

Introduction

La création d'un système d'information n'est jamais évidente et c'est même inconcevable sans une bonne méthode. C'est dans cette logique que s'inscrit la méthode MERISE qui permet de structurer et d'organiser les données de manière optimale tant au niveau de l'exploitation des données qu'au niveau des améliorations et de la maintenance de la base de données.

C'est dans cet esprit que nous nous intéresserons à la méthode MERISE dans l'ordre chronologique de la démarche à adopter.

Nous présenterons la méthode MERISE avant d'en étudier le modèle conceptuel de données, ses extensions puis les formes normales. Nous poursuivrons par l'étude du modèle logique de données pour terminer par le modèle physique de données.

I - Présentation générale

- Un modèle conceptuel de données est indépendant de la technologique utilisée.
- Différents Systèmes de Gestion de Bases de Données (SGBD) existent. Ils ont tous des particularités.

A - Présentation de la méthode MERISE

La méthode MERISE est dite méthode d'analyse, de conception et de réalisation de projets. Elle date des années 1978 et 1979. Elle résulte d'une demande du ministère de l'Industrie (en 1977) qui souhaitait obtenir une méthode de conception de système d'information. Ce sont le Centre Techniques de l'Equipement et le Centre d'Etudes techniques de l'Equipement qui sont à l'origine de cette méthode. De plus, la méthode MERISE permet de schématiser des systèmes d'informations parfois complexes et permet par conséquent de clarifier les choses.

Aujourd'hui, 70% des entreprises prétendent l'utiliser pour dans leur démarche de création de base de données relationnelles.

B - Principe de fonctionnement

Elle s'articule sur trois niveaux pouvant eux même comprendre deux modèles :

- Niveau conceptuel :
 - Le modèle conceptuel des données (MCD)
 - Le modèle conceptuel des traitements (MCT)
- Niveau organisationnel :
 - Le modèle logique des données (MLD)
 - Le modèle logique des traitements (MLT)
- Niveau physique

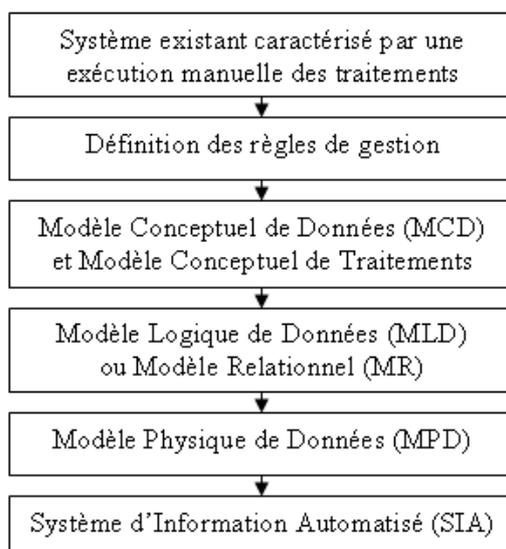
C - Objectifs de la méthode MERISE

Cette méthode d'analyse vise à concevoir un système d'information en séparant le traitement des données pour assurer la pérennité du projet. En effet, cela permet de modifier les données ou les traitements sans pour autant refaire le projet dans sa totalité.

D - Phases de réalisation d'un projet avec la méthode MERISE

La création ou la maintenance d'un système d'information se fait par étapes successives. Chaque étape doit être approuvée pour passer à la suivante.

Présentons un schéma des phases de réalisation d'un projet avec la méthode MERISE :



La définition des règles de gestion correspond au recueil des besoins. Cela permet d'établir les objectifs du SI en listant les données à y intégrer et en cherchant à le circonscrire.

La création du MCD permet de prendre en considération les règles de gestions énoncés ainsi que de structurer les données entre elles. Le MCD se nomme également modèle Entité-Association. Le MCT permet de lister les traitements à effectuer.

Le modèle logique correspond à la création des relations à partir du MCD.

Le modèle physique correspond à l'implémentation du MLD sur un Serveur de Bases de Données Relationnelles SGBDR.

