|  |  |  |  |
| --- | --- | --- | --- |
| **Auteur :** Charles M. | | **Approbateurs :** Adrien F, Adrien D, Sébastien B | |
| **Révision** | | | |
| **Date:** 21/01/2019 | **Nature de la modification** | | **Version :** 1.0 |
| Création du dossier | |



<title>Dossier de conception </title>

<meta charset= “utf-8”>

<meta name=”projet” content=”CAASis”>

<meta name=”keywords” content=”HTML, CSS, PHP, JavaScript, MySQL”>

Web Project

Dossier de conception

<head>

</head>

Table des matières

[I. Dictionnaire de données 4](#_Toc535878061)

[II. Matrice des dépendances 5](#_Toc535878062)

[III. Modèle conceptuel de données 6](#_Toc535878063)

[1. User, location statut 6](#_Toc535878064)

[2. Events et ideas 6](#_Toc535878065)

[3. Images 8](#_Toc535878066)

[4. Commentaires 9](#_Toc535878067)

[5. Shop 10](#_Toc535878068)

[IV. Modèle logique de données 11](#_Toc535878069)

Table des illustrations

[Figure 1 Dictionnaire de données 4](#_Toc535868427)

[Figure 2 Matrice des dépendances 5](#_Toc535868428)

[Figure 3 MCD partie User, location et statut 6](#_Toc535868429)

[Figure 4 MCD partie Events et ideas 6](#_Toc535868430)

[Figure 5 MCD partie images 8](#_Toc535868431)

[Figure 6 MCD partie comments 9](#_Toc535868432)

[Figure 7 MCD partie shop 10](file:///C:\Users\Adrien\Desktop\ProjetWeb\Rapport%20Bases%20de%20Données%20-%20CAASis.docx#_Toc535868433)

[Figure 8 MLD 11](#_Toc535868434)

# Dictionnaire de données

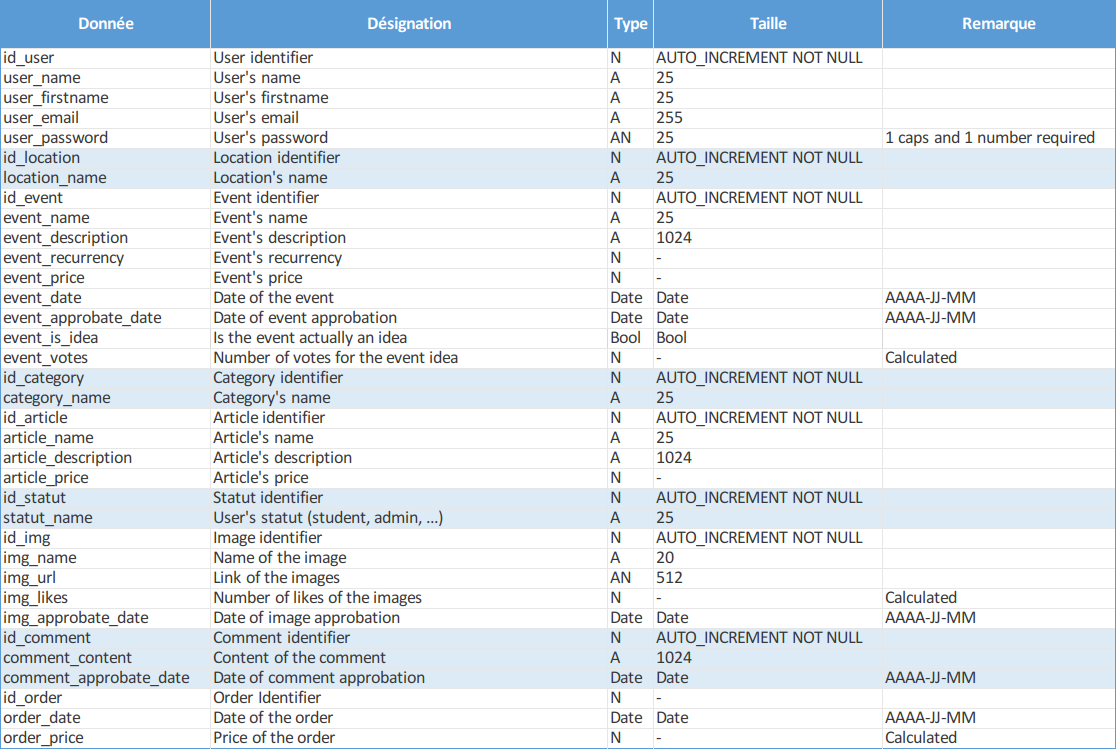


Figure 1 Dictionnaire de données

Voici la liste des données utilisées dans notre base de données. Cette base est le résultat de notre première imagination de la base de données et est donc une liste de toutes les données dont nous avons besoin pour le bon fonctionnement de notre système d’information.

