
EtuDoc

Modèle Conceptuel et Logique de Données – NA17

Version 2

Solène HOULLIEZ - Justine MARLOW - Thomas MEUROU

AUTOMNE 2017 - UTC

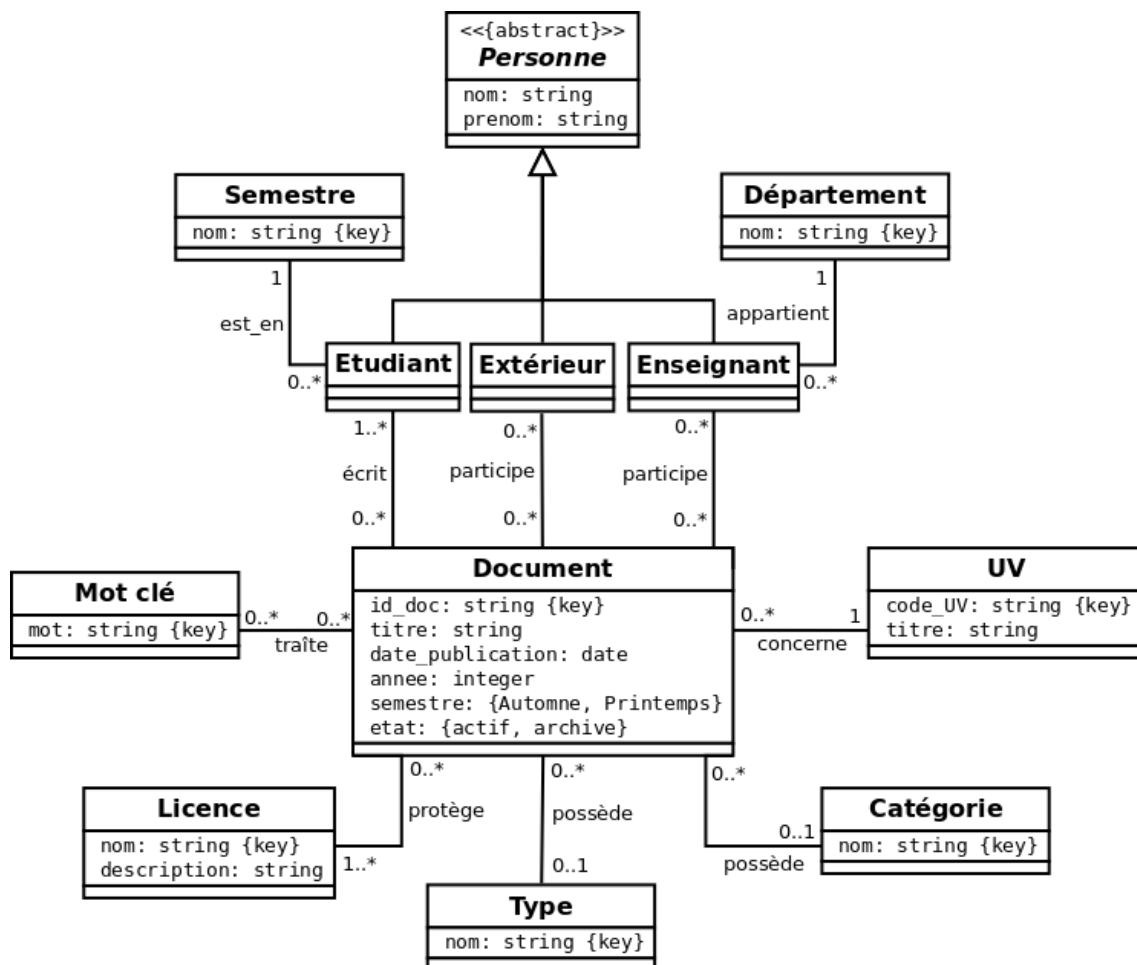
Version	Date de rendu	Note de révision
1	22/09/2017	Version initiale
2	29/09/2017	MCD : Ajout de la classe Personne et de la relation d'héritage Corrections de noms d'association et de cardinalités Suppression des parties inutiles MLD : Mise à jour du MLD en fonction des modifications apportées au MCD Ajout de l'explication de la transformation de l'héritage Correction de la partie Détails de la normalisation Ajout de tableaux avec des données d'exemple



Ce document est mis à disposition selon les termes de la Licence Creative Commons
Attribution - Pas d'Utilisation Commerciale - Pas de Modification 3.0 France.
<http://creativecommons.org/licenses/by-nc-nd/3.0/fr/>

Modèle Conceptuel de données

UML



Notes concernant la notion d'utilisateur et d'administrateur

Nous avons décidé de ne pas intégrer la gestion des accès et des droits (utilisateur, administrateur) dans le modèle conceptuel de données car cela fait partie, selon nous, de la sécurité et non du service de base de données sur lequel s'appuie EtuDc.

