

Projet 9

Présenté par Mathieu Morin, Patrice Dumontier-Houle, Guillaume Riou et Nicolas Hurtubise

Introduction

Le projet que nous avons mené à terme consiste en un système de questions à choix multiples, pouvant être adapté à élèves de tous les niveaux. Notre projet vise à rassembler des connaissances pertinentes à la réussite d'un cours afin que les étudiants puissent tester leurs connaissances de manière efficace et interactive.

Notre système est conçu pour accumuler de l'information à propos des réponses des étudiants aux questions de manière à ce qu'un professeur puisse cibler la matière moins comprise. En utilisant cet outil, les professeurs pourront structurer leurs cours en fonctions des difficultés des élèves de manière à passer moins de temps à réviser les concepts les mieux assimilés, et plus de temps à focaliser sur les sujets que les élèves saisissent le moins.

Modèle Entité Relation

Le Modèle Entité Relation de notre projet comporte l'entité *professeur*, qui est identifiée uniquement par son id. Ses propriétés sont son prénom, son nom et son mot de passe. Un professeur peut donner 0 ou plusieurs cours et a accès à une section spéciale du site en se logguant avec ses identifiants.

Une entité *cours* est représentée par son sigle (par exemple IFT2935) et a comme attribut un nom de cours.

Chaque cours est subdivisé en parties de cours (entité *partie_cours*), qui sont identifiées par leur id et qui ont comme informations le nom de la partie (name).

Chaque entité *question* est associée à une partie de cours. Une question a un id comme identifiant et a contient son énoncé (content), le nombre de fois qu'une personne a répondu correctement à la question (success) et le nombre de fois qu'une personne a répondu incorrectement à la question (failures). Ces nombres sont particulièrement utiles pour les professeurs désirant faire des statistiques sur les sujets les mieux compris et les plus difficilement compris par les élèves.

Les entités *réponse* consistent en des choix de réponse offerts à une question donnée et sont identifiées par un id. Chaque réponse contient l'énoncé de réponse (texte), ainsi qu'une valeur de vérité indiquant s'il s'agit d'une réponse valide ou non (0 pour faux, 1 pour vrai).

De plus, chaque question est classifiée par sa catégorie, qui est identifiée par son id, et qui a comme information son nom (name).

Les catégories sont liées à elles-mêmes via une relation N vers 1, permettant ainsi une hiérarchie de catégories et de sous-catégories. Par exemple, on pourra avoir la catégorie "Normalisation des bases de données", qui contiendra le sujet plus spécifique "Formes normales", lui-même composé des sous-sous-catégories "1FN", "2FN", "3FN", etc.

Modèle Relationnel

Le modèle relationnel de notre projet est calqué sur notre Modèle Entité Relation, mais est beaucoup plus proche du modèle utilisé par notre application.

D'un point de vue du SQL, nos définitions de relations sont les suivantes :

