

PROJET SGBD

Emission des pièces en euro

Étape : Conception de la BD

(Modèle Entité/Association & Implantation/Alimentation)

**Massiles
Julien
Clément
Dorian**

**GHERNAOUT
FURET
FLAMBARD
LEMERCIER**

Table des matières:

Table des matières:	2
Description du projet:	3
Références :	3
Schéma/Modèle entités associations:	4
Description des entités:	5
Première entité, PieceModele:	5
Deuxième entité, PieceCaracteristique:	7
Troisième entité, PieceTranche :	9
Quatrième entité, PiecePays:	11
Cinquième entité, Collectionneur:	13
Description des associations:	15
Première association, Definir:	15
Deuxième association, Avoir:	16
Troisième association, Emettre:	17
Quatrième association, Collection:	18
Modèle Relationnel:	20
Comparaison avec le modèle de l'étape précédente:	21
Observations:	22
Conclusion:	23

Description du projet:

Le projet consiste à créer une base de données pour modéliser les émissions des pièces en euro.

Dans ce rapport, on retrouvera une décomposition suivant le modèle entité/association des informations autour de ce thème.

Le modèle entité/association concrétise une répartition de l'information qui assure une facilité au niveau d'implémentation/alimentation/maintenance de la base de données.

Références :

[Lien vers la description du sujet telle qu'elle est donnée dans l'énoncé.](#)

[Lien vers les attendus de l'étape 1 du projet.](#)

[Lien vers travail effectué lors de l'étape 1 du projet.](#)

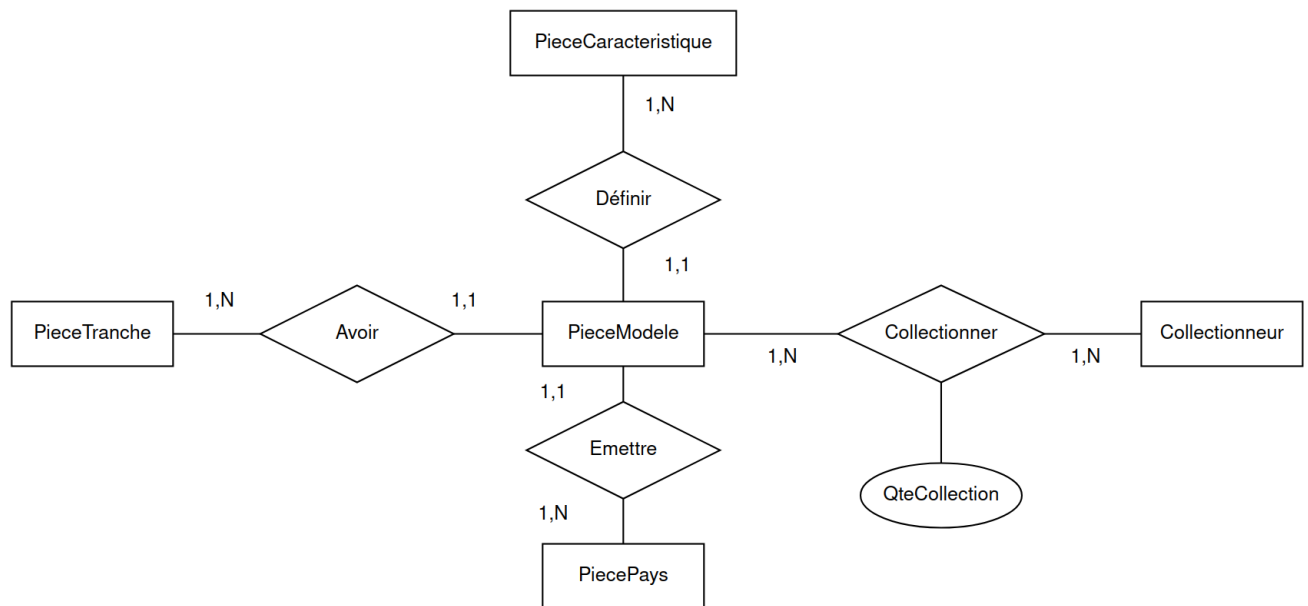
[Lien vers les attendus de l'étape 2 du projet.](#)

