III - Visualisation des données

Vues :

1)

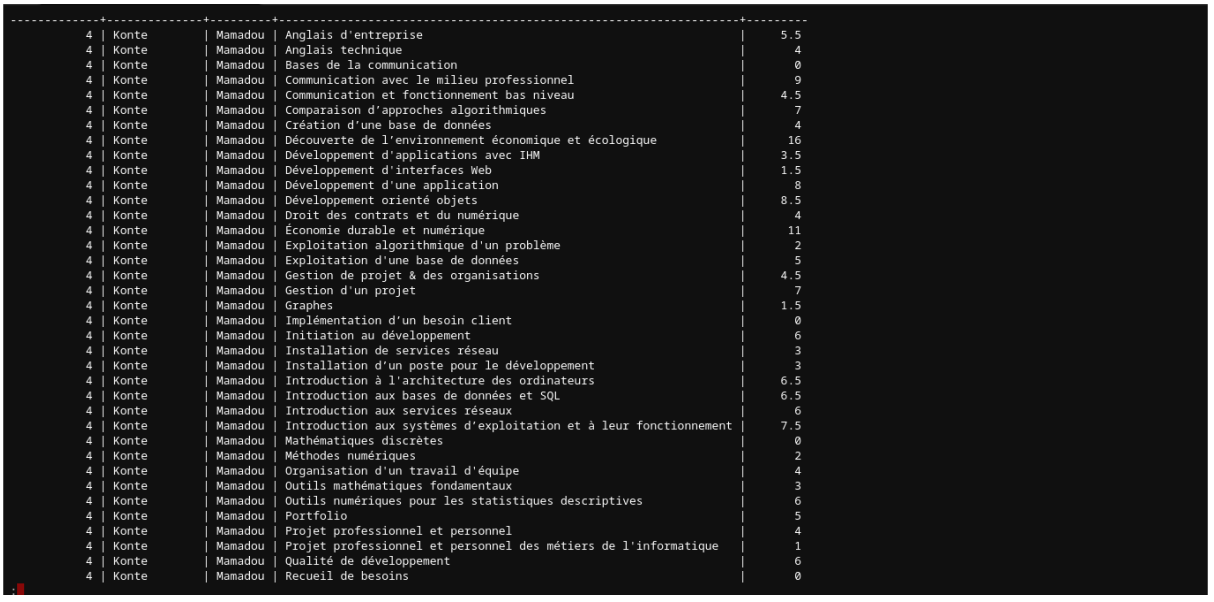
```
CREATE VIEW Moyennes_matiere
AS
SELECT e.id_personne, p.nom as nom_etudiant, p.prenom, m.nom as
nom_matiere, avg(note) as moyenne
FROM etudiant e, matiere m, controle c, notes n, personne p
WHERE m.id_matiere=c.id_matiere AND p.id_personne=e.id_personne AND
c.id_controle=n.id_controle AND n.id_personne=e.id_personne and
m.id_matiere=n.id_matiere
GROUP BY e.id_personne, p.nom, p.prenom, m.nom;
```

2) *Moyenne_matiere* renvoi la moyenne de toutes les compétences pour tous les étudiants

Si dans la requètes du terminal nous faisons ça :

```
postgres=# select * from  moyennes_matiere where id_personne=4;
postgres=#
```

