

Mini Projet - BDR

Rendu final

Système de gestion des écoles

Groupe 6

Professeur : René Rentsch

Assistant : Sébastien Rosat

28.01.2022

Table des matières

1. Introduction	4
2. Description du projet	4
3. Modèle Entité-Associations	5
4. Modèle relationnel.....	6
4.1. Description des tables.....	7
5. Description de l'application	8
5.1. Utilisation de l'application.....	8
6. Bugs connus	10
7. Remarques	11
7.1. Triggers	11
7.2. Schémas ER et EA	11
7.3. Données de tests.....	11
8. Conclusion.....	11

1. Introduction

Le but de ce document est de présenter notre mini-projet réalisé durant le cours BDR lors du semestre d'automne 2021. Le but de ce mini-projet est de réaliser une application complète de base de données relationnelle. Ce projet s'est étalé sur plusieurs mois et différentes étapes ont été effectuées avant la réalisation finale de l'application web. La première étape a consisté à définir le cahier des charges. Cette première analyse nous a permis de définir les besoins en données ainsi que les besoins fonctionnels. Dans un second temps, nous avons réalisé la modélisation conceptuelle et relationnelle de la base de données tout en définissant les diverses contraintes d'intégrité puis le projet s'est terminé par la création de l'application web ainsi que toutes les requêtes SQL nécessaires faites à la base de données.

2. Description du projet

Dans le cadre de ce mini-projet du cours BDR, nous avons décidé de créer une application pour la gestion d'une école. Le but de cette dernière est de permettre aux écoles une gestion simplifiée de tout le secteur "Formation et éducation".

La structure d'une école étant relativement complexe, nous l'avons séparée en deux axes principaux. D'un côté la partie liée à la formation et l'éducation des étudiants et de l'autre la partie liée à la gestion de l'école (personnel administratif et technique). Pour des raisons de simplification, ce mini-projet s'adresse donc uniquement à la partie formation et éducation.

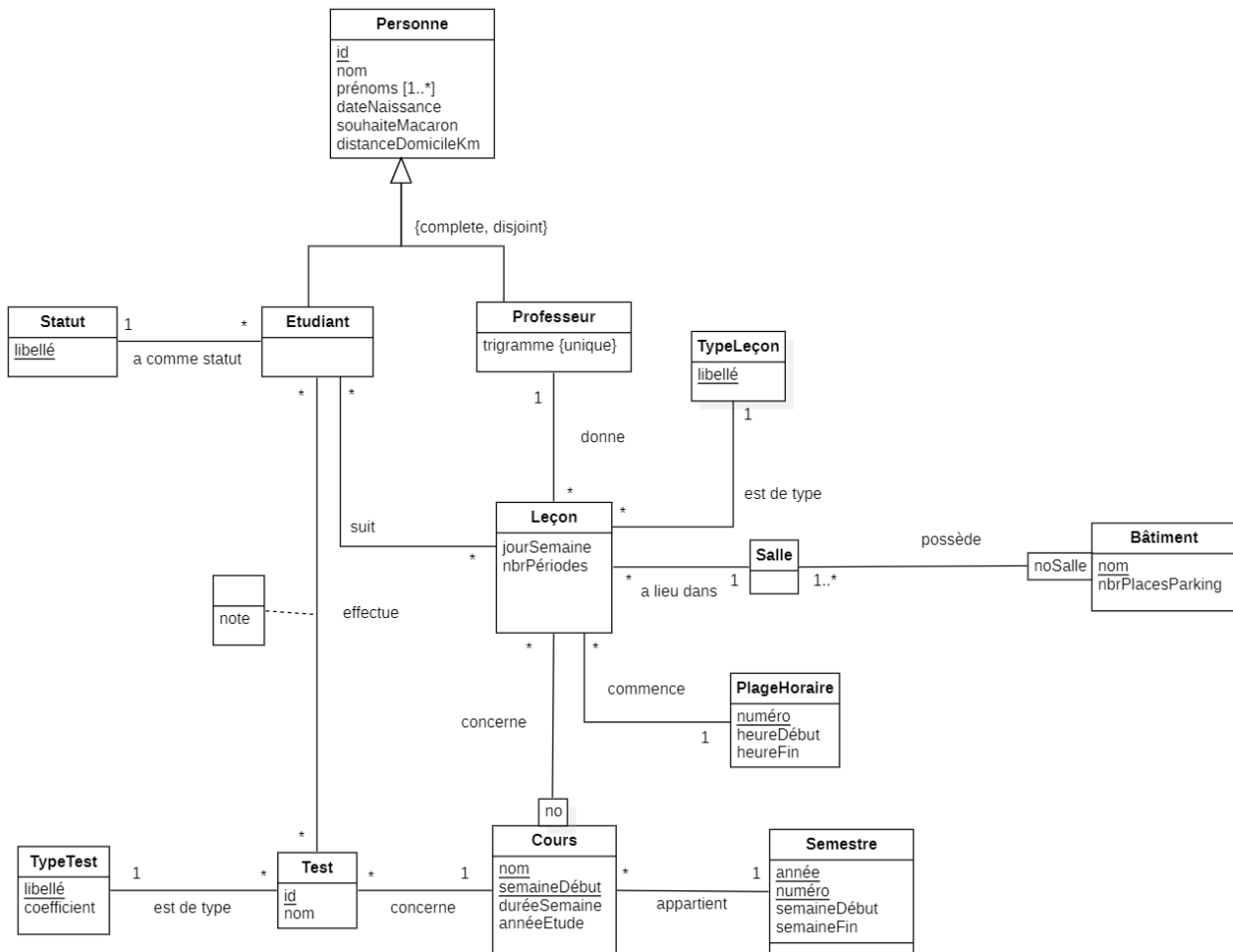
Ce programme permet donc de gérer toutes les entités liées à la formation dans une école à savoir :