Remplissage de la Base de Données :

https://fr.wikipedia.org/wiki/Tirage_des_pi%C3%A8ces_de_monnaie_en_euro

Schéma/Modèle entités associations:

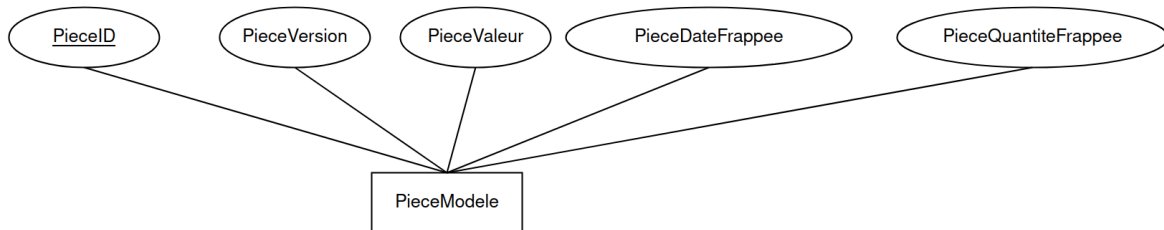
Note: Contrairement à la première étape, on a plutôt décidé de nommer la relation qui concrétise l'entité de Pièce comme PieceModele, étant donné que c'est plus expressif et pertinent. Nous avons aussi pris la liberté d'enlever la relation PieceEmission de l'étape 1 et de la regrouper avec PieceModele.



Description des entités:

Première entité, PieceModele:

Schéma/Visuel:



Description:

Elle correspond au modèle de la pièce, et pas à une pièce physique. Une autre façon, moins formelle, de voir ça est de le considérer comme le moule avec lequel la pièce a été frappée.

Attributs:

- PieceID: C'est la clé primaire, ça identifie un modèle.

*Clé primaire, de type entier, qui s'auto-incrément.
Par conséquent, unique et non nulle.*

- PieceVersion: C'est la version du modèle, comme la version Mariane française.

Chaîne de caractère, non nulle.

- PieceValeur: C'est la valeur qu'aura toutes pièces suivant le modèle courant.

De type entier, non nul et supérieur à 0. On se satisfera par le type entier et cela en utilisant des valeurs en centimes d'euro.

Par ailleurs, on pourra ajouter une contrainte d'appartenance à un ensemble de valeurs bien définies, pour ne pas se retrouver avec des pièces à valeurs monétaires non-usuelles.

Par contre, cela limitera l'utilisation de notre base de données dans le futur, dans le cas où une nouvelle pièce aura une valeur monétaire non incluse dans notre ensemble.

- *PieceDateFrappee: C'est la date d'utilisation du modèle. (C'est aussi la date ou le modèle est sorti, car sans production de pièces, le modèle de celles-ci n'existe pas encore.)*

De type date, non nulle.