On peut avoir juste les moyennes de l'élève selectionné par son *id*



```
------------+-------------+---------+-----------------------------------------------------------------+---------
         4 | Konte       | Mamadou | Anglais d'entreprise                                            |     5.5
         4 | Konte       | Mamadou | Anglais technique                                               |       4
         4 | Konte       | Mamadou | Bases de la communication                                       |       0
         4 | Konte       | Mamadou | Communication avec le milieu professionnel                      |       9
         4 | Konte       | Mamadou | Communication et fonctionnement bas niveau                      |     4.5
         4 | Konte       | Mamadou | Comparaison d'approches algorithmiques                          |       7
         4 | Konte       | Mamadou | Création d'une base de données                                  |       4
         4 | Konte       | Mamadou | Découverte de l'environnement économique et écologique          |      16
         4 | Konte       | Mamadou | Développement d'applications avec IHM                           |     3.5
         4 | Konte       | Mamadou | Développement d'interfaces Web                                  |     1.5
         4 | Konte       | Mamadou | Développement d'une application                                 |       8
         4 | Konte       | Mamadou | Développement orienté objets                                    |     8.5
         4 | Konte       | Mamadou | Droit des contrats et du numérique                              |       4
         4 | Konte       | Mamadou | Économie durable et numérique                                   |      11
         4 | Konte       | Mamadou | Exploitation algorithmique d'un problème                        |       2
         4 | Konte       | Mamadou | Exploitation d'une base de données                              |       5
         4 | Konte       | Mamadou | Gestion de projet & des organisations                           |     4.5
         4 | Konte       | Mamadou | Gestion d'un projet                                             |       7
         4 | Konte       | Mamadou | Graphes                                                         |     1.5
         4 | Konte       | Mamadou | Implémentation d'un besoin client                               |       0
         4 | Konte       | Mamadou | Initiation au développement                                     |       6
         4 | Konte       | Mamadou | Installation de services réseau                                 |       3
         4 | Konte       | Mamadou | Installation d'un poste pour le développement                   |       3
         4 | Konte       | Mamadou | Introduction à l'architecture des ordinateurs                   |     6.5
         4 | Konte       | Mamadou | Introduction aux bases de données et SQL                        |     6.5
         4 | Konte       | Mamadou | Introduction aux services réseaux                               |       6
         4 | Konte       | Mamadou | Introduction aux systèmes d'exploitation et à leur fonctionnement |     7.5
         4 | Konte       | Mamadou | Mathématiques discrètes                                         |       0
         4 | Konte       | Mamadou | Méthodes numériques                                             |       2
         4 | Konte       | Mamadou | Organisation d'un travail d'équipe                              |       4
         4 | Konte       | Mamadou | Outils mathématiques fondamentaux                               |       3
         4 | Konte       | Mamadou | Outils numériques pour les statistiques descriptives            |       6
         4 | Konte       | Mamadou | Portfolio                                                       |       5
         4 | Konte       | Mamadou | Projet professionnel et personnel                               |       4
         4 | Konte       | Mamadou | Projet professionnel et personnel des métiers de l'informatique |       1
         4 | Konte       | Mamadou | Qualité de développement                                        |       6
         4 | Konte       | Mamadou | Recueil de besoins                                              |       0
(
```