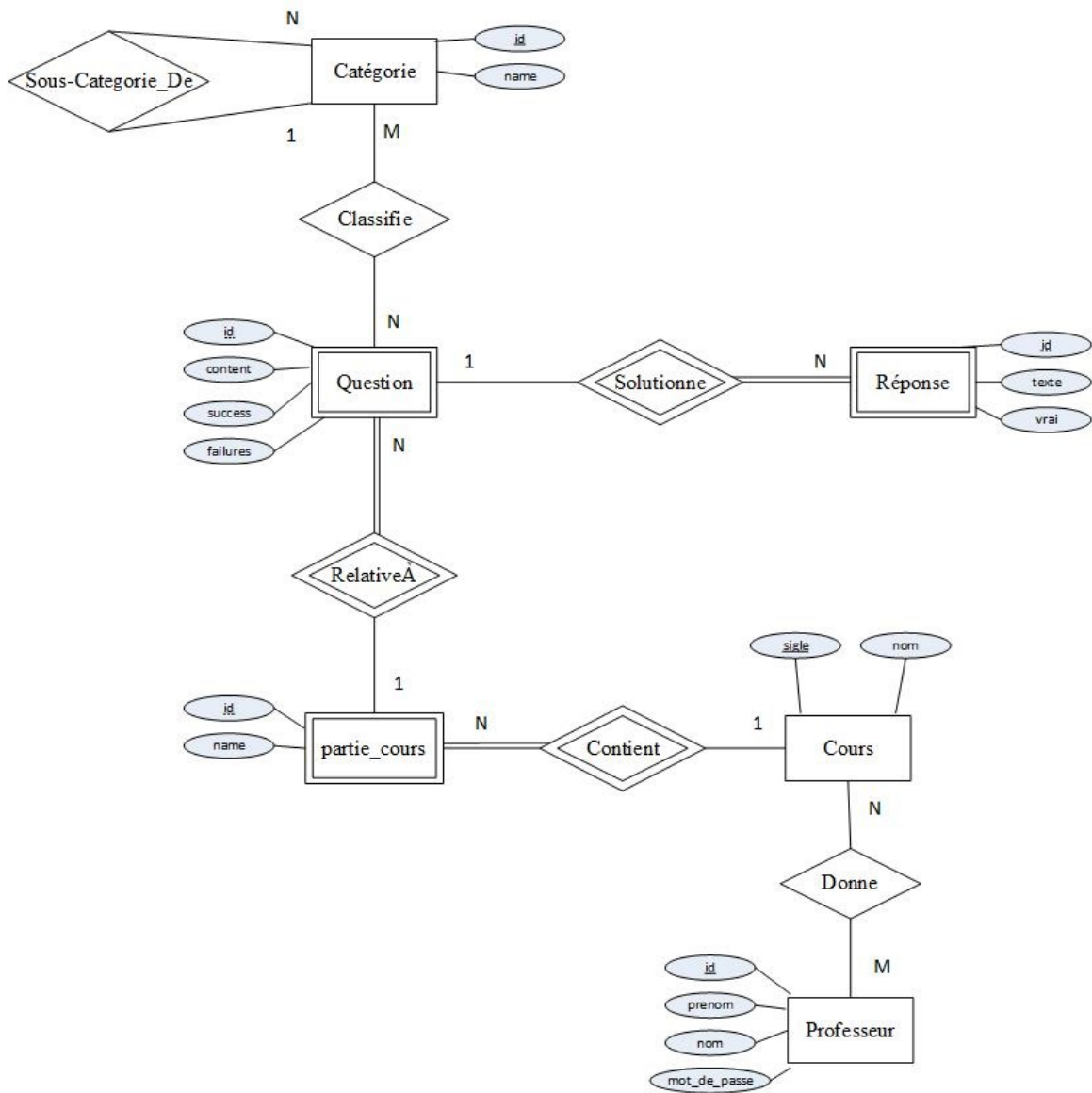


Figure 1: Diagramme Entité Relation

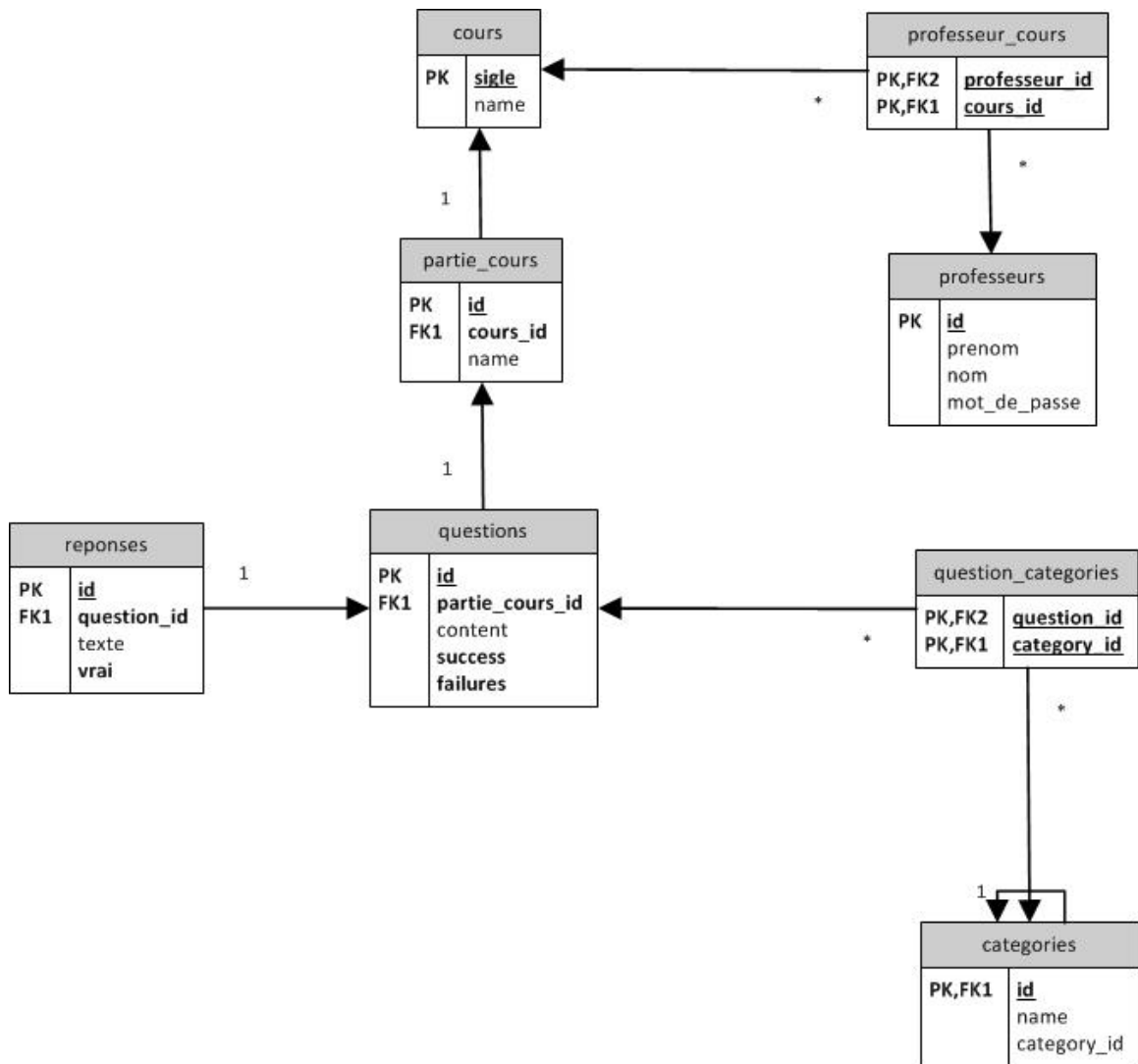


Figure 2: Diagramme Relationnel

```

-- Professeurs
CREATE TABLE professeurs(
    id number(10) NOT NULL,
    prenom char(30),
    nom char(30),
    mot_de_passe char(30),
    PRIMARY KEY(id)
);

-- Cours
CREATE TABLE cours(
    sigle char(7) NOT NULL,
    name varchar(127) NOT NULL,
    PRIMARY KEY(sigle)
);

-- Cours donnés par un professeur
CREATE TABLE professeur_cours(
    professeur_id number(10) NOT NULL,
    cours_id char(7) NOT NULL,
    FOREIGN KEY(professeur_id) REFERENCES professeurs(id),
    FOREIGN KEY(cours_id) REFERENCES cours(sigle),
    PRIMARY KEY (professeur_id, cours_id)
);

-- Partie de cours = Chapitre/section dans un cours
CREATE TABLE partie_cours(
    id number(10) NOT NULL,
    name varchar(127),
    cours_id char(7) NOT NULL,
    FOREIGN KEY(cours_id) REFERENCES cours(sigle),
    PRIMARY KEY(id)
);

-- Questions à choix multiples
CREATE TABLE questions(
    id number(10) NOT NULL,
    content varchar(127) NOT NULL,
    partie_cours_id number(10) NOT NULL,
    success number(10) DEFAULT 0 NOT NULL,
    failures number(10) DEFAULT 0 NOT NULL,
    FOREIGN KEY(partie_cours_id) REFERENCES partie_cours(id),
    PRIMARY KEY(id)
);

-- Réponses des questions
CREATE TABLE reponses(
    id number(10) NOT NULL,
    texte varchar(127),
    question_id number(10) NOT NULL,
    vrai number(1) NOT NULL,