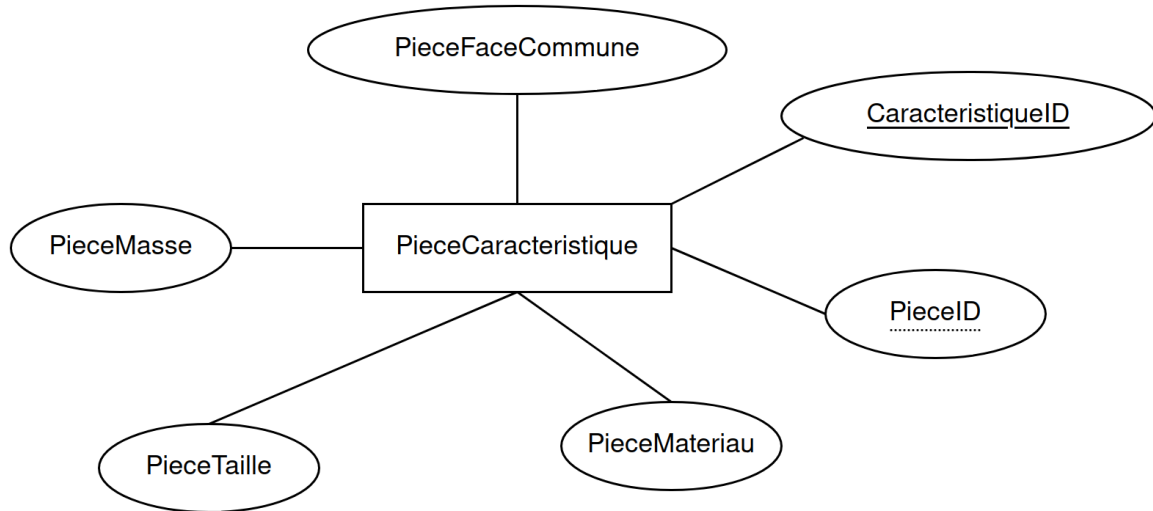
On peut ajouter une contrainte sur les valeurs des dates. Par exemple: la date doit être après Janvier 1999 et avant l'année d'aujourd'hui.

- *PieceQuantiteFrappee: C'est le nombre de fois que le modèle courant a été frappé.*

De type entier, potentiellement entier long, non nul et supérieur à 0.

Deuxième entité, PieceCaracteristique:

Schéma/Visuel:



Description:

Cette relation regroupe toutes les caractéristiques physiques d'un modèle de pièce.

Attributs:

- CaracteristiqueID: c'est la clé primaire de notre table, et identifie un ensemble de caractéristiques attribuables à un modèle de pièce.

Clé primaire de type entier qui s'auto-incrémente. Par conséquent, non nulle et unique.

- #PieceID: c'est une clé étrangère qui fait référence à la clé primaire de la relation PieceModele et donc cela nous permet d'établir une relation 1-N entre ces deux relations.

Se référer à la table PieceModele précédemment définie.

- PieceFaceCommune: C'est la description de la face 'pile' (avec la valeur) de la pièce qui peut être commune à d'autres pièces malgré une version différente.

Chaîne de caractère, non nulle.

- PieceMasse: la masse des pièces suivant le modèle qui regroupe les caractéristiques courantes.

De type entier, non nul et supérieur à 0. On se satisfera du type entier et cela en utilisant des valeurs en gramme arrondies.

On peut potentiellement ajouter des contraintes de valeurs. Par exemple: la masse ne doit pas dépasser les 100g.

- PieceTaille: la taille du diamètre des pièces suivant le modèle qui regroupe les caractéristiques courantes.

De type entier, non nul et supérieur à 0. On se satisfera du type entier et cela en utilisant des valeurs en millimètre arrondies.

On peut potentiellement ajouter des contraintes de valeurs. Par exemple: la taille ne doit pas dépasser les 1000 mm.

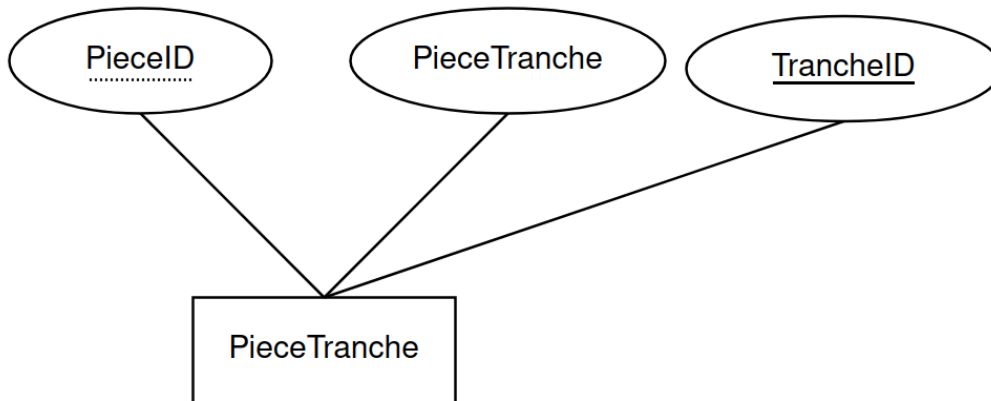
- PieceMateriau: le matériau de construction des pièces suivant le modèle qui regroupe les caractéristiques courantes.