Sinon sans indications particulières on a tout :

```
 1 | Zidee       | Johann   | Exploitation algorithmique d'un problème                          |    15
 1 | Zidee       | Johann   | Exploitation d'une base de données                                |    12
 1 | Zidee       | Johann   | Gestion de projet & des organisations                             |    13
 1 | Zidee       | Johann   | Gestion d'un projet                                               |     3
 1 | Zidee       | Johann   | Graphes                                                           |   2.5
 1 | Zidee       | Johann   | Implémentation d'un besoin client                                 |    20
 1 | Zidee       | Johann   | Initiation au développement                                       |    11
 1 | Zidee       | Johann   | Installation de services réseau                                   |    20
 1 | Zidee       | Johann   | Installation d'un poste pour le développement                     |     3
 1 | Zidee       | Johann   | Introduction à l'architecture des ordinateurs                     |  11.5
 1 | Zidee       | Johann   | Introduction aux bases de données et SQL                          |  14.5
 1 | Zidee       | Johann   | Introduction aux services réseaux                                 |    17
 1 | Zidee       | Johann   | Introduction aux systèmes d'exploitation et à leur fonctionnement |  17.5
 1 | Zidee       | Johann   | Mathématiques discrètes                                           |     0
 1 | Zidee       | Johann   | Méthodes numériques                                               |    12
 1 | Zidee       | Johann   | Organisation d'un travail d'équipe                                |    18
 1 | Zidee       | Johann   | Outils mathématiques fondamentaux                                 |     3
 1 | Zidee       | Johann   | Outils numériques pour les statistiques descriptives              |   7.5
 1 | Zidee       | Johann   | Portfolio                                                         |    15
 1 | Zidee       | Johann   | Projet professionnel et personnel                                 |     4
 1 | Zidee       | Johann   | Projet professionnel et personnel des métiers de l'informatique   |     3
 1 | Zidee       | Johann   | Qualité de développement                                          |    19
 1 | Zidee       | Johann   | Recueil de besoins                                                |    15
 2 | Eyala       | Seraphin | Anglais d'entreprise                                              |   5.5
 2 | Eyala       | Seraphin | Anglais technique                                                 |    14
 2 | Eyala       | Seraphin | Bases de la communication                                         |    16
 2 | Eyala       | Seraphin | Communication avec le milieu professionnel                        |    19
 2 | Eyala       | Seraphin | Communication et fonctionnement bas niveau                        |    12
 2 | Eyala       | Seraphin | Comparaison d'approches algorithmiques                            |    20
 2 | Eyala       | Seraphin | Création d'une base de données                                    |    20
 2 | Eyala       | Seraphin | Découverte de l'environnement économique et écologique            |    12
 2 | Eyala       | Seraphin | Développement d'applications avec IHM                             |  11.5
 2 | Eyala       | Seraphin | Développement d'interfaces Web                                    |    18
 2 | Eyala       | Seraphin | Développement d'une application                                   |     8
 2 | Eyala       | Seraphin | Développement orienté objets                                      |     9
 2 | Eyala       | Seraphin | Droit des contrats et du numérique                                |    15
 2 | Eyala       | Seraphin | Économie durable et numérique                                     |    18
 2 | Eyala       | Seraphin | Exploitation algorithmique d'un problème                          |     6
```

1)

```sql
Create view moyennes_semestre1
as
select e.id_personne, p.nom as nom_etudiant, p.prenom, s.id_semestre,
ROUND(CAST(avg(note) AS numeric), 2) as moyenne
FROM etudiant e, matiere m, controle c, notes n, personne p, semestre s
WHERE m.id_matiere=c.id_matiere AND p.id_personne=e.id_personne AND
c.id_controle=n.id_controle AND n.id_personne=e.id_personne and
m.id_matiere=n.id_matiere and s.id_semestre=m.id_semestre and
m.id_semestre=n.id_semestre and n.id_semestre=c.id_semestre and
s.id_semestre='S1'
group by e.id_personne, p.nom, p.prenom, s.id_semestre
```

2) *Moyenne_semestre1* renvoi la moyenne de chaque personne dans le semestre 1.

```
postgres=# select * from moyennes_semestre1;
 id_personne | nom_etudiant |  prenom  | id_semestre | moyenne
-------------+--------------+----------+-------------+---------
           1 | Zidee        | Johann   | S1          |   11.96
           2 | Eyala        | Seraphin | S1          |   17.30
           3 | Pommier      | Melvyn   | S1          |   13.22
           4 | Konte        | Mamadou  | S1          |    4.78
           5 | Martin       | Sophie   | S1          |   10.52
           6 | Leroy        | David    | S1          |   10.35
           7 | Morin        | Nathan   | S1          |   11.26
           8 | Gauthier     | Zoe      | S1          |   10.48
           9 | Roy          | Camille  | S1          |   11.52
          10 | Beaulieu     | Oceane   | S1          |   11.26
          11 | Hughes       | Leo      | S1          |   11.30
          12 | Parker       | Ava      | S1          |   10.96
(12 rows)
```

1)

```
Create view moyennes_semestre2
as
select e.id_personne, p.nom as nom_etudiant, p.prenom, s.id_semestre,
ROUND(CAST(avg(note) AS numeric), 2) as moyenne
FROM etudiant e, matiere m, controle c, notes n, personne p, semestre s
WHERE m.id_matiere=c.id_matiere AND p.id_personne=e.id_personne AND
c.id_controle=n.id_controle AND n.id_personne=e.id_personne and
m.id_matiere=n.id_matiere and s.id_semestre=m.id_semestre and
m.id_semestre=n.id_semestre and n.id_semestre=c.id_semestre and
s.id_semestre='S2'
group by e.id_personne, p.nom, p.prenom, s.id_semestre
```