E - Vocabulaire

A chaque phase de la réalisation du projet correspond certains termes ou expressions :

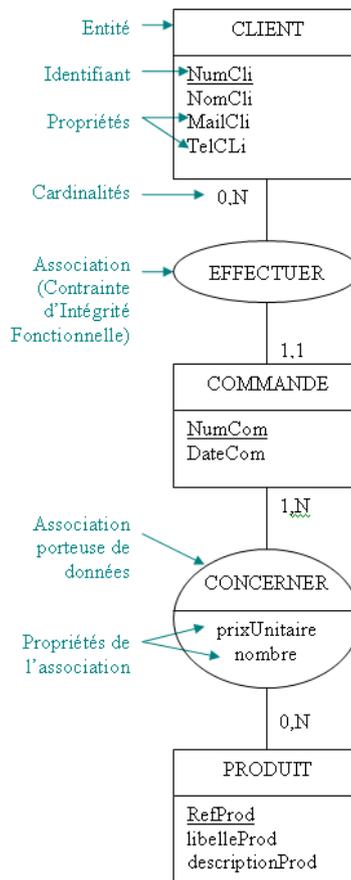
Niveau Conceptuel	Niveau Logique	Niveau Physique
Entité	Relation	Table
Propriété	Attribut	Champ
Identifiant	Attribut clé primaire	Champ
Association 1,1 - 1,N	Attribut clé étrangère	Champ
Association 1,N - 1,N ou supérieur	Relation avec les clefs étrangères en clefs primaire	Table
Occurrence	n-uplet	Enregistrement

II - Le modèle Conceptuel de Données (MCD)

A - Présentation du MCD

Le MCD se repose sur le modèle Entité - Association. Ce dernier permet de représenter de manière schématique la manière dont s'articulent les données à implémenter par la suite.

Pour la compréhension de cette page, vous pourrez regarder le MCD suivant :



B - L'entité

L'entité permet de modéliser un ensemble d'objets concrets ou abstraits de même nature ou un ensemble d'individus de même nature et ayant un intérêt pour le domaine étudié. L'entité est nommée à l'aide d'un substantif dans le MCD. Elle compte une ou plusieurs propriétés au sein desquels on trouve nécessairement son identifiant. Chaque élément d'une entité est nommé "occurrence".

C - La propriété

C'est la plus petite partie d'information manipulée et qui permet de caractériser l'association ou l'entité. La propriété peut être :

- simple : un numéro de client, un nom, une adresse mail.
- composée : une date, une adresse.

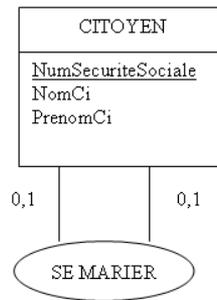
D - L'identifiant

L'identifiant est une propriété de l'entité qui permet l'identification de chacune des occurrences de l'entité. Cette propriété est soulignée dans le MCD.

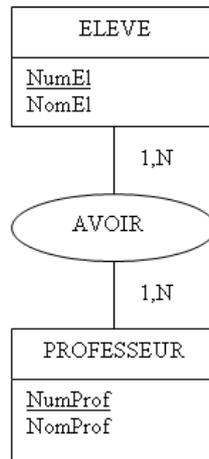
E - L'association

L'association permet de relier une ou plusieurs entités entre elles. Ces liaisons sont énoncées via les règles de gestion. Contrairement à l'entité, l'association se nomme avec un verbe. Il existe différents type d'associations.

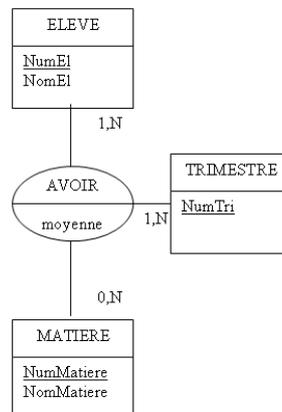
1 - L'association réflexive (ou récursive) :



2 - L'association binaire (ou à 2 pattes) :



3 - L'association ternaire (ou à 3 pattes) :



F - Les cardinalités

Les cardinalités se trouvent sous la forme d'un couple de valeurs (minimum, maximum).

- La cardinalité minimum correspond au nombre minimal de fois que chaque occurrence de l'entité participe aux occurrences de l'association. Elle prend généralement la valeur 0 ou 1.
- La cardinalité maximum correspond au nombre maximal de fois où chaque occurrence de l'entité participe aux occurrences de l'association. Elle est au moins égale à 1. L'infini est noté "n".

III - Concepts avancés de la méthode MERISE

Dans un souci de regrouper les extensions de la méthode MERISE, cette partie vous proposera les concepts avancés aux niveaux conceptuel et relationnel.

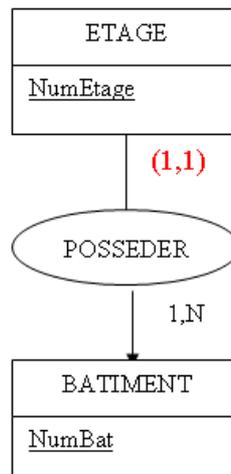
A - L'identification relative

1 - Présentation

Si un identifiant ne comporte que des propriétés de son entité, on le nomme "identifiant absolu". Les identifiants absolus se rencontrent dans le cas d'entités définies indépendamment les unes des autres. D'autres entités sont identifiées par l'intermédiaire d'une ou plusieurs autres entités. Cela s'appelle "l'identification relative" ou encore "agrégation".

