

« Bases de données »
STEVE KREMER

Travaux pratiques n° 4 : SQL Solutions

Remarques préliminaires

Les solutions proposées dans ce document représentent *une solution possible*. D'autres reformulations des requêtes sont bien sûr envisageables.

Les solutions ont été implantées et testées utilisant la base de données POSTGRESQL.

Création des tables

```
create table employé (mat int not null, nom char(20), prénom char(20),  
primary key (mat));
```

```
create table local (num int not null, primary key (num));
```

```
create table bureau (numlocal int, numtel int not null,  
primary key (numlocal), unique (numtel), foreign key (numlocal)  
references local(num));
```

```
create table auditoire (numlocal int, nbplace int,  
primary key (numlocal), foreign key (numlocal) references local(num));
```

```
create table secrétaire (matempl int, numlocal int,  
primary key (matempl), foreign key (matempl) references employé(mat),  
foreign key (numlocal) references local(num));
```

```
create table professeur (matempl int, numlocal int,  
primary key (matempl), unique (numlocal),  
foreign key (matempl) references employé(mat),  
foreign key (numlocal) references local(num));
```

```
create table cours (intitulé char(20), code int, nbheures int,  
primary key (code));
```

```
create table donne (matempl int, codecours int,
```

```
primary key (matempl,codecours),
foreign key (matempl) references professeur(matempl),
foreign key (codecours) references cours(code));
```

```
create table année (nom char(40), code int,
primary key (code));
```

```
create table programme (codecours int,
codeannée int,
numlocal int not null,
primary key (codecours,codeannée),
foreign key (codecours) references cours(code),
foreign key (codeannée) references année(code));
```

```
create table étudiant (mat int, nom char(20) not null,
prénom char(20) not null, codeannée int not null,
primary key (mat),
foreign key (codeannée) references année(code));
```

Remarquons que les attributs appartenant à une clef primaire, sont déclarés implicitement comme étant `not null`, contrairement aux attributs uniques, ou appartenant à des clefs étrangères.

Requêtes

```
select intitulé, nbheures from cours,programme,étudiant where
cours.code=programme.codecours and
programme.codeannée=étudiant.codeannée and
étudiant.nom='Gastier' and étudiant.prénom='Leroy';
```

```
select mat, nom, prénom from cours,programme,étudiant where
cours.code=programme.codecours and
programme.codeannée=étudiant.codeannée and
cours.intitulé='Théorie de la poutre';
```

```
select count(*) as nb_étudiants from étudiant;
```

```
select professeur.matempl, employé.nom, employé.prénom, count(*) from
professeur,employé,donne where professeur.matempl=employé.mat
and donne.matempl=employé.mat group by
professeur.matempl, employé.nom, employé.prénom;
```

```
select E.mat, E.nom from employé E, professeur P where E.mat=P.matempl
and not exists (select * from donne D1, donne D where E.mat=D.matempl
and D1.codecours=D.codecours and D1.matempl<>E.mat);
```

```
select T.numlocal, count(*) from (select * from secrétaire S union
```

```
select * from professeur P) as T group by T.numlocal order by  
T.numlocal;
```

```
select année.nom, année.code from année where année.code in  
(select codeannée from étudiant group by codeannée having count(mat)>5);
```

```
select sum(nbplace) from auditoire;
```

```
select distinct année.code as code_année, cours.code as code_cours  
from année, programme, auditoire, étudiant, cours,  
(select codeannée, count(distinct mat) as nbr from étudiant  
group by codeannée) as étudiant2 where  
année.code=programme.codeannée and  
programme.numlocal=auditoire.numlocal and  
cours.code=programme.codecours and  
auditoire.nbplace<étudiant2.nbr and  
année.code=étudiant.codeannée;
```