



BASES DE DONNÉES

Synoptique et aide-mémoire

Par: Vanessa GIACOMONI

Mars 2016

Table des matières

1.I.OBJET DU DOCUMENT:.....	3
2.II.NOTION DE BASE DE DONNÉES:.....	3
3.LES TYPES D'ORGANISATIONS:.....	3
4.LES SYSTÈMES DE GESTION DE BASES DE DONNÉES:.....	4
5.LES BASES DE DONNEES RELATIONNELLES:.....	5
5.1.LE MODÈLE CONCEPTUEL:.....	5
5.1.1.LES ENTITES:.....	5
5.1.2.FORMALISME DE REPRÉSENTATION DES ENTITÉS:.....	6
5.1.3.REPRÉSENTATION DES ATTRIBUTS RÉPÉTITIFS:.....	7
5.1.4.LES ASSOCIATIONS:.....	8
5.1.5.FORMALISME DE REPRÉSENTATION DES ASSOCIATIONS:.....	9
5.1.6.LES SCHÉMAS ENTITÉS-ASSOCIATIONS:.....	10
5.2.LE MODÈLE LOGIQUE:.....	11
5.2.1.REPRÉSENTATION D'UNE ENTITÉ:.....	11
5.2.2.NOTIONS DE CLEF PRIMAIRE:.....	12
5.2.3.REPRÉSENTATION GRAPHIQUE D'UNE TABLE DANS LE MLD:.....	12
5.2.4.REPRÉSENTATION ALGÈBRIQUE D'UNE TABLE:.....	12
5.2.5.REPRÉSENTATION D'UNE ASSOCIATION DANS LE MLD:.....	13
5.2.6.LES FORMES NORMALES:.....	15
5.3.LE MODÈLE PHYSIQUE (M.P.D):.....	18
6.EXEMPLES:.....	19
6.1.EXEMPLE DE MCD: Ce M.C.D décrit l'organisation (Système d'Information) d'une bibliothèque.....	19
6.2.EXEMPLE DE M.L.D:.....	20
6.3.EXEMPLE DE MODÈLE PHYSIQUE (SQL):.....	21

1.I.OBJET DU DOCUMENT:

Ce document est une présentation synoptique des principaux concepts liés aux BASES DE DONNEES. Après une rapide présentation du concept général de base de donnée informatique et aux différentes organisations possibles des données, le document traite plus particulièrement les Système De Gestion de Bases de Données RELATIONNELLE (SGBD relationnelles).

2.II.NOTION DE BASE DE DONNÉES:

Physiquement, une Base de Donnée informatique (BD) est constituée d'un ensemble de fichiers. Cependant, contrairement à un ensemble de fichiers quelconques (fichiers "à plat"), les données contenues dans ces fichiers sont ORGANISÉES et COHERENTES ENTRE ELLES (Ces données sont liées entre elles par des RELATIONS et ne se "contredisent pas").

Les DONNÉES d'une BASE DE DONNEES se caractérisent par:

- Leur UNICITÉ (une donnée est représentée une fois et une seule);
- Leur NON AMBIGUÏTÉ (une donnée est parfaitement identifiée et ne peut être confondue avec une autre);
- Leur COHERENCE (deux données d'une même BD ne se contredisent pas);
- Leur INTÉGRITÉ (leur existence et leur valeur est protégée par des mécanismes de contrôle d'accès).

3.LES TYPES D'ORGANISATIONS:

Il existe 4 types principaux d'organisation:

- Les BD HIÉRARCHIQUES: les données sont organisées en arborescence suivant des relations de type "père-fils";
- Les BD RÉSEAU: les données sous la forme de GRAPHEs aux parcours prédéterminés;
- Les BD RELATIONNELLES: les données sont structurées entre elles d'une manière dynamique par des ASSOCIATIONS qui peuvent être de nature diverses;
- Les BD OBJETS: ce sont des BD relationnelles dans lesquelles, contrairement aux BD relationnelles classiques, les données peuvent être des structures complexes (par exemple, des objets rassemblant des données internes et des méthodes agissant sur ces données).

