le cnam
Grand Est

Développement web (2)

Sites dynamiques
et développement côté serveur

NFA017 (4 ECTS)

Séance 06
Modélisation d'une base de données
2022

Le Cnam en Grand Est - Centre régional
4 avenue du Docteur Heydenreich CS 65228 F 54052 Nancy Cedex
tél +33 (0)3 83 85 49 00 www.cnam-grandest.fr

Le plus grand soin a été apporté à la réalisation de ce support pédagogique afin de vous fournir une information complète et fiable. Cependant, le Cnam Grand-Est n'assume de responsabilités, ni pour son utilisation, ni pour les contrefaçons de brevets ou atteintes aux droits de tierces personnes qui pourraient résulter de cette utilisation.

Les exemples ou programmes présents dans cet ouvrage sont fournis pour illustrer les descriptions théoriques. Ils ne sont en aucun cas destinés à une utilisation commerciale ou professionnelle.

Le Cnam ne pourra en aucun cas être tenu pour responsable des préjudices ou dommages de quelque nature que ce soit pouvant résulter de l'utilisation de ces exemples ou programmes.

Tous les noms de produits ou autres marques cités dans ce support sont des marques déposées par leurs propriétaires respectifs.

Ce support pédagogique a été rédigé par Alexandre ECUVILLON et mis en page par Simon MAHIEUX, enseignants au Cnam Grand-Est.

Table des matières

1. La méthode Merise

Merise est un sujet particulièrement vaste, dont l'étude approfondie dépasse le cadre de cette unité d'enseignement. Dans ce cours, nous vous proposons un survol de la méthode Merise avant de nous focaliser sur les outils qu'elle propose pour modéliser les bases de données.

1.1 Merise : généralités

Merise est une méthode d'analyse, de conception et de gestion de projet informatique.

Merise est née vers 1978-1979 à la suite d'une concertation lancée en 1977 par le ministère de l'industrie français pour choisir des sociétés de service en informatique capables de créer une nouvelle méthode de conception de système d'information. La méthode est spécifiquement française et n'a jamais réussi à rayonner en dehors de l'hexagone.

Contrairement à une idée répandue, Merise n'est pas un acronyme ; son origine provient d'une analogie avec le merisier « qui ne peut porter de beaux fruits que si on lui greffe une branche de cerisier : ainsi en va-t-il des méthodes informatiques bien conçues, qui ne produisent de bons résultats que si la greffe sur l'organisation réussit » (tirée de la préface du livre « La méthode Merise - Principes et outils » rédigée par Jacques Lesourne).

Plutôt qu'une notation d'analyse informatique « technique » - comme UML - Merise est surtout une démarche qui permet d'établir un Système d'Information ; elle se focalise sur la gestion de l'entreprise, ses processus, en laissant de côté l'aspect informatique. Il convient de noter que Merise a une approche très centrée sur les « data » (données).

Merise est de nos jours largement utilisée en entreprise (en dépit de son âge) mais pas toujours dans sa globalité. Il n'est pas rare qu'on réduise – à tort – la méthode aux outils qu'elle propose pour concevoir des bases de données (MCD, MLD, MPD).

Merise est une méthode tandis qu'UML est une notation !
Merise a toujours été en concurrence avec d'autres systèmes de modélisation. Il est courant de la comparer à UML. Attention, Merise et UML ne se situent pas au même niveau : l'une est une méthode tandis que l'autre n'est « qu'une » notation. On dit qu'UML est une notation car son objectif est de fournir un ensemble d'outils de modélisation à base de pictogrammes. UML ne cherche donc pas à donner de réponses sur le plan de la gestion de projet. Au contraire, si Merise propose également des notations au travers de différents schémas, son périmètre est plus vaste car elle intègre aussi bien des éléments de démarche (une façon de cheminer) que de gestion de projet.

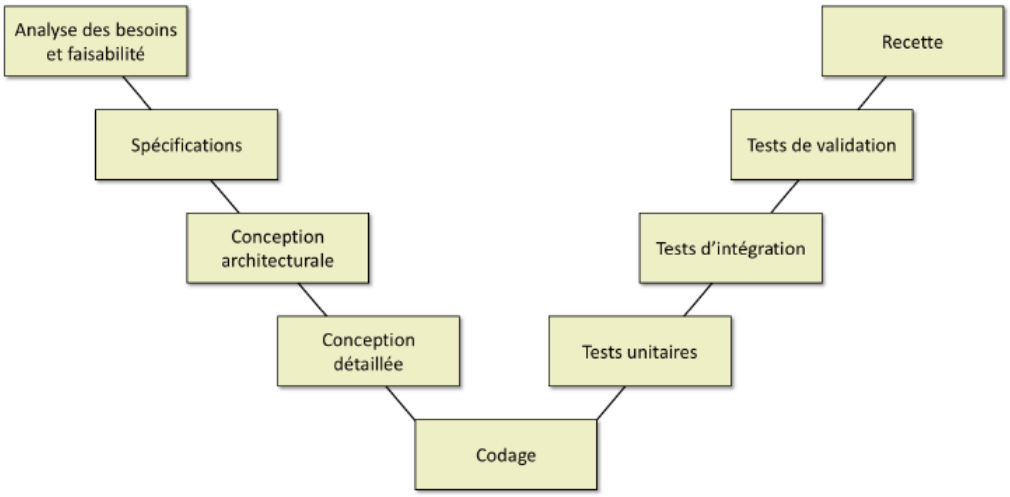
On peut donc dire, très synthétiquement, que Merise c'est : des cycles de vie projets, des documents-types et des diagrammes.

1.2 Le cycle de vie

Merise reconnaît macroscopiquement que le cycle de vie d'un projet se découpe en trois grandes phases :

1. **Gestation et conception :** naissance de l'idée, réalisation d'études de plus en plus détaillées visant à spécifier les fonctionnalités qui seront mises en oeuvre.
2. **Réalisation et exploitation :** développement technique de la solution, tests, intégration, déploiement en production.
3. **Maintenance et mort :** utilisation de l'application, maintenances correctives et évolutives puis retrait de la solution (mort).

Il existe plusieurs modèles de cycle de vie. Le plus connu et le plus traditionnel d'entre eux est le « cycle en V », qui décrit une démarche par étapes successives, la validation de l'une entraînant le début de la suivante.



Le cycle en V

1.3 Les groupes de travail

Merise identifie lors d'un projet trois groupes de travail typiques.

Groupe de travail	Rôles
Comité de pilotage	• Veiller à la participation de toutes les directions de l'entreprise • Définir, et faire comprendre à tous, les objectifs du Schéma Directeur. • Valider les actions du Comité de projet. • Prendre les décisions de choix d'un scénario.
Comité de projet	• Planifier et faire respecter le plan global du projet. • Assurer la cohérence des diverses actions des groupes de travail. • Réaliser les aides à décision du groupe de pilotage. • Tenir à jour et à disposition la documentation du projet. • Présenter, animer, promouvoir le système projeté.
Groupes d'utilisateurs	• Identifier les forces, les faiblesses, les limites du système existant. • Apporter la connaissance du terrain, des besoins. • Comprendre la formalisation du système projeté. • Veiller au réalisme du système projeté par rapport à la connaissance de l'entreprise et de son environnement.