- Les étudiants
- Les professeurs
- Les salles de cours et les bâtiments
- Les cours et les leçons
- Les horaires
- Les travaux écrits ainsi que leurs notes

L'application web a été réalisée avec le langage PHP ainsi qu'une base de données PostgreSQL. Nous avons également utilisé Docker afin de faire tourner le tout dans des containers pour des raisons de simplifications lors du processus de développement entre les divers membres du groupe. La gestion de cette application se fait donc via une page web simple et moderne.

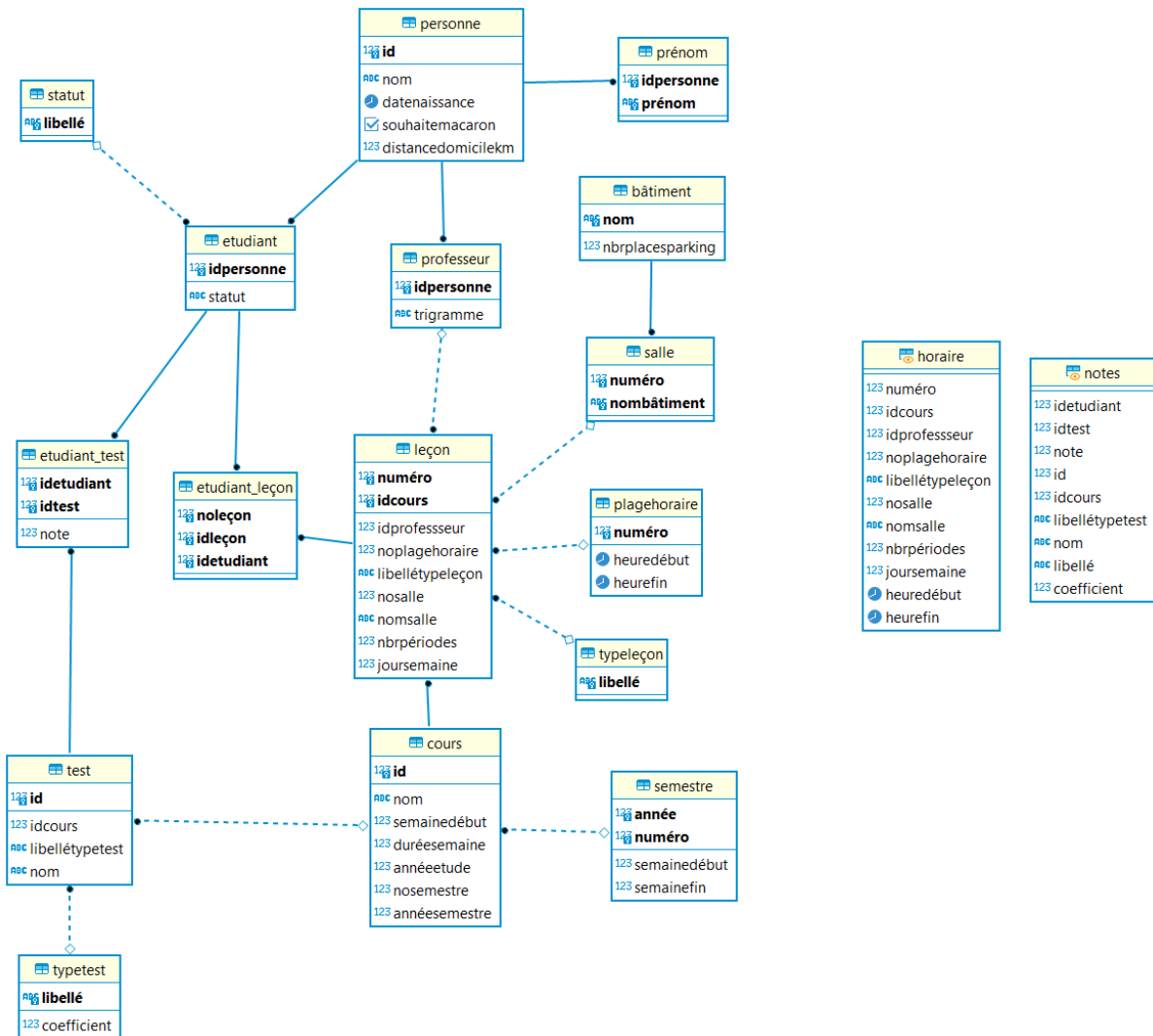
3. Modèle Entité-Associations

Le modèle EA ci-dessous a été créé au moyen du logiciel Star-UML. Ce diagramme a été créé lors de la phase n°2 du projet. Suite à cela, quelques modifications de noms d'attributs ont été effectuées mais la structure générale n'a pas été modifiée.



4. Modèle relationnel

Le modèle relationnel ci-dessous a été obtenu grâce au logiciel DBeaver une fois le script SQL exécuté. Les deux tables sans lien sont des vues utilisées par l'application.



4.1. Description des tables

- **Personne**

Notre base de données contient des étudiants ainsi que des professeurs. Pour factoriser un maximum les informations, nous avons décidé de créer une table Personne. Les étudiants et les professeurs héritent donc de cette table ainsi que les attributs