

# Modélisation, normalisation et représentation

Amina, KACIMI

Valentin, DEHAINAULT

Minh-Hoang, DANG

December 14, 2018

## Préambule

Lors de la phase précédente, nous avons nettoyé la base de données. Les données brutes ont été traitées et transférées dans une table accueillante. Cette table, nommée **Correction**, est à l'image du fichier CSV, ne possède pas de clé et ses attributs sont nullables.

Dans cette phase, nous allons normaliser cette table. A la fin, la table **Correction** va laisser la place pour les sous-tables.

## Déterminer les formes de normales

Avant de commencer, nous avons déterminé la clé primaire pour la table **Correction**. Nous avons remarqué que l'attribut **Article** est présent dans toutes les lignes. L'attribut **Discriminant** sert à compléter l'attribut **Article**. De ce logique, **Article** et **Discriminant** forment la clé primaire de **Correction**.

ReferenceCindoc	varchar	integer(10)	N
Serie	varchar	varchar(255)	N
<b>Article</b>		<b>integer(10)</b>	
<b>Discriminant</b>		<b>integer(10)</b>	
Ville		varchar(255)	N
CodePostal		varchar(5)	N
Sujet		varchar(255)	N
DescDet		varchar(255)	N
Date		date	N
NoteBP		varchar(255)	N
FicNum		varchar(22)	N
Idx_Ico		varchar(255)	N
TailleCli		varchar(255)	N
N_V		varchar(3)	N
C_G		varchar(3)	N
Remarques		varchar(255)	N
CoordX		float(10)	N
CoordY		float(10)	N

Powered By Visual Paradigm Community Edition

## Première forme normale

Correction n'est pas en 1FN. Plusieurs attributs restent divisibles.






















Attribut	Problème	Solution
Date	Les données sont présentes sous forme 'mois/année', 'année'. Nous souhaitons faire des requêtes concernant chaque élément (mois, année)	Un attribut pour mois (numeric), Un attribut pour année (numeric)
idx_per	Les données sont présentes sous forme "nom_oeuvre,type_oeuvre".	Un attribut pour nom_personne et un attribut pour type_oeuvre.

## Deuxième forme normale

La table **Correction** est maintenant en 1FN. Les dépendances fonctionnelles sont définies comme suivent:

Attribut maitre	Attributs auxiliaires
Ville	CoordX, CoordY
Date	DateMois, DateAnnee
Idx_Pers	OeuvreNom, OeuvreType
Cliche	N_V, C_G, TailleCli
ReferenceCindoc, Serie	Sujet, NoteBP, FicNum, DescDet, Remarques

Elle est également en 2FN car il n'y a pas de DF sur une partie de la clé. mais pas en 3FN, car il y a des DF hors de la clé.

Correction			
	ReferenceCindoc	varchar	integer(10) N
	Serie	varchar	varchar(255) N
	Article		integer(10)
	Discriminant		integer(10)
	Ville	varchar	varchar(255) N
	CodePostal	varchar	varchar(5) N
	Sujet	varchar	varchar(255) N
	DescDet	varchar	varchar(255) N
	DateMois	integer	integer(2) N
	DateAnnee	integer	integer(4) N
	NoteBP	varchar	varchar(255) N
	FicNum	varchar	varchar(22) N
	OeuvreNom	varchar	varchar(255) N
	OeuvreType	varchar	varchar(255) N
	Idx_Ico	varchar	varchar(255) N
	TailleCli	varchar	varchar(255) N
	N_V	varchar	varchar(3) N
	C_G	varchar	varchar(3) N
	Remarques	varchar	varchar(255) N
	CoordX	float	float(10) N
	CoordY	float	float(10) N

## Troisième forme normale

Pour obtenir la table Correction en 3FN, il faut dans un premier temps diviser chacune des DF en des tables distinctes. Ainsi on obtient les tables suivantes :

Nom de table	Colonnes de la table
Ville	<u>idVille</u> , nomVille, CoordX, CoordY
Date	<u>idDate</u> , DateMois, DateAnnee
Oeuvre	<u>idOeuvre</u> , OeuvreNom, OeuvreType
Cliche	<u>idCliche</u> , N_V, C_G, TailleCli
Photo	<u>idPhoto</u> , Sujet, NoteBP, FicNum, DescDet, Remarques
Article	<u>idArticle</u> , <u>Discriminant</u> , ReferenceCindoc, Serie, # idPhoto, #idCliche, #idOeuvre, #idDate, #idVille

Ainsi cet ensemble de données est en 3FN car on a transformé toutes les dépendances fonctionnelles existantes en tables qui sont référencées entre elles par des clés étrangères.

Toutefois cette base peut encore être optimisée en terme d'espace car il reste des redondances d'informations au sein des nouvelles tables, en divisant à nouveau les informations nous obtenons les tables suivantes :

Nom de table	Colonnes de la table
Ville	<u>idVille</u> , nomVille, CoordX, CoordY
Date	<u>idDate</u> , DateMois, DateAnnee
IndexPersonne	<u>idOeuvre</u> , nomOeuvre, #idTypeOeuvre
Photo	<u>Article</u> , <u>Discriminant</u> , NoteBP, FicNum, DescDet, NbreCli, Remarques, ReferenceCindoc, N_V, C_G, #idVille, #idDate, #idSerie.
Sujet	<u>idSujet</u> , nomSujet
TypeOeuvre	<u>idTypeOeuvre</u> , nomTypeOeuvre
Serie	<u>idSerie</u> , nomSerie .
Cliche	<u>idCliche</u> , Taille
Cliche_Photo	<u>idCliche</u> , <u>PhotoArticle</u> , <u>PhotoDiscriminant</u>

IndexIconographique_Photo	<u>idlco</u> , <u>PhotoArticle</u> , <u>PhotoDiscriminant</u>
Photo_Sujet	<u>PhotoArticle</u> , <u>PhotoDiscriminant</u> , <u>SujetIdSujet</u>

Pour donner un exemple de ce à quoi ces tables servent, prenons le cas de Sujet. Comme le sujet est souvent composé de plusieurs informations dans la table Article, il est judicieux de créer une table Sujet contenant un id et le sujet afin d'éviter les répétitions d'un même sujet qui serait plus coûteuse que la répétition d'un id (entier).

### Diagramme final :

