

S2.04 Exploitation d'une Base de Données Rendu final v2

Yassine BELLAGRAA

Faty LO

Marie Faty

TABLE DES MATIÈRES

S2.04 Exploitation d'une Base de Données - Rendu final v2

I. Modélisation de données	3
II. Visualisation de données	10
III. Restrictions d'accès aux données	16

I. Modélisation de données

1. Etablir un cahier des charges.

Une base de données pour quoi faire ?

On va chercher à gérer les notes des étudiants en BUT en prenant en compte le système de notation (SAE, ressource, compétence) et également contrôler les accès de chacun.

Quelles seront les futurs tables ?

- Table Prof
- Table Ressource
- Table Competence
- Table Controle
- Table SAE
- Table Rendu
- Table Notes
- Table Etudiant

Comment reconnaître les clés primaires et les clés étrangères sur le futur modèle de données ?

Clé primaire : ce qui est en gras

Clé étrangère : ce qui est en italique

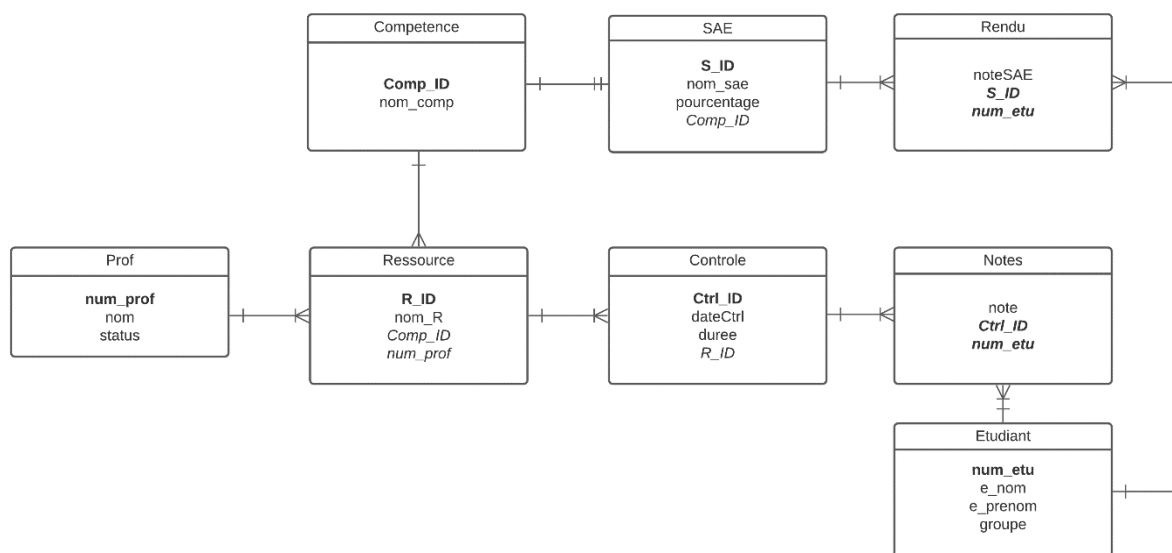
Quel est notre objectif ?

Notre objectif est d'apprendre à analyser, comparer et stocker les notes d'étudiants en BUT.

Nous verrons également comment visualiser ces données.

Pour finir, nous définirons des restrictions d'accès aux futurs utilisateurs qui auront accès à notre base de données.

2. Etudier un modèle de données et réaliser une Base de Données à partir de ce modèle.



bd_notes
Prof
Ressource
Competence
SAE
Rendu
Notes
Etudiant
Controle

Prof	
num_prof	Numéro d'identification d'un prof (suite de 6 chiffres)
nom	Nom de famille du prof
status	Enseignant, Enseignante, etc.