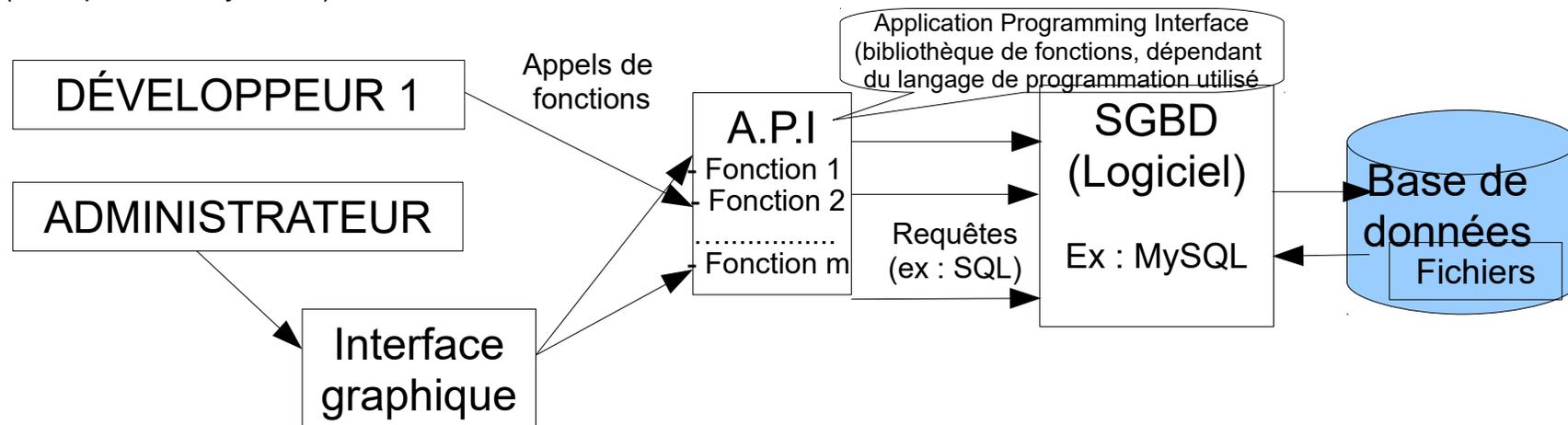
4.LES SYSTÈMES DE GESTION DE BASES DE DONNÉES:

Les SGBD sont des LOGICIELS qui permettent de gérer des bases de données d'un certain type. Ils permettent:

- De gérer le CONTENU (écrire, lire, supprimer) une donnée, sauvegarder ou restaurer les contenus;
- De mettre en oeuvre les MÉCANISMES liés au type de base de données concerné;
- De contrôler l'ACCÈS à la base de données.

L'utilisateur accède généralement à un SGBD par deux moyens:

- Les DÉVELOPPEURS accèdent à un SGBD par un interface de programmation (A.P.I) qui dépend du langage informatique qu'ils utilisent;
- Les ADMINISTRATEURS utilisent un interface graphique en ligne qui leur permet d'activer les différentes fonctions du SGBD (exemple: PHPMyAdmin).



5.LES BASES DE DONNEES RELATIONNELLES:

Une BD RELATIONNELLE peut être décrite selon trois niveaux d'analyse :

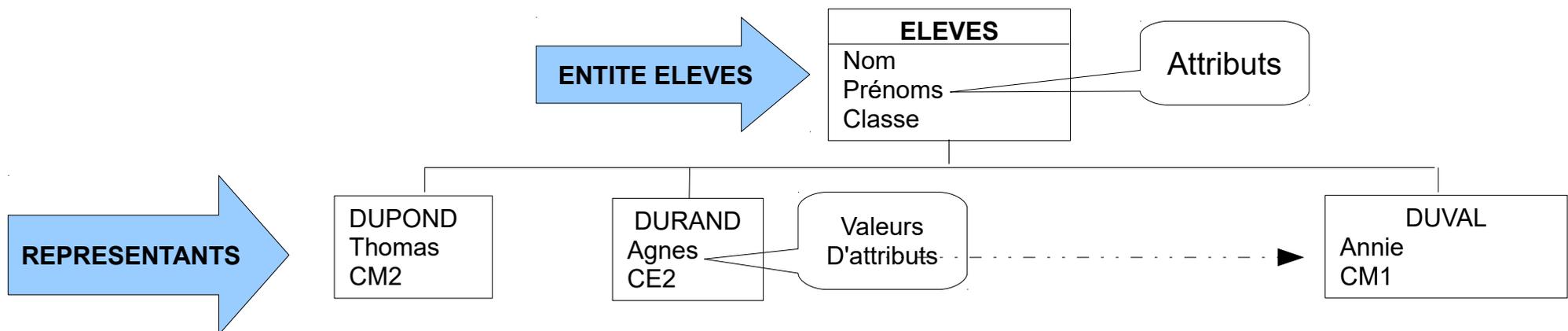
- Le