```

```

        FOREIGN KEY(question_id) REFERENCES questions(id),
        PRIMARY KEY(id)
    );

-- Une catégorie peut avoir des sous-catégories de façon récursive
CREATE TABLE categories(
    id number(10) NOT NULL,
    name varchar(127),
    category_id number(10) DEFAULT NULL,
    FOREIGN KEY(category_id) REFERENCES categories(id),
    PRIMARY KEY(id)
);

-- Associations entre catégories et questions
CREATE TABLE question_categories(
    question_id number(10) NOT NULL,
    category_id number(10) NOT NULL,
    FOREIGN KEY(question_id) REFERENCES questions(id),
    FOREIGN KEY(category_id) REFERENCES categories(id),
    PRIMARY KEY (question_id, category_id)
);

```

L'Application

L'application est divisée en sections suivant les fonctionnalités suivantes :

1. Réponse à des questions aléatoires (qui peuvent être tirées au sort parmi les catégories, les cours, les parties de cours, etc.)
2. Recherche de questions et de concepts dans la base de données
3. Visualisation de statistiques générales
4. Visualisation de statistiques propres à chaque cours (page protégée en accès, restreinte aux professeurs concernés)

1. Réponse à des questions aléatoires

La principale fonctionnalité de l'application est de permettre aux élèves de tirer au sort des questions relatives à des concepts vus en cours pour valider leur compréhension partielle ou totale. Les élèves peuvent donc choisir de répondre aléatoirement :

1. À des questions relatives à une partie de cours
2. À des questions tirées de l'intégrité d'un cours
3. À des questions spécifiquement reliées à un des concept de la matière

Lorsque le choix est fait, l'élève se voit présenté une question et des choix de réponse. Il peut alors sélectionner le choix qui lui semble juste et confirmer sa réponse. L'application affiche alors si la question a été réussie (en vert) ou échouée (en rouge), note le score dans le navigateur web de l'élève pour futures références et permet de passer à une autre question dans le même thème.

project9	Accueil	Stats	Professeurs	🏆 Mon Score :0%	🔄 Reset	
----------	---------	-------	-------------	-----------------	---------	--

Questions aléatoires à choix multiples

Par cours

🎓 Bases de données (IFT2935)

- Première partie : Introduction
- Deuxième partie : Modèle Entité Relation
- Troisième partie : Modèle Relationnel
- Quatrième partie : L'algèbre et le calcul Relationnel

» Toutes les questions du cours

🎓 Programmation (IFT1235)

Par catégorie

- 📁 Normalisation des bases de données
 - 📁 Formes normales
- 📁 Les modèles de données
 - 📁 Modèle entité-relation
 - 📁 Modèle relationnel
- 📁 Questionnement
 - 📁 QCM
- 📁 Syntaxe
 - 📁 Expression Conditionnelles
- 📁 Programmation Orientée Objet
 - 📁 Méthodes
 - 📁 Objets

Rechercher des questions

Entrez les mots-clés qui vous intéressent

ex.: algèbre relationnel



Afficher seulement les résultats contenant tous les mots-clés

Figure 3: Sélection d'un type de questions aléatoires : partie de cours, intégrité d'un cours ou catégories et concepts

project9	Accueil	Stats	Professeurs	Mon Score : 57%	Reset
----------	---------	-------	-------------	-----------------	-------

Quel est le symbol pour l'opération de SÉLECTION ?

☐ Sigma

☐ Pi

Confirmer



Veuillez sélectionner une réponse

project9 - 2015

Figure 4: Affichage d'une question aléatoire


project9	Accueil	Stats	Professeurs	Mon Score : 59%	Reset
----------	---------	-------	-------------	-----------------	-------

Quel est le symbol pour l'opération de SÉLECTION ?

☒ Sigma

☐ Pi

Confirmer



Yay ! Vous avez choisi la bonne réponse !

Prochaine question →

project9 - 2015

Figure 5: Lorsque la réponse est sélectionnée, l'application confirme s'il s'agit de la bonne réponse

2. Recherche de questions et de concepts dans la base de données

En plus de choisir des concepts via une liste de cours et de catégories, il semblait intéressant d'ajouter la possibilité de rechercher dans la base de données en vue de trouver plus facilement des sujets spécifiques.

The screenshot shows the 'project9' application interface. At the top, there is a navigation bar with links: 'Accueil', 'Stats', 'Professeurs', 'Mon Score : 59%', and 'Reset'. Below this, there is a section titled 'Rechercher des questions'. It contains a search input field with the text 'algèbre relationnel'. Below the input field is a toggle switch labeled 'Afficher seulement les résultats contenant tous les mots-clés'. There are four panels displaying search results: 'Questions trouvées', 'Catégories trouvées', 'Cours trouvés', and 'Parties de cours trouvées'. The 'Questions trouvées' panel shows one result: '1 Nommez un types de contrainte d'intégrité relationnelle'. The 'Parties de cours trouvées' panel shows two results: '1 Troisième partie : Modèle Relationnel' and '2 Quatrième partie : L'algèbre et le calcul Relationnel'.