Competence	
Comp_ID	Code de la compétence (C1, C2, ...)
nom_comp	Nom de la compétence

Ressource	
R_ID	Code de la ressource (R2.01, R1.08, ...)
nom_R	Nom de la ressource
Comp_ID	Clé primaire de la table Competence
num_prof	Clé primaire de la table Prof

SAE	
S_ID	Code de la SAE (S2.04, S1.05, ...)
nom_sae	Nom de la SAE
pourcentage	Pourcentage de la SAE dans une compétence
<i>Comp_ID</i>	Clé primaire de la table Competence

Controle	
Ctrl_ID	Numéro d'identification d'un contrôle (4 chiffres)
dateCtrl	Date de réalisation du contrôle
duree	Court, Moyen ou Long
<i>R_ID</i>	Clé primaire de la table Ressource

Rendu	
noteSAE	Note obtenue à une SAE
S_ID	Clé primaire de la table SAE
num_etu	Clé primaire de la table Etudiant

Notes	
note	Note obtenue à une ressource
Ctrl_ID	Clé primaire de la table Controle
num_etu	Clé primaire de la table Etudiant

Etudiant	
num_etu	Numéro étudiant
e_nom	Nom de l'étudiant
e_prenom	Prénom de l'étudiant
groupe	Groupe de l'étudiant (ex : Draco, Phoenix, ...)

3. Définir les règles de gestion de ces données et leurs mises en œuvre par des procédures stockées.

(Voir Partie 1)

4. Fournir un script de création de la base de données.

Ouvrons un terminal.

Pour créer une base de données, nous allons d'abord nous connecter sur notre base de données par défaut, dans notre cas postgres.

```
psql -d postgres -U postgres
```

À partir de là, nous allons créer une nouvelle base de données que l'on nommera bd_note :

```
create database bd_notes;
```

Fermons le terminal.

Pour allons maintenant nous connecter à bd_notes.

La commande à saisir est :

```
psql -U postgres -d bd_notes
```

Lorsque nous sommes dans notre base de données, nous pouvons créer nos tables à partir du script préalablement créé.

Nous pouvons directement copier-coller le script ci-dessous dans le terminal.

Si les tables ont bien été créées, nous aurons la liste des tables avec la commande \d

Script de création :

```
DROP TABLE IF EXISTS Prof CASCADE;  
DROP TABLE IF EXISTS Ressource CASCADE;  
DROP TABLE IF EXISTS Competence CASCADE;  
DROP TABLE IF EXISTS SAE CASCADE;  
DROP TABLE IF EXISTS Controle CASCADE;  
DROP TABLE IF EXISTS Notes CASCADE;  
DROP TABLE IF EXISTS Etudiant CASCADE;  
DROP TABLE IF EXISTS Rendu CASCADE;
```

```
CREATE TABLE Prof  
(  
    num_prof int primary key,  
    nom varchar(30),  
    status varchar(30)  
);
```

```
CREATE TABLE Competence  
(  
    Comp_ID varchar(2) primary key,  
    nom_comp text  
);
```

```
CREATE TABLE Ressource  
(  
    R_ID varchar(5) primary key,  
    nom_R text,  
    Comp_ID varchar(2),  
    num_prof int,  
    foreign key (Comp_ID) references Competence(Comp_ID),  
    foreign key (num_prof) references Prof(num_prof)  
);
```

CREATE TABLE SAE

```
(  
  S_ID varchar(5) primary key,  
  nom_sae text,  
  pourcentage int,  
  Comp_ID varchar(2),  
  foreign key (Comp_ID) references Competence(Comp_ID)  
);
```

CREATE TABLE Controle

```
(  
  Ctrl_ID int primary key,  
  dateCtrl date,  
  duree varchar(10),  
  R_ID varchar(5),  
  foreign key (R_ID) references Ressource(R_ID)  
);
```

CREATE TABLE Etudiant

```
(  
  num_etu int primary key,  
  e_nom varchar(30),  
  e_prenom varchar(30),  
  groupe varchar(20)  
);
```

CREATE TABLE Notes

```
(  
  note decimal(4,2),  
  Ctrl_ID int,  
  num_etu int,  
  foreign key (Ctrl_ID) references Controle(Ctrl_ID),  
  foreign key (num_etu) references Etudiant(num_etu),  
  primary key(num_etu, Ctrl_ID)
```


);

CREATE TABLE Rendu

```
(  
  noteSAE decimal(4,2),  
  S_ID varchar(5),  
  num_etu int,  
  foreign key (S_ID) references SAE(S_ID),  
  foreign key (num_etu) references Etudiant(num_etu),  
  primary key(num_etu, S_ID)  
);
```

II. Visualisation de données

1. Définir un ensemble de données dérivées à visualiser.

Nous avons décidé de choisir un ensemble de données dérivées à visualiser :

- L'ensemble des ressources attribuées à un prof.
- L'ensemble des ressources attribuées à une compétence.
- La moyenne de chacune des ressources de chaque étudiant.
- Toutes les notes de SAE de chaque étudiant.

