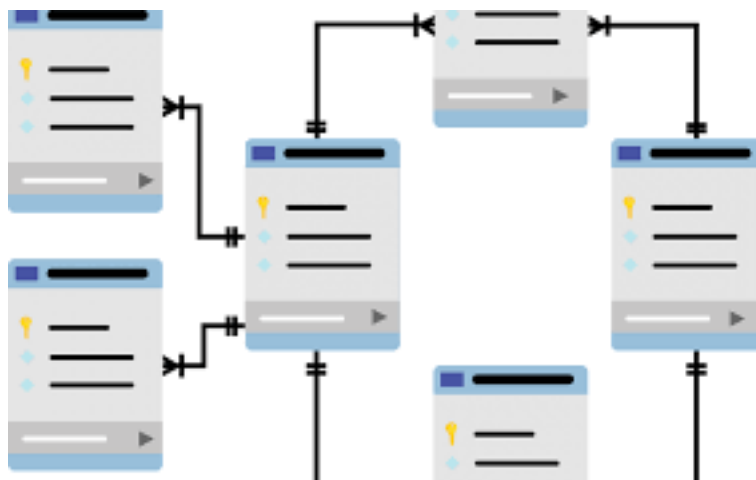


RAPPORT DE PROJET DE BASE DE DONNÉES

Conception de base de données d'un établissement d'éducation supérieure



Rand ASSWAD
Génie Mathématique

A l'attention de :
Mme. Nathalie Chaignaud

Contents

1	Introduction	2
1.1	Le projet en quelques mots	2
1.2	Le modèle de l'établissement	2
2	Conception	3
2.1	Modèle Entité/Association	3
2.2	Les relations entre les entités	4
2.3	Schéma relationnel	4
2.4	Schéma relationnel final	5
3	Implémentation	6
3.1	Environnement	6
3.2	Les requêtes	7

1 Introduction

1.1 Le projet en quelques mots

Le système d'éducation supérieure dans le monde a beaucoup évolué dans les dernières décennies à cause de plusieurs facteurs, notamment l'ouverture à l'international et les développements en technologie.

Dans le monde d'aujourd'hui, les établissements d'enseignement supérieur s'ouvrent de plus en plus à d'autres pays à travers des programmes d'échanges académiques et des projets en collaboration.

De plus, la technologie est de plus en plus présente dans l'éducation. Grâce à Internet, ils existent aujourd'hui des opportunités d'éducation en ligne, ce qui a donné naissance à des plateformes d'enseignement offrant des programmes diplômants.

Par conséquent, il a fallu développer des systèmes communs d'évaluation de compétences académiques. La solution adoptée par la majorité des établissements d'enseignement supérieur est le **crédit académique** qui est une unité de mesure de compétences obtenues liée au nombre d'heures passées pour acquérir ces compétences.

Ce système a permis les écoles et les universités de donner plus de libertés aux étudiants dans leurs parcours académiques.

Bien que ce système soit plus libre et démocratique, il est certainement plus complexe à mettre en place et nécessite un modèle bien défini.

Ce projet vise donc à modéliser une base de données relationnelles pour un établissement d'éducation supérieur *libre*, la base de données conçue est utilisable par un établissement scolaire réel ou une plateforme d'éducation complète. Ce modèle n'existe pas encore, mais s'inspire beaucoup du système existant adopté dans beaucoup de pays.

1.2 Le modèle de l'établissement

La philosophie principale de l'établissement est *la liberté de l'éducation*; dans le sens où les étudiants ont la possibilité de choisir leurs matières en toute liberté. L'établissement offre des formations en domaines différents (sciences de la nature et sciences sociales/humaines). La question qui se pose donc : *comment obtenir un diplôme*?

