

Modèle relationnel et script SQL

Mini projet BDR

Système de gestion des écoles

Groupe 6

Professeur : René Rentsch

Assistant : Sébastien Rosat

17.12.2021

Table des matières

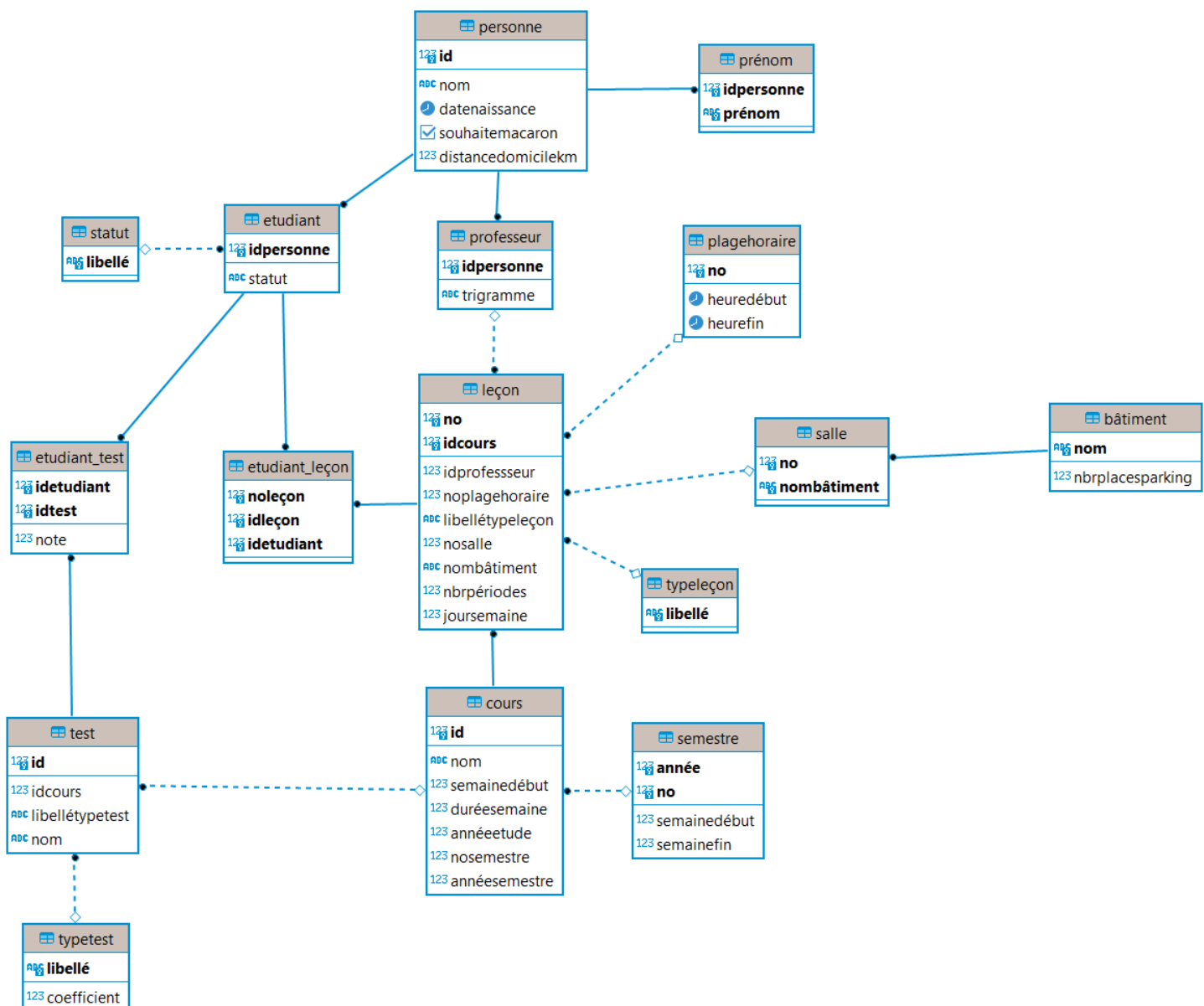
1. Introduction	4
2. Modèle relationnel.....	4
3. Contraintes d'intégrité	5
4. Remarques	5
5. Conclusion	6