Maîtrise d'ouvrage (MOA) / maîtrise d'oeuvre (MOE)
La maîtrise d'ouvrage (MOA) est l'entité porteuse du besoin, définissant l'objectif du projet, son calendrier et le budget consacré à ce projet. Le résultat attendu du projet est la réalisation d'un produit, appelé ouvrage. Le maître d'ouvrage est responsable de l'expression fonctionnelle des besoins mais n'a pas forcément les compétences techniques liées à la réalisation dudit ouvrage.

La maîtrise d'oeuvre (MOE) est l'entité retenue par le maître d'ouvrage pour réaliser l'ouvrage, dans les conditions de délais, de qualité et de coût fixées par ce dernier conformément à un contrat. La maîtrise d'oeuvre est donc responsable des choix techniques inhérents à la réalisation de l'ouvrage conformément aux exigences de la maîtrise d'ouvrage. Il désigne une personne physique chargée du bon déroulement du projet : il s'agit du chef de projet.

1.4 Les étapes de la méthode Merise

Merise est une méthode visant à créer des systèmes d'information : c'est une démarche méthodologique de développement de système d'information.

Les différentes étapes sont :
- Le schéma directeur
- L'étude préalable
- L'étude détaillée fonctionnelle
- L'étude détaillée technique
- La réalisation
- La mise en oeuvre
- La maintenance

A chaque étape sont attendus des livrables, c'est-à-dire un ensemble de documents, typiquement des études.

1.4.1 Le schéma directeur

Il consiste à étudier le système d'information (choix d'organisation, choix stratégiques…) pour le découper en domaines d'activité. Au niveau organisationnel, le système d'information est défini. Au niveau stratégique, les choix de matériels, de logiciels et des architectures sont effectués. Au cours de cette étape, on établit un plan directeur (ou plan de développement) qui sera utilisé au cours des différentes étapes à suivre. Il fixe notamment les besoins en personnel, en matériel et définit la stratégie à mettre en place.

Le schéma directeur concerne donc des choix d'orientation macroscopiques, stratégiques, dans une vision moyen-long terme. Il engage typiquement l'avenir de l'entreprise et est du ressort de ses dirigeants.

1.4.2 L'étude préalable

L'étude préalable est une étape essentielle de la démarche. Elle donne aux responsables les moyens de décider des solutions à adopter pour l'informatisation d'un système. Elle permet de mesurer les risques encourus par l'entreprise et, dans cette optique, peut se solder par un refus catégorique de la mise en place de la solution informatique (cas « extrême »).

Cette étape a pour objectif de définir le domaine sur lequel porte le projet d'informatisation. Et ainsi, elle met en évidence les nouvelles procédures à mettre en place, les postes de

travail touchés par cette réorganisation, le degré d'automatisation, les moyens informatiques à mettre en œuvre ...

On peut décomposer cette étape en quatre parties :

- Etude d'opportunité
- Etude de l'existant
- Schéma global de la solution
- Etude de la solution

1.4.2.1 Etude d'opportunité

Lorsqu'un projet paraît risqué ou mal maîtrisé, on réalisera souvent une étude d'opportunité. Elle consiste à étudier sommairement les gains attendus, les difficultés possibles, les risques associés, les coûts, les grandes options de mise en oeuvre, etc. Il s'agit à son issue de valider l'orientation : soit le projet n'est pas jugé opportun et on s'arrête là, soit il semble intéressant et on s'engage dans des études un peu plus poussées.

1.4.2.2 Etude de l'existant

Cette étude a pour but d'établir un bilan de l'organisation actuelle de l'entreprise – bilan qu'on juge nécessaire de connaître pour bien concevoir le système futur. Les diverses activités de l'entreprise y sont décrites ainsi que les informations véhiculées. A cette étape, un diagramme de circulation des flux est réalisé : on y précise les flux, leurs sens et les différents acteurs (internes et externes). Le rapport est soumis aux utilisateurs pour validation.

1.4.2.3 Schéma global de la solution

A ce niveau, on couche les grandes orientations de la solution cible. Au niveau de la gestion, sont déterminées les activités qui ne sont pas touchées par la réorganisation, les activités à modifier (à adapter à la nouvelle gestion) et les nouvelles activités (dues à la nouvelle gestion). Au niveau organisationnel, sont déterminées les tâches manuelles et les tâches automatisées. Au niveau technique, il faut définir les différents matériels et logiciels nécessaires à la mise en place de la nouvelle organisation.

1.4.2.4 Etude de la solution

La solution est présentée aux décisionnaires avec tous les éléments nécessaires à la prise de décision (bilan qualitatif, bilan quantitatif, mise en évidence des nouveaux services, des améliorations…). Un bilan économique de la nouvelle solution est également présenté (coût de la mise en place…). Le plan de développement est complété : il comporte à présent le scénario de mise en œuvre. Sont également présentés à ce niveau de l'étude le découpage en sous-projets et le calendrier de réalisation. A la fin de cette étape, un bilan de l'étude préalable est réalisé. Elle débouche alors sur un GO (feu vert de la direction pour poursuive) ou NO-GO (feu rouge, le projet est stoppé).

1.4.3 L'étude détaillée fonctionnelle

A cette étape, le dossier réalisé précédemment est validé et l'étude des données et des traitements peut être approfondie. Un dossier concernant les spécifications suivantes est porté à validation :

- Interface (enchaînement des écrans, des menus de choix…)
- Règles de sécurité
- Descriptif des traitements, des différents contrôles de validité, etc.
- Descriptif des états (chaque sortie est ajoutée au dossier en exemple)

Un plan de développement plus précis est réalisé : il propose un planning de développement et de mise en oeuvre. Le Modèle Conceptuel des Données et le Modèle Logique des Données sont généralement conçus au cours de cette étude. L'étude préalable a permis la validation de la solution dans ses grandes lignes tandis que l'étude détaillée va permettre la validation des spécifications fonctionnelles propres au projet.

1.4.4 L'étude détaillée technique

L'étude technique a pour but de préparer la réalisation. Elle doit lever les dernières contraintes et établir les choix qui orienteront la réalisation. Une description de l'environnement technique doit permettre de préciser :

- Le type de matériel retenu et son système d'exploitation
- Le choix du système de gestion de données
- Les outils de développement (langage, atelier de génie logiciel, etc.)
- Les plans de tests et jeux d'essais

Une description de l'architecture du logiciel :

- Structure des sous-programmes
- Découpage en unités de traitement

Une description du modèle physique des données :

- Description des procédures de sécurité
- Répartition physique des données sur les sites

1.4.5 La réalisation

Cette phase, qui aboutit au logiciel testé et prêt à l'utilisation, est constituée de quatre étapes :

- Production du logiciel (« codage »)
- Tests unitaires correspondant à l'unité de traitement
- Tests d'intégration dans l'ensemble du logiciel
- Eventuelle optimisation du Modèle Physique des Données

1.4.6 La mise en œuvre

- Mise en place des nouveaux équipements
- Mise en place des moyens humains
- Mise en place des fichiers et de la documentation
- Livraison des logiciels
- Tests des utilisateurs (validation)
- Déploiement de la nouvelle application
- Fonctionnement en grandeur réelle
- Réception de l'application

1.4.7 La maintenance

La maintenance correspond aux adaptations du logiciel nécessaires tout au long de sa vie.