Figure 6: Recherche d'un sujet dans la base de données

On peut donc rechercher un ou plusieurs termes parmi les énoncés de questions, les catégories, les cours et les parties de cours. L'option de raffiner la recherche selon les mots-clés permet de rechercher parmi les éléments contenant *l'un des mots-clés entrés* (via un **OR SQL**), ou encore parmi les éléments contenant *tous les mots-clés entrés* (via un **AND SQL**).

3. Visualisation de statistiques générales

La section *Stats*, accessible depuis le menu du haut, permet de donner une vue d'ensemble de la base de données, notamment en ce qui a trait aux cours et aux professeurs utilisant le système, ainsi qu'à des statistiques personnalisées.

Grâce à cette section, les élèves pourront visualiser leur taux de succès et adapter leurs efforts en conséquence.

4. Visualisation de statistiques propres à chaque cours

La section *Professeurs* permet de visualiser des statistiques propres à chaque cours et est réservée aux professeurs authentifiés.

Les professeurs souhaitant avoir une idée de l'état de l'apprentissage de leurs élèves devront donc se logger dans le système en indiquant leur Prénom, Nom et Mot de passe secret.

Une fois authentifiés, les professeurs verront dans l'ordre :

project9

Accueil

Stats

Professeurs

Mon Score :59%

Reset

Méthodes

Objets

Rechercher des questions

Entrez les mots-clés qui vous intéressent

algèbre relationnel

☒

Afficher seulement les résultats contenant tous les mots-clés

Questions trouvées

Catégories trouvées

Cours trouvés

Parties de cours trouvées

1

Quatrième partie :

L'algèbre et le calcul Relationnel

project9 - 2015

Figure 7: Il est possible de faire une recherche plus précise en demandant à utiliser tous les termes de recherche

1. Le taux de réussite global pour toutes leurs questions
2. Le taux de réussite pour chacune de leurs questions

Les questions sont ordonnées par cours ainsi que par taux de réussite. Il est ainsi facile de cibler rapidement quels concepts sont bien assimilés et quels concepts donnent du fil à retordre aux élèves.

Données utilisées

Les données réalistes utilisées pour des fins de test sont insérées dans la base de données de la façon suivante :

-- Valeurs de test --

```

INSERT INTO categories VALUES(1, 'Normalisation des bases de données', NULL);
INSERT INTO categories VALUES(2, 'Formes normales', 1);
INSERT INTO categories VALUES(3, 'Les modèles de données', NULL);
INSERT INTO categories VALUES(4, 'Modèle entité-relation', 3);
INSERT INTO categories VALUES(5, 'Modèle relationnel', 3);
INSERT INTO categories VALUES(6, 'Algèbre relationnel', 5);
INSERT INTO categories VALUES(9, 'Questionnement', NULL);
INSERT INTO categories VALUES(10, 'QCM', 9);
INSERT INTO categories VALUES(11, 'Bases de données', NULL);
INSERT INTO categories VALUES(15, 'Syntaxe', NULL);
INSERT INTO categories VALUES(16, 'Expression Conditionelles', 15);
INSERT INTO categories VALUES(17, 'Programmation Orientée Objet', NULL);
INSERT INTO categories VALUES(18, 'Méthodes', 17);
INSERT INTO categories VALUES(19, 'Objets', 17);

```

project9	Accueil	Stats	Professeurs	🏆 Mon Score :59%	↺ Reset	
-----------------	---------	-------	-------------	------------------	---------	--

Statistiques

Statistiques du site

Réponses par cours

Nom	Sigle	Nombre de réponses
Programmation	IFT1235	19
Bases de données	IFT2935	18

Moyenne de questions par cours

Nombre moyen
8.5

Nombre de questions par professeur

Prénom	Nom	Nombre de questions
Claude	Frasson	9
Esma	Aïmeur	9

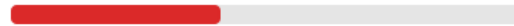
Mes statistiques



Taux de réussite : 59%



Nombre de bonne réponses : 13



Nombre d'échecs : 9

C'est bien...

Vous êtes sur une bonne voie, continuez à travailler sur vos cours et vos notes en seront récompensées !

Figure 8: Les statistiques affichées permettent d'avoir une idée d'ensemble de la réussite ainsi que de l'état du site

project9	Accueil	Stats	Professeurs	🏆 Mon Score :57%	↺ Reset	
----------	---------	-------	-------------	------------------	---------	--

Login

Prénom

Nom

Mot de passe

project9 - 2015

Figure 9: Page de login pour les professeurs

project9	Accueil	Stats	Professeurs	🏆 Mon Score :57%	↺ Reset	
----------	---------	-------	-------------	------------------	---------	--

Bonjour, Claude Frasson !

Vos cours

Sigle	Nom	Nombre de questions
IFT2935	Bases de données	9

Vos collègues

Professeur	Sigle	Cours
Esma Aïmeur	IFT2935	Bases de données

Mes questions



Taux de réussite moyen des questions: 57 %

Taux de réussite par question

IFT2935 - Bases de données



Qu'est-ce qu'une base de données ? (100 % sur 1 réponses)



Quel est le symbol pour l'opération de SÉLECTION ? (100 % sur 1 réponses)



Pourquoi a-t-on besoin de bases de données ? (50 % sur 4 réponses)