2) *Moyenne_semestre2* renvoi la moyenne de chaque personne dans le semestre 2.

```
postgres=# select * from moyennes_semestre2;
 id_personne | nom_etudiant |  prenom  | id_semestre | moyenne
-------------+--------------+----------+-------------+---------
           1 | Zidee        | Johann   | S2          |   11.57
           2 | Eyala        | Seraphin | S2          |   10.68
           3 | Pommier      | Melvyn   | S2          |   11.61
           4 | Konte        | Mamadou  | S2          |    4.93
           5 | Martin       | Sophie   | S2          |   11.96
           6 | Leroy        | David    | S2          |   12.43
           7 | Morin        | Nathan   | S2          |   12.79
           8 | Gauthier     | Zoe      | S2          |   14.39
           9 | Roy          | Camille  | S2          |   13.04
          10 | Beaulieu     | Oceane   | S2          |   11.61
          11 | Hughes       | Leo      | S2          |   12.57
          12 | Parker       | Ava      | S2          |    4.18
(12 rows)
```

1)

```
CREATE VIEW Moyennes_groupe
AS
SELECT g.nom as nom_groupe,e.id_personne, p.nom as nom_etudiant,
p.prenom, s.id_semestre, ROUND(CAST(avg(note) AS numeric), 2) as
moyenne
FROM etudiant e, matiere m, controle c, notes n, personne p, groupe g,
semestre s
WHERE m.id_matiere=c.id_matiere AND p.id_personne=e.id_personne AND
c.id_controle=n.id_controle AND n.id_personne=e.id_personne and
m.id_matiere=n.id_matiere and g.id_personne=e.id_personne and
s.id_semestre=m.id_semestre and m.id_semestre=n.id_semestre and
n.id_semestre=c.id_semestre
GROUP BY nom_groupe, e.id_personne, p.nom, p.prenom, s.id_semestre;
```

2) *Moyenne_groupe* renvoi la moyenne de chaque personne de chaque groupe.

```
postgres=# select * from moyennes_groupe ;
 nom_groupe | id_personne | nom_etudiant |  prenom  | id_semestre | moyenne
------------+-------------+--------------+----------+-------------+---------
 Shango     |           5 | Martin       | Sophie   | S1          |   10.52
 Shango     |           5 | Martin       | Sophie   | S2          |   11.96
 Shango     |           6 | Leroy        | David    | S1          |   10.35
 Shango     |           6 | Leroy        | David    | S2          |   12.43
 Shango     |           7 | Morin        | Nathan   | S1          |   11.26
 Shango     |           7 | Morin        | Nathan   | S2          |   12.79
 Shango     |           8 | Gauthier     | Zoe      | S1          |   10.48
 Shango     |           8 | Gauthier     | Zoe      | S2          |   14.39
 Whaitiri   |           1 | Zidee        | Johann   | S1          |   11.96
 Whaitiri   |           1 | Zidee        | Johann   | S2          |   11.57
 Whaitiri   |           2 | Eyala        | Seraphin | S1          |   17.30
 Whaitiri   |           2 | Eyala        | Seraphin | S2          |   10.68
 Whaitiri   |           3 | Pommier      | Melvyn   | S1          |   13.22
 Whaitiri   |           3 | Pommier      | Melvyn   | S2          |   11.61
 Whaitiri   |           4 | Konte        | Mamadou  | S1          |    4.78
 Whaitiri   |           4 | Konte        | Mamadou  | S2          |    4.93
 Zeus       |           9 | Roy          | Camille  | S1          |   11.52
 Zeus       |           9 | Roy          | Camille  | S2          |   13.04
 Zeus       |          10 | Beaulieu     | Oceane   | S1          |   11.26
 Zeus       |          10 | Beaulieu     | Oceane   | S2          |   11.61
 Zeus       |          11 | Hughes       | Leo      | S1          |   11.30
 Zeus       |          11 | Hughes       | Leo      | S2          |   12.57
 Zeus       |          12 | Parker       | Ava      | S1          |   10.96
 Zeus       |          12 | Parker       | Ava      | S2          |    4.18
(24 rows)
```