Deux types de maintenance existent.

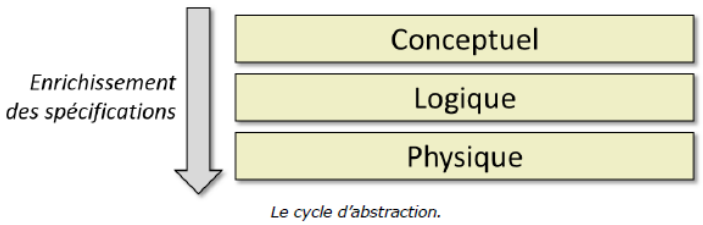
- La maintenance corrective : correction d'une anomalie (de conception ou de réalisation)
  - Erreur de conception : elle est due à une incohérence dans l'analyse et nécessite de revoir cette dernière
  - Erreur de réalisation : elle est due à une mauvaise compréhension ou un oubli lors de la réalisation
- La maintenance évolutive : modifications impliquées par une évolution de l'organisation

1.5 Le cycle d'abstraction et ses modèles

Le cycle d'abstraction propose trois niveaux d'intérêt (conceptuel, organisationnel, physique) du plus abstrait au plus concret. L'objectif est de commencer par prendre les grandes décisions métiers avant d'entrer dans les détails des impacts organisationnels puis techniques.

A chaque niveau, Merise propose des modèles (diagrammes) de plus en plus détaillés (« zoom ») :

- Au niveau conceptuel, Merise s'attache aux invariants de l'organisme étudié du point de vue métier. Les choix organisationnels et techniques y sont totalement absents.
- Au niveau logique (ou organisationnel), les modèles conceptuels sont précisés et complétés avec les résultats des choix d'organisation.
- En fin au niveau physique, on aborde de manière concrète (technique) la mise en œuvre du système.



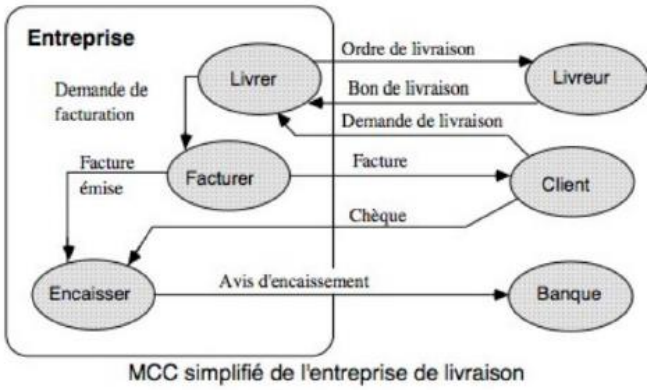
Le cycle d'abstraction.

Ce chapitre vous propose de survoler les différents modèles proposés par Merise.

1.5.1 Le niveau conceptuel

1.5.1.1 Le Modèle Conceptuel de Communication (MCC)

Le **M**odèle **C**onceptuel de **C**ommunication décrit les échanges entre les systèmes internes ou externes (on le nomme aussi « diagramme des flux »).

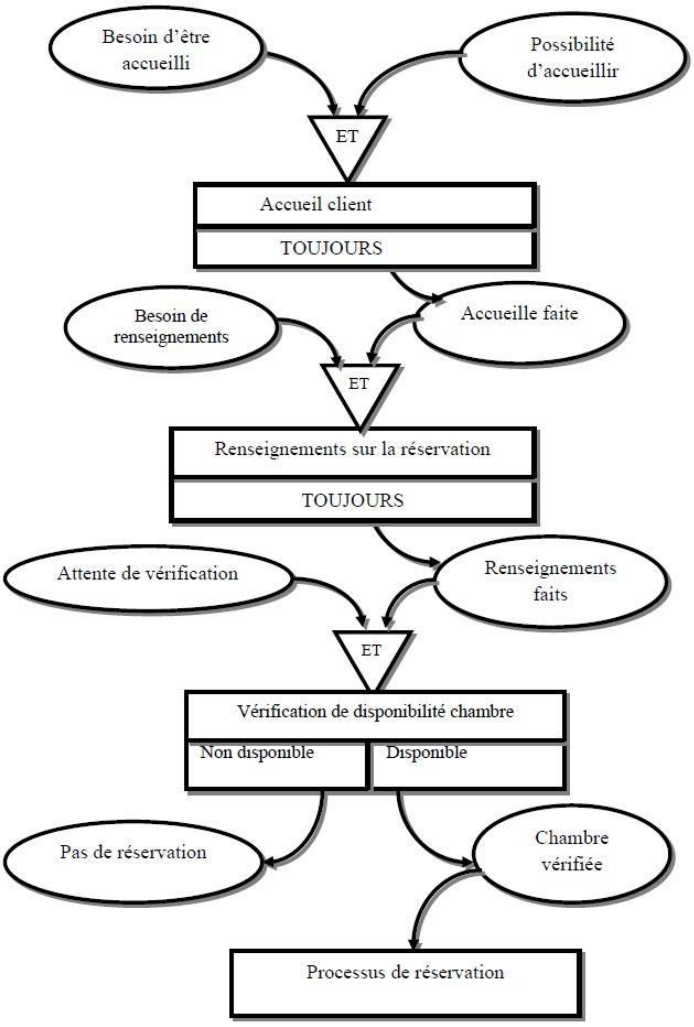


MCC simplifié de l'entreprise de livraison

Exemple de MCC.

1.5.1.2 Le Modèle Conceptuel des Traitements (MCT)

Le **M**odèle **C**onceptuel des **T**raitements décrit les opérations qui sont réalisées en fonction d'événements (dynamique du Système d'Information). Il représente les événements, les résultats, les opérations et les synchronisations.

