 FROM etudiants, resultat
3 WHERE etudiants.ne = resultat.ne
4 AND resultat.id_module = 1402
5 AND resultat.annee = 2016 ;
```

5. Requête affichant les numéros étudiant, les noms et prénoms des étudiants de L1.

Corrigé :

```
1 SELECT ne, nom, prenom
2 FROM etudiants, formation
3 WHERE etudiants.id_formation = formation.id_formation
4 AND formation.type = 'L1' ;
```

6. Requête affichant l'identifiant et le libellé de tous les modules de licence (« L1 », « L2 », « L3 »).

Corrigé :

```
1 SELECT id_module, libelle_m
2 FROM module, formation
3 WHERE module.id_formation = formation.id_formation
4 AND formation.type in('L1', 'L2', 'L3');
```

```

5
6  /* ou bien :
7  AND (formation.type = 'L1'
8  OR formation.type = 'L2'
9  OR formation.type = 'L3')
10 */

```

7. Requête affichant les noms et prénoms des étudiants inscrits en M1 et ayant suivi le module « Informatique ».

Corrigé :

```

1  SELECT  E.nom, E.prenom
2  FROM    etudiants as E, formation as F, module as M
3  WHERE   E.id_formation= F.id_formation
4  AND     F.id_formation = M.id_formation
5  AND     F.type = 'M1'
6  AND     M.libelle_m = 'Informatique' ;
7
8  /* ou bien */
9
10 SELECT  E.nom, E.prenom
11 FROM    etudiants e
12 JOIN    formation f ON f.id_formation==e.id_formation
13 JOIN    module m ON m.id_formation=f.id_formation
14 WHERE   f.type="M1" AND m.libelle_m="Informatique"

```

8. Requête affichant l'identifiant du module 4526 ainsi que les notes moyenne, minimale et maximale obtenues dans ce module en 2016.

Corrigé :

```

1  SELECT  id_module ,
2          AVG(note) as note moyenne ,
3          MIN(note) as note minimale ,
4          MAX(note) as note maximale
5  FROM    resultat
6  WHERE   id_module = 4526
7  AND     annee = 2016 ;

```

9. Requête affichant le nombre d'étudiants salariés suivant la formation dont le libellé est « Physique de rêve ».

Corrigé :

```

1  SELECT  COUNT(*) as 'nombre_etudiants_salaries'
2  FROM    etudiants, formation
3  WHERE   etudiants.id_formation = formation.id_formation
4  AND     salaire = 1
5  AND     libelle_f = 'Physique_de_reve';

```

10. Requête affichant le libellé et le nombre d'étudiants dans chaque formation.

Corrigé :

```

1  SELECT  libelle_f, COUNT(*) as 'nombre_etudiants'

```

```

2 FROM      etudiants , formation
3 WHERE      etudiants.id_formation = formation.id_formation
4 GROUP BY  libelle_f ;

```

11. Requête affichant le numéro d'étudiant, le nom, le prénom et la moyenne générale de Manel sayadi en 2014.

Corrigé :

```

1 SELECT  E.ne, E.nom, E.prenom, AVG(note) AS 'Moyenne_generale'
2 FROM      etudiants as E, resultat as R
3 WHERE      E.ne = R.ne
4 AND        annee = 2004
5 AND        E.prenom = 'Manel'
6 AND        E.nom = 'Sayadi' ;

```

12. Requête affichant la liste (numéro d'étudiants, noms, prénoms) ainsi que les moyennes générales de tous les étudiants ayant validé leur année en 2015, quelque soit la formation.

Corrigé :

```

1 SELECT  E.ne, E.nom, E.prenom, AVG(R.note) AS 'Moyenne_generale'
2 FROM      etudiants as E, resultat as R
3 WHERE      E.ne = R.ne
4 AND        R.annee = 2015
5 GROUP BY  E.ne, E.nom, E.prenom
6 HAVING     AVG(R.note) >= 10 ;

```

Pour plus d'informations :

GROUP BY: https://www.techonthenet.com/sql/group_by.php

sql: www.sql.sh