Les règles suivantes répondent à cette question :

- L'année est divisée en trois trimestres : l'automne (**AUT**), le printemps (**PRN**), et l'été (**ETE**). Il est possible de commencer ou terminer ses études n'importe quel trimestre.
- Chaque matière est associée à un certain nombre de crédits lié à la quantité de cours et de travail nécessaires pour la valider.
- Un diplôme est obtenu lors de la validation d'un certain nombre de crédits divisés en deux catégories : crédits essentiels et crédits complémentaires.
 - **Crédits essentiels** : Crédits à obtenir en validant des matières associées à ce programme. On dira que ces matières appartiennent à ce programme. Une matière peut appartenir à plusieurs programmes. L'étudiant n'est pas obligé de valider toutes les matières qui appartiennent à un programme pour obtenir le diplôme, il suffit d'obtenir le nombre minimum de crédits essentiels par ces matières.
 - **Crédits complémentaires** : Ce sont des crédits d'enrichissement personnel. Les étudiants obtiendront ces crédits en validant des matières de leurs choix (hors programme).
- Il n'y a pas de contraintes sur le nombre de crédits/matières qu'un(e) étudiant(e) obtient en un trimestre.
- Un(e) étudiant(e) peut suivre plusieurs programmes diplômants à la fois (ou aucun).
- Les étudiants ont la liberté de viser un ou plusieurs diplômes ou de juste suivre des matières pour s'enrichir.
- L'existence d'un quota pour chaque matière entraîne un critère de sélection des étudiants, ce critère est déterminé par plusieurs facteurs:
 - Les étudiants qui visent des diplômes contenant tel matière sont prioritaires.
 - Le nombre de fois qu'un(e) étudiant(e) ait suivi la matière antérieurement est un facteur négatif.
 - L'avis du conseiller est un facteur très important.
- Le choix de viser un diplôme est analogue au projet professionnel, il est donc dynamique et évolue au cours des trimestres. On donne alors aux étudiants la possibilité de changer de diplômes pendant la première année à chaque fin de trimestre, il est aussi possible de changer de diplôme plus tard sous réserve d'avoir au moins 50% de matières communes déjà validées.
- Certaines matières demandent la prédisposition de certaines notions exprimées en fonction d'autres matières dites **les prérequis** de la matière. Les matières sans prérequis sont dites des *matières élémentaires*.
- Afin de garder le niveau académique de l'établissement **les matières** sont catégorisées par **domaines** avec le poste du *responsable académique* qui veille sur le bon déroulement des matières de ce domaine. Il se peut que le responsable soient des enseignants.