- L'entité permettant l'identification est nommée "entité agrégeante".
- L'entité identifiée se nomme "entité agrégée".

L'identification relative se note de la manière suivante :



Remarque : l'identification relative n'existe que si les cardinalités exprimant l'identification relative sont (1,1) et s'il y a stabilité dans le temps. (Un étage ne peut pas changer de bâtiment).

2 - Passage au relationnel

Les règles de passage au schéma relationnel s'appliquent pour obtenir :

R1 : ETAGE(NumEtage, #NumBat);

R2 : BATIMENT(NumBat);

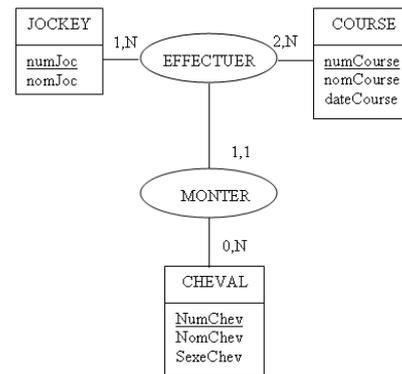
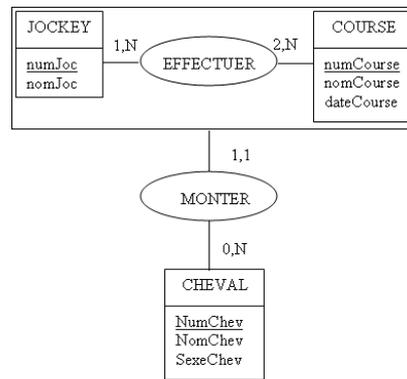
B - L'agrégation

1 - Présentation

Elle consiste à considérer globalement des entités et une association comme un objet unique. Cet objet peut ensuite être lié à d'autres entités ou agrégations via des relations. On peut également parler "d'association d'associations".

L'agrégation peut être représentée de deux manières différentes.

Un exemple d'agrégation est le suivant :



2 - Passage au relationnel

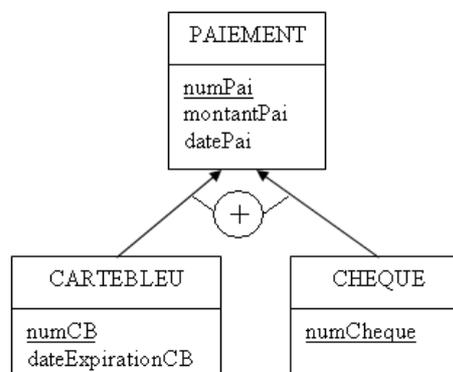
Les règles de passage au schéma relationnel s'appliquent pour obtenir :

- R1 : JOCKEY(numJoc, nomJoc);
- R2 : COURSE(numCourse, nomCourse, dateCourse);
- R3 : CHEVAL(NumChev, NomChev, SexeChev);
- R4 : EFFECTUER(#numJoc, #numCourse, #numChev);

C - La généralisation - spécialisation

1 - Présentation

Il peut être utile de factoriser des données communes à plusieurs entités au sein d'une seule entité mère. Par exemple, une entreprise souhaite enregistrer les moyens de paiements de ses clients. Ces derniers peuvent être faits en chèque ou par carte bleue.



On dit que les entités CHEQUE et CARTEBLEU sont des sous-types de l'entité PAIEMENT (qui est le sur-type).

On peut aussi dire que les entités CHEQUE et CARTEBLEU sont des entités spécialisées issues de l'entité générique (PAIEMENT).

Les entités spécialisées héritent des attributs de l'entité générique.

2 - Passage au modèle relationnel

Le passage de la généralisation - spécialisation au schéma relationnel est un peu plus complexe que celui des concepts évoqués précédemment. En effet, il existe trois solutions.

a - Première solution

On traduit uniquement l'entité générique pour obtenir :

R1 : PAIEMENT(numPai, montantPai, datePai, #numCB, dateExpirationCB, #numCheque);

b - Seconde solution

On traduit les entités spécialisées pour obtenir :

R1 : PAIEMENT_CARTEBLEU(numPai, montantPai, datePai, #numCB, dateExpirationCB);

R2 : PAIEMENT_CHEQUE(numPai, montantPai, datePai, #numCheque);

c - Troisième solution

On traduit les entités spécialisées et l'entité générique

R1 : PAIEMENT(numPai, montantPai, datePai);

R2 : CARTEBLEU(numCB, dateExpirationCB, #numPai);

R3 : CHEQUE(numCheque, #numPai);

C'est la troisième solution qui est généralement retenue car elle conserve les avantages de la généralisation - spécialisation pour le modèle relationnel puis physique. De plus, elle est plus évolutive que les deux solutions précédentes même si elle est plus complexe à utiliser en SQL.