Procédures :

1)

```sql
CREATE OR REPLACE function VoirMoyenne(id_etudiant INT, semestres
VARCHAR)
RETURNS VOID AS $$
DECLARE
  moyenne NUMERIC;
BEGIN
  -- Sélectionner les informations de l'étudiant
  SELECT ROUND(CAST(avg(note) AS numeric), 2) INTO moyenne
  FROM etudiant e
  INNER JOIN personne p ON p.id_personne = e.id_personne
  INNER JOIN notes n ON n.id_personne = e.id_personne
  INNER JOIN controle c ON c.id_controle = n.id_controle
  INNER JOIN matiere m ON m.id_matiere = n.id_matiere
  INNER JOIN semestre s ON s.id_semestre = m.id_semestre
  WHERE e.id_personne = id_etudiant AND s.id_semestre = semestres;
 -- Afficher le numéro d'étudiant
  RAISE NOTICE 'Pour l''étudiant numéro: %', id_etudiant;
 -- Afficher la moyenne de l'étudiant
  RAISE NOTICE 'La moyenne de l''étudiant est : %', moyenne;
END;
$$ LANGUAGE plpgsql;
```

2) La procédure VoirMoyenne déclare moyenne, pour ensuite la calculer pour l'étudiant dont son id est donné en paramètre, pour le semestre donné également en paramètre.

Le numéro d'identifiant de l'étudiant est donc donné avec sa moyenne grâce aux RAISE NOTICE.

```
postgres=# select voirmoyenne(1,'S1');
NOTICE:  Pour l'étudiant numéro: 1
NOTICE:  La moyenne de l'étudiant est : 11.96
 voirmoyenne
-------------

(1 row)

postgres=# select voirmoyenne(1,'S2');
NOTICE:  Pour l'étudiant numéro: 1
NOTICE:  La moyenne de l'étudiant est : 11.57
 voirmoyenne
-------------

(1 row)

postgres=# select voirmoyenne(2,'S2');
NOTICE:  Pour l'étudiant numéro: 2
NOTICE:  La moyenne de l'étudiant est : 10.68
 voirmoyenne
-------------

(1 row)

postgres=# select voirmoyenne(10,'S1');
NOTICE:  Pour l'étudiant numéro: 10
NOTICE:  La moyenne de l'étudiant est : 11.26
 voirmoyenne
-------------

(1 row)
```

IV - Restrictions d'accès aux Données

1)

```
CREATE FUNCTION AjouterNote( p_id_etudiant int, p_id_type varchar,
p_id_controle int,p_id_matiere INT,p_id_semestre varchar, p_note
DECIMAL(4, 2), p_id_responsable INT
)
RETURNS VOID
AS $$
BEGIN
    IF NOT EXISTS (
        SELECT 1
        FROM matiere m, controle c
        WHERE m.id_matiere = p_id_matiere
          AND m.id_personne = p_id_responsable
    ) THEN
        RAISE EXCEPTION 'Le professeur n''est pas le référent de cette
matière.';
    END IF;
    IF NOT EXISTS (
        SELECT 1
        FROM controle c
        WHERE c.id_controle= p_id_controle
    ) THEN
        RAISE EXCEPTION 'Le contrôle n''existe pas';
    END IF;

    INSERT INTO notes (id_personne, id_type, id_controle,
id_matiere,id_semestre, note)
```

```
    VALUES (p_id_etudiant, p_id_type, p_id_controle, p_id_matiere,
p_id_semestre, p_note );
END;
$$ LANGUAGE plpgsql;
```