1. Introduction

Le but de ce document est de présenter la modélisation de la base de données de notre projet du cours BDR au format relationnel. Ce modèle a été obtenu en convertissant le modèle « Entité-Relation » présenté dans le document « Modélisation conceptuelle » lors de la phase précédente. Pour rappel, l'objectif de cette base de données est de permettre de créer un système de gestion des écoles (SMS – School Management System).

2. Modèle relationnel

Le modèle relationnel ci-dessous a été obtenu grâce au logiciel DBeaver une fois le script SQL exécuté. Vous trouverez le script joint à ce document.



3. Contraintes d'intégrité

Suite à la création du modèle relationnel, certaines des contraintes d'intégrité décrites dans la modélisation conceptuelle ont pu être implémentées dans le script SQL. Cependant, certaines contraintes n'ont pu être modélisées et le seront par la suite au moyen de « triggers » ou « procédures stockées ».

Les contraintes d'intégrité suivantes ont été exprimées dans le script SQL :

- Un numéro de semestre vaut soit 1 soit 2
- Une note d'un test doit être comprise entre 1 et 6
- Un semestre ne peut pas avoir un numéro différent de 1 ou 2
- La date de naissance d'une personne ne peut pas être plus grande que la date du jour
- La distance domicile-école en km doit être une valeur supérieure ou égale à 0
- Il n'est pas possible de créer un semestre avec une année déjà passée

La liste des contraintes d'intégrité suivantes sont celles qui n'ont pas pu être intégrées au script SQL car ne pouvant pas être exprimées au moyen d'instructions SELECT.

- Un étudiant ou un professeur ne peuvent pas participer à plusieurs leçons qui ont lieu en même temps
- Un étudiant ne peut pas avoir un test d'un cours qu'il ne suit pas
- Un étudiant ne peut pas suivre 2 cours de semestre différents mais d'une même année
- Les heures des différentes périodes ne peuvent pas se chevaucher
- Une leçon ne peut pas avoir lieu dans la même salle à la même heure qu'une autre leçon
- Une leçon ne peut pas durer plus longtemps qu'il n'y a de périodes dans la journée
- La semaine de début d'un cours doit être entre la semaine de début et celle de fin de son semestre respectif
- La semaine de début + la durée en semaine d'un cours ne doit pas excéder la semaine de fin de son semestre respectif

4. Remarques

- Pour des raisons de logique, un bâtiment doit obligatoirement avoir une ou plusieurs salles de cours et une salle de cours doit appartenir à un bâtiment. De ce fait on ne peut pas créer de salle sans référencer un bâtiment et on ne peut pas créer de bâtiment sans référencer de salle. Afin de résoudre ce problème, il sera possible de créer des bâtiments sans salle.
- Nous avons fait le choix d'utiliser des « SMALLINT » pour la distance domicile-école de la table **Personne**. Ce choix s'explique car nous ne voyons pas l'intérêt d'utiliser des nombres à virgules car cela ne semble pas nécessaire d'avoir une telle précision.

- Nous avons décidé de créer un « Domain » appelé « type_libellé » du type « VARCHAR(30) » pour nous permettre de factoriser un maximum car à plusieurs reprises dans le script nous utilisons des libellés.
- Pour la table **Cours**, nous avons fait le choix d'utiliser une clé artificielle plutôt qu'une clé composite. Ce choix a été pris afin de simplifier les choses et de nous permettre d'éviter l'utilisation d'une clé primaire composée de 4 champs.

5. Conclusion

Pour conclure, nous pouvons remarquer que la modélisation conceptuelle nous a permis d'intégrer au script une bonne partie des contraintes d'intégrités. Cependant, les contraintes restantes seront représentées par la suite tout comme les contraintes d'intégrités liées à l'héritage.