Figure 10: Page de statistiques réservée aux professeurs

```

INSERT INTO professeurs VALUES(1, 'Claude', 'Frasson', 'ift2935');
INSERT INTO professeurs VALUES(5, 'Denis', 'Bluteau', 'ift1235');
INSERT INTO professeurs VALUES(6, 'Anna', 'Francoeur', 'ift1235');
INSERT INTO professeurs VALUES(4, 'Esma', 'Aïmeur', 'ift2935');

INSERT INTO cours VALUES('IFT2935', 'Bases de données');
INSERT INTO cours VALUES('IFT1235', 'Programmation');

INSERT INTO professeur_cours VALUES(1, 'IFT2935');
INSERT INTO professeur_cours VALUES(4, 'IFT2935');
INSERT INTO professeur_cours VALUES(5, 'IFT1235');
INSERT INTO professeur_cours VALUES(6, 'IFT1235');
INSERT INTO professeur_cours VALUES(4, 'IFT1235');

INSERT INTO partie_cours VALUES(1, 'Première partie : Introduction', 'IFT2935');
INSERT INTO partie_cours VALUES(2, 'Deuxième partie : Modèle Entité Relation',
'IFT2935');
INSERT INTO partie_cours VALUES(3, 'Troisième partie : Modèle Relationnel',
'IFT2935');
INSERT INTO partie_cours VALUES(4, 'Quatrième partie : L''algèbre et le calcul
Relationnel', 'IFT2935');
INSERT INTO partie_cours VALUES(10, 'Partie 1 : Introduction', 'IFT1235');
INSERT INTO partie_cours VALUES(11, 'Partie 2 : Structures de contrôle',
'IFT1235');
INSERT INTO partie_cours VALUES(12, 'Partie 3 : Les procédures', 'IFT1235');
INSERT INTO partie_cours VALUES(13, 'Partie 4 : Programmation orientée objet',
'IFT1235');

INSERT INTO questions(id, content, partie_cours_id) VALUES(1, 'Qu''est-ce
qu''une base de données ?', 1);
INSERT INTO questions(id, content, partie_cours_id) VALUES(2, 'Pourquoi a-t-on
besoin de bases de données ?', 1);
INSERT INTO questions(id, content, partie_cours_id) VALUES(3, 'Qu''est-ce
qu''un attribut clé ?', 2);
INSERT INTO questions(id, content, partie_cours_id) VALUES(4, 'Qu''est-ce
qu''un attribut composite ?', 2);
INSERT INTO questions(id, content, partie_cours_id) VALUES(5, 'Nommez un type
de contrainte d''intégrité relationnelle', 3);
INSERT INTO questions(id, content, partie_cours_id) VALUES(6, 'Quel est le
symbol pour l''opération de SÉLECTION ?', 3);
INSERT INTO questions(id, content, partie_cours_id) VALUES(7, 'Quel est
l''utilité de la theta jointure ?', 4);
INSERT INTO questions(id, content, partie_cours_id) VALUES(8, 'Existe-t-il un
opérateur nommé sesqui-jointure ?', 4);
INSERT INTO questions(id, content, partie_cours_id) VALUES(9, 'La 2-forme
normale consiste à n''avoir aucun attribut non clé dépendant d''une partie de
la clé', 1);

```

```

INSERT INTO questions(id, content, partie_cours_id) VALUES(10, 'Qu''est-ce qui
est une relation dans un modèle E-R?', 2);
INSERT INTO questions(id, content, partie_cours_id) VALUES(11, 'Qu''est-ce
qu''une jointure?', 4);
INSERT INTO questions(id, content, partie_cours_id) VALUES(12, 'Qu''elle est la
différence entre un modèle E-R et un modèle relationnel?', 3);
INSERT INTO questions(id, content, partie_cours_id) VALUES(13, 'Qu''elle est la
différence entre une jointure et une  $\theta$  jointure?', 4);
INSERT INTO questions(id, content, partie_cours_id) VALUES(14, 'Pourquoi n''y
a-t-il pas d''attributs multivalués dans un modèle relationnel?', 3);
INSERT INTO questions(id, content, partie_cours_id) VALUES(15, 'Qu''elle est
l''utilité de l''implantation de la  $\theta$  jointure dans une base de donnée?',
4);
INSERT INTO questions(id, content, partie_cours_id) VALUES(16, 'Qu''est-ce
qu''un attribut clé étrangère?', 2);
INSERT INTO questions(id, content, partie_cours_id) VALUES(17, 'Combien
d''attributs peuvent être clés d''une table relationnelle?', 2);
INSERT INTO questions(id, content, partie_cours_id) VALUES(18, 'Comment se
nomme l''opération servant à faire la jointure de deux tables selon un critère
quelconque sous forme de booléen?', 4);
INSERT INTO questions(id, content, partie_cours_id) VALUES(19, 'Pourquoi
devons-nous comprendre l''algèbre relationnel avant d''apprendre le langage
SQL?', 4);
INSERT INTO questions(id, content, partie_cours_id) VALUES(20, 'Qu''elle est la
différence entre une entité et une relation?', 2);
INSERT INTO questions(id, content, partie_cours_id) VALUES(30, 'Laquelle de ces
définitions représente une variable ?', 10);
INSERT INTO questions(id, content, partie_cours_id) VALUES(31, 'Qu''es-ce
qu''un interpréteur ?', 10);
INSERT INTO questions(id, content, partie_cours_id) VALUES(32, 'Que
devrait-t-on mettre au début d''un switch ?', 11);
INSERT INTO questions(id, content, partie_cours_id) VALUES(33, 'Quelle est la
différence entre un while et un do while ?', 11);
INSERT INTO questions(id, content, partie_cours_id) VALUES(34, 'En python, quel
mot clé permet de retourner une valeur d''une fonction ?', 12);
INSERT INTO questions(id, content, partie_cours_id) VALUES(35, 'En java, quel
est le type déclaré d''une fonction ne retournant rien ?', 12);
INSERT INTO questions(id, content, partie_cours_id) VALUES(36, 'Qu''es-ce
qu''un objet ?', 13);
INSERT INTO questions(id, content, partie_cours_id) VALUES(37, 'Quel est le
premier paramètre de toute méthode d''un objet en python ?', 13);

INSERT INTO reponses VALUES(1, 'Un ordinateur distant contenant un programme',
1, 0);
INSERT INTO reponses VALUES(2, 'Une structure servant à organiser des données',
1, 1);
INSERT INTO reponses VALUES(3, 'Pour organiser efficacement des données', 2, 1);
INSERT INTO reponses VALUES(4, 'Pour mettre des programmes sur internet', 2, 0);
INSERT INTO reponses VALUES(5, 'Un attribut permettant d''accéder à la
structure interne de la base', 3, 0);
INSERT INTO reponses VALUES(6, 'Un attribut identifiant un tuple', 3, 1);

