

Interrogation de Bases de données relationnelles
21 Novembre (Durée : 30')
Sans documents

1.1		2
1.2		3
1.3		4
1.4		3
2		8

1 Gestion de notes d'étudiants

Une base de données de gestion de notes d'étudiants comporte les tables suivantes:

etudiant (numetu, nom, prenom, datenaiss, adresse)

matiere (codemat, libelle, coeff)

epreuve (numepreuve, datepreuve, coeff, codemat)

notation (numetu, numepreuve, note)

La table *etudiant* liste tous les etudiants . *numetu* est la clé primaire de la table *etudiant*, c'est un numero d'identification. Les autres colonnes décrivent les nom, prénom, date de naissance et adresse de l'étudiant. Attention, il peut y avoir plusieurs étudiants de même nom, mais ils ont des numéros différents.

La table *matiere* décrit les matières enseignées. *codemat* est la clé primaire de la table *matiere*, c'est un code d'identification. Les autres colonnes décrivent le libellé et le coefficient de la matière.

La table *epreuve* décrit les épreuves passées ou à venir. *numepreuve* est la clé primaire de la table *epreuve*, c'est un numero d'identification. Les autres colonnes décrivent la date de l'épreuve, son coefficient et le code de la matière sur laquelle porte l'épreuve.

La table *notation* décrit les notes obtenues par les étudiants aux épreuves qu'ils ont passées. Le présence du tuple (*numetu,numepreuve,note*) dans la table signifie que l'étudiant d'identifiant *numetu* a obtenu la note *note* à l'épreuve d'identifiant *numepreuve*.

1. Ecrire une requête qui renvoie le nom de chaque étudiant avec le nombre d'épreuves qu'il a passé?

```
select nom , count(numepreuve) from etudiant , notation
where notation.numetu=etudiant.unmetu
group by etudiant.numetu, nom
```

2. Ecrire une requête qui renvoie le nom de chaque étudiant avec le nombre d'épreuves qu'il a passé avec succès (c'est à dire avec une note supérieure ou égale à 12)?

```
select nom , count(numepreuve) from etudiant , notation
where notation.numetu=etudiant.unmetu and note >=12
group by numetu, nom
```

3. Ecrire une requête qui renvoie le nom du ou des étudiants ayant passé le plus grand nombre d'épreuves avec succès (c'est à dire avec une note supérieure ou égale à 12)?

```
select nom from etudiant where numetu in
(select etudiant.numetu
 from notation , etudiant
 where notation.numetu=etudiant.numetu and note >=12
 group by etudiant.numetu,nom
 having count (numepreuve) >= all (select count(numepreuve) from notation
                                where note >=12
                                group by notation.numetu))
```

4. Ecrire une requête qui renvoie le libellé de chaque matière avec la date de la dernière épreuve pour cette matière. Si aucune épreuve n'est prévue pour une matière, celle ci doit tout de même apparaître, sans date.

```
select libelle, max(dateepreuve)
 from matiere full join epreuve on (matiere.codemat=epreuve.codemat)
 group by libelle
```

2 Gestion de fichiers unix

Les informations concernant une hiérarchie de fichiers unix sont stockées dans une table fichiers (*id*, *name*, *parent_id*) , où *id* est un identifiant du fichier ou du répertoire, *name* son nom et *parent_id* l'identifiant du repertoire parent s'il existe. Le tuple correspondant au répertoire racine est (1,"*null*")

Ecrire une requête dont le résultat est l'ensemble des tuples (*id,name,path*) où *id* est l'identifiant du fichier ou du répertoire, *name* son nom et *path* son chemin d'accès depuis la racine.

Par exemple si la table fichiers contient

id	name	parent_id
1	"	null
2	rep1	1
3	rep2	2
4	fich1	3

Le résultat de la requête devra être

id	name	path
1	"	/
2	rep1	/rep1
3	rep2	/rep1/rep2
4	fich1	/rep1/rep2/fich1

On rappelle qu'en SQL la concaténation se note ||

```
CREATE TABLE fichiers(  
    id SERIAL PRIMARY KEY,  
    name VARCHAR,  
    parent_id INTEGER REFERENCES fichiers(id)  
);
```

```
WITH RECURSIVE path(name, path, id, parent_id) AS (  
    SELECT name, '', id, parent_id FROM fichiers WHERE id = 1  
    UNION  
    SELECT  
        fichiers.name,  
        parentpath.path || '/' || fichiers.name,  
        fichiers.id, fichiers.parent_id  
    FROM fichiers, path as parentpath  
    WHERE fichiers.parent_id = parentpath.id)  
SELECT id, name, path FROM path;
```