Nous retrouvons également une désignation indiquant à quoi sert/correspond la donnée, son type, sa taille ainsi qu’une éventuelle remarque.

# Matrice des dépendances

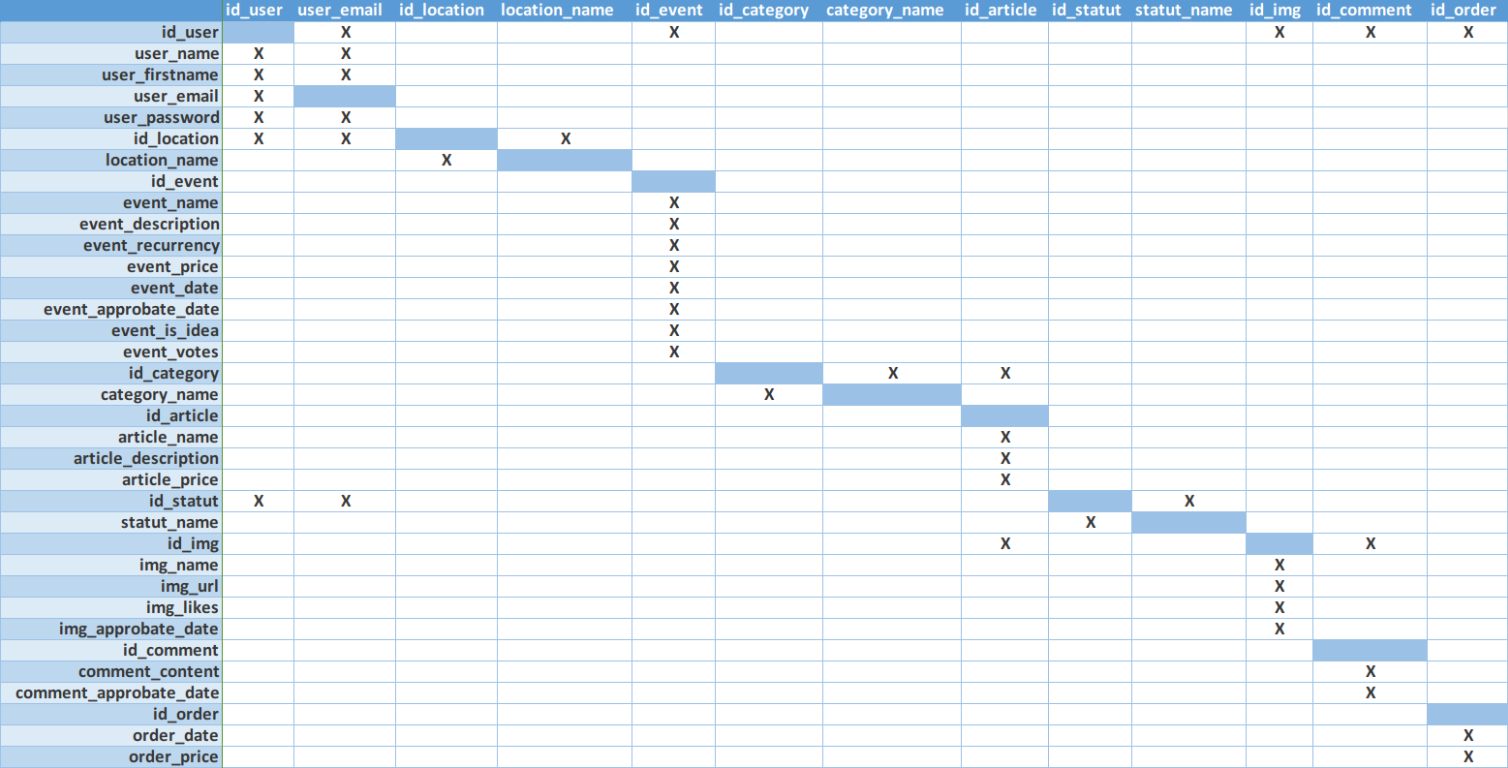


Figure 2 Matrice des dépendances

Certaines données parmi celles listées dans le dictionnaire de données dépendent d’autres données. C’est en fonction de ces dépendances que nous allons créer des futurs modèles. Voici donc ci-dessus une matrice montrant toutes les dépendances existantes entre nos données (représentées par des croix).