2. Décrire des procédures, vues ou vues matérialisées pour accéder à ces données.

Insertions

Pour pouvoir observer les résultats des vues que nous avons créées, nous ferons des INSERT INTO dans le but de compléter nos tables créées à partir du script de la partie I.

```
INSERT INTO Prof VALUES (322616, 'Abir', 'Enseignant'),
(245165, 'Lebbah', 'Enseignant'),
(644232, 'Hebert', 'Enseignant'),
(644240, 'Nassiet', 'Enseignante'),
(644243, 'Clertant', 'Enseignant'),
(843164, 'Toure', 'Enseignante'),
(564678, 'Capdeville', 'Enseignante'),
(645203, 'De Saint Sernin', 'Enseignant'),
(616452, 'Couleau', 'Enseignante'),
(475635, 'Pisanu', 'Enseignant'),
(643981, 'Azzag', 'Enseignant'),
(645662, 'Butelle', 'Enseignant'),
(843618, 'Dubacq', 'Enseignant'),
(319462, 'Pekergin', 'Enseignant');
INSERT INTO Competence VALUES ('C1', 'Realiser un developpement d
application'),
('C2', 'Optimiser des applications informatiques'),
('C3', 'Administrer des systemes informatiques communicants complexes'),
('C4', 'Gerer des donnees de l information'),
('C5', 'Conduire un projet'),
('C6', 'Travailler dans une equipe informatique');
INSERT INTO Ressource VALUES ('R1.01', 'Initiation au developpement', 'C1',
643981),
('R1.02', 'Developpement d interfaces web', 'C5', 843618),
('R1.03', 'Introduction a l architecture des ordinateurs', 'C3', 319462),
```

```

('R1.04', 'Introduction aux systemes d exploitation et a leur
fonctionnement', 'C3', 843618),
('R1.05', 'Introduction aux bases de donnees et SQL', 'C4', 245165),
('R1.06', 'Mathematiques discretes', 'C2', 644232),
('R1.07', 'Outils mathematiques fondamentaux', 'C2', 644243),
('R1.08', 'Gestion de projet', 'C5', 843164),
('R1.09', 'Economie durable et numerique', 'C6', 843164),
('R1.10', 'Anglais technique', 'C3', 564678),
('R1.11', 'Bases de la communication', 'C5', 616452),
('R1.12', 'Projet professionnel et personnel', 'C6', 616452),
('R2.01', 'Initiation au developpement', 'C1', 643981),
('R2.02', 'Developpement d applications avec IHM', 'C1', 475635),
('R2.03', 'Qualite de developpement', 'C1', 475635),
('R2.04', 'Communication et fonctionnement bas niveau', 'C3', 645662),
('R2.05', 'Introduction aux services reseaux', 'C3', 645662),
('R2.06', 'Exploitation d une base de donnees', 'C4', 322616),
('R2.07', 'Graphes', 'C2', 644232),
('R2.08', 'Outils numeriques pour les statistiques descriptives', 'C4',
644240),
('R2.09', 'Methodes numeriques', 'C2', 644232),
('R2.10', 'Gestion de projet et des organisations', 'C5', 843164),
('R2.11', 'Droit des contrats et du numerique', 'C6', 645203),
('R2.12', 'Anglais d entreprise', 'C6', 564678),
('R2.13', 'Communication avec le milieu professionnel', 'C6', 616452),
('R2.14', 'Projet professionnel et personnel', 'C6', 616452);
INSERT INTO SAE VALUES ('S1.01', 'Implementation d un besoin client', 40,
'C1'),
('S1.02', 'Comparaison d approches algorithmiques', 40, 'C2'),
('S1.03', 'Installation d un poste pour le developpement', 40, 'C3'),
('S1.04', 'Creation d une base de donnees', 40, 'C4'),
('S1.05', 'Recueil de besoins', 40, 'C5'),
('S1.06', 'Decouverte de l environnement economique et ecologique', 40,
'C6'),
('S2.01', 'Developpement d une application', 38, 'C1'),
('S2.02', 'Exploration algorithmique d un probleme', 38, 'C2'),
('S2.03', 'Installation de services reseau ', 38, 'C3'),
('S2.04', 'Exploitation d une base de donnees', 38, 'C4'),
('S2.05', 'Gestion d un projet', 38, 'C5'),
('S2.06', 'Organisation d un travail d equipe', 38, 'C6');
INSERT INTO Controle VALUES (6366, '2022-04-19', 'Moyen', 'R2.06'),
(3156, '2022-01-21', 'Long', 'R1.01'),
(4566, '2022-05-24', 'Moyen', 'R2.01'),
(3424, '2022-05-31', 'Long', 'R2.06'),
(8324, '2022-01-19', 'Court', 'R1.03'),
(2438, '2022-01-19', 'Long', 'R1.09'),
(8348, '2022-01-12', 'Moyen', 'R1.07'),
(8921, '2022-06-14', 'Long', 'R2.02'),
(8622, '2022-06-14', 'Court', 'R2.05'),
(8123, '2022-06-14', 'Moyen', 'R2.09');
INSERT INTO Etudiant VALUES (12100104, 'Benaissa', 'Younes', 'Pegasus'),
(12100105, 'Botseke', 'Joe', 'Pegasus'),
(12100106, 'Paillot', 'Jefferson', 'Pegasus'),
(12100107, 'Hammache', 'Kais', 'Pegasus'),
(12100108, 'Baskaran', 'Sarujan', 'Draco'),
(12100109, 'Mahrez', 'Lilia', 'Draco'),
(12100110, 'Camara', 'Moustapha', 'Draco'),
(12100111, 'Buzura', 'Darius', 'Draco'),