qui leur sont communs. Il s'agit d'un héritage de type complet disjoint car toute instance de Personne est au moins instance de Professeur ou Etudiant, mais il est aussi disjoint car une personne ne peut pas être en même temps étudiant et professeur.

- **Etudiant**

Un étudiant hérite d'une personne ainsi que tous ses attributs. Un étudiant possède en plus un statut qui définit si l'étudiant a terminé ses études, si elles sont en cours, s'il a abandonné ou s'il a arrêté pour cause d'échec. De plus, un étudiant peut avoir une note pour chaque test d'un cours qu'il suit. Un étudiant peut également suivre une ou plusieurs leçons tout au long de la semaine.

- **Statut d'un étudiant**

En plus des attributs hérités de la table Personne, les étudiants possèdent ou non une cause de départ. Cette cause permet de savoir la raison pour laquelle l'étudiant n'est plus aux études. Les différentes causes de départ sont : « Réussite, Echec, Abandon »

- **Professeur**

Un professeur hérite d'une personne ainsi que tous ses attributs. Un professeur possède en plus un trigramme qui est unique. De plus, un professeur donne des leçons spécifiques.

- **Semestre**

Permet de modéliser un semestre d'étude. Un semestre possède une année ainsi qu'un numéro (valant 1 ou 2). De plus un semestre possède une semaine de début et de fin.