2 Conception

2.1 Modèle Entité/Association

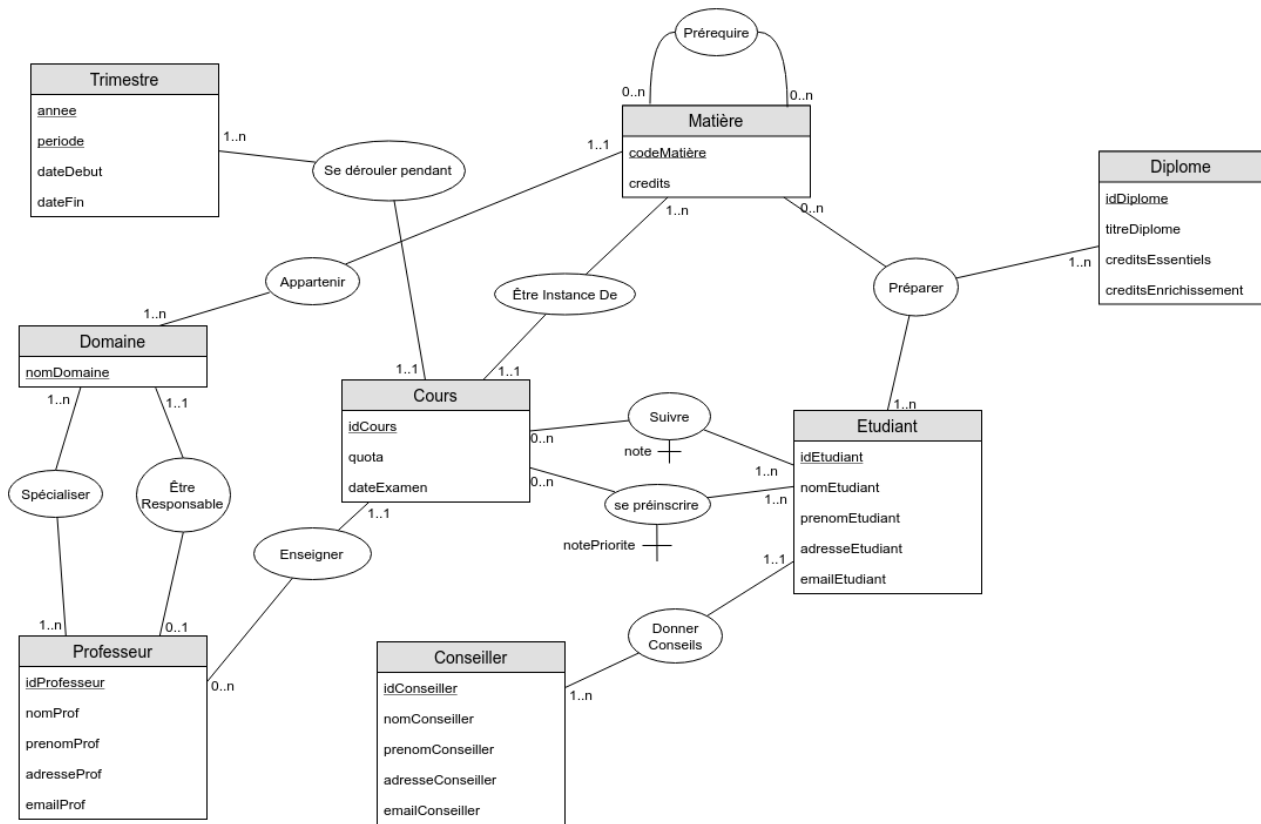


Figure 1: Modèle Entité/Association

Trimestre : Cette entité correspond à une période scolaire identifiée par une clé composée de 2 attributs : *année* + *période*. La période est l'une des trois : l'automne, le printemps ou l'été, il est possible d'associer une valeur numérique ou un code à chaque période afin de minimiser la taille de stockage. Un trimestre est décrit par sa date de début et sa date de fin.

Cours : L'entité cours représente une instance concrète d'une *Matière*. Elle est identifiée par la clé unique *idCours* et contient les attributs *quota* et *dateExamen*.

Professeur : Un professeur a un identifiant unique (*idProfesseur*) et ses informations de contact en tant qu'attributs.

Domaine : Un domaine est identifié par son nom car il est forcément unique (logiquement, si les domaines sont bien appelés ils portent des noms distinctes).

Matière : L'entité *Matière* a un identifiant unique et un certain nombre de *crédits*.

Diplôme : Un diplôme est identifié par la clé *idDiplome* et est caractérisé par les attributs *titreDiplome*, *creditsEssentiels*, et *creditsEnrichissement*.

Etudiant : Identifié par la clé *idEtudiant* et caractérisé par ses informations de contact.

Conseiller : Identifié par la clé *idConseiller* et caractérisé par ses informations de contact.

2.2 Les relations entre les entités

Relations binaires

Se dérouler pendant : Un cours se déroule pendant un trimestre définitivement, plusieurs cours se déroulent pendant un trimestre.

Enseigner : Un cours est enseigné par un professeur, un professeur peut enseigner plusieurs cours. Il est possible qu'un professeur n'enseigne pas des cours s'il est responsable académique.

Spécialiser : Un professeur est spécialisé dans un ou plusieurs domaines, dans un domaine spécifique il y a un ou plusieurs professeurs spécialisés.

Être responsable : Un professeur peut être le responsable académique dans un de ces domaines de spécialité, le domaine a un professeur comme responsable académique.

Appartenir : Une matière appartient à un domaine académique, et un domaine contient au moins une matière.

Prérequière : Une matière peut avoir une ou plusieurs matières prérequisées, et une matière peut être prérequisée par une ou plusieurs matières.

Être instance de : Un cours est une instance d'une matière sur un trimestre, une matière peut être instanciée plusieurs fois par plusieurs cours.

Suivre : Un étudiant suit un ou plusieurs cours et obtient une note dans chaque cours. Un cours peut être suivi par un ou plusieurs étudiants.