Chaîne de caractère, non nul.

Potentiellement, on peut ajouter une contrainte d'appartenance à un ensemble de valeurs bien définie en utilisant un "CHECK IN".

Troisième entité, PieceTranche :

Schéma/Visuel:



Description:

Cette relation stockera toutes les tranches possibles que peut avoir une pièce. Notamment, on peut avoir une tranche lisse, cannelée, striée, chevronnée, ornée de motifs, de symboles ou d'écritures en relief et/ou en creux.

Attributs:

- #PieceID: ceci est une clé étrangère qui fait référence à la clé primaire de la relation PieceModèle, et donc avec cet attribut; on arrive à mettre en place une relation 1-N entre ces deux relations.

Pour le type, il faut se référer à la relation PieceModele, précédemment définie.

- TrancheID: ceci est la clé primaire de cette relation, et elle nous permet d'identifier une tranche spécifique.

Clé primaire de type entier, non nulle et qui s'auto incrémente.

- PieceTranche: ceci stock l'information de la tranche.

De type chaîne de caractère, non nulle.

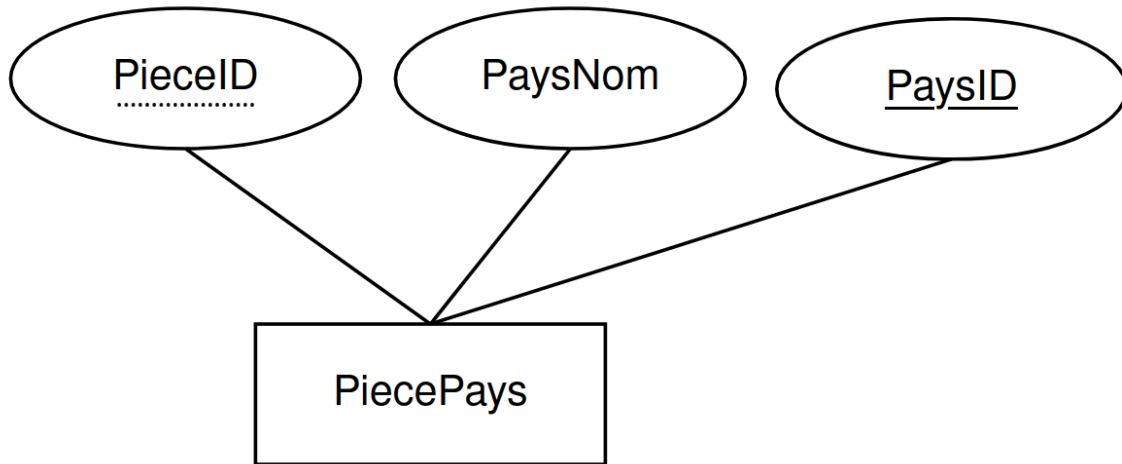
Potentiellement, on pourra ajouter des contraintes d'appartenance à un ensemble de valeurs. Par exemple

on peut s'assurer que la tranche est une des valeurs énoncées lors de la description de la table.

Note: Dans cette relation, on pourra se passer de TrancheID, car en réalité PieceTranche sera unique pour chaque modèle de pièce. Par contre, on a décidé de l'ajouter à la relation pour être consistant sur la création des relations/tables.

Quatrième entité, PiecePays:

Schéma/Visuel:



Description:

Cette relation stockera les pays qui ont émis un modèle de pièces.