# Modèle conceptuel de données

## User, location statut

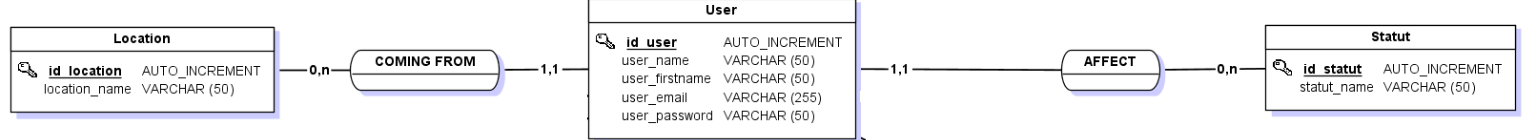


Figure 3 MCD partie User, location et statut

Nous avons créé une entité user avec comme données le nom, l'e-mail, le prénom et le mot de passe. Ces champs sont propres aux utilisateurs, c'est pourquoi nous les avons regroupés dans cette entité, contrairement aux champs location et statut.

La location est le campus dans laquelle se situe l'utilisateur mais, à l'avenir, si le nombre de centres est décuplé par exemple, il pourrait être nécessaire de prendre en compte également la région ou le pays : ce qui serait possible facilement grâce à l'entité location.

Le raisonnement est le même pour le statut, nous l'avons placé dans une entité à part question d'évolutivité.

## Events et ideas

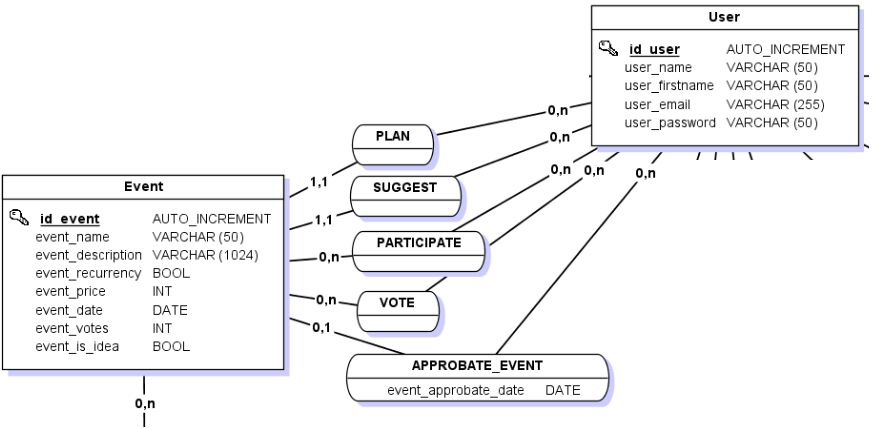


Figure 4 MCD partie Events et ideas

Nous avons choisi d'avoir une seule entité pour les événements et les idées, contenant : le nom, description, récurrence, prix, date, et les votes. Le champ des votes est calculé et permet de nous retourner directement le nombre de votes pour les événements. Il n'est pas indispensable puisqu'il y a la relation vote mais il permet une meilleure cohérence des données. Un champ (event\_is\_idea) nous permet de différencier les idées des événements.

Une idée est moins complète qu'un événement et aura donc beaucoup de champs vides. Cependant, lorsque les idées seront approuvées, nous allons compléter/modifier l'enregistrement plutôt que d'en créer un nouveau ce qui permettra d'avoir moins de données que si nous avions séparé les idées des événements.

Les associations plan et suggest ont pour but la sécurité. Elles permettront de savoir par qui a été transformé l'idée en événement et de qui elle provient. En cas de questions par exemple, cela permet d’avoir des personnes responsables en quelques sortes de l’évènement.

Un utilisateur peut indiquer qu'il participe à un événement et plusieurs peuvent participer au même événement. Nous avons donc une relation n à n permettant de stocker qui participe (ou a participé) à quels événements.

De même pour les votes, l'association vote nous permettra de savoir qui a voté pour telle ou telle idée.

La relation approbate\_event permettra également de garder en mémoire quels tuteurs ont approuvé chacun des événements. Nous retrouvons également des relations d'approbation pour les images ainsi que les commentaires.