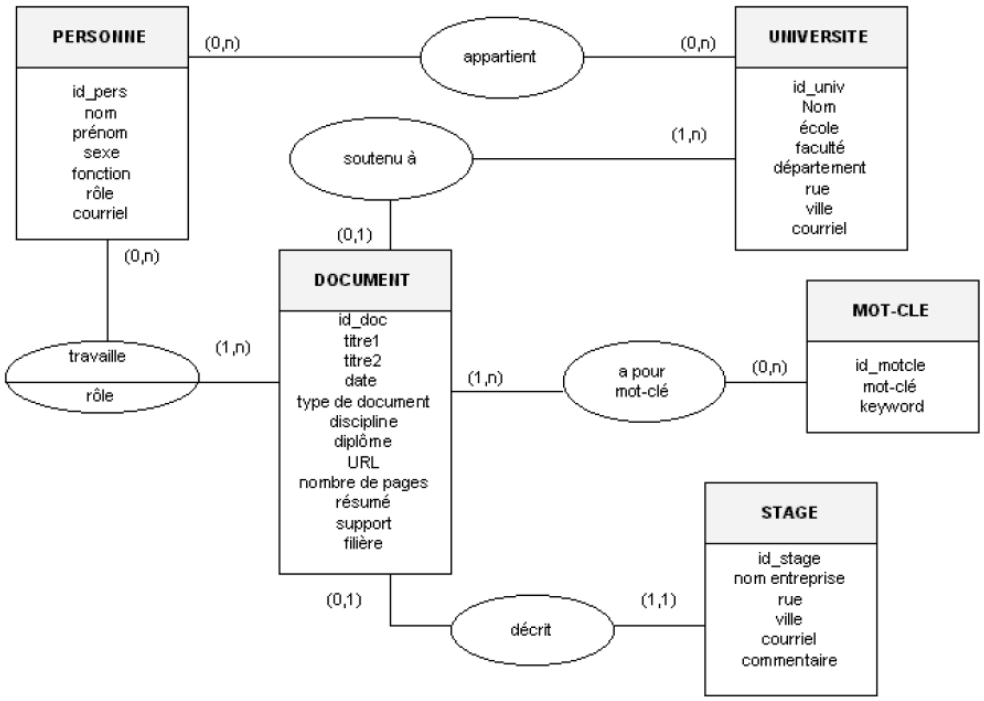


Exemple de MCT

1.5.1.3 Le Modèle Conceptuel des Données (MCD)

Le **M**odèle **C**onceptuel des **D**onnées décrit la sémantique c'est à dire le sens attaché aux données et à leurs rapports et non à l'utilisation qui peut en être faite. Avant la construction de ce modèle, il convient de réaliser l'inventaire des données (le dictionnaire des données) dont on élimine les données calculées, les redondances et les synonymes. Le modèle est validé à l'aide de règles de vérification.

Le MCD est, de loin, le modèle le plus connu et employé de Merise car il débouche à terme sur la modélisation de la base de données. La démarche consiste une fois établi à l'enrichir / détailler via un MLD puis un MPD (MCD, MLD et MPD sont abordés en détail un peu plus loin dans ce cours).



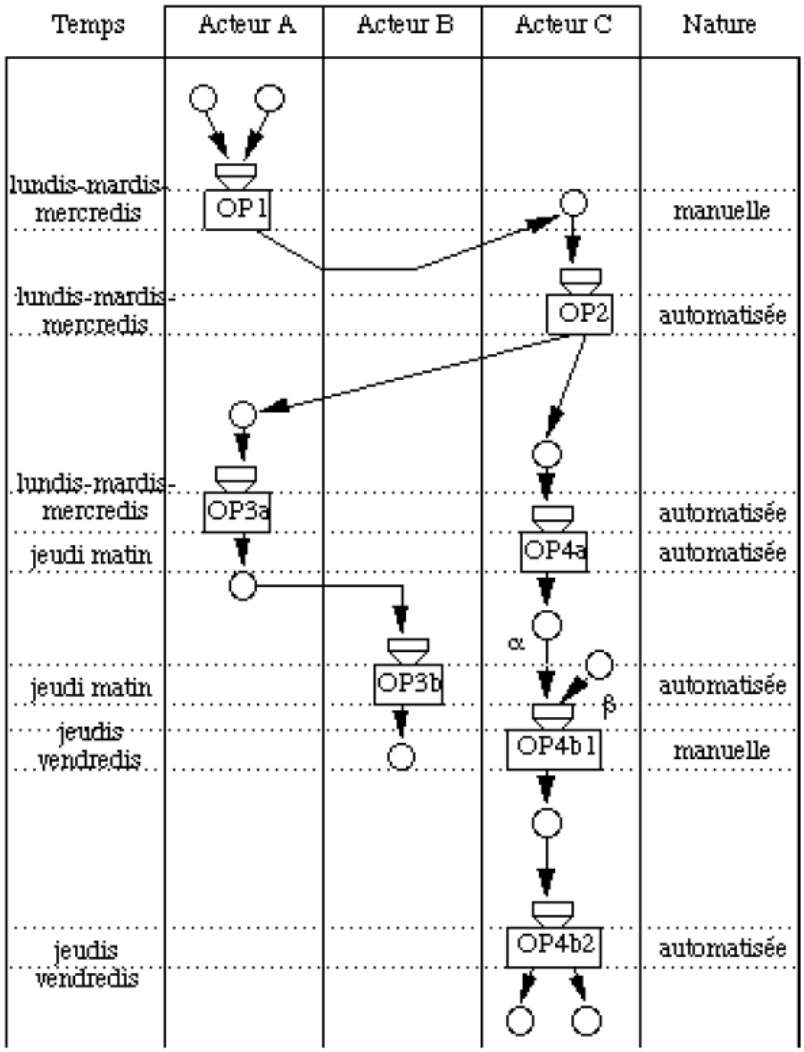
MCD: catalogue informatisé de travaux d'étudiants

Exemple de MCD.

1.5.2 Le niveau organisationnel

1.5.2.1 Le Modèle Organisationnel des Traitements

Le but de ce modèle est de fournir une représentation de l'organisation de l'entreprise. Les concepts d'événement et de résultat présents dans la description conceptuelle sont repris dans la description organisationnelle. On y ajoute la notion de poste de travail et la notion de temps.



Exemple de MOT.

1.5.2.2 Le Modèle Logique des Données (MLD)

Le **M**odèle **L**ogique des **D**onnées (également appelé MOD : **M**odèle **O**rganisationnel des **D**onnées) est une traduction du modèle conceptuel des données. Il intègre les choix d'organisation des données (fichiers classiques, modèle hiérarchique, modèle relationnel).

Exemple de MLD.

1.5.3 Le niveau physique

Il consiste à apporter des solutions physiques aux problèmes : il est la représentation des moyens qui vont être mis en œuvre.

Le **M**odèle **P**hysique des **D**onnées (MPD) est la traduction du Modèle Logique des Données (MLD). Il exprime la modélisation dans le système de gestion de base de données retenu (exemple : Oracle, MySQL, SQL Server, etc.). Autrement dit, le MPD est le script SQL de création technique de la base de données.

2. Le Modèle Conceptuel des Données

Le niveau conceptuel (dans lequel le MCD est créé) correspond à une formalisation du système d'information indépendante de toute contrainte d'organisation. A ce niveau, il faut distinguer la formalisation des données mémorisées dans la base de données (aspect statique) et celle des traitements (aspect dynamique). Au niveau conceptuel, la formalisation des données constitue le **M**odèle **C**onceptuel des **D**onnées, celle des traitements constitue le **M**odèle **C**onceptuel des **T**raitements.

Le MCD (associé à la démarche Merise) doit son succès aux qualités suivantes :

- Une approche naturelle et facilement compréhensible
- Une représentation claire et complète
- Une construction directe des résultats
- Une représentation synthétique des données

Nous aborderons tout d'abord le formalisme du Modèle Conceptuel des Données puis la manière de concevoir ce modèle.

2.1 Concepts de base

2.1.1 Entité

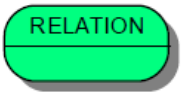
Définition : L'entité est un élément concret ou abstrait du système d'information qui a une existence propre et sur lequel on souhaite enregistrer des informations qui lui sont spécifiques.