- **Cours**

Permet de modéliser un cours. Chaque cours possède un nom, une semaine de début, une durée en semaine, une année d'étude, De plus, un cours concerne un semestre

- **Test**

Permet de modéliser un test. Chaque test possède un nom et est d'un type spécifique défini dans la table **TypeTest** (TE, TP). De plus, un test concerne un cours spécifique. De plus, chaque type de test possède son propre coefficient qui sera ensuite utilisé pour le calcul des moyennes.

- **Leçon, TypeLecon et PlageHoraire**

La leçon est le point central de la base de données. Chaque leçon est liée à un ou plusieurs étudiants ainsi qu'à un seul professeur. Les leçons ont lieu dans une salle et concernent un cours spécifique. Chaque leçon possède un type (Cours, Labo) présent dans la table **TypeLecon**. Pour modéliser de manière efficace les horaires d'une leçon (heure de début, durée), nous avons décidé d'utiliser en lieu et place de valeurs sous la forme « d'heures » des entiers permettant de modéliser le numéro de chaque période de la journée. Ainsi, une leçon commence à une certaine période de la journée et dure N périodes pour un jour donné. Nous avons également créé une table **Plage Horaire** permettant de faire le lien entre les numéros de chaque période avec une heure précise. Grâce à cette technique, il est tout à fait possible d'ajouter des périodes à une journée ou de changer les heures de chaque période sans que cela ait des effets négatifs sur les données déjà insérées.

- **Salle et bâtiment**

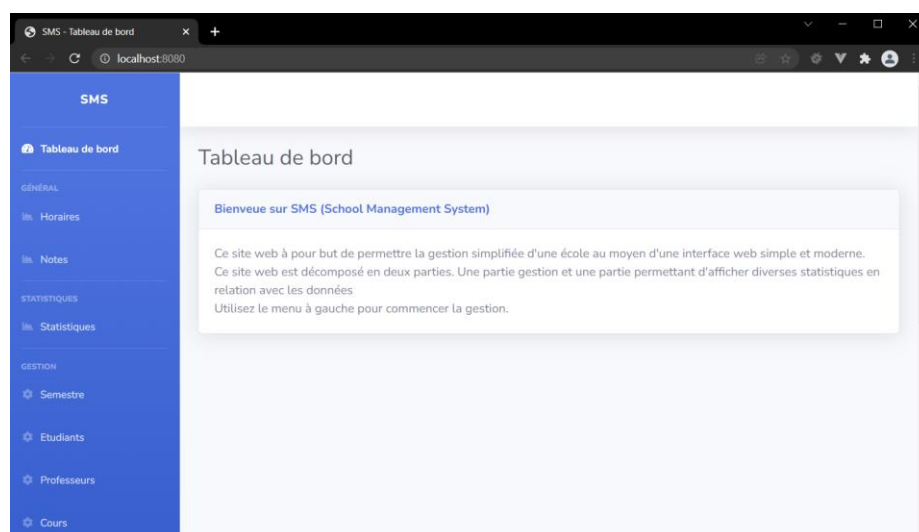
Ces deux tables permettent de modéliser les bâtiments de l'école ainsi que leurs salles. Un bâtiment possède un nom ainsi qu'un nombre de places de parking. Une salle possède un numéro, est liée à un bâtiment et est utilisée ou non par un cours.

5. Description de l'application

Pour la réalisation de l'application, nous avons décidé de créer une application web. La réalisation de cette application a été faite au moyen du langage web PHP ainsi que l'outil de conteneurisation Docker. Pour la réalisation de l'interface, et sachant que cette partie n'est pas fondamentale pour ce projet, nous avons décidé d'utiliser un « template » gratuit trouvé sur Internet. Cela nous a permis de gagner du temps à ce niveau tout en ayant une interface moderne et agréable.