## Images

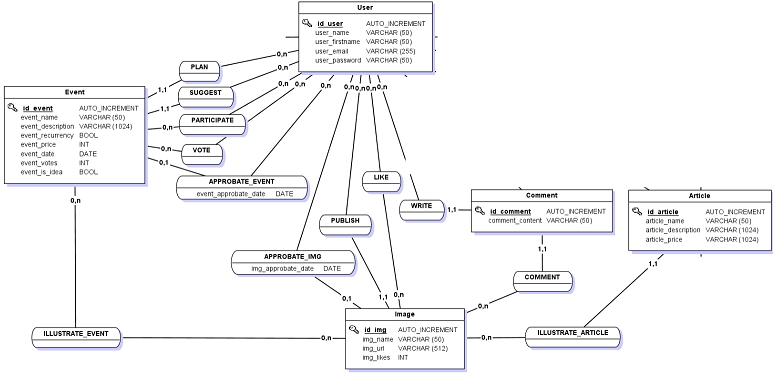


Figure 5 MCD partie images

Les images ont leur propre entité contenant leur nom, lien et nombre de likes. L'utilité du champ nombre de like est la même que celle du nombre de votes. Une image peut illustrer un événement ou bien un article, d'où les relations illustrate. Il y en a 2 ce qui nous permet de filtrer les photos en fonction de ce qu'elles illustrent, dans le but de récupérer seulement les photos des événements au format PDF par exemple.

Les images sont liées aux utilisateurs puisqu'ils les publient (même principe que les planifications d'événements), les likes (même principe que les votes) et également approuvées.

## Commentaires

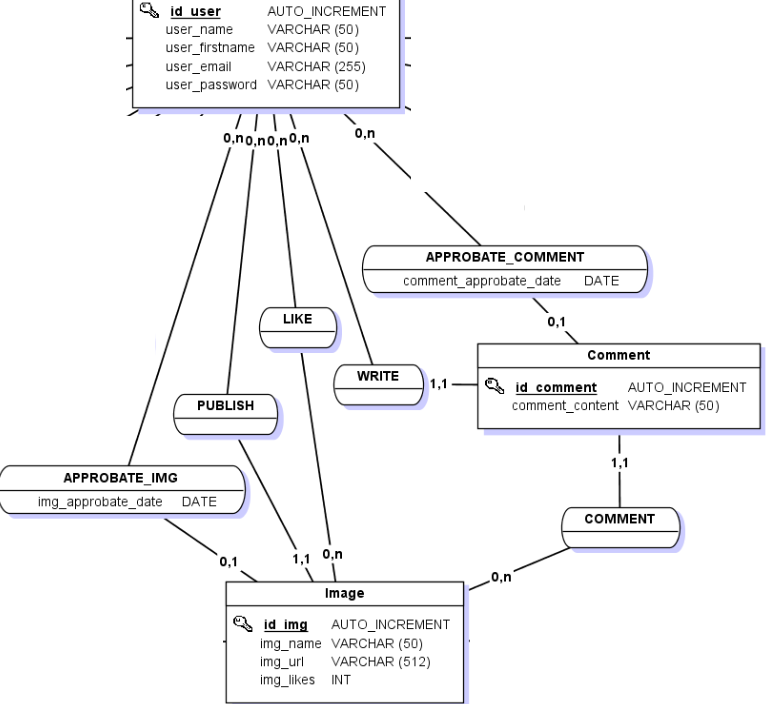


Figure 6 MCD partie comments

Les utilisateurs peuvent commenter les photos, d'où l'entité "comment". Un commentaire est dans un premier temps écrit par un utilisateur, puis approuvé par un autre utilisateur, dont le rang lui permet de le faire (d'où les relations write et approbate\_coment).

Ils sont liés aux images grâce à l’association comment, qui lie les entités commentaires et images.

## Shop

Un utilisateur peut remplir son panier shopping cart d'articles. Cette association permet de garder en mémoire les paniers tant que les commandes n'ont pas été faites.

Quand cela sera le cas, il sera supprimé pour ne pas garder de données inutiles.

À partir de son panier l'utilisateur peut passer commande d'où l'entité order, qui nous permet également de stocker toutes les commandes effectuées.

Il y a un certain nombre de catégories (qui peut évoluer) pour les articles d'où l'entité category liée aux articles par la relation bind.

Figure 7 MCD partie shop

Nous avons fait une dernière modification (qui n’apparaît pas dans les captures d’écran) concernant les événements récursifs. Il était nécessaire de savoir à quelle fréquence ils se répétaient. Nous avons donc créé un nouveau champ event\_frequency dans ce but, puisque la date et le fait que l’événement soit récursif ne suffisaient pas.