3 - Contraintes de couverture et de disjonction

a - Définitions

Il y a couverture si chaque occurrence d'une entité générique est une occurrence d'une entité spécialisée.

Il y a disjonction si aucun objet ne peut être une occurrence de plusieurs entités spécialisées.

b - Notations

Comme vous avez pu le voir dans le schéma ci-dessus (III C 1), il existe différentes notations en fonction de la couverture et de la disjonction.

En voici un tableau récapitulatif :

	Couverture	Non couverture
Disjonction	Partition notée "+"	Exclusion notée "X"
Non Disjonction	Totalité notée "T"	Pas de contrainte

IV - Les formes normales

A - Problématique

Il peut arriver qu'il existe une redondance des informations. Ceci augmente la consommation en mémoire de la base de données ainsi que la saisie des données. De plus, la mise à jour des données peut être longue voir incohérente dans certains cas. Il apparaît alors nécessaire de normaliser les relations. Dans le meilleur des cas, ces problèmes doivent être détectés au niveau conceptuel voir logique. Cette normalisation va passer par l'application d'un ensemble de règles sur l'ensemble des relations.

B - Présentation

Dans les années 1970, Codd a mis en place l'ensemble des règles de normalisation. Chaque forme normale correspond à une règle. Il existe six règles et donc six formes normales qui s'imbriquent. Nous n'étudierons

que les quatre premières formes normales dont la Boyce Code Normal Form qui correspond à la 3ème forme normale étendue.

C - La première forme normale

1 - Conditions :

Une relation est en première forme normale (1FN) si :

- Elle possède une clef primaire.
- Ses attributs sont atomiques c'est à dire qu'ils ne peuvent pas être décomposés et ne représentent pas une liste. (exemple : adresse n'est pas atomique car elle peut être décomposée en rue, code postal, ville).

2 - Exemple :

CITOYEN(NumSecuSociale, Nom, Adresse) n'est pas en 1FN.

Après correction, nous obtenons la relation suivante :

CITOYEN(NumSecuSociale, Nom, Rue, CodePostal, Ville) est en 1FN.

D - La seconde forme normale

1 - Conditions :

Une relation est en seconde forme normale (2FN) si :

- Elle est en 1FN.
- Tous les attributs n'appartenant pas à la clef primaire dépendent de la totalité de cette dernière.

2 - Exemple :

Considérons l'exemple la relation CONCERNER :

COMMANDE(NumCom, DateCom, #NumCli);

PRODUIT(RefProd, descriptionProd);

CONCERNER(#NumCom, #RefProd, PrixUnitaire, Nombre, libelleProd);

Elle est bien en 1FN mais l'attribut libelleProd ne dépend que d'une partie de la clef primaire #RefProd.

Cette relation n'est donc pas en 2FN.

Après correction, nous obtenons les relations suivantes :

COMMANDE(NumCom, DateCom, #NumCli);

PRODUIT(RefProd, libelleProd, descriptionProd);

CONCERNER(#NumCom, #RefProd, PrixUnitaire, Nombre);

E - La troisième forme normale

1 - Conditions :

Une relation est en troisième forme normale (3FN) si :

- Elle est en 2FN.
- Tous les attributs n'appartenant pas à la clef primaire ne dépendent que de cette dernière et non d'autres attributs non clefs.

2 - Exemple :

Considérons l'exemple suivant :

CLIENT(NumCli, NomCli, MailCli, TelCli, NumRegionCli, NomRegionClient);

Cette relation est bien en 1FN et en 2FN. Par contre, elle n'est pas en 3FN car l'attribut NomRegionClient dépend fonctionnellement de NumRegionCli.

Après correction, nous obtenons les relations suivantes :

```
CLIENT(NumCli, NomCli, MailCli, TelCli, #NumRegionCli);  
REGION(NumRegion, NomRegion);
```

F - La Boyce Codd Normal Form

1 - Conditions :

Une relation est en Boyce Code Normal Form (BCNF) si :

- Elle est en 3FN.
- Il n'y a pas de dépendance fonctionnelle autre qu'à partir de la clef primaire.