```

```
(12100112, 'Caillaud', 'Maxime', 'Phoenix') ,
(12100113, 'Houmimid', 'Hani', 'Phoenix'),
(12100114, 'Lannuzel', 'Dylan', 'Phoenix'),
(12100115, 'Hayek', 'Sofiene', 'Phoenix'),
(12100116, 'Idier', 'Laetitia', 'Andromeda'),
(12100117, 'Salemkour', 'Tinhinane', 'Andromeda'),
(12100118, 'Loquemanique', 'Jonathan', 'Andromeda'),
(12100119, 'Talbi', 'Samir', 'Andromeda'),
(12100120, 'Agsous', 'Mokrane', 'Cygnus'),
(12100121, 'Amgad', 'Elabsy', 'Cygnus'),
(12100122, 'Ouyed', 'Neil', 'Cygnus'),
(12100123, 'Abdul', 'Malak', 'Cygnus');
```

```
INSERT INTO Notes VALUES (15, 6366 , 12100104),
```

```
(14, 8921, 12100104),
(17, 4566, 12100104),
(16, 8622, 12100104),
(12, 3156, 12100104),
(13, 3424, 12100105),
(12, 8921, 12100105),
(9, 8123, 12100105),
(10, 8324, 12100105),
(17, 8622, 12100105),
(14, 6366, 12100106),
(12, 8348, 12100106),
(11, 8324, 12100106),
(10, 8921, 12100106),
(19, 3424, 12100106),
(9, 6366, 12100107),
(8, 4566, 12100107),
(11, 8324, 12100107),
(15, 8123, 12100107),
(20, 3156, 12100107),
(11, 3424, 12100108),
(12, 4566, 12100108),
(18, 8348, 12100108),
(10, 8123, 12100108),
(16, 8324, 12100108),
(14, 6366, 12100109),
(14, 8622, 12100109),
(15, 4566, 12100109),
(16, 8324, 12100109),
(11, 3156, 12100109),
(7, 3424, 12100110),
(9, 8921, 12100110),
(10, 8622, 12100110),
(11, 8348, 12100110),
(17, 4566, 12100110),
(11, 6366, 12100111),
(15, 8622, 12100111),
(13, 8324, 12100111),
(18, 8123, 12100111),
(12, 4566, 12100111),
(10, 3424, 12100112),
(13, 8324, 12100112),
(13, 2438, 12100112),
(15, 8348, 12100112),
(10, 8123, 12100112),
```

```

(13, 6366, 12100113),
(19, 8921, 12100113),
(10, 2438, 12100113),
(11, 3156, 12100113),
(16, 4566, 12100113),
(18, 3424, 12100114),
(19, 8123, 12100114),
(12, 2438, 12100114),
(10, 8622, 12100114),
(14, 3156, 12100114),
(19, 6366, 12100115),
(10, 2438, 12100115),
(14, 8324, 12100115),
(12, 8348, 12100115),
(15, 8622, 12100115),
(11, 3424, 12100116),
(14, 8348, 12100116),
(8, 8622, 12100116),
(11, 4566, 12100116),
(9, 8123, 12100116),
(8, 3424, 12100117),
(14, 8921, 12100117),
(11, 8123, 12100117),
(10, 3156, 12100117),
(9, 8348, 12100117),
(16, 6366, 12100118),
(11, 4566, 12100118),
(12, 3156, 12100118),
(10, 8123, 12100118),
(14, 8324, 12100118),
(15, 3424, 12100119),
(10, 3156, 12100119),
(6, 8348, 12100119),
(5, 8622, 12100119),
(19, 2438, 12100119),
(9, 6366, 12100120),
(14, 8921, 12100120),
(11, 4566, 12100120),
(8, 8123, 12100120),
(13, 8324, 12100120),
(6, 3424, 12100121),
(10, 4566, 12100121),
(11, 8348, 12100121),
(15, 8324, 12100121),
(9, 8123, 12100121),
(12, 6366, 12100122),
(11, 4566, 12100122),
(14, 8348, 12100122),
(16, 2438, 12100122),
(10, 8324, 12100122),
(17, 3424, 12100123),
(19, 8123, 12100123),
(18, 8921, 12100123),
(20, 8324, 12100123),
(19, 8622, 12100123);
INSERT INTO Rendu VALUES (15.5, 'S2.06', 12100104),
(13, 'S2.04', 12100105),