# Modèle logique de données

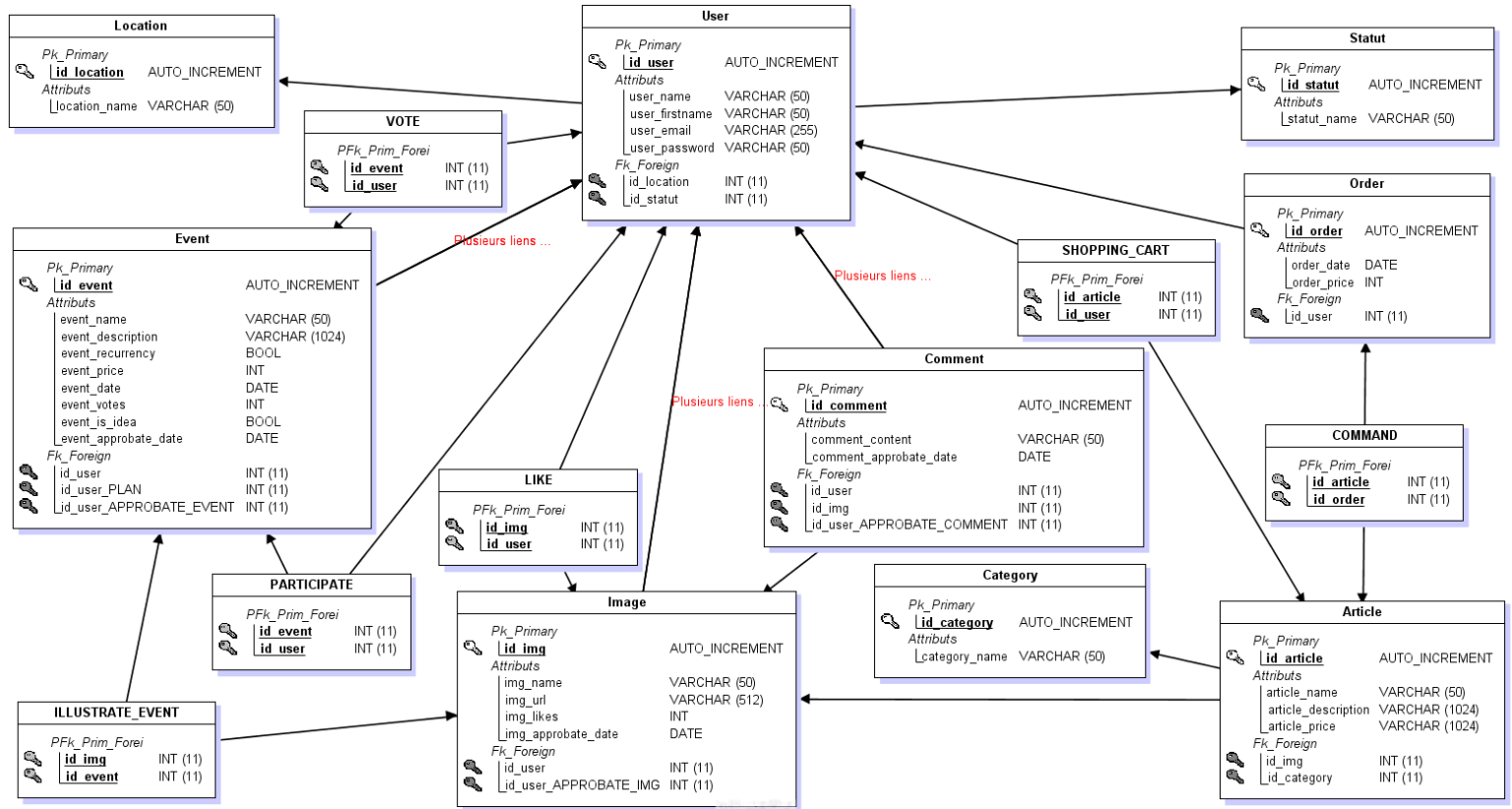


Figure 8 MLD

Le modèle logique de données est le résultat du MCD précédemment crée. Il montre clairement la forme qu’aura la base de données en présentant toutes les tables de la base ainsi que toutes les relations entre elles (clés primaires et étrangères).

On constate que certaines relations ont créé des clés étrangères dans des tables et d’autres ont fait apparaître des tables, contenant plusieurs clés primaires comme par exemple la table participate qui nous permet de stocker quels utilisateurs participent ou ont participé à quels événements.

Notre base de données est donc composée au total de 15 tables. Sachant que toutes les informations relatives à l’utilisateur (table user, location et statut) se trouvent sur une base de données nationale contrairement au reste qui se trouve sur une base de données locale.