S2.04 Exploitation d'une Base de Données - Partie 2 et 3

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, Nom, Prenom

    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 offre la possibilité d'observer 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;
```

Il est possible d'observer les moyennes respectives de chaque groupe au sein de la promotion grâce à cette vue.

```
bd_user1=> select * from moy_groupes;
 nomgroupe | moyenne
-----------+---------
 Tlaloc    |   11.33
 Indra     |   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;
```

En examinant la dernière vue, on peut observer les moyennes individuelles des étudiants. Il est clair que Giorgi Dikhamidze a obtenu des notes de 4 et 14 en BDD, ce qui implique logiquement qu'il a une moyenne 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

```
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('Tlaloc');
 moy_grp_specifique
--------------------
 (Tlaloc,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,Giorgi,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('Tlaloc');
    etudiant_grp
---------------------
 (Dikhamidze,Giorgi)
 (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 Tlaloc plus précisément.

```
bd_user1=> select * from notes_groupe('Tlaloc')
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

1) Dans cette troisième partie nous allons définir les règles d'accès aux données.

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

```
CREATE ROLE administrateur ADMIN role_membre PASSWORD "iut_villetaneuse_2023_admin";
CREATE ROLE responsable_matiere ADMIN role_membre PASSWORD "iut_villetaneuse_2023_matiere";
CREATE ROLE enseignant ADMIN role_membre PASSWORD "iut_villetaneuse_2023";
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;
```

2) Maintenant nous allons voir les procédures et les vues pour mettre en œuvre les règles de la base de données.

```sql
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;
```

Voici la vue notes_etudiant qui affiche les informations sur les notes d'un étudiant spécifique.

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

Voici la vue etudiants_de_son_grp qui contient les informations des étudiants appartenant au même groupe que l'utilisateur actuel.