Représentation : L'entité est représentée par un rectangle.

2.1.2 Relation

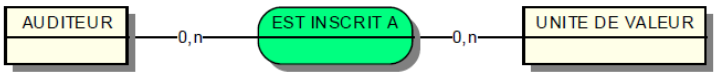
Définition : Une relation est la prise en charge par le système d'information d'une association entre deux ou plusieurs entités.

Représentation : La relation est représentée par un ovale.

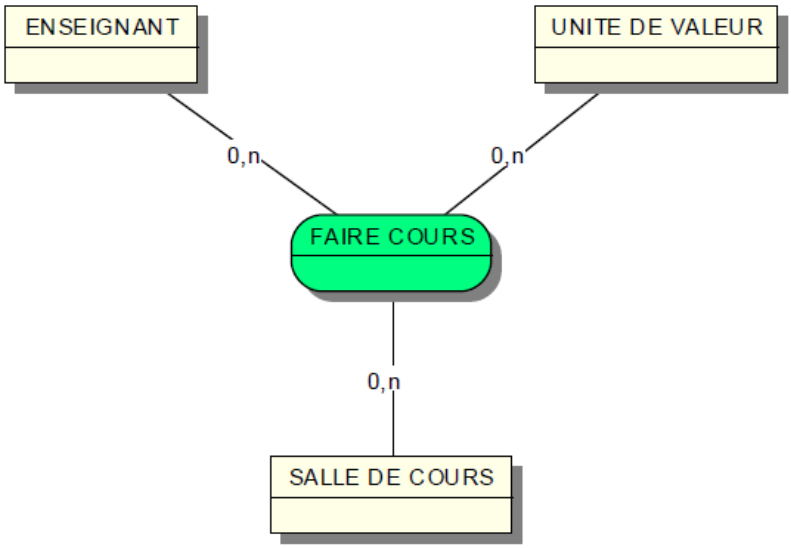


Remarque : Une relation peut être également appelée « association ».

Exemples :
Association entre deux entités (dimension 2) :



Association entre trois entités (dimension 3) :



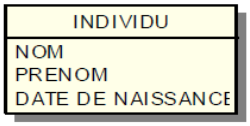
Remarque : Les annotations portées sur les liaisons entre relations et entités sont appelées cardinalités. Leur signification sera précisée au chapitre 2.2.

2.1.3 Propriété

Définition : Une propriété est une donnée élémentaire attribut d'une entité ou d'une relation.

Représentation :

Exemple :



L'entité « individu » représentée ci-dessus est caractérisée par trois propriétés : « nom », « prénom » et « date de naissance ».

2.1.4 Occurrence

2.1.4.1 Occurrence d'une propriété

Définition : Une occurrence de propriété est une valeur que peut prendre une propriété.

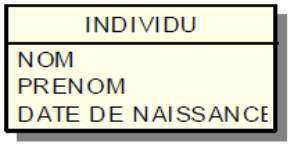
Exemple :
La propriété nommée « Nom de l'individu » peut prendre les valeurs suivantes :
DUPONT
DURAND
PIERRE
…

DUPONT, DURAND et PIERRE sont des occurrences de la propriété « Nom de l'individu ».

2.1.4.2 Occurrence d'une entité

Définition : Une occurrence d'entité est un ensemble ayant une existence propre d'occurrences de ses propriétés.

Exemple :
Soit l'entité suivante :

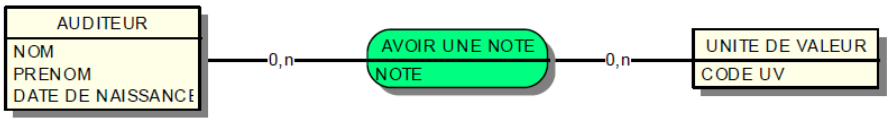


L'ensemble {DUPONT, JEAN, 23/10/1970} est une occurrence de l'objet INDIVIDU car « DUPONT » est une occurrence de la propriété NOM, « JEAN » une occurrence de PRENOM et « 23/10/1970 » une occurrence de DATE DE NAISSANCE.

2.1.4.3 Occurrence d'une relation (d'une association)

Définition : Une occurrence de relation est constituée d'une et d'une seule occurrence de chacune des entités associées. L'occurrence de chacune des propriétés de l'association est en relation avec les occurrences des entités associées.

Exemple :



Soient les occurrences suivantes :

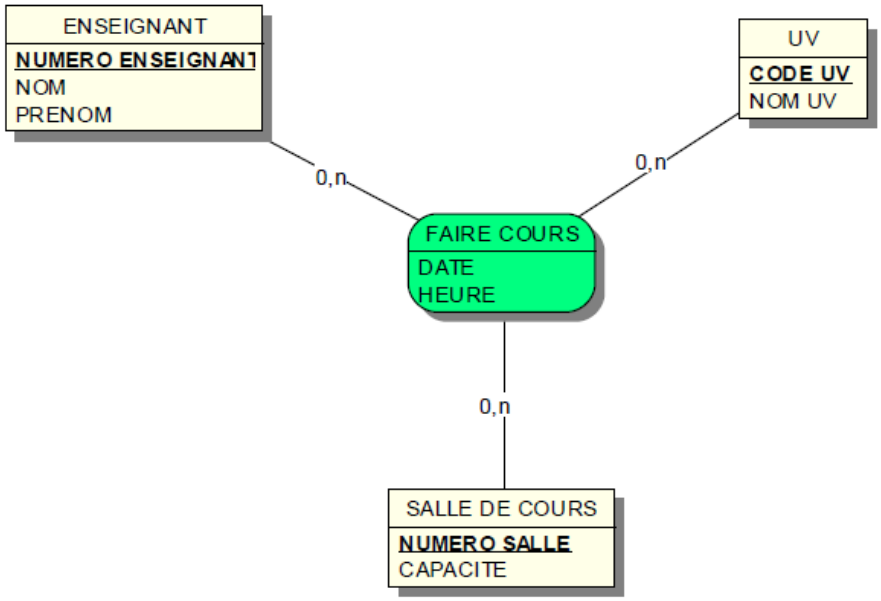
- {DUPONT, JEAN, 23/10/1970} : une occurrence de l'entité « AUDITEUR ».
- {FRANÇAIS} et {MATHEMATIQUES} : deux occurrences de l'entité « UNITE DE VALEUR »
- 12 peut être une occurrence de « NOTE » associée aux occurrences {DUPONT, JEAN, 23/10/1970} et {FRANÇAIS} ; 13 une autre occurrence de « NOTE » associée aux occurrences {DUPONT, JEAN, 23/10/1970} et {MATHEMATIQUES}

2.1.5 Identifiant

Définition : L'identifiant (également appelé clé) est une propriété particulière (ou l'association de plusieurs propriétés) qui permet de caractériser ses occurrences de façon unique.

Représentation : L'identifiant d'une entité est l'ensemble des propriétés soulignées.

Exemple :



Remarque :
Un identifiant doit être unique. Il n'est pas rare qu'aucune propriété ou ensemble de propriétés ne puisse être considéré comme identifiant. Dans ce cas, on ajoute un identifiant fictif sans signification particulière (exemple : NUMERO_ENSEIGNANT, CODE_UV)

Au chapitre 3, nous verrons les règles de passage du Modèle Conceptuel des Données au modèle relationnel : c'est à ce moment que nous déterminerons la manière de créer les identifiants d'associations.