2) Dans cette fonction, on teste d'abord si la personne qui ajoute la note est un responsable d'une la matière dont il veut ajouter la note. Si ça n'est pas le cas, un message d'erreur apparait pour signaler qu'il n'est pas le référent de cette matière. Ensuite on teste si le contrôle, pour lequel le référent veut rentrer une note, existe ou non car on ne peut pas rentrer de note pour un contrôle pas déjà fait. Si ça n'est pas le cas, un RAISE EXCEPTION apparait disant que le contrôle n'existe pas.
Lorsque tous les tests sont passés, un INSERT est effectué dans la table note.

```
postgres=# select ajouternote(4,'R',200,101,'S2',16,28);
ERROR:  Le professeur n'est pas le référent de cette matière.
CONTEXT:  PL/pgSQL function ajouternote(integer,character varying,integer,integer,character varying,numeric,integer) line 9 at RAISE
postgres=# select ajouternote(4,'R',200,101,'S2',16,26);
ERROR:  Le controle n'existe pas
CONTEXT:  PL/pgSQL function ajouternote(integer,character varying,integer,integer,character varying,numeric,integer) line 16 at RAISE
```

Sinon :

```
postgres=# select ajouternote(3,'R',200,101,'S1',16,26);
 ajouternote
-------------

(1 row)
```

```
        12 | S       |         550 |         220 | S2          |    8
        12 | P       |         551 |         221 | S2          |    1
         3 | R       |         200 |         101 | S1          |   16
(613 rows)
```

1)

```
CREATE FUNCTION Ajoutercontrole( p_id_responsable INT,p_id_controle
int, p_id_type varchar,p_id_semestre varchar,p_id_matiere INT, p_nom
varchar,p_date varchar
)

RETURNS VOID
AS $$
BEGIN
    IF EXISTS (
        SELECT 1
        FROM controle c
        WHERE c.id_controle= p_id_controle
    ) THEN
```

```
        RAISE EXCEPTION 'Le contrôle existe.';
    END IF;
    IF not EXISTS (
        SELECT 1
        FROM matiere m, controle c
        WHERE m.id_matiere = p_id_matiere
          AND m.id_personne = p_id_responsable
    ) THEN
        RAISE EXCEPTION 'Le professeur n''est pas le référent de cette
matière.';
    END IF;

    INSERT INTO controle (id_controle, id_type, id_semestre,
id_matiere, nom, date_eval)
    VALUES (p_id_controle, p_id_type, p_id_semestre,p_id_matiere,
p_nom,p_date);
END;
$$ LANGUAGE plpgsql;
```

2) Dans cette fonction, on teste d'abord si le contrôle, pour lequel le référent veut rentrer une note, existe ou non car on ne peut pas rentrer de note pour un contrôle pas déjà fait. Si ça n'est pas le cas, un RAISE EXCEPTION apparait disant que le contrôle n'existe pas

   Ensuite on teste si la personne qui ajoute la note est un responsable d'une la matière dont il veut ajouter la note. Si ça n'est pas le cas, un message d'erreur apparait pour signaler qu'il n'est pas le référent de cette matière.

   Lorsque tous les tests sont passés, un INSERT est effectué dans la table controle. Cette fonction est la suite logique de la première car, effectivement, si nous voulons ajouter des notes, il faut d'abord ajouter des contrôles.

```
postgres=# select ajoutercontrole(28,200,'R','S1',101,'controle','26/07/2023');
ERROR:  Le controle existe.
CONTEXT:  PL/pgSQL function ajoutercontrole(integer,integer,character varying,character varying,integer,character varying,character varying) line 8 at RAISE
postgres=# select ajoutercontrole(28,201,'R','S1',101,'controle','26/07/2023');
ERROR:  Le professeur n'est pas le référent de cette matière.
```

```
postgres=# select ajoutercontrole(26,200,'R','S1',101,'controle','26/07/2023');
 ajoutercontrole
-----------------

(1 row)
```

Sinon quand tout est bon :

```
        549 | S       | S2          |       219 | SAE-2.19       | 09/06/2023
        550 | S       | S2          |       220 | SAE-2.20       | 23/05/2023
        551 | P       | S2          |       221 | Portfolio      | 14/06/2023
        200 | R       | S1          |       101 | controle       | 26/07/2023
(52 rows)
```