Se préinscrire : Un ou plusieurs étudiants se pré-inscrivent dans un cours, et en fonction d'une note de priorité en respectant le quota du cours un ou plusieurs étudiants suivent le cours. La liste de pré-inscrits peut contenir un ou plusieurs étudiants, s'il n'y a pas de pré-inscrits le cours ne s'ouvre pas.

Donner conseils : Un conseiller donne des conseils à un ou plusieurs étudiants concernant leur choix des matières, chaque étudiant est pris en conseil d'un conseiller.

Relation ternaire

Préparer :

- Une matière appartient à un ou plusieurs ou aucun programme diplômant, et un programme contient une ou plusieurs matières. (Dans le sens défini en Section 1.2)
- Un étudiant prépare un ou plusieurs ou aucun diplôme, un diplôme peut être préparé par un ou plusieurs étudiants.
- Un étudiant prépare une ou plusieurs matières, et une matière est préparée par un ou plusieurs ou aucun étudiant.

2.3 Schéma relationnel

On passe aux schémas relationnels à partir de notre modèle Entité/Association.

2.3.1 Entités

- Trimestre(**année**, **période**, dateDebut, dateFin)
- Cours(**idCours**, quota, dateExamen)
- Professeur(**idProfesseur**, nomProf, prenomProf, adresseProf, emailProf)
- Domaine(**nomDomaine**)
- Matière(**codeMatière**, crédits)
- Etudiant(**idEtudiant**, nomEtudiant, prenomEtudiant, adresseEtudiant, emailEtudiant)
- Conseiller(**idConseiller**, nomConseiller, prenomConseiller, adresseConseiller, emailConseiller)
- Diplome(**idDiplome**, titreDiplome, creditsEssentiels, creditsEnrichissement)

2.3.2 Associations

Associations binaires 1..n

- Cours(**idCours**, quota, dateExamen, codeMatiere, annee, periode, idProfesseur)
- Domaine(**nomDomaine**, idProfRespo)
- Matiere(**codeMatiere**, crédits, nomDomaine)
- Etudiant(**idEtudiant**, nomEtudiant, prenomEtudiant, adresseEtudiant, emailEtudiant, idConseiller)

Associations binaires n..n

- Specialiser(**idProfesseur**, **nomDomaine**)
- Prerequisite(**codeMatiere**, **codePrerequise**)
- SePreinscrire(**idCours**, **idEtudiant**)
- Suivre(**idCours**, **idEtudiant**)

Association ternaire

- Preparer(**idEtudiant**, **codeMatiere**, **Diplome**)

2.3.3 Dénormalisation

- La relation **Trimestre** peut être supprimée en la déplaçant vers la relation **Cours** car elle porte peu d'attributs et elle résulte d'une entité faible.
- La relation **Domaine** peut être supprimée facilement en la déplaçant vers les relations **Professeur** et **Cours**.

On est tenté de supprimer la relation **Conseiller** car elle est une entité faible et n'est reliée qu'à l'entité **Etudiant**. Néanmoins, il est très important d'avoir un tableau existant dans la base de donnée pour les personnes (Etudiants, Professeurs, Conseillers) sans avoir à faire des requêtes pour obtenir tel liste, je fais donc le choix de garder cette relation.

2.3.4 Modification sur le pré-rapport

Quand j'ai commencé la deuxième partie du projet, j'ai reconsidéré les choix que j'avais effectué, j'expliquerais brièvement les changements que j'ai effectué:

- Dans la dénormalisation, j'avais supprimé la relation **Matiere** car elle est instanciée par la relation **Cours**, finalement j'ai préféré la garder afin de distinguer ces deux relations et de rendre les tableaux plus lisibles.
- Pour simplifier, j'ai renommé les entités qui portent des noms comme *prenomConseiller* en *prenom*.
- J'ai fusionné les deux relations **Préinscription** et **Suivre** dans la relation **InscriptionCours** qui fait l'affaire.
- J'ai remplacé l'association ternaire par deux associations binaires plus simples, car on n'utilise jamais les deux relations dans la même requête. En effet, la relation ternaire était un peu forcée.