```

```
(18, 'S2.05', 12100106),
(14, 'S2.01', 12100107),
(11, 'S2.02', 12100108),
(16, 'S2.06', 12100109),
(18.5, 'S2.04', 12100110),
(17, 'S2.03', 12100111),
(13.5, 'S2.06', 12100112),
(16.5, 'S2.03', 12100113),
(12, 'S2.02', 12100114),
(20, 'S2.01', 12100115),
(14, 'S2.05', 12100116),
(15.5, 'S2.05', 12100117),
(12.5, 'S2.06', 12100118),
(14.5, 'S2.03', 12100119),
(12, 'S2.02', 12100120),
(19.5, 'S2.01', 12100121),
(10.5, 'S2.01', 12100122),
(11, 'S2.01', 12100123);
```

Nous avons décidé par la suite que nous ferons principalement des vues pour visualiser nos données dérivées

L'ensemble des ressources attribuées l'ensemble des profs :

```
CREATE VIEW ressources_prof AS select r.R_ID,
p.nom from Ressource r,
Prof p where p.num_prof = r.num_prof group by p.nom,
r.R_ID order by r.R_ID asc;
select * from ressources_prof;
```

L'ensemble des ressources attribuées à une compétence :

```
CREATE VIEW ressources_competence AS select r.R_ID,
c.nom_comp,
c.Comp_ID from Competence c,
Ressource r where r.Comp_ID = c.Comp_ID group by c.Comp_ID,
r.R_ID,
c.nom_comp order by r.R_ID asc;
select * from ressources_competence;
```

La moyenne de chacune des ressources de chaque étudiant :

```
CREATE VIEW moyenneEtu_parRessources AS select c.R_ID,  
e.num_etu, e.e_nom,  
e.e_prenom,  
avg(n.note) as "moyenne_etudiant_par_ressources" from Notes n,  
Etudiant e,  
Controle c where n.num_etu = e.num_etu and c.Ctrl_ID = n.Ctrl_ID group by  
e.num_etu,  
c.R_ID, e.e_nom,  
e.e_prenom order by e.e_nom ;  
  
select * from moyenneEtu_parRessources;
```

Toutes les notes de SAE de chaque étudiant :

```
CREATE VIEW notes_sae AS select r.S_ID,  
e.num_etu, e.e_nom,  
e.e_prenom,  
r.noteSAE from Rendu r,  
Etudiant e where e.num_etu = r.num_etu group by r.S_ID,  
e.num_etu, e.e_nom, e.e_prenom,  
r.noteSAE order by e.e_nom ;  
  
select * from notes_sae;
```