2.2 Cardinalités

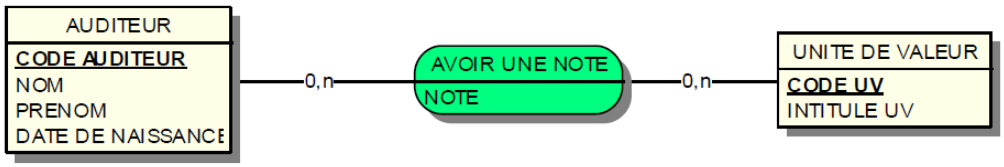
2.2.1 Définitions

La cardinalité d'une entité par rapport à une association s'exprime par deux nombres appelés cardinalité minimum et cardinalité maximum.

La cardinalité minimale peut être égale à 0 ou à 1. Si la cardinalité est égale à 0, c'est qu'il existe au moins une occurrence de l'entité qui ne participe pas aux occurrences de l'association. Si la cardinalité est égale à 1, chaque occurrence de l'entité participe aux occurrences de l'association.

La cardinalité maximale exprime le nombre maximum de fois où une occurrence de l'entité participe aux occurrences de l'association. On la note égale à n, elle peut être égale à 1 ou à tout autre nombre strictement positif (quand le nombre d'occurrences est quantifiable).

Exemple :



Un auditeur peut ne pas avoir de note, c'est à dire qu'il peut exister un auditeur qui n'a pas passé d'examen (cardinalité minimum = 0). Un auditeur peut être inscrit à plusieurs unités de valeur et avoir une note pour chacune des valeurs auxquelles il est inscrit (cardinalité maximum = n). Il est possible que personne ne se présente à l'examen pour une unité de valeur (cardinalité minimum = 0). Plusieurs notes peuvent être attachées à une même unité de valeur (cardinalité maximum = n).

2.3 Contraintes d'intégrité

Une contrainte d'intégrité est une règle sur les valeurs permises pour une (ou plusieurs) propriétés. Il en existe plusieurs types, de complexité croissante.

2.3.1 Contrainte d'intégrité associée à une rubrique

Définition : Une contrainte d'intégrité associée à une rubrique (à une propriété) est l'ensemble des valeurs possibles (occurrences) que peut prendre cette rubrique.

Exemples : Une note doit être comprise entre 0 et 20, un prix doit être positif …

2.3.2 Contrainte d'intégrité sémantique

Définition : Une contrainte d'intégrité sémantique correspond à une règle sémantique fournie par l'utilisateur permettant à l'analyste de modéliser des contraintes spécifiques. Elles sont divisées en deux catégories : les contraintes statiques et les contraintes de transition.

2.3.2.1 Contrainte d'intégrité sémantique statique

Ce type de contrainte concerne l'état du système d'information indépendamment du temps. Les contraintes simples font intervenir une ou plusieurs occurrences dont on compare directement les composantes. Les contraintes complexes (également appelées ensemblistes) font intervenir des ensembles de données (occurrences) dont on fait la moyenne, la somme …

Exemples : La date de facturation est supérieure à la date de commande *(contrainte simple)*. La somme des salaires (masse salariale) ne doit pas dépasser un seuil donné *(contrainte ensembliste)*.

2.3.2.2 Contrainte d'intégrité sémantique de transition

Les contraintes de transition concernent l'évolution dans le temps des données du système d'information. Les contraintes de transition simples font intervenir une ou plusieurs rubriques qui varient dans le temps. Les contraintes complexes (également appelées

ensemblistes) font intervenir des données variant dans le temps dont on fait la moyenne, la somme …

Exemples : Le salaire d'un employé varie dans le temps mais ne peut pas diminuer *(contrainte simple)*. La moyenne des coûts de production des produits de l'entreprise ne doit pas dépasser de plus de 7 % la moyenne de l'année précédente *(contrainte ensembliste)*.

2.3.3 Contrainte d'intégrité associée à plusieurs rubriques

Une contrainte d'intégrité associée à plusieurs rubriques est appelée **D**épendance **F**onctionnelle (DF).

2.3.3.1 Dépendances fonctionnelles

Définition : Deux rubriques (ou propriétés) sont en dépendance fonctionnelle si la connaissance d'une d'entre elles permet la connaissance de l'autre.

Notation : A → B signifie que B dépend fonctionnellement de A. En d'autres termes, la connaissance d'une occurrence de la propriété A permet la connaissance d'une occurrence de la propriété B.

Exemple :
Numéro INSEE → sexe
Numéro INSEE → année de naissance
Code d'un auditeur → Nom de cet auditeur

2.3.3.2 Propriétés des dépendances fonctionnelles

Réflexivité :	a → a		
Projection :	a → b + c	➔	a → b et a → c
Augmentation :	a → b	➔	∀c, a + c → b
Additivité :	a → b et a → c	➔	a → b + c
Transitivité :	a → b et b → c	➔	a → c
Pseudo-transitivité :	a → b et b + c → d	➔	a + c → d

2.3.3.3 Dépendances non fonctionnelles

Définition : Deux propriétés sont en dépendance non fonctionnelle si la connaissance d'une occurrence de la première :

- Ne détermine la connaissance d'aucune occurrence de la seconde.
- Ou détermine la connaissance de plusieurs valeurs de la seconde.

Exemple : La marque d'un véhicule permet la connaissance d'une gamme de véhicules

Remarque : Les contraintes de ce type ne sont pas prises en compte dans le MCD.

2.3.4 Contrainte d'intégrité associée au MCD

2.3.4.1 Règles de normalisation

Le modèle conceptuel des données sera toujours présenté en troisième forme normale. Les formes normales s'emboitent les unes dans les autres, tant et si bien que le respect d'une forme normale de niveau supérieur implique le respect des formes normales des niveaux inférieurs. Un MCD qui respecte la 3ème forme normale est donc considéré comme « correctement modélisé ».

1 FN

Une entité est en première forme normale (1 FN) si et seulement si :
- Toutes ses propriétés sont *élémentaires*.
- Elle possède un identifiant (une clef).

Une propriété est dite *élémentaire* (ou scalaire) si, dans le système d'information, elle ne peut pas être décomposée en plusieurs propriétés – c'est-à-dire si cette propriété n'est pas « divisible ». Ainsi si une propriété « dimension » d'une entité « meuble » implique la largeur, la hauteur et la profondeur, elle n'est pas élémentaire et on aurait dû la décomposer en trois propriétés distinctes.

2 FN

Une entité est en deuxième forme normale si et seulement si :
- Elle est en première forme normale.
- Toutes ses propriétés sont en dépendance fonctionnelle élémentaire de l'identifiant.

Soient les propriétés P_1, P_2 et P_3. Une dépendance fonctionnelle ($P_1 \rightarrow P_2$) est dite élémentaire si et seulement si il n'existe pas de propriété (P_3) incluse dans P_1 en dépendance fonctionnelle avec P_2 ($P_3 \rightarrow P_2$).

