

PROJET SGBD
émission des pièces en euro
Étape : Conception de la BD
(Modèle Logique)

GROUPE :

Massiles GHERNAOUT

Julien FURET

Clément FLAMBARD

Dorian LEMERCIER

Table des matières :

Description de la relation mère:

Relation:

Description des domaines de définition des attributs:

Couverture minimale de la relation mère:

Ensemble de dépendances fonctionnelles :

Explication et description des dépendances fonctionnelles:

Clé minimale:

Forme normale de la relation mère:

Algorithme de Bernstein - Décomposition de la relation mère:

- 1 - Première étape: décomposer l'ensemble des dépendances fonctionnelles en sous ensembles qui contiennent que des dépendances fonctionnelles avec le même côté gauche:
- 2 - Créer une relation pour chaque sous ensemble:
- 3 - Vérifier que les relations obtenues sont en 3NF:
- 4 - Vérifier que la clé minimale de la relation mère est bien dans une des relations obtenues, si non, l'ajouter dans une relation à elle seule:
- 5 - Voir si les relations obtenues sont en BCNF:
- 6 - Conclusion sur l'algorithme de Bernstein:

Description de la relation mère :

Relation:

PieceEuro(

PieceID, PieceValeur, PieceMateriau, PieceDateFrappe, PieceVersion,
PieceTaille, PieceMasse, PieceTranche, PieceFaceCommune,
PieceQuantiteFrappe, PaysNom, CollectionneurID, CollectionneurNom,
CollectionneurPrenom, CollectionneurNombrePieces
)

Description des domaines de définition des attributs:

PieceID : entier, correspond à l'identifiant de la pièce.

PieceValeur : entier, correspond à la valeur monétaire de la pièce.

PieceMateriau : chaîne de caractères, correspond au matériau utilisé lors de la fabrication de la pièce.

PieceDateFrappe : date, correspond à la date où la pièce a été frappée.

PieceVersion : chaîne de caractères, correspond à la série d'édition de la pièce.

PieceTaille : entier, correspond à la taille de la pièce soit son diamètre.

PieceMasse : entier, correspond au poids de la pièce.

PieceTranche : chaîne de caractères, correspond à la caractéristique de la tranche de la pièce.

PieceFaceCommune : chaîne de caractères, correspond à la face en commun de toutes les pièces en euros déterminées par la

PieceQuantiteFrappe : entier, correspond au volume d'émission d'une pièce.

PaysNom : chaîne de caractère, correspond au nom du pays.

CollectionneurID : entier, correspond à l'identifiant du collectionneur.

CollectionneurNom : chaîne de caractère, correspond au nom du collectionneur.

CollectionneurPrenom : chaîne de caractère, correspond au prénom du collectionneur.

CollectionneurNombrePieces : entier, correspond au nombre de pièces que le collectionneur possède.

Couverture minimale de la relation mère :

Ensemble de dépendances fonctionnelles :

PieceValeur, PieceDateFrappe -> PieceFaceCommune.

PieceVersion -> PaysNom.

PieceValeur, PieceDateFrappe -> PieceMateriau, PieceMasse, PieceTaille.

PieceVersion, PieceValeur, PieceDateFrappe -> PieceQuantiteFrappe.

PieceVersion, PieceValeur -> PieceTranche.

PieceID -> PieceValeur, PieceVersion, PieceDateFrappe, PieceMasse, PieceTaille,
PieceTranche, PieceMateriau, PieceFaceCommune, PieceQuantiteFrappe.

CollectionneurID -> CollectionneurPrenom, CollectionneurNom

PieceID, CollectionneurID -> CollectionneurNombrePieces

Explication et description des dépendances fonctionnelles:

1. PieceID -> PieceValeur, PieceVersion, PieceDateFrappe, PieceMasse, PieceTaille, PieceTranche, PieceMateriau, PieceFaceCommune, PieceQuantiteFrappe.

Explication:

PieceID c'est la clé et l'identifiant d'une pièce, donc c'est tout à fait normal qu'elle détermine tout ce que la pièce a.

Note: de même pour CollectionneurID.

2. PieceID, CollectionneurID -> CollectionneurNombrePieces.

Explication:

Pour connaître le volume d'une collection quelconque, on a juste besoin de son propriétaire et de l'objet de la collection.