2.4 Schéma relationnel final

- Cours(**code**, matiere, quota, quotaExamen, dateDebut, dateExamen, enseignant, annee, periode)
- Professeur(**id**, nom, prenom, adresse, email, dateNaissance)
- Etudiant(**id**, nom, prenom, adresse, email, dateNaissance, idConseiller)
- Conseiller(**id**, nom, prenom, adresse, email, dateNaissance)
- Diplome(**id**, titre, niveau, creditsEssentiels, creditsEnrichissement)
- PreRequisition(**matiere**, matiereRequise)
- InscriptionCours(**cours**, **etudiant**, suivreCours, inscrit, noteConseiller, noteObtenue)
- Preparer(**etudiant**, **diplome**)
- Contenir(**diplome**, **matiere**)

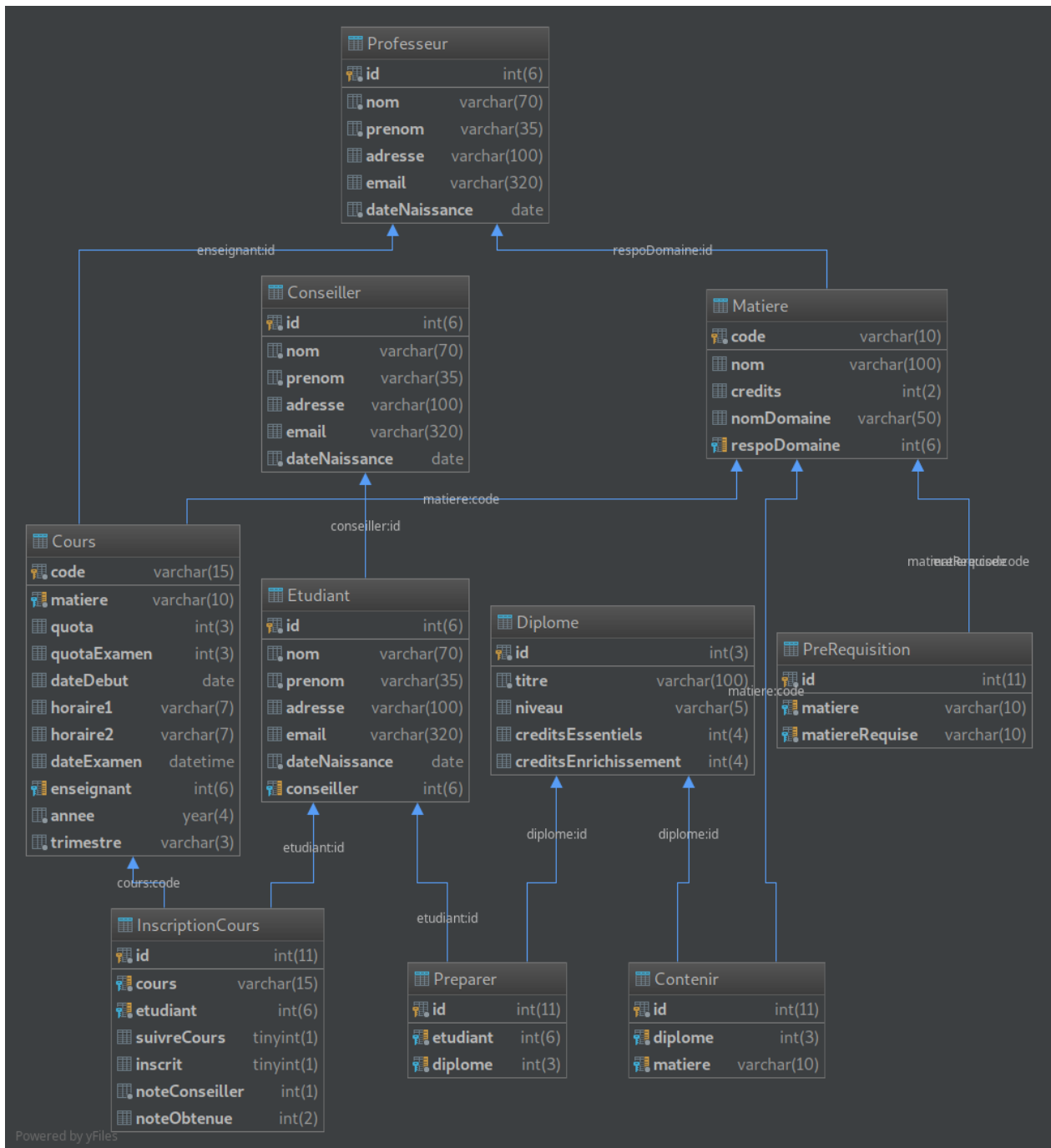


Figure 2: Schéma Relationnel

3 Implémentation

3.1 Environnement

Pour des raisons de portabilité, j'ai sauvegardé les scripts SQL permettant de créer la base de donnée, et la populer par les tableaux, des données aléatoires pour tester et les requêtes dans ce rapport (*repertoire: db*).

Pour faciliter la connection, j'ai créé des scripts shell qui lancent les requêtes SQL à partir du terminal (*repertoire: bin*).

Les tableaux dans ce rapport sont compilé en directe à l'aide des libraries R **knitr** et **DBI**.

Le code source du projet y compris ce rapport est disponible sur: github.com/rand-asswad/gm4_bdd_univ

Le rapport est aussi disponible en format HTML (dynamique) sur: rand-asswad.github.io/gm4_bdd_univ/rapport/index.html

Connection à la base de données:

```
library(DBI)
univ <- dbConnect(RMariaDB::MariaDB(), dbname="univ", username="root")
dbListTables(univ)

## [1] "Conseiller"      "Contenir"        "Cours"
## [4] "Diplome"         "Etudiant"        "InscriptionCours"
## [7] "Matiere"         "PreRequisition"  "Preparer"
## [10] "Professeur"      "Valide"          "matieresDiplome"
## [13] "matieresEtudiant"
```