On peut exprimer ce principe plus simplement par « un attribut non-clé ne doit pas dépendre d'une partie de la clé », c'est-à-dire qu'un attribut non-clé n'est "devinable" avec une partie de la clé.

Exemple :
Une entité a une clé compose de plusieurs propriétés dont l'une est « numéro de département ». Elle dispose en outre d'une propriété non-clé « nom de département ». Cette entité n'est pas en 2ème forme normale, car la connaissance du numéro de département permet de déduire le nom du département.

3 FN

Une entité est en troisième forme normale si et seulement si :
- Elle est en deuxième forme normale.
- Toutes ses propriétés sont en dépendance fonctionnelle élémentaire directe de l'identifiant.

Soient les propriétés P_1, P_2 et P_3. Une dépendance fonctionnelle ($P_1 \rightarrow P_2$) est dite directe si et seulement si il n'existe pas de propriété (P_3) telle que $P_1 \rightarrow P_3$ et $P_3 \rightarrow P_2$.

Plus simplement dit : « un attribut non-clé ne doit pas dépendre d'un ou plusieurs attributs ne participant pas à la clé ». C'est donc le même principe que la 2ème forme normale, mais appliqué à tous les attributs, pas seulement ceux clé.

Remarque : certains analystes utilisent une quatrième forme normale appelée Forme Normale de Boyce-Codd (BCFN). Son énoncé est le suivant : « si une entité a un identifiant concaténé, un des éléments qui le compose ne doit pas dépendre d'une autre propriété ».

2.3.4.2 Contraintes d'intégrité structurelles

Intégrité d'entité : La valeur de la propriété « identifiant » doit être non nulle et unique.

Intégrité référentielle : Pour toute occurrence d'une relation, les individus intervenants doivent obligatoirement exister. Une contrainte d'intégrité fonctionnelle représente un lien qui ne peut pas être mis à jour (impossibilité de créer un enregistrement « fils » dans une table qui dépend fonctionnellement d'une autre table si l'enregistrement « maître » n'existe pas).

3. Le Modèle Logique des Données

Le Modèle Logique des Données (également appelé « modèle relationnel » ou « schéma relationnel ») est indépendant du choix du logiciel de gestion. Le but de ce modèle est de créer une architecture des données exploitable au niveau physique.

Ce modèle est une traduction directe du modèle conceptuel des données avec pertes ou modifications d'informations :

- Perte de la notion de cardinalité.
- Modification des entités (le cas échéant : ajout de propriétés).
- Les dépendances fonctionnelles et les contraintes d'intégrité fonctionnelles n'apparaissent plus ; en fait, elles sont transformées !

Les objectifs du modèle relationnel sont :

- L'indépendance physique.
- La perte des redondances de données (gain d'espace disque).
- La manipulation des données aisée.

Le Modèle Logique des Données est associé (est « spécifique ») à un Système de Gestion de Base de Données Relationnel tels que Oracle, Access ou Paradox. Par opposition, le MCD était encore totalement indépendant du type de base de données retenue.

3.1 Définitions, terminologies

3.1.1 Relation

Définition : Une relation est représentée comme une table contenant des colonnes (attributs). Chaque colonne comporte des valeurs appartenant à un domaine. Une ligne de la table est appelée tuple.

Exemple : La relation « PERSONNE »

CodePers	NomPers	PrenomPers	DateNais	CodeSexe
01	DUPONT	Jean	12/04/1978	1
02	DURAND	Jacques	01/02/1968	1
03	GIAL	Hélène	25/06/1979	2

3.1.2 Domaine

Définition : Un domaine de valeurs est un ensemble de valeurs d'un type dit élémentaire (exemples : entier, réel, caractère, chaîne de caractères …). La notion de type élémentaire s'oppose à celle de type complexe (exemples : listes, enregistrements …).

Exemple : Le domaine associé à « CodeSexe » est l'ensemble de valeurs entières {0, 1}.

3.1.3 Attribut

Définition : Un attribut est le nom d'une colonne d'une relation. Il est obligatoirement associé à un domaine. L'attribut permet notamment de manipuler les valeurs contenues dans les colonnes.

Exemple : « CodePers » est un attribut de la relation « PERSONNE ».

Remarque : Un même nom d'attribut peut figurer dans différentes relations.

3.1.4 Schéma d'une relation

Définition : Le schéma d'une relation est un nom suivi de l'ensemble des associations « Attribut – Domaine » qui le constitue. On le note : R (A1 : D1, A2 : D2 , …, An : Dn).

Exemple : PERSONNE (CodePers : string, NomPers : string, PrenomPers : string, DateNais : date, CodeSexe : char)

Remarque : Dans la pratique, on oublie de mentionner le domaine. Ainsi, l'exemple précédent se note : PERSONNE (CodePers, NomPers, PrenomPers, DateNais, CodeSexe)

3.1.5 Instance d'une relation

Définition : Une instance de relation est un sous-ensemble de l'ensemble des valeurs que peut prendre une relation. Cet ensemble se définit par le produit cartésien de tous les domaines des attributs d'une relation.

Remarques :

- L'ordre des lignes (des tuples) dans une relation n'a pas d'importance.
- Il ne peut pas y avoir deux lignes identiques dans une même relation.
- Certains attributs (à l'exception des identifiants) peuvent ne rien contenir.

3.1.6 Tuple

Définition : Un tuple est un ensemble de valeurs correspondant à une ligne de la relation.

Exemple : {01, DUPONT, Jean, 12/04/1978, 1} est un tuple de la relation « PERSONNE »

3.1.7 Clé primaire

Définition : Une clé primaire est un attribut particulier (ou un ensemble d'attributs) d'une relation qui permet de distinguer les tuples d'une relation. On l'appelle également identifiant. Pour repérer la clé, on la souligne dans la relation.

Exemple : L'attribut « CodePers » est la clé primaire de la relation « PERSONNE ».

Remarque : Une clé composée de plusieurs attributs est appelée clé concaténée.

3.1.8 Clé étrangère

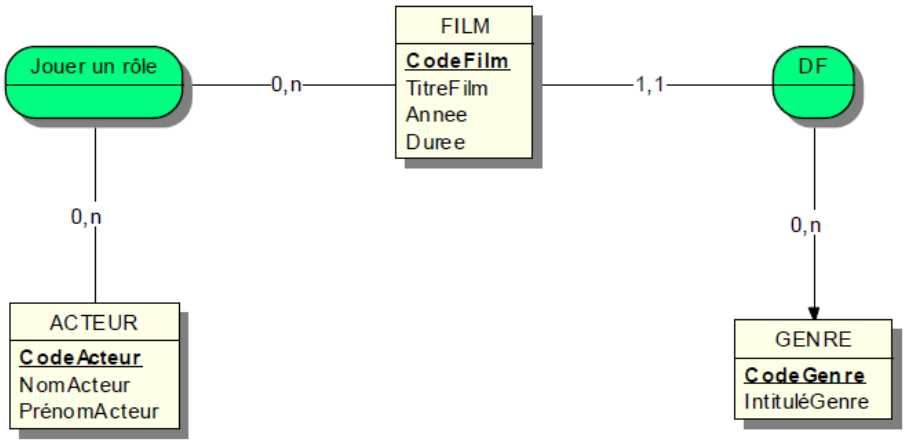
Définition : Une clé étrangère est un attribut d'une relation qui est clé primaire dans une autre relation.

