

# Cahier de conception - Base de données

Anaïs Pignet - Alicia Rannou - Lucas Trebouet Voisin - Maxime Poirier - Cheikh Saliou NDIAYE

---

Dans le cadre du module de Base de Données enseigné à l'Enssat en première année, nous avons dû développer une base de données pour la gestion des clés des salles de l'école, et plus particulièrement de leurs emprunts par les étudiants, les professeurs ou le personnel.

Ce document décrit les choix que nous avons été amenés à faire lors de la conception de cette base de données.

Nos choix d'implémentation se basent sur l'analyse fonctionnelle réalisée lors de l'étude du cahier des charges du projet de Technologies Web. Nous avons ainsi choisi de chaîner les différents objets de la manière suivante : Une clé est attachée à une canon et à zéro ou plusieurs trousseaux, un canon est attaché à une porte et une porte est attachée à une salle.

## Clé primaire de la table Borrowing

Un emprunt a une clé primaire 'id' de type int, et non le couple utilisateur - trousseau (``user`(`id`),`keychain`(`id`))`). Pour des raisons de praticité dans l'implémentation web, un seul id est préférable.

## Un trousseau n'est empruntable qu'une seule fois

Pour emprunter à nouveau le même trousseau, un nouveau trousseau identique (avec les mêmes clés) est inséré en base. Cela nous permet de garder un historique de tous les trousseaux composés et leurs emprunts, et nous évite de manipuler des statuts "emprunté" ou "libre" sur des trousseaux.

## Multiplicité des canons et clés

Si un canon d'un même fournisseur et de même longueur est installé sur deux portes, on considère que c'est le même canon. Nous avons également raisonné de cette façon pour les clés. Ainsi un `lock` n'est associé qu'à une seule `enssat_key` (en plus des éventuels passes partiels ou totaux).

Cela pose en revanche un problème pour compter les exemplaires de clés et déterminer que toutes les clés ouvrant une porte ont été perdues par exemple. Il nous aurait fallu ajouter un attribut donnant le nombre d'exemplaires restant d'une clé ou d'un canon.

### **L'association trousseau - clé**

Cette association donne lieu à une création de table. En effet un trousseau se compose de plusieurs clés, et chaque clé peut être dans plusieurs trousseaux. Nous avons donc créé la table `keys_keychain`.

### **La table Room**

Cette table n'a pas d'id de type int. Nous avons choisi de mettre en clé primaire le nom de la salle (par exemple 005V). Puisque c'est la seule donnée de cette table, créer un id serait ajouter des données inutiles puisque le nom d'une salle est déjà suffisamment discriminant pour que deux salles ne portent pas le même nom.

### **La table Provider**

Pour éviter des doublons, l'identifiant du fournisseur est choisi comme clé primaire pour la table Provider.

### **La table Enssat\_lock**

La table Enssat\_lock a pour clé primaire son id. Nous avons choisis de mettre les attributs `doorId`, `providerId` comme attributs non nuls car cela nous permet d'avoir une intégrité des données.

### **La table Enssat\_key**

L'identifiant de la clé est choisi comme clé primaire pour la relation `enssat_key`. L'attribut `lockId` est un attribut non nul qui constitue une clé étrangère.

### **La table user**

Pour éviter des doublons, l'identifiant de l'utilisateur est choisi comme clé primaire pour la table user. Les attributs `ur1identifiant` et `enssatPrimaryKey` doivent être uniques. Ce choix permet d'éviter d'avoir des utilisateurs différents avec le même `ur1identifiant/enssatPrimaryKey`.

### **Les contraintes d'inclusions**

`Enssat_lock [providerId] c Provider[id]`

`Door [ roomId ] c room[ name ]`

Enssat\_key [lockId]  $\subset$  enssat\_lock[id]

Keys\_keychain [keyId]  $\subset$  enssat\_key[id]

Keys\_keychain [keychainId]  $\subset$  keychain [ id ]

Borrowing[keychainId]  $\subset$  keychain [id]

Borrowing[userId]  $\subset$  user [id]