

## SAE 2.04 - Exploitation d'une Base de Données

### Partie 2 et 3

#### Base de donnée Promotion

Afin de mieux visualiser les vues et procédures définies, nous avons créé un ensemble de données fictif.

```
bd_user1=> SELECT * from Etudiant;
```

etudiant_id	nom	prenom	nomgroupe
0	Dikhamidze	Giorgi	Cygnus
1	Dujardin	Jean	Draco
2	Norris	Chuck	Cygnus
3	Booba	Kaaris	Phoenix
4	Benzema	Karim	Cygnus

(5 lignes)

```
bd_user1=> SELECT * from Moyenne;
```

moyenne_id	moyenne
0	0

(1 ligne)

```
bd_user1=> SELECT * from Matiere;
```

matiere_id	matiere	coefficient	competence_id
0	BDD	30	0
1	Dev Java	60	1
2	Qualité dév	15	1

(3 lignes)

```
bd_user1=> SELECT * from Controle;
```

controle_id	matiere_id	controle
0	0	Petit controle BDD
1	0	Controle BDD 2
2	1	Controle Java

(3 lignes)

```
bd_user1=> SELECT * from Notes;
```

etudiant_id	controle_id	moyenne_id	note
0	0	0	4.00
0	1	0	14.00
0	2	0	16.00
1	0	0	12.00

(4 lignes)

```
bd_user1=> SELECT * from Groupe;
```

groupe_id	nomgroupe
0	Cygnus
1	Pegasus
2	Phoenix
3	Draco
4	Andromeda

(5 lignes)

```
bd_user1=> SELECT * from Competences;
```

competence_id	matiere_id	competence
0	0	Gérer des données
1	1	Réaliser un dév app.

(2 lignes)

Après réflexion, la relation/table moyenne n'a pas été utilisée car nous n'avons pas eu besoin de stocker les moyennes.

#### I / Deuxième partie

##### 1) Vues

```
CREATE view nombre_etudiants as
  SELECT count(*)
    from etudiant as nombre_etudiants;
```

Nous avons décidé de créer une vue qui compte le nombre d'étudiants dans la promo.

```
bd_user1=> select * from nombre_etudiants;
count
-----
      5
(1 ligne)
```

```
CREATE view notes_matiere AS
  SELECT Matiere,Controle>Note
    FROM Matiere natural join Controle
      natural join Notes natural join Etudiant
  order by matiere, controle;
```

Cette vue répertorie les notes dans chaque matière

```
bd_user1=> select * from notes_matiere;
matiere | controle | note | nom | prenom
-----+-----+-----+-----+-----
BDD      | Controle BDD 2 | 14.00 | Dikhamidze | Giorgi
BDD      | Petit controle BDD | 4.00 | Dikhamidze | Giorgi
BDD      | Petit controle BDD | 12.00 | Dujardin | Jean
Dev Java | Controle Java | 16.00 | Dikhamidze | Giorgi
(4 lignes)
```

```
CREATE view controles_matiere AS
  SELECT matiere,controle
    FROM Matiere natural join Controle
  where matiere.matiere_id=controle.matiere_id;
```

Quant à cette vue, elle permet de voir tous les contrôles de chaque matière.

```
bd_user1=> select * from controles_matiere;
matiere | controle
-----+-----
BDD      | Petit controle BDD
BDD      | Controle BDD 2
Dev Java | Controle Java
(3 lignes)
```

```
CREATE view moy_groupes AS
  SELECT Etudiant.nomgroupe, avg(note)::decimal(4,2) as moyenne
  from notes natural join etudiant natural join groupe
  group by nomgroupe
  order by moyenne;
```

Cette vue permet de voir les différentes moyennes de chaque groupe présent dans la promotion.

```
bd_user1=> select * from moy_groupes;
 nomgroupe | moyenne
-----+-----
 Cygnus    |    11.33
 Draco     |    12.00
(2 lignes)
```

```
CREATE VIEW moyennes_matiere AS
  SELECT Etudiant.Etudiant_id, Nom, Prenom, Matiere.matiere,
  round(avg(note),2) as moyenne
  FROM Etudiant natural join Notes natural join
  Controle natural join Matiere
  WHERE (Matiere.Matiere_id=Controle.Matiere_id
  AND Controle.Controle_id =Notes.Controle_id
  AND Notes.Etudiant_id =Etudiant.Etudiant_id)
  GROUP BY Etudiant.etudiant_id, Nom, Prenom, Matiere.matiere;
```

La dernière vue permet de voir la moyenne de chaque étudiant. On voit que l'étudiant Giorgi Dikhamidze a eu une note de 4 et 14 en BDD et que donc logiquement sa moyenne est de 9 dans cette matière.

```
bd_user1=> select * from moyennes_matiere;
 etudiant_id | nom      | prenom | matiere | moyenne
-----+-----+-----+-----+-----
          0 | Dikhamidze | Giorgi | Dev Java |    16.00
          1 | Dujardin   | Jean   | BDD      |    12.00
          0 | Dikhamidze | Giorgi | BDD      |     9.00
(3 lignes)
```

## 2) Procédures

Note: les fonctions retournant un tableau par “return setof record” semblent ne pas séparer les données en colonnes

```
CREATE or REPLACE FUNCTION moy_grp_specifique(in grp varchar, out
nomgroupe varchar, out moyenne numeric)
returns setof record
AS $$
    SELECT Etudiant.nomgroupe, avg(note)::decimal(4,2) as moyenne
    from notes natural join etudiant natural join groupe
    where nomgroupe=grp
    group by nomgroupe;

$$ language SQL;
```