Exemple :
PERSONNE (**CodePers**, NomPers, PrenomPers, **CodeFonction**)
FONCTION (**CodeFonction**, IntituléFonction)

« CodeFonction » est une clé étrangère de la relation « PERSONNE ». Elle permet d'associer une personne et une fonction (telle personne a telle fonction).

3.2 Exemple

Soit le modèle conceptuel des données suivant :



La représentation relationnelle sera composée de quatre tables :

FILM				
CodeFilm	*TitreFilm*	*Annee*	*Duree*	*CodeGenre*
01	Titanic	1997	3h00	1
…	…	…	…	…

*FILM(**CodeFilm**, TitreFilm, Annee, Duree, **CodeGenre**)*

ACTEUR		
CodeActeur	*NomActeur*	*PrénomActeur*
01	DiCaprio	Leonardo
02	Winslet	Kate
…	…	…

*ACTEUR(**CodeActeur**, NomActeur, PrénomActeur)*

JOUER_UN_ROLE	
CodeFilm	*CodeActeur*
01	01
01	02
…	…

*JOUER_UN_ROLE(**CodeFilm, CodeActeur**)*

GENRE	
CodeGenre	*IntituléGenre*
1	Drame
…	…

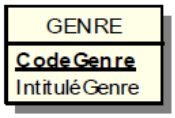
*GENRE(**CodeGenre**, IntituléGenre)*

3.3 Règles de passage du MCD au MLD

3.3.1 Entité

Chaque entité du modèle conceptuel des données est transformée en relation dans le modèle relationnel. Les attributs de l'entité deviendront les attributs de la relation. L'identifiant devient clé de la relation.

Exemple :

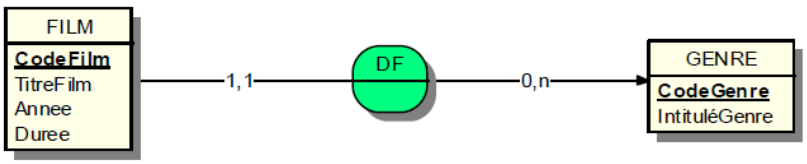


*GENRE(**CodeGenre**, IntituléGenre)*

3.3.2 Association « un à plusieurs »

Ce type d'association se traduit par l'introduction de la clé primaire de la relation ayant pour cardinalité maximale n en tant que clé étrangère de l'autre relation.

Exemple :

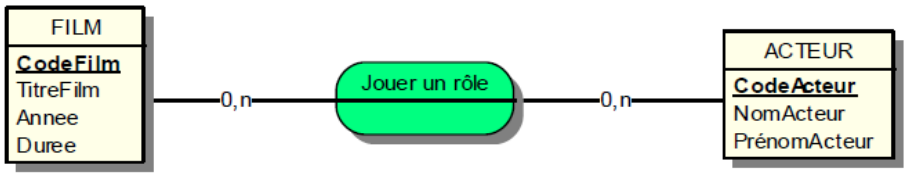


*GENRE(**CodeGenre**, IntituléGenre)*
*FILM(**CodeFilm**, TitreFilm, Annee, Duree, **CodeGenre**)*

3.3.3 Association binaire « plusieurs à plusieurs »

Cette association est prise en compte en créant une nouvelle relation qui a pour clé primaire la concaténation des clés primaires des deux relations associées. Les attributs de l'association sont insérés dans cette relation.

Exemple :

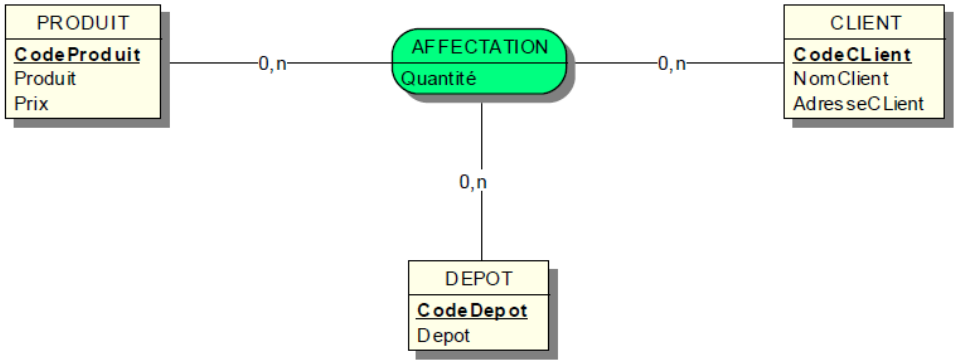


*ACTEUR(**CodeActeur**, NomActeur, PrénomActeur)*
*FILM(**CodeFilm**, TitreFilm, Annee, Duree, **CodeGenre**)*
*JOUER_UN_ROLE(**CodeFilm, CodeActeur**)*

3.3.4 Association N-aire

Cette association est prise en compte en créant une nouvelle relation qui a pour clé primaire la concaténation des clés primaires des N relations associées. Les attributs de l'association sont insérés dans cette relation.

Exemple :



*PRODUIT(**CodeProduit**, Produit, Prix)*
*CLIENT(**CodeClient**, NomClient, AdresseClient)*
*DEPOT(**CodeDepot**, Depot)*
*AFFECTATION(**CodeProduit, CodeClient, CodeDepot**, Quantité)*

Ce schéma permet d'affecter à un client un produit provenant de plusieurs dépôts. On aurait pu imaginer qu'un client ne puisse s'approvisionner qu'auprès d'un seul dépôt. Cette solution modifie le schéma relationnel.

*PRODUIT(**CodeProduit**, Produit, Prix)*
*CLIENT(**CodeClient**, NomClient, AdresseClient)*
*DEPOT(**CodeDepot**, Depot)*
*AFFECTATION(**CodeProduit, CodeClient,** Quantité, CodeDepot)*

Remarque : Selon les besoins du système d'information, les deux solutions sont envisageables mais n'ont pas le même sens (possibilités différentes).

4. Le Modèle Physique de Données et les AGL

Le Modèle Physique de Données consiste à implémenter une base de données dans un SGBDR, ce qui revient à employer un script SQL.

Dans la pratique, on utilisera un Atelier de Génie Logiciel (AGL) pour automatiquement générer le MPD (scripts SQL) à partir du MLD. Pour les plus connus et les plus exhaustifs, on pourra citer « Power Designer » (anciennement nommé « Power AMC » et encore plus anciennement « AMC Designor ») et « Win'Design » (tous deux payants). Les outils gratuits sont nombreux, changeants, et de qualité très variable – citons par exemple « AnalyseSI » ou « JMerise ».

En général, un AGL qui implémente Merise permet de « dessiner » le MCD. Une fonctionnalité permet ensuite d'en tirer automatiquement le MLD puis de l'ajuster. Finalement, un outil permet de générer automatiquement le script SQL (MPD) à partir du MLD, en fonction du SGBDR ciblé.