```

```

INSERT INTO reponses VALUES(7, 'Un attribut comportant plusieurs sous-données',
4, 1);
INSERT INTO reponses VALUES(8, 'Un attribut référencé à plusieurs endroits', 4,
0);
INSERT INTO reponses VALUES(9, 'Unique', 5, 1);
INSERT INTO reponses VALUES(10, 'Additif', 5, 0);
INSERT INTO reponses VALUES(11, 'Pi', 6, 0);
INSERT INTO reponses VALUES(12, 'Sigma', 6, 1);
INSERT INTO reponses VALUES(13, 'Elle sert à joindre deux tables sur une
condition', 7, 1);
INSERT INTO reponses VALUES(14, 'Elle sert à éliminer des colonnes dans un
tuple', 7, 0);
INSERT INTO reponses VALUES(15, 'Vrai', 8, 0);
INSERT INTO reponses VALUES(16, 'Faux', 8, 1);
INSERT INTO reponses VALUES(17, 'Vrai', 9, 1);
INSERT INTO reponses VALUES(18, 'Faux', 9, 0);
INSERT INTO reponses VALUES(19, 'Une variable est un paramètre permettant de
changer le flot d''exécution d''un programme', 30, 0);
INSERT INTO reponses VALUES(20, 'Une variable est un emplacement mémoire
réservé dans lequel on peut storer des données', 30, 1);
INSERT INTO reponses VALUES(21, 'Un interpréteur est un programme permettant
d''exécuter du code non compilé', 31, 1);
INSERT INTO reponses VALUES(22, 'Un interpréteur permet de transférer un
programme sur un serveur distant', 31, 0);
INSERT INTO reponses VALUES(23, 'La variable que l''on veut tester', 32, 1);
INSERT INTO reponses VALUES(24, 'Le code qu''on veut exécuter', 32, 0);
INSERT INTO reponses VALUES(25, 'Dans un do while, le code s''exécute au moins
une fois', 33, 1);
INSERT INTO reponses VALUES(26, 'Il n''y a aucune différence', 33, 0);
INSERT INTO reponses VALUES(27, 'Un do while est plus optimisé', 33, 0);
INSERT INTO reponses VALUES(28, 'give', 34, 0);
INSERT INTO reponses VALUES(29, 'return', 34, 1);
INSERT INTO reponses VALUES(30, 'end', 34, 0);
INSERT INTO reponses VALUES(31, 'null', 35, 0);
INSERT INTO reponses VALUES(32, 'void', 35, 1);
INSERT INTO reponses VALUES(33, 'nil', 35, 0);
INSERT INTO reponses VALUES(34, 'Une donnée complexe regroupant plusieurs types
de données ainsi que des fonctions', 36, 1);
INSERT INTO reponses VALUES(35, 'Une représentation en mémoire d''un élément de
base de donnée', 36, 0);
INSERT INTO reponses VALUES(36, 'self', 37, 0);
INSERT INTO reponses VALUES(37, 'l''objet lui-même', 37, 1);
INSERT INTO reponses VALUES(38, 'Tout', 10, 0);
INSERT INTO reponses VALUES(39, 'Les losanges seulement', 10, 1);
INSERT INTO reponses VALUES(40, 'Une union suivie d''une sélection', 11, 1);
INSERT INTO reponses VALUES(41, 'Une union suivie d''une projection', 11, 0);
INSERT INTO reponses VALUES(42, 'La perspective de l''entité', 12, 1);
INSERT INTO reponses VALUES(43, 'Les relations entre entités', 12, 0);
INSERT INTO reponses VALUES(44, 'Les possibilités de condition de sélection qui
suit l''union', 13, 0);
INSERT INTO reponses VALUES(45, 'La condition de sélection qui suit l''union',

```

```

13, 1);
INSERT INTO reponses VALUES(46, 'Il y en a, la question n''a pas de sens.', 14,
0);
INSERT INTO reponses VALUES(47, 'Puisque chaque valeur identifie une relation
différente.', 14, 1);
INSERT INTO reponses VALUES(48, 'L''opportunité de faire des jointures avec des
conditions variées.', 15, 1);
INSERT INTO reponses VALUES(49, 'Impressionner le client.', 15, 0);
INSERT INTO reponses VALUES(50, 'Un attribut qui réfère à une autre table
relationnelle.', 16, 1);
INSERT INTO reponses VALUES(51, 'Un attribut clé qui réfère à une autre table
relationnelle.', 16, 0);
INSERT INTO reponses VALUES(52, '1 attribut clé seulement.', 17, 0);
INSERT INTO reponses VALUES(53, 'Autant que nécessaire.', 17, 1);
INSERT INTO reponses VALUES(54, 'La sesqui-jointure', 18, 0);
INSERT INTO reponses VALUES(55, 'La thêta jointure', 18, 1);
INSERT INTO reponses VALUES(56, 'Puisque l''algèbre relationnel aide à faire
des requêtes complexes.', 19, 0);
INSERT INTO reponses VALUES(57, 'Puisque l''algèbre relationnel aide à rendre
simple les notions complexes des requêtes SQL.', 19, 1);
INSERT INTO reponses VALUES(58, 'L''entité peut être perçue comme étant une
relation, mais pas nécessairement l''inverse.', 20, 1);
INSERT INTO reponses VALUES(59, 'Une entité est concrète alors qu''une relation
est abstraite et les deux sont distinctes.', 20, 0);
INSERT INTO reponses VALUES(60, 'Une structure servant à organiser des
données', 1, 1);