2 - Exemple :

Considérons l'exemple suivant :

```
BATIMENT(CodeCiviqueBat, NomRue, CodeCommune, NbrEtage);
```

L'exemple ci-dessus est en 1FN ; 2FN et 3FN mais il n'est pas en BCNF car il y a une dépendance fonctionnelle entre l'attribut CodeCommune et CodeCiviqueBat.

Après correction, nous obtenons les relations suivantes :

```
BATIMENT(CodeCiviqueBat, #CodeCommune, NomRue, NbrEtage);  
COMMUNE(CodeCommune, NomCommune);
```

Notons que nous sommes ici dans le cas d'une identification relative.

V - Le modèle logique de données

A - Présentation

Il semble important de rappeler que le modèle logique de données (ou modèle relationnel) se situe au niveau organisationnel de la méthode MERISE. La modélisation logique des données permet d'organiser les données provenant du MCD. C'est en 1970 que le modèle relationnel a été défini par E.F. Codd. Le modèle relationnel possède plusieurs caractéristiques :

- Il dispose d'un algèbre permettant la manipulation des tables et des relations.
- Il permet de créer des collections de relations.
- Un modèle sera qualifié de relationnel s'il permet le parcours de la structure des données en utilisant des chemins non prédéterminés mais créés dynamiquement par des requêtes.
- Les concepts du modèle relationnel proviennent de la théorie des ensembles.
- La relation peut être représentée par un tableau de données avec les colonnes comme attributs et les lignes comme tuple.
- La clé primaire se composant d'un ou plusieurs attribut(s) permet de retrouver un tuple unique.
- La clef primaire peut être simple ou composée.

B - Règles de passage du MCD au MLD

- Chaque entité devient une relation.
 - L'identifiant devient clef primaire.
 - Les propriétés deviennent attributs.
- Lorsqu'une entité est source de cardinalités comme 0,1 ou 1,1 alors la règle précédente s'applique. De plus, l'identifiant de l'entité père dans l'association devient clef étrangère dans l'entité fils.
- Une association de type 0,n - 1,n (ou 0,n - 0,n ou 1,n - 1,n) devient une relation dont :

- La clef primaire se compose des identifiants des entités engagés dans la relation.
- Les attributs sont les propriétés portées par l'association.

Nous remarquerons que les clefs primaires sont soulignées et les clefs étrangères sont précédées de "#".

C - Exemple de passage d'un MCD à un MLD

Reprenons l'exemple de la partie "I".

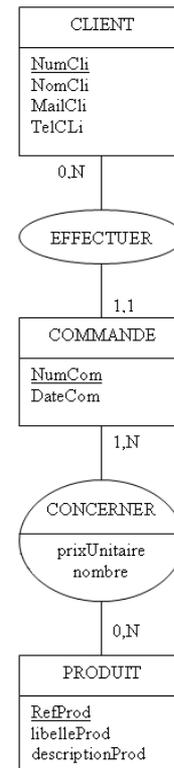
Par convention, nous noterons chaque relation R.

R1 : CLIENT(NumCli, NomCli, MailCli, TelCli);

R2 : COMMANDE(NumCom, DateCom, #NumCli);

R3 : PRODUIT(RefProd, libelleProd, descriptionProd);

R4 : CONCERNER(#NumCom, #RefProd, PrixUnitaire, Nombre);



VI - Le modèle physique de données

Il s'agit de la dernière étape avant l'obtention d'un système d'information l'implémentation du modèle relationnel dans le système de gestion de bases de données (SGBDR).

Bien que l'implémentation se fasse au travers d'une interface, il reste possible d'opérer en SQL (Structured Query Language) pour créer nos bases de données, tables, vues, utilisateurs, procédures stockées et autres objets de bases de données.

Conclusion

Dans ce chapitre, nous avons vu comment modéliser une base de données à l'aide de la méthode MERISE. Bien que certaines personnes habituées à cette méthode passent directement à l'implémentation du modèle physique de données, ceci est controversé. En effet, la méthode MERISE nécessite une démarche par étape qui favorise la qualité de chaque modèle avec ses différents niveaux de validations. Il est préférable de se rendre compte d'une erreur de modélisation tôt dans l'avancement du projet plutôt qu'une fois le modèle physique implémenté dans un système de gestion de bases de données relationnelles en production.

De plus, il sera toujours plus aisé de maintenir et de faire évoluer un modèle physique de données sain qu'un système mal pensé. Dans le cas d'un système sain, nous pourrions éventuellement procéder à la rétro-conception dans le but de revenir du modèle physique de données au modèle conceptuel de données pour comprendre les règles de gestion au cas où nous n'étions pas présent lors de la modélisation du système.