En fonction du groupe renseigné, cette procédure retourne la moyenne de celle-ci.

```
bd_user1=> select moy_grp_specifique('Cygnus');
moy_grp_specifique
-----
(Cygnus,11.33)
(1 ligne)
```

```
CREATE or REPLACE FUNCTION moy_matiere( in mat varchar, out matiere
varchar, out decimal(4,2))
AS $$
    select matiere,avg(note)::decimal(4,2) as moyenne
    from matiere natural join controle
        natural join notes natural join etudiant
    where matiere=mat
    group by matiere;

$$ language SQL;
```

La procédure moy\_matiere donne la moyenne de la matière indiquée dans la promotion BUT Informatique 1ère année.

```
bd_user1=> select moy_matiere('BDD');
moy_matiere
-----
(BDD,10.00)
(1 ligne)
```

```
CREATE or REPLACE FUNCTION moy_general_etudiant( in id integer, out nom
varchar,out prenom varchar,out decimal(4,2))
AS $$
    select nom,prenom, avg(note)::decimal(4,2) as moyenne
    from notes natural join etudiant
    where etudiant_id=id
    group by nom,prenom;

$$ language SQL;
```

Cette procédure, comme son nom l'indique, retourne la moyenne générale de l'étudiant.

```
bd_user1=> select moy_general_etudiant(0);
           moy_general_etudiant
-----
(Dikhamidze,Georgi,11.33)
(1 ligne)
```

```
CREATE or REPLACE FUNCTION etudiant_grp( in grp varchar, out nom
varchar, out prenom varchar)
returns setof record
AS $$
    SELECT nom,prenom
    FROM Etudiant
        natural join Groupe
    where etudiant.nomgroupe=grp;

$$ language SQL;
```

etudiant\_grp() permet de lister tous les étudiants d'un groupe

```
bd_user1=> select etudiant_grp('Cygnus');
           etudiant_grp
-----
(Dikhamidze,Georgi)
(Norris,Chuck)
(Benzema,Karim)
(3 lignes)
```

---

```
CREATE or REPLACE FUNCTION notes_groupe(in grp varchar, out matiere
                                         varchar, out controle varchar, out note numeric,
                                         out nom varchar, out prenom varchar)
returns setof record
AS $$
    select Matiere, Controle, Note, nom, prenom
    from matiere natural join controle
        natural join notes natural join etudiant
    where etudiant.nomgroupe=grp;

$$ language SQL;
```

---

Cette ultime procédure regroupe toutes les notes d'un groupe spécifique, ici le groupe Cygnus plus précisément.

```
bd_user1=> select * from notes_groupe('Cygnus')
bd_user1-> ;*
```

matiere	controle	note	nom	prenom
BDD	Petit controle BDD	4.00	Dikhamidze	Giorgi
BDD	Controle BDD 2	14.00	Dikhamidze	Giorgi
Dev Java	Controle Java	16.00	Dikhamidze	Giorgi

(3 lignes)

---

## II/ Troisième partie

Nous avons créé des rôles dont nous n'avons pas eu la possibilité de tester.

```
CREATE ROLE administrateur ADMIN role_membre PASSWORD
'iut_villetaneuse_2022_admin';
CREATE ROLE responsable_matiere ADMIN role_membre PASSWORD
'iut_villetaneuse_2022_matiere';
CREATE ROLE enseignant ADMIN role_membre PASSWORD
'iut_villetaneuse_2022';
CREATE ROLE etudiant;
```

Par la suite, nous avons rédigé quelques permissions.

```
GRANT (INSERT,UPDATE,DELETE,SELECT) ON * TO administrateur WITH GRANT
OPTION;
GRANT (INSERT,UPDATE,DELETE,SELECT) ON (Competences,Matiere) TO
responsable_matiere WITH GRANT OPTION;
GRANT (INSERT,UPDATE,DELETE,SELECT) ON (Controle, Notes) TO enseignant
WITH GRANT OPTION;
GRANT SELECT ON (notes_etudiant, Matiere, Competences, Etudiant,
Groupe) TO etudiant WITH GRANT OPTION;

REVOKE INSERT,UPDATE,DELETE ON * FROM etudiant;
```

```
CREATE view notes_etudiant AS
    SELECT Matiere,Controle,Note
        FROM Matiere natural join Controle
            natural join Notes natural join Etudiant
        WHERE nom=current_user OR prenom=current_user;
```

Pour cette vue, l'étudiant Dikhamidze utilise la fonction pour visualiser ses notes aux contrôles qu'il a passés dans différentes matières.

```
bd_user1=> select * from notes_etudiant;
matiere | controle | note
-----+-----+-----
BDD | Petit controle BDD | 4.00
BDD | Controle BDD 2 | 14.00
Dev Java | Controle Java | 16.00
(3 lignes)
```

```
CREATE view etudiants_de_son_grp AS
  SELECT etudiant_id,nom,prenom
    FROM etudiant
   WHERE nomgroupe=(select nomgroupe from etudiant where
nom=current_user);
```

Ici, on se place dans la situation où Karim Benzema veut voir la liste des étudiants de son groupe.

```
bd_user1=> select * from etudiants_de_son_grp;
 etudiant_id |      nom      | prenom
-----+-----+-----
          0 | Dikhamidze    | Giorgi
          2 | Norris        | Chuck
          4 | Benzema       | Karim
(3 lignes)
```

Par souci d'équilibrage et de bon fonctionnement de la base de données, nous n'avons pas rédigé d'autres permissions et procédures et avons donc essayé de rédiger les idées théoriques de ces derniers.

Pour qu'un enseignant ne puisse saisir des notes seulement dans sa matière, il aurait fallu créer une table qui associerait une matière à son enseignant et éventuellement créer un trigger avant l'insertion de notes.