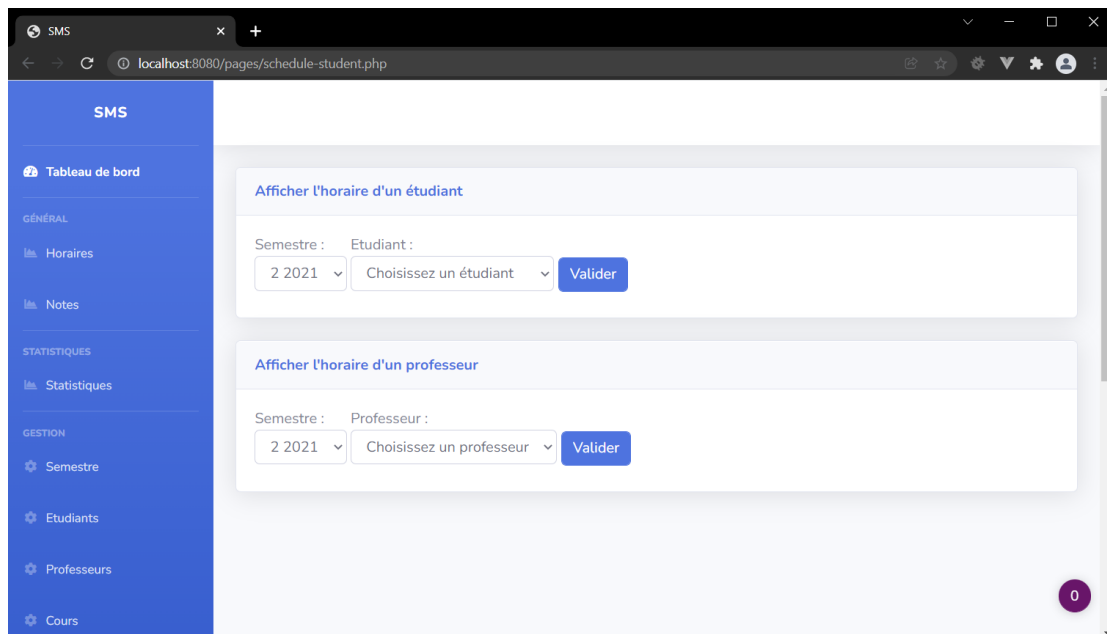
5.1. Utilisation de l'application

Pour lancer l'application sur votre ordinateur, merci de consulter le document « Guide d'installation » vous expliquant les étapes à réaliser. Une fois le projet lancé, vous tomberez sur la page suivante :



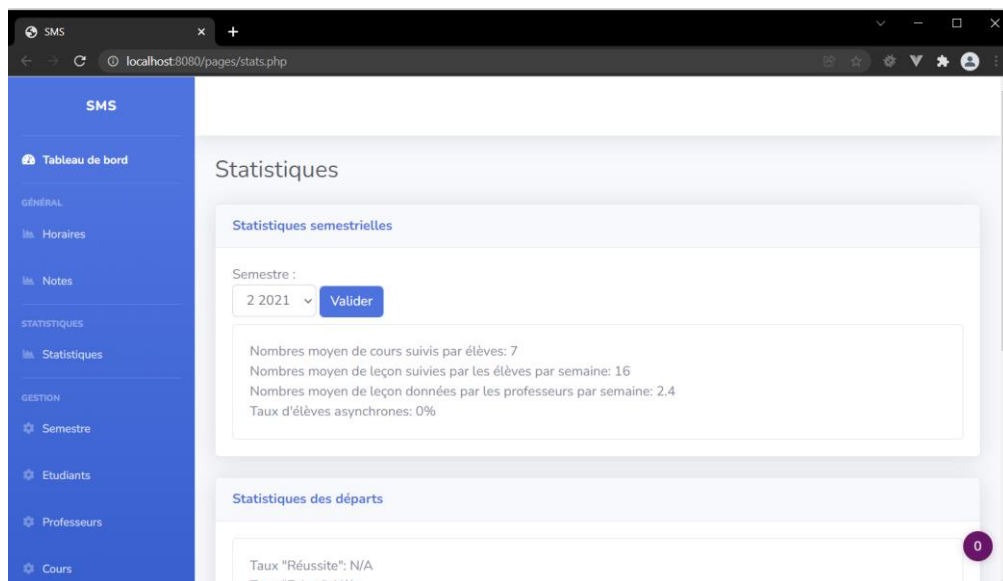
Pour naviguer sur le site, vous trouverez sur la gauche un menu vous permettant de naviguer à travers les diverses pages du site. Ce menu est décomposé en 3 parties

- **Général** – Permet d'obtenir les horaires et les notes des divers élèves



L'image ci-dessus présente la page permettant d'afficher les horaires des étudiants ou des professeurs. Il est possible que certains professeurs ou élèves ne possèdent pas d'horaire pour un semestre spécifique. En effet, pour des raisons de temps, certains élèves (par exemple de 3^{ème} année) n'ont que leur horaire de 3^{ème} année et non ceux de 1^{ère} et 2^{ème}.