III. Restrictions d'accès aux données

1. Définir des règles d'accès aux données.

- Un étudiant ne peut consulter que ses propres notes pour les ressources
- Un étudiant ne peut consulter que ses propres notes pour les SAE
- Un enseignant doit pouvoir saisir les notes de ses contrôles
- Un enseignant ne peut pas modifier une note qui ne fait partie de ses contrôles

Selon le choix fait il faudra refaire des vues

2. Décrire des procédures, vues pour mettre en œuvre ces règles.

Avant de créer des vues, on crée un rôle 'etudiant' et un rôle 'prof' dans le but d'éviter d'attribuer les privilèges manuellement pour chaque étudiant ou chaque professeur.

Pour cela on se connecte à la BDD bd_notes avec l'utilisateur

postgres : `psql -d bd_notes -U postgres`

On crée le rôle 'etudiant' avec : `CREATE ROLE etudiant LOGIN;`

On crée le rôle 'prof' avec : `CREATE ROLE prof LOGIN;`

On crée des utilisateurs : un étudiant et un prof.

On va se baser sur les INSERT INTO réalisés précédemment pour choisir un numéro qui définira un utilisateur.

Nous avons donc choisi :

- 644243 : M. CLERTANT, prof

- 12100111 : Darius Buzura, étudiant

On crée les utilisateurs dans bd_notes et on leur attribuera un mot de passe pour qu'ils puissent se connecter.

On leur attribut également un rôle avec IN ROLE :

```
CREATE user "644243" with password 'EuDS2+H' IN ROLE prof;  
CREATE user "12100111" with password '!jf7qAE' IN ROLE etudiant;
```

On peut voir qu'après l'exécution de ces commandes, on observe CREATE ROLE. Un utilisateur est aussi un rôle.

On peut maintenant définir des restrictions d'accès sur les rôles etudiant et prof.

Nous sommes toujours connecté en temps que postgres pour pouvoir attribuer les privilèges (GRANT).

- Un étudiant ne peut consulter que ses propres notes pour les ressources

On va créer une vue pour donner à l'étudiant à ses propres notes de ressources

```
psql -d bd_notes -U postgres
```

```
CREATE view notes_etudiant_R AS  
SELECT e_nom Nom, e_prenom Prenom, note, nom_R as ressource  
FROM Etudiant natural join Notes natural join Controle natural join  
Ressource WHERE num_etu = current_user::int  
ORDER BY (nom_R) ASC;
```

On autorise l'étudiant à voir ses notes (notes_etudiant_R)

```
grant SELECT on notes_etudiant_R to etudiant;
```

On se reconnecte en temps qu'étudiant :

```
psql -d bd notes -U "12100111"
bd_notes=> select * from notes;
ERREUR: droit refusé pour la table notes
bd_notes=> select * from notes_etudiant_r;
  nom  | prenom | note |
-----+-----+-----+
Buzura | Darius | 11.00 | Exploitation d une base de donnees
Buzura | Darius | 12.00 | Initiation au developpement
Buzura | Darius | 13.00 | Introduction a l architecture des ordinateurs
Buzura | Darius | 15.00 | Introduction aux services reseaux
Buzura | Darius | 18.00 | Methodes numeriques
(5 lignes)
```

- Un étudiant ne peut consulter que ses propres notes pour les SAE

On va créer une vue pour donner accès à l'étudiant à ses propres notes de SAE.

```
psql -d bd_notes -U postgres
```

```
CREATE view notes_etudiant_S AS
SELECT e_nom Nom, e_prenom Prenom, notesSAE as note , nom_sae as SAE
FROM Etudiant natural join Rendu natural join SAE
WHERE num_etu = current_user::int
ORDER BY (nom_sae) ASC;
```

On autorise l'étudiant à voir ses notes (notes_etudiant_S)

```
grant SELECT on notes_etudiant_S to etudiant;
```

On se reconnecte en temps qu'étudiant :

```
psql -d bd notes -U "12100111"
bd_notes=> select * from rendu;
ERREUR: droit refusé pour la table rendu
bd_notes=> select * from notes_etudiant_s;
  nom  | prenom | note |
-----+-----+-----+
Buzura | Darius | 17.00 | Installation de services reseau
(1 ligne)
```