3.2 Les requêtes

3.2.1 Les vues

Les tableaux de la base de données sont permettent d'obtenir toutes les informations nécessaires par des requêtes, il n'y a donc aucun besoin de d'avoir d'autres tableaux. En revanche, il est certainement pratique d'avoir des **vues** qu'on utilisera souvent dans nos requêtes, par exemple une liste des matières validés (par étudiant).

```
DROP VIEW IF EXISTS Valide;
```

```
CREATE VIEW Valide AS
SELECT Etudiant.id as etudiant, Cours.matiere
FROM InscriptionCours
INNER JOIN Cours ON InscriptionCours.cours = Cours.code
INNER JOIN Etudiant ON InscriptionCours.etudiant = Etudiant.id
WHERE noteObtenue >= 10;
```

En rajoutant quelques colonnes à cette vue, on obtient:

```
SELECT Etudiant.nom, Etudiant.prenom, Matiere.code, Matiere.nom, Matiere.credits
FROM Valide
INNER JOIN Etudiant ON Valide.etudiant = Etudiant.id
INNER JOIN Matiere ON Valide.matiere = Matiere.code
```

Table 1: Displaying records 1 - 10

nom	prenom	code	nom..4	credits
Jacquet	Anaëlle	Comb2	Optimisation combinatoire	4
Jacquet	Anaëlle	Elec0	Electromagnétisme et courant alternatif	2
Jacquet	Anaëlle	EnB22	Anglais niveau B2.2	1
Jacquet	Anaëlle	LinAlg2	Algèbre Linéaire – approfondissement	3
Jacquet	Anaëlle	Mesure0	Théorie des ensembles	1
Jacquet	Anaëlle	Proba0	Initiation à la probabilité (discrète)	1
Jacquet	Anaëlle	Techno1	Systèmes d'exploitations, logiciels (culture générale)	1
Jacquet	Anaëlle	Thermo1	Modèles thermodynamiques	2
Jacquet	Anaëlle	Ana0	Analyse (suites et fonctions de variables réelles en dimension 1)	1
Jacquet	Anaëlle	Complex1	Analyse complexe	3