```

```

INSERT INTO question_categories VALUES(9, 2);
INSERT INTO question_categories VALUES(3, 4);
INSERT INTO question_categories VALUES(10, 4);
INSERT INTO question_categories VALUES(20, 4);
INSERT INTO question_categories VALUES(3, 5);
INSERT INTO question_categories VALUES(4, 5);
INSERT INTO question_categories VALUES(5, 5);
INSERT INTO question_categories VALUES(12, 5);
INSERT INTO question_categories VALUES(14, 5);
INSERT INTO question_categories VALUES(15, 5);
INSERT INTO question_categories VALUES(16, 5);
INSERT INTO question_categories VALUES(17, 5);
INSERT INTO question_categories VALUES(6, 6);
INSERT INTO question_categories VALUES(7, 6);
INSERT INTO question_categories VALUES(8, 6);
INSERT INTO question_categories VALUES(11, 6);
INSERT INTO question_categories VALUES(12, 6);
INSERT INTO question_categories VALUES(13, 6);
INSERT INTO question_categories VALUES(18, 6);
INSERT INTO question_categories VALUES(19, 6);
INSERT INTO question_categories VALUES(1, 11);
INSERT INTO question_categories VALUES(2, 11);
INSERT INTO question_categories VALUES(15, 11);
INSERT INTO question_categories VALUES(16, 11);

```



```

INSERT INTO question_categories VALUES(32, 15);
INSERT INTO question_categories VALUES(33, 16);
INSERT INTO question_categories VALUES(34, 15);
INSERT INTO question_categories VALUES(35, 15);
INSERT INTO question_categories VALUES(36, 19);
INSERT INTO question_categories VALUES(37, 18);

```

```

INSERT INTO question_categories
  SELECT id, 10 FROM questions;

```

Requêtes de l'application

L'application exécute les requêtes suivantes sur la base de données

----- Page d'accueil -----

-- Sélection de tous les cours

```
SELECT * FROM cours;
```

-- Sélection des parties de cours

```
SELECT * FROM partie_cours;
```

-- Sélection récursive des catégories

```

SELECT cat.name
FROM categories cat
LEFT JOIN categories parent ON parent.id=cat.category_id
START WITH cat.category_id IS NULL
CONNECT BY PRIOR cat.id=parent.id;

```

*-- Note : dans les dernières version d'Oracle, il est possible de retrouver
 -- le niveau de récursion de la requête via une requête du type :
 -- WITH children_categories(id, name, level) AS (SELECT ...
 -- Mais la version d'Oracle SQL du DIRO ne permet pas une telle chose.
 -- Une telle requête est utilisée dans notre application*

----- API de recherche -----

*-- Requête originale telle qu'elle est exécutée en python :
 -- SELECT id, content AS text FROM questions WHERE ' + op.join(['LOWER(content)
 LIKE ?' for i in range(len(words))]) + ' LIMIT 10;
 -- Où 'op' est soit 'OR', soit 'AND', pour permettre de rechercher en
 considérant *tous les mots-clés* ou *au moins un des mots-clés*
 -- Le 'LIMIT 10' sert à éviter d'avoir trop de résultats à envoyer lors d'une
 requête à l'API (par exemple, en recherchant
 -- "a", ce qui sélectionne la majorité des entrées de la base de données), mais
 n'est pas disponible facilement dans la version d'Oracle
 -- SQL du DIRO (les dernières versions permettent de faire 'FETCH FIRST 10 ROWS
 ONLY'*

*-- Les requêtes suivantes sont des exemples compilés en SQL pour 'op' = AND et
 pour 'words' = ['algèbre', 'relationnel'] :
 SELECT id, content AS text FROM questions WHERE LOWER(content) LIKE '%algèbre%'*

```

AND LOWER(content) LIKE '%relationnel%';
SELECT id, name AS text FROM categories WHERE LOWER(name) LIKE '%algèbre%' AND
LOWER(name) LIKE '%relationnel%';
SELECT sigle AS id, name AS text FROM cours WHERE LOWER(name) LIKE '%algèbre%'
AND LOWER(name) LIKE '%relationnel%';
SELECT id, name AS text FROM partie_cours WHERE LOWER(name) LIKE '%algèbre%'
AND LOWER(name) LIKE '%relationnel%';

----- Tirage de questions aléatoire -----
-- Sélection de questions par sigle de cours
-- Exemple de requête compilée pour le cours 'IFT2935'
SELECT * FROM questions
    WHERE partie_cours_id IN (
        SELECT id FROM partie_cours
        WHERE cours_id='IFT2935'
    );

-- Sélection des questions relatives à un partie d'un cours
-- Exemple de requête compilée pour partie_cours_id=1
SELECT * FROM questions WHERE partie_cours_id=1;

-- Sélection des ID de catégories enfants d'une catégorie donnée
-- Exemple compilé pour les enfants de la catégorie 1
SELECT cat.id
FROM categories cat
LEFT JOIN categories parent ON parent.id=cat.category_id
START WITH cat.category_id = 1
CONNECT BY PRIOR cat.id=parent.id;

-- Sélection des questions relatives à des catégories données
-- Exemple compilé pour les catégories 17 et 19
SELECT * FROM questions question
JOIN question_categories
ON question_categories.question_id=question.id
AND question_categories.category_id IN (17, 19);

-- Sélection d'une question précise
-- Exemple compilé pour la question 1
SELECT * FROM questions WHERE id=1;

-- Sélection d'une catégorie et de ses catégories parentes
-- Exemple compilé pour trouver la catégorie 5 et ses parents
SELECT cat.id
FROM categories cat
LEFT JOIN categories parent ON parent.id=cat.category_id
START WITH cat.id = 5
CONNECT BY cat.id = PRIOR cat.category_id;

-- Sélection des réponses d'une question dans un ordre aléatoire
-- Exemple compilé pour trouver les réponses de la question 1
SELECT * FROM reponses WHERE question_id=1 ORDER BY dbms_random.value;