3. PieceValeur, PieceDateFrappe -> PieceFaceCommune.

Explication:

Normalement, les pièces en euro de même valeurs ont la même face commune, mais il s'avère que lors d'un changement majeur au sein de l'UE, cette face commune soit modifiée.

4. PieceVersion -> PaysNom.

Explication:

Chaque version est émise par un unique pays.

5. PieceValeur, PieceDateFrappe -> PieceMateriau, PieceMasse, PieceTaille.

Explication:

Selon la valeur de la pièce et la date de fabrication, les caractéristiques de la pièce varient, soient le matériau, la masse et la taille.

6. PieceVersion, PieceValeur, PieceDateFrappe -> PieceQuantiteFrappe.

Explication:

Connaissant la version d'une pièce, sa valeur et sa date de fabrication, on peut retrouver le volume d'émission.

7. PieceVersion, PieceValeur -> PieceTranche.

Explication:

La tranche de la pièce (relief ou/et inscription) dépend exclusivement de sa valeur et aussi de sa version.

Clé minimale:

- (PieceID, CollectionneurID)

Justification:

PieceID nous permet d'avoir tous les attributs sur **Piece**, mais pas les attributs sur le **Collectionneur** ou la **Collection**.

CollectionneurID nous permet d'avoir tous les attributs sur le **Collectionneur** mais pas sur **Piece** ou la **Collection**.

Donc, avec (**PieceID**, **CollectionneurID**) on peut avoir accès à tous les attributs de notre relation mère. C'est notre point d'entrée vers le graph de notre relation mère.

Forme normale de la relation mère:

1NF ? oui car tous les attributs ont des types simples.

2NF ? non, car il y a des attributs qui dépendent d'une partie de la clé. Notamment **PieceValeur** qui dépend de **PieceID**, ou bien **CollectionneurNom** qui dépend de **CollectionneurID**.

3NF ? Non. Même pas en 2NF.

Algorithme de Bernstein - Décomposition de la relation mère :

1 - Première étape: décomposer l'ensemble des dépendances fonctionnelles en sous ensembles qui contiennent que des dépendances fonctionnelles avec le même côté gauche:

On nommera nos ensembles avec des lettres de l'alphabet.

A := {
PieceID -> PieceValeur, PieceVersion, PieceDateFrappe.
}

B := {
PieceValeur, PieceDateFrappe -> PieceFaceCommune
PieceValeur, PieceDateFrappe -> PieceMateriau, PieceMasse, PieceTaille.
}

C := {
PieceVersion, PieceValeur -> PieceTranche.
}

D := {
PieceVersion -> PaysNom.
}

E := {
PieceVersion, PieceValeur, PieceDateFrappe -> PieceQuantiteFrappe
}

F := {
CollectionneurID -> CollectionneurPrenom, CollectionneurNom
}

G := {
PieceID, CollectionneurID -> CollectionneurNombrePieces
}

2 - Créer une relation pour chaque sous ensemble:

Pour A :

On nomme la relation **Piece**, et elle aura les attributs ci-dessous:

Piece(PieceID, PieceValeur, PieceVersion, PieceDateFrappe)

Pour B :

On nomme la relation **PieceCaracteristique**, et elle aura les attributs ci-dessous:

PieceCaracteristique(PieceValeur, PieceDateFrappe, PieceFaceCommun, PieceMateriau, PieceMasse, PieceTaille)

Pour C :

On nomme la relation **PieceTranche**, et elle aura les attributs ci-dessous:

PieceTranche(PieceVersion, PieceValeur, PieceTranche)

Pour D :

On nomme la relation **PiecePays**, et elle aura les attributs ci-dessous:

PiecePays(PieceVersion, PaysNom)

Pour E :

On nomme la relation **Emission**, et elle aura les attributs ci-dessous:

Emission(PieceVersion, PieceValeur, PieceDateFrappe, PieceQuantiteFrappe)

Pour F :

On nomme la relation **Collectionneur**, et elle aura les attributs ci-dessous:

Collectionneur(CollectionneurID, CollectionneurPrenom, CollectionneurNom)

Pour G :

On nomme la relation **Collection**, et elle aura les attributs ci-dessous:

Collection(PieceID, CollectionneurID, CollectionneurNombrePieces)

3 - Vérifier que les relations obtenues sont en 3NF:

Note: Toutes les relations sont en 1NF.

Piece(PieceID, PieceValeur, PieceVersion, PieceDateFrappe) :

- La clé est (PieceID).
- Elle est en 2NF, car la clé est atomique.
- Elle est en 3NF, car il n'y a pas de dépendance fonctionnelle entre les attributs non clés.