- **Statistiques** – Permet d'obtenir diverses statistiques sur les données



L'image ci-dessus présente la page des statistiques. Ces stats sont obtenues pour chaque semestre. Il est possible que certains semestres n'aient pas de statistiques car ces derniers ne possèdent pas tous des données.

- **Gestion** – Ces diverses pages permettent de gérer les insertions, suppressions et modifications des données des diverses tables présentées précédemment.

The screenshot shows a web browser window with the URL `localhost:8080/pages/lessons-view.php`. The application has a blue sidebar menu with the following items: **Tableau de bord**, **GÉNÉRAL** (Horaires, Notes), **STATISTIQUES** (Statistiques), and **GESTION** (Semestre, Etudiants, Professeurs, Cours). The main content area is titled 'Ajouter une leçon' and contains a form with the following fields: Cours (MAD - 1/2021), Professeur (BOULÉ Ernest), Heure de début (08:45), Type leçon (Cours), and Salle (Salle 101 - Cheseaux). Below these fields are 'Nombre de périodes' (1) and 'Jour semaine' (Lundi), followed by an 'Ajouter' button. Below the form is a section titled 'Liste des leçons' containing a table with the following data:

Cours	Professeur	Plage horaire	Type leçon	Salle	Bâtiment	Jour semaine	Action
POO	CAISSE Patrick	08:45 - 10:00	Cours	902	Cheseaux	Vendredi	<div>Supprimer Modifier Ajouter étudiants</div>

L'image ci-dessus présente une des pages de gestion. De manière générale, ces pages se présentent sous la forme d'un tableau listant les diverses valeurs dans la table associée. Tout à droite du tableau, une colonne « action » permet d'effectuer les diverses actions sur les données.

6. Bugs connus

A ce jour, aucun bug connu n'est à signaler.

7. Remarques

7.1. Triggers

Les triggers suivants n'ont pas été réalisés car d'autres triggers de même type ont déjà été créés :

- La semaine de début d'un cours doit être entre la semaine de début et celle de fin de son semestre respectif

7.2. Schémas ER et EA

Les schémas EA et ER créés lors des phases 2 et 3 n'ont pas été grandement modifié lors de la conception de l'application. Seuls les attributs nommés « no » ont été renommés en « numéro » pour des raisons de noms réservés.

7.3. Données de tests

Comme demandé, une série de données de tests a été créée. Cependant, pour des raisons de temps à disposition certains étudiants ou professeurs n'ont pas tous leurs cours. Ex : un étudiant de 3^{ème} année possède uniquement ses cours de 3^{ème} et non ceux de 1^{ère} et 2^{ème}. Il est donc possible lors de la visualisation des horaires que certains étudiants n'aient pas de cours pour un ou plusieurs semestres spécifiques. Il en va de même pour les tests. Certains étudiants ont des cours sans avoir de note associée. Dans les deux cas, cela ne pose aucun problème au bon fonctionnement de l'application ni même pour les statistiques.

8. Conclusion

Pour conclure, nous pouvons remarquer que les phases préliminaires de conception de la base de données (modèle EA puis modèle ER) durant lesquelles nous avons passé une grande partie de ce mini-projet nous ont permis de créer une structure de base de données suffisamment solide tel qu'il n'a pas été nécessaire, par la suite, de la modifier. De plus, la base de données étant suffisamment bien structurée, cela nous a grandement aidé lors de la conception de l'application. En ce qui concerne l'insertion des données et comme signalé au préalable par M. Rentsch, cette étape nous a demandé de nombreuses heures de travail afin d'obtenir une base de travail cohérente. Pour la phase de développement et une fois quelques données de test insérées, il a été facile et relativement rapide de mettre en place le site web tel que proposé.