Voici une vue de diplômes et le nombre de matières qu'elles contiennent.

```
DROP View IF EXISTS matieresDiplome;
```

```
CREATE VIEW matieresDiplome AS
SELECT Diplome.id, COUNT(*) AS nbMatiere
FROM Contenir
INNER JOIN Diplome on Contenir.diplome = Diplome.id
GROUP BY Contenir.diplome;
```

Afin de mieux visualiser cette vue, on rajoutes quelques colonnes du tableau *Diplome*.


```
SELECT Diplome.titre, Diplome.niveau, matieresDiplome.nbMatiere, Diplome.creditsEss
FROM matieresDiplome
INNER JOIN Diplome ON matieresDiplome.id = Diplome.id;
```

Table 2: Displaying records 1 - 10

titre	niveau	nbMatiere	creditsEssentiels
Mathématiques fondamentales et appliquées	BSc	56	150
Mathématiques et sciences de données	BSc	57	150
Informatique	BSc	57	150
Ingénierie de logiciel	BSc	59	150
Mathématiques et sciences de données	BSc	59	150
Physiques appliquées	BSc	58	150
Génie électronique	BEng	57	150
Génie énergétique	BEng	55	150
Génie généraliste	BEng	58	150
Analyse de signal	MEng	41	105

3.2.2 Les sélections

Les sélections sont spécifiques à une ligne d'un tableau, donc dans le code source j'ai créé des *procédures* qui prennent la clé primaire en entrée pour effectuer ces requêtes.

Dans ce rapport, j'ai testé avec les valeurs suivantes:

```
idEtudiant <- 1
idDiplome <- 5
domaine <- "Combinatoires"
codeCours <- "AUT18Ana01"
```

On sélectionnent les matières qu'un étudiant pourra prendre. C'est-à-dire, les matières que l'étudiant n'a pas encore validées *et* il/elle a déjà validé ses matières requises.

```
SELECT Matiere.code, Matiere.nom, Matiere.credits
FROM Matiere
INNER JOIN PreRequisition ON Matiere.code = PreRequisition.matiere
WHERE Matiere.code NOT IN (SELECT matiere FROM Valide WHERE Valide.etudiant = ?i
AND PreRequisition.matiereRequise = ANY (SELECT matiere FROM Valide WHERE Valide
```

Table 3: Displaying records 1 - 10

code	nom	credits
Ana1	Analyse (suites et fonctions de variables réelles en dimension n)	2
Complex2	Analyse complexe – approfondissement	4
Data3	Big data – avancé	4
DevWeb0	Créations de sites pour les non informaticiens	1
Elec1	Electromagnétisme avancé	3
Fluid2	Turbulances, dyphasique, etc	4
Fourier1	La famille de transformations de fourier	3
Fourier2	Approfondissement dans l'analyse de fourier	4
Meca0	Physique Newtonienne (base de mécanique du point + applications)	1
Mesure1	Théorie de la mesure	3

On sélectionnent les cours qu'un étudiant peut prendre cette année.

```
SELECT Cours.code, Cours.matiere, Matiere.nom, Matiere.credits, Cours.trimestre, Co
FROM Cours
INNER JOIN Matiere ON Cours.matiere = Matiere.code
INNER JOIN PreRequisition ON Cours.matiere = PreRequisition.matiere
```

```
WHERE Matiere.code NOT IN (SELECT matiere FROM Valide WHERE Valide.etudiant = ?id)
AND PreRequisition.matiereRequis = ANY (SELECT matiere FROM Valide WHERE Valide.etudiant = ?id)
AND Cours.annee = Year(curdate());
```