PieceCaracteristique(PieceValeur, PieceDateFrappe, PieceFaceCommun, PieceMateriau, PieceMasse, PieceTaille) :

- La clé est (PieceValeur, PieceDateFrappe).
- Elle est en 2NF, car il y a aucune partie de la clé qui détermine un autre attribut non clé.
- Elle est en 3NF, car il n'y a aucune dépendance fonctionnelle entre les attributs non clés.

PieceTranche(PieceVersion, PieceValeur, PieceTranche) :

- La clé est (PieceVersion, PieceValeur).
- Elle est en 2NF, car il y a aucune partie de la clé qui détermine un autre attribut non clé.
- Elle est en 3NF, car il n'y a aucune dépendance fonctionnelle entre les attributs non clés.

PiecePays(PieceVersion, PaysNom) :

- La clé est (PieceVersion).
- Elle est en 2NF, car la clé est atomique.
- Elle est en 3NF, car il y a qu'un seul attribut non clé.

Emission(PieceVersion, PieceValeur, PieceDateFrappe, PieceQuantiteFrappe) :

- La clé est (PieceVersion, PieceValeur, PieceDateFrappe).
- Elle est en 2NF, car il y a aucune partie de la clé qui détermine un autre attribut non clé.
- Elle est en 3NF, car il y a qu'un seul attribut non clé.

Collectionneur(CollectionneurID, CollectionneurPrenom, CollectionneurNom) :

- La clé est (CollectionneurID).
- Elle est en 2NF, car la clé est atomique.
- Elle est en 3NF, car il n'y a aucune dépendance fonctionnelle entre les attributs non clés.

Collection(PieceID, CollectionneurID, CollectionneurNombrePieces) :

- La clé est (PieceID, CollectionneurID).
- Elle est en 2NF, car il y a aucune partie de la clé qui détermine un autre attribut non clé.
- Elle est en 3NF, car il y qu'un seul attribut non clé.

4 - Vérifier que la clé minimale de la relation mère est bien dans une des relations obtenues, si non, l'ajouter dans une relation à elle seule:

Pour notre cas, la clé minimale de la relation mère est (PieceID, CollectionneurID) et elle est bien présente dans la relation **Collection**.

5 - Voir si les relations obtenues sont en BCNF:

Afin qu'une relation soit en BCNF, il ne faut pas y avoir d'attribut non-clé qui détermine une partie de la clé.

Piece(PieceID, PieceValeur, PieceVersion, PieceDateFrappe) :

- La clé est (PieceID) et aucun autre attribut ne la détermine.
Elle est bien en BCNF.

PieceCaracteristique(PieceValeur, PieceDateFrappe, PieceFaceCommune, PieceMateriau, PieceMasse, PieceTaille) :

- La clé est (PieceValeur, PieceDateFrappe, PieceFaceCommune), et aucun autre attribut ne la détermine.
Elle est bien en BCNF.

PieceTranche(PieceVersion, PieceValeur, PieceTranche) :

- La clé est (PieceVersion, PieceValeur) et aucun autre attribut ne la détermine.
Elle est bien en BCNF.

PiecePays(PieceVersion, PaysNom) :

- La clé est (PieceVersion) et aucun autre attribut ne la détermine.
Elle est bien en BCNF.

Emission(PieceVersion, PieceValeur, PieceDateFrappe, PieceQuantiteFrappe) :

- La clé est (PieceVersion, PieceValeur, PieceDateFrappe), et aucun autre attribut ne la détermine.
Elle est bien en BCNF.

Collectionneur(CollectionneurID, CollectionneurPrenom, CollectionneurNom) :

- La clé est CollectionneurID, et aucun autre attribut ne la détermine.
Elle est bien en BCNF.

Collection(PieceID, CollectionneurID, CollectionneurNombrePieces) :

- La clé est PieceID, CollectionneurID, et aucun autre attribut ne la détermine.
Elle est bien en BCNF.

6 - Conclusion sur l'algorithme de Bernstein :

L'algorithme de Bernstein nous a permis d'avoir des sous-relations à notre relation-mère suivant une décomposition qui assure la troisième forme normale, et dans notre cas, la BCNF est aussi validée.

Par ailleurs, cette décomposition est sans perte de dépendances fonctionnelles, et donc, sans perte d'informations (SPI/SPD)