

# TD 1 - Optimisation des Bases de données

Du cahier des charges à la base de données

## Cahier des charges

### Contexte

L'objectif de ce sujet est de créer une application permettant de gérer un ensemble d'articles à l'aide d'une base de données pour accéder à un ensemble d'informations sur des articles disponibles (titres, résumé, auteurs...) et avoir accès à une classification des articles à l'aide de notes et de tags associés à ces articles.

### Description des articles

Un article possède plusieurs attributs comme son titre, qui doit être unique, ses auteurs, un résumé et un type comme "démonstration", "recherche-long", "recherche-court" etc. Un article avec un titre donné peut être publié sous différents supports de publication : conférence, journal, thèse, livre etc. Tout support a donc un type (conférence, journal etc.) et un nom. On doit pouvoir déterminer l'année de publication d'un article sur un support donné, sachant qu'un article peut être publié dans plusieurs conférences et/ou journaux la même année ou à des années différentes. Néanmoins, un article donné, sur un support donné ne sera publié qu'une seule année.

Un auteur est caractérisé par son adresse email. Il possède aussi un nom, un prénom, et parfois un site web personnel. Le ou les laboratoires auxquels il appartient (un chercheur peut travailler pour différents laboratoires en même temps) lui sont associés. Un laboratoire est caractérisé par son nom et possède un sigle (par exemple LRI pour Laboratoire de Recherche en Informatique), une adresse, et éventuellement une adresse url indiquant son site web.

### Notation et Annotation des articles

Les utilisateurs de notre outil sont toujours des auteurs d'au moins un article de la base mais tous les auteurs ne sont pas nécessairement utilisateurs de notre outil. Le système ne prévoit pas d'identification par mot de passe. On aimerait permettre aux utilisateurs de notre outil de noter et d'étiqueter (= annoter) les articles gérés par le système (ces notes et tags ne dépendent pas du support de publication). La provenance (l'utilisateur ayant donné ce tag ou cette note) des tags et des notes est stockée dans le système. Cette provenance peut être utilisée dans les requêtes. Il est possible de trouver par exemple tous les auteurs ayant notés leurs propres articles.

Un utilisateur peut apposer des tags (= des étiquettes) sur n'importe quel article. Un tag n'est caractérisé que par sa valeur. A priori, n'importe quelle chaîne de caractères peut faire office de valeur à un tag. Chaque utilisateur peut aussi attribuer une note (entier compris entre 0 et 5) à des articles. Un utilisateur ne peut noter qu'une seule fois chaque article.

## 1 Modélisation

Modélisez l'énoncé ci-dessus à l'aide d'un diagramme de classes UML.

## 2 Conception

Après avoir comparé votre schéma avec celui donné en correction, déduisez le schéma relationnel lié à ce dernier.

## 3 Requêtes algébriques

Exprimer en algèbre les requêtes suivantes.

1. Quelle est l'adresse postale du Laboratoire LRI ?
2. Quels sont les articles écrits par Susan Davidson ?
3. Quels sont les auteurs qui ne sont pas utilisateurs ?
4. Qui (identifiants) a annoté au moins un article de susan@cis-upenn.edu ?
5. Qui (identifiants) n'a annoté que des articles de susan@cis-upenn.edu ?
6. Qui (identifiants) n'a annoté aucun des articles de susan@cis-upenn.edu ?

## 4 Interrogation SQL (simple)

Ecrivez en SQL les requetes suivantes

1. Qui (noms et emails) a noté au moins un article ?
2. Quelles annotations ont été données par Peter Buneman ?
3. Quels sont les articles (titres) dont Susan Davidson est l'un des co-auteurs ?