Table 4: Displaying records 1 - 10

code	matiere	nom	credits	trimestre	ens
ETE18Elec11	Elec1	Electromagnétisme avancé	3	ETE	
PRN18Optique11	Optique1	Optique ondulatoires	3	PRN	
AUT18Fourier21	Fourier2	Approfondissement dans l'analyse de fourier	4	AUT	
ETE18Fourier21	Fourier2	Approfondissement dans l'analyse de fourier	4	ETE	
AUT18Mesure11	Mesure1	Théorie de la mesure	3	AUT	
PRN18Mesure11	Mesure1	Théorie de la mesure	3	PRN	
AUT18Topo01	Topo0	Initiation à la topologie (générale et algébrique)	2	AUT	
ETE18Topo01	Topo0	Initiation à la topologie (générale et algébrique)	2	ETE	
PRN18Topo01	Topo0	Initiation à la topologie (générale et algébrique)	2	PRN	
AUT18Proba11	Proba1	Probabilité discrète et continue, var aléatoires, loi des grandes nombres	2	AUT	

```
SELECT Diplome.titre, Diplome.niveau, COUNT(1) as nbValide, matieresDiplome.nbMatiere
FROM Contenir
INNER JOIN Diplome ON Contenir.diplome = Diplome.id
INNER JOIN Matiere ON Contenir.matiere = Matiere.code
INNER JOIN Valide ON Matiere.code = Valide.matiere
INNER JOIN matieresDiplome ON Diplome.id = matieresDiplome.id
WHERE Valide.etudiant = ?idEtudiant
GROUP BY Contenir.diplome
ORDER BY nbValide DESC;
```

Table 5: Displaying records 1 - 10

titre	niveau	nbValide	nbTotal
Génie généraliste	BEng	18	58
Technicien en informatique	Tech	17	39
Mathématiques et sciences de données	BSc	16	59
Génie de modélisation et calculs	MEng	13	41
Mathématiques fondamentales et appliquées	BSc	13	56
Mathématiques fondamentales et appliquées	MSc	13	42
Génie électronique	BEng	12	57
Mathématiques et sciences de données	BSc	12	57
Analyse de signal	MEng	12	41
Informatique	BSc	11	57

```
SELECT Matiere.code, Matiere.nom, Matiere.credits
FROM Contenir
INNER JOIN Matiere ON Contenir.matiere = Matiere.code
WHERE Matiere.code NOT IN (SELECT Valide.matiere FROM Valide WHERE Valide.etudiant = ?idEtudiant)
AND Contenir.diplome = ?idDiplome;
```

Table 6: Displaying records 1 - 10

code	nom	credits
Ana1	Analyse (suites et fonctions de variables réelles en dimension n)	2
Ana2	Equation différentielles ordinaires	3
Comb1	Combinatoires avancées	3
Comp0	Automates, langages, regex	4
Comp1	Langages et compilateurs	4

code	nom	credits
Comp2	Compilateurs et assembleur	4
Control1	Théorie de contrôle	3
Control2	Système dynamiques, théorie de contrôle	3
Data0	Base de données et SQL	2
Data1	Base de données avancé	3

```

SELECT Etudiant.id, Etudiant.prenom, Etudiant.nom, InscriptionCours.noteConseiller,
FROM InscriptionCours
INNER JOIN Etudiant ON InscriptionCours.etudiant = Etudiant.id
WHERE InscriptionCours.cours = ?codeCours
ORDER BY InscriptionCours.noteConseiller DESC;

```

Table 7: Displaying records 1 - 10

id	prenom	nom	noteConseiller	suivreCours
92	Diego	Chevalier	9	1
77	Alexia	Rousseau	8	1
74	Françoise	Olivier	8	1
45	Constant	Laurent	8	1
40	Kylian	Meunier	7	1
65	Mélissa	Pierre	7	0
37	Baptiste	Garcia	6	0
88	Alexis	Rolland	6	0
68	Constant	Le goff	6	1
13	Félix	Renaud	5	0

```

SELECT DISTINCT Professeur.id, Professeur.nom, Professeur.prenom
FROM Cours
INNER JOIN Professeur ON Cours.enseignant = Professeur.id
INNER JOIN Matiere ON Cours.matiere = Matiere.code
WHERE Matiere.nomDomaine = ?domaine;

```

Table 8: 4 records

id	nom	prenom
18	Humbert	Capucine
17	Noel	Lauriane
16	Roger	Élouan
15	Guillot	Antoine