Modèle Logique de données

Ensemble des tables

Classes :

```
Semestre( #nom : string)
Etudiant ( #id : integer, nom : string, prenom : string, statut => Semestre )
    avec statut NOT NULL
Departement( #nom : string)
Enseignant ( #id : integer, nom : string, prenom : string, departement =>Departement )
    avec departement NOT NULL
Exterieur ( #id : integer, nom : string, prenom : string )
UV ( #code : string, titre : string )
Type ( #nom : string )
Mot_cle ( #nom : string )
Licence ( #nom : string, description : string )
Categorie ( #nom : string )
Document ( #id_doc : string, titre : string, date_publication : date, annee : integer,
    semestre : {Automne, Printemps}, etat : {actif, archive},
    uv => UV, type => Type, categorie => Categorie )
    avec uv NOT NULL
```

Classes association :

```
Ecrit ( #id_doc => Document, #id_etu => Etudiant )
Participe_enseignant ( #id_doc => Document, #id_enseignant => Enseignant )
Participe_exterieur ( #id_doc => Document, #id_exterieur => Exterieur )
Protege ( #doc => Document, #licence => Licence )
Traite ( #doc => Document, #mot => Mot_cle )
```

Détails de la transformation de l'héritage

Nous avons choisi de transformer l'héritage en dupliquant les attributs de la classe mère *Personne* (nom, prenom) dans chacune des classes filles *Etudiant*, *Enseignant* et *Exterieur*. En effet, cette solution nous paraît la moins lourde en mémoire (plus légère qu'intégrer les attributs des classes filles à la classe mère et de spécifier les attributs à projeter) et la plus adaptée au fait que la classe *Personne* est abstraite (intégrer des clés étrangères vers la classe mère au sein des classes filles indiquerait que la classe *Personne* puisse être instanciée).

Détails de la normalisation

Chacune de nos tables possède une clé primaire, signalée par '#' qui définit de manière stable et unique chaque ligne de la table à laquelle elle appartient. De plus, chaque attribut est monoévalué, c'est à dire qu'un attribut ne peut avoir qu'une seule valeur par ligne et aucun attribut choisi n'est décomposable en plusieurs attributs significatifs. Ainsi notre modèle, suite à la normalisation, est en première forme normale.



D'autre part, le modèle est en 1NF et les attributs non clés sont totalement dépendant de la clé ainsi le modèle est en deuxième forme normale.

En outre, aucun attribut non clé ne dépend d'un autre attribut non clé, et comme le modèle est déjà en deuxième forme normale, on peut alors affirmer que notre modèle est en troisième forme normale.

Contraintes à ajouter au MLD

Nous avons traduit une cardinalité 1..N comme une cardinalité 0..N. Ainsi, le modèle logique de données ne permet pas d'imposer qu'il y ait au moins une association entre les entités Etudiant et Document ainsi qu'entre Licence et Document. Il faudra donc être particulièrement vigilant au niveau de l'utilisation de la base de données : lors de l'ajout d'un tuple dans la table Document, il faudra également ajouter systématiquement un tuple dans la table Ecrit et dans la table Protege.

Tableaux avec des données d'exemple

Semestre :

nom
GI01

Etudiant :

id	nom	prenom	statut
1	Marlow	Justine	GI01
2	Houillez	Solène	GI01
3	Meurou	Thomas	GI01

Departement :

nom
Informatique

Enseignant :

id	nom	prenom	statut
1	Crozat	Stéphane	Informatique

Exterieur :

id	nom	prenom
1	Turing	Alan

UV :

code	titre
NA17	Base de données

Type :

nom
texte

Categorie :

nom
Note de clarification

Mot_cle :

nom
base de données
conception

Licence :

nom	description
UTCDoc	Licence de l'UTC

Document :

id_doc	titre	date_publication
2017-A-NA17-Note	Note de clarification de projet	2017-09-24
annee	semestre	etat
2017	Automne	actif
uv	type	categorie
NA17	texte	Note de clarification

Ecrit :

id_doc	id_etu
2017-A-NA17-Note	1
2017-A-NA17-Note	2
2017-A-NA17-Note	3

Participe_enseignant :

id_doc	id_enseignant
2017-A-NA17-Note	1

Participe_exterieur :

id_doc	id_exterieur
2017-A-NA17-Note	1

Protege :

doc	licence
2017-A-NA17-Note	UTCDoc

Traite :

doc	mot
2017-A-NA17-Note	base de données
2017-A-NA17-Note	conception