```

```

----- Statistiques -----
-- Nombre de réponses de questions par cours
SELECT sigle AS Sigle, cours.name AS Nom, COUNT(reponses.id) AS
Nombre_de_reponses
FROM cours
JOIN partie_cours ON partie_cours.cours_id=cours.sigle
JOIN questions ON partie_cours.id=questions.partie_cours_id
JOIN reponses ON reponses.question_id=questions.id
GROUP BY cours.sigle, cours.name
ORDER BY cours.sigle;

-- Moyenne de questions dans la base de données / cours
SELECT AVG(cnt) AS Nombre_moyen FROM (
    SELECT COUNT(questions.id) as cnt
    FROM cours
    JOIN partie_cours ON partie_cours.cours_id=cours.sigle
    JOIN questions ON partie_cours.id=questions.partie_cours_id
    GROUP BY cours.sigle
);

-- Nombre de questions par professeur
SELECT prenom AS Prenom, nom AS Nom, COUNT(questions.id) AS Nombre_de_questions
FROM professeurs professeur
JOIN professeur_cours ON professeur.id=professeur_cours.professeur_id
JOIN cours ON professeur_cours.cours_id=cours.sigle
JOIN partie_cours ON partie_cours.cours_id=cours.sigle
JOIN questions ON questions.partie_cours_id=partie_cours.id
GROUP BY professeur.id, prenom, nom;

----- Mise à jour des statistiques de bonnes/mauvaises réponses -----
-- Augmente de 1 le nombre de bonne réponses à une question
-- Exemple compilé pour la question 1
UPDATE questions SET success=success+1 WHERE id = 1;
-- Augmente de 1 le nombre de mauvaises réponses à une question
-- Exemple compilé pour la question 1
UPDATE questions SET failures = failures + 1 WHERE id = 1;

----- Statistiques réservés aux professeurs -----
-- Recherche d'un professeur via son nom, prénom et mot de passe (pour
l'authentification)
-- Si la requête ne donne aucun résultat, on peut assumer que
l'authentification a échoué
-- Exemple compilé pour 'Claude Frasson', avec le mot de passe 'ift2935'
SELECT * FROM professeurs WHERE nom='Frasson' AND mot_de_passe='ift2935' AND
prenom='Claude';

-- Nombre de questions par cours
-- Exemple compilé pour professeur.id=1
SELECT COUNT(questions.id) AS nbr_questions, cours.name AS name, cours.sigle AS
id, cours.sigle AS sigle

```

```

FROM cours
JOIN professeur_cours ON professeur_cours.professeur_id=1
AND cours.sigle=professeur_cours.cours_id
JOIN partie_cours ON partie_cours.cours_id=cours.sigle
JOIN questions ON questions.partie_cours_id=partie_cours.id
GROUP BY cours.name, cours.sigle;

-- Taux de réussite moyen sur toutes les questions du professeur
-- Exemple compilé pour professeur_id=1
SELECT ((SUM(success)*100)/NULLIF(SUM(success) + SUM(failures), 0)) AS
average_reussite
FROM questions
JOIN professeur_cours ON professeur_cours.professeur_id = 1
JOIN partie_cours ON questions.partie_cours_id = partie_cours.id
JOIN cours ON partie_cours.cours_id=cours.sigle
WHERE cours.sigle=professeur_cours.cours_id;

-- Taux de réussite par question
-- Exemple compilé pour professeur_id=1
SELECT sigle, content, (success*100)/NULLIF(success + failures, 0) AS rapport,
(success + failures) AS nbr_answers
FROM questions
JOIN partie_cours ON questions.partie_cours_id = partie_cours.id
JOIN cours ON partie_cours.cours_id=cours.sigle
JOIN professeur_cours ON professeur_cours.professeur_id = 1
AND cours.sigle=professeur_cours.cours_id
ORDER BY rapport DESC, nbr_answers DESC;

-- Sélection des collègues d'un professeur
-- Exemple compilé pour professeur_id=1
SELECT * FROM professeurs professeur
JOIN professeur_cours ON professeur_cours.professeur_id=professeur.id
JOIN cours ON professeur_cours.cours_id=cours.sigle
WHERE cours.sigle IN (
    SELECT cours_id FROM professeurs
    JOIN professeur_cours pc ON pc.professeur_id=professeurs.id AND
professeur_id=1
) AND professeur.id != 1;

```

Conclusion

Durant ce projet, nous avons découvert la facilité avec laquelle python peut permettre de développer des applications webs mais nous avons surtout pu constater le défi que représente la gestion d'une base de donnée dans une application non triviale. En effet, dans de tels entreprises, la base de donnée joue un rôle central tant dans la conservation des données que dans l'accès à celles-ci.

Nous avons choisis d'utiliser SQLite pour ce projet puisque la légèreté et la facilité d'installation de ce système nous a rendu la tâche considérablement plus facile et rapide. Les différences syntaxiques entre le SQL accepté par les systèmes Oracle et celui accepté par SQLite ne nous ont pas posés

de problèmes majeurs puisque l'apprentissage de l'algèbre relationnelle nous a pourvu d'un cadre théorique solide et invariant sur lequel nous pouvons nous baser dans toutes situations.

Notre projet est ainsi complet et serait utilisable par les professeurs souhaitant aider leurs élèves via un système web interactif. Il y a cependant toujours place à l'amélioration, certaines fonctionnalités telles que des statistiques de bonne réponses par étudiant pour permettre un profilage plus précis des difficultés particulières des utilisateurs du système.