Attributs:

- #PieceID: ceci est une clé étrangère qui fait référence à la clé primaire de la relation PieceModèle, et donc avec cet attribut, on arrive à mettre en place une relation 1-N entre ces deux relations.

Pour le type, il faut se référer à la relation PieceModele, précédemment définie.

- PaysID: ceci la clé primaire de cette relation et elle nous permet d'identifier un pays.

Clé primaire de type entier, qui s'auto-incrémente.

- PaysNom: ceci stockera le nom du pays.

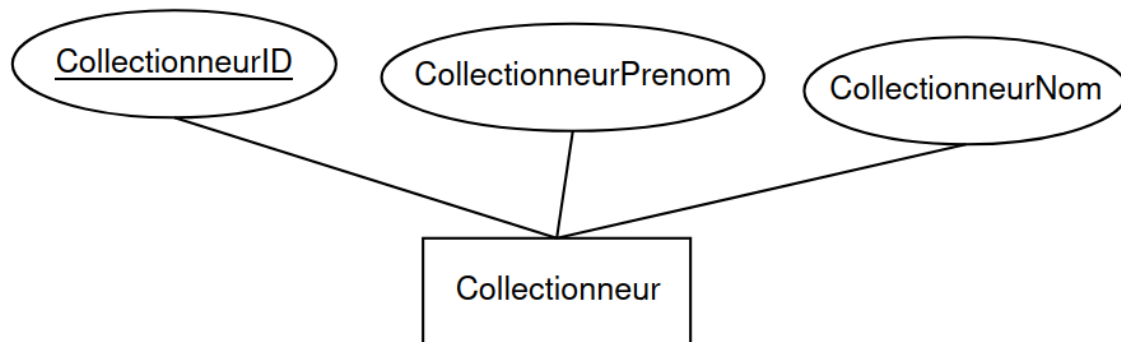
Chaîne de caractère non nulle.

On peut potentiellement ajouter une contrainte d'appartenance à un ensemble de valeurs qui s'assurerait que le nom du pays est bien dans l'Europe/EU.

Note: Dans cette relation, on pourra se passer de PaysID car en réalité PaysNom sera unique pour chaque modèle de pièce. Par contre, on a décidé de l'ajouter à la relation pour être consistant sur la création des relations/tables.

Cinquième entité, Collectionneur:

Schéma/Visuel:



Description:

Cette relation stockera les différents collectionneurs de pièces.

Attributs:

- CollectionneurID: C'est notre clé primaire, et elle permet d'identifier un collectionneur.

Clé primaire de type entier, qui s'auto-incrémente. Par conséquent non nulle et unique.

- CollectionneurNom: C'est le nom du collectionneur.

Chaîne de caractère non nulle.

- CollectionneurPrenom: C'est le prénom du collectionneur.

Chaîne de caractère non nulle.

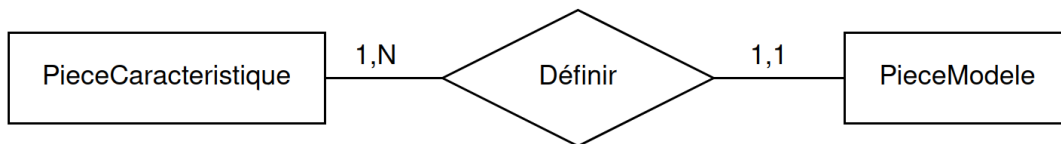
Note: on aurait pu aussi ajouter la date de naissance, et d'autres attribut supplémentaires, mais le thème est sur les pièces en euro, donc l'entité "collectionneur" n'est pas aussi importante pour l'enrichir en informations.

Par ailleurs, avec l'état de la table actuel, on peut déduire un attribut "virtuel" qui est le nom complet en concaténant le nom et le prénom.

Description des associations:

Première association, Définir:

Schéma/Visuel:



Description:

Les pièces et leurs modèles par conséquent ont des caractéristiques spécifiques qui définissent le modèle et aussi les pièces physiques issues de celui-ci.

Attributs:

C'est une association 1-N, donc l'association ne peut avoir d'attributs.

Cardinalité:

Un ensemble de caractéristiques peut décrire/définir un ou plusieurs modèles de pièces.

Un modèle de pièce peut être décrit/défini par un seul groupe de caractéristiques.

Deuxième association, Avoir:

Schéma/Visuel:

**Description:**

Une pièce a forcément une tranche. Par ailleurs, toutes les pièces qui suivent le même modèle ont la même tranche en conséquence. Par de ce fait, un modèle est associé à une tranche.

Attributs:

C'est une association 1-N, donc l'association ne peut avoir d'attributs.

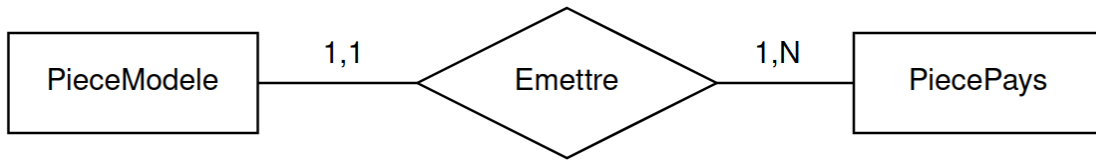
Cardinalité:

Chaque modèle de pièce a une seule tranche.

Chaque tranche peut être attribuée à une ou plusieurs types de pièces physiques, et à leurs modèles par conséquent.

Troisième association, Emettre:

Schéma/Visuel:

**Description:**

Un pays émet des pièces en les fabriquant, et pas conséquent, le modèle de ces pièces est émis par ce même pays.

Attributs:

C'est une association 1-N, donc l'association ne peut avoir d'attributs.

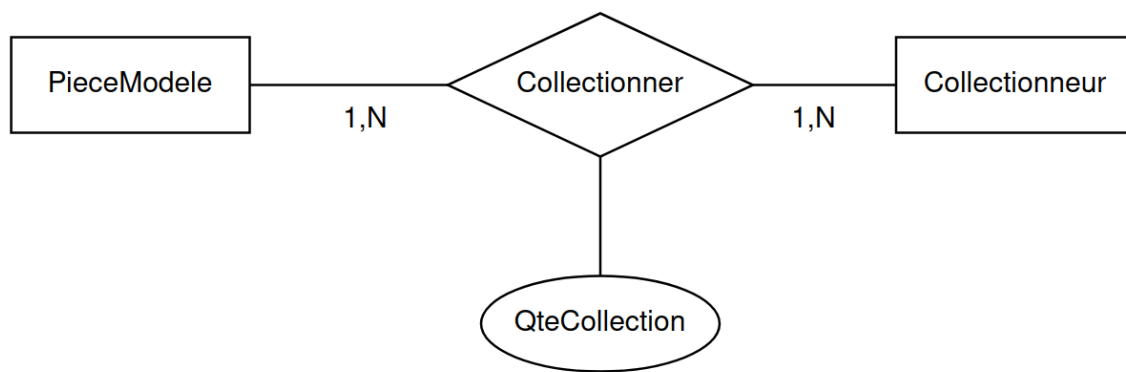
Cardinalité:

Chaque modèle est émis par un seul pays.

Chaque pays peut émettre un ou plusieurs modèles.

Quatrième association, Collection:

Schéma/Visuel:



Description:

Un collectionneur est un passionné de pièces, et il compte en collectionner le plus possible.

Chaque collectionneur rassemble un tas de pièces physiques d'un modèle donné qui lui fait une collection suivant ce modèle.

Les collectionneurs ont tendance à collectionner plusieurs choses, par conséquent, chaque collectionneur peut avoir plusieurs collections, ce qui signifie la présence de plusieurs modèles.

Attributs:

Ceci est une association N-M, ce qui signifie qu'on aura une table de jointure qui la concrétise dans notre base de données.

Par conséquent, on peut bien avoir des attributs, et dans ce cas on a "QteCollection" comme un attribut.

Cet attribut ajoutera l'information de volume à une collection d'un certain collectionneur vis-à-vis d'un modèle de pièce.

Cet attribut sera de type entier, et doit être supérieur à 0 et donc non nulle.

Cardinalité:

Chaque collectionneur peut avoir une ou plusieurs collections de pièces, par conséquent, il collectionne potentiellement plusieurs modèles de pièces.

Chaque modèle de pièce, une fois émis, peut se retrouver dans les collections de différents collectionneurs.

Par ailleurs, le nombre de collectionneurs ayant des pièces de ce modèle suit/indique la rareté de ce modèle.

Modèle Relationnel:

PieceModele(

PieceID: clé primaire entier,
 PieceVersion: chaîne de caractère,
 PieceValeur: entier,
 PieceDataFrappee: date,
 PieceQuantiteFrappee: entier long,
)

PieceCaracteristique(

CaracteristiquesID: clé primaire entier,
 #PieceID: clé étrangère entier,
 PieceFaceCommune: chaîne de caractères,
 PieceMateriau: chaîne de caractères,
 PieceMasse: entier,
 PieceTaille: entier,
)

PieceTranche(

 #PieceID: clé étrangère entier,
 TrancheID: clé primaire entier,
 PieceTranche: chaîne de caractères,
)

PiecePays(

PaysID: clé primaire entier,
 #PieceID: clé étrangère entier,
 PaysNom: chaîne de caractères,
)

Collectionneur(

CollectionneurID: clé primaire entier,
 CollectionneurNom: chaîne de caractères,
 CollectionneurPrenom: chaîne de caractères,
)

Collectionner qui est notre table de jointure pour la relation N-M entre Collectionneur et PieceModele.

Collection(

 #CollectionneurID: clé étrangère entier,
 #PieceID: clé étrangère entier,
 (#CollectionneurID, #PieceID):
 clé primaire (entier, entier)
 QteCollection: entier,
)

Comparaison avec le modèle de l'étape précédente:

Lors de l'étape précédente, les relations obtenues étaient plus centrées sur la redondance est la façon dont les données sont réparties que l'aspect implémentation et pratique.

Au bout de la deuxième étape, nous pensons que le modèle Entité/Association permet d'avoir une approche plus pragmatique et plus simple à comprendre et donc à maintenir.

Aussi, le fait que tout se joue autour des clés primaires/clés étrangères permet de totalement abstraire le contenu des relations et de pouvoir changer celui-ci sans avoir à reconstruire la base de données, il est juste nécessaire d'adapter les requêtes à nos changements. Toutefois, cela ne s'applique pas aux relations obtenues lors de l'étape précédente, car celles-ci sont construites à partir de dépendances fonctionnelles qui elles même sont construites à partir des attributs/contenu de la relation.

Un autre point fort du modèle entité/association est le fait qu'il adopte une approche du haut vers le bas de la pyramide "top-to-bottom approach", qui permet de se focaliser sur les entités et leurs associations en premier lieu et puis à la répartition des attributs et de l'information. Cela permet de respecter le jargon métier plus facilement et de suivre le principe "domain driven design/development".

Observations:

Pour ce qui est des observations/cas limites, nous voulons noter:

- Le fait que l'Union européenne peut changer, peut causer une incohérence au niveau des informations présentes dans la base. Par contre, la date pourra expliquer ces incohérences. De même pourra être dit pour quelques autres attributs, (la valeur par exemple).
- La notion de version n'est pas aussi claire et nette que les autres, et nous comptons sur la phase de l'alimentation pour l'éclaircir.

Conclusion:

Pour conclure, ce rapport est à la fois la synthèse, mais aussi la documentation. Par conséquent, le code fourni à côté, reflétera ce qui est exprimé/défini dans ce rapport.

Par ailleurs, nous voulons bien marquer le fait que le modèle entité-association est plus simple à mettre en place, à adapter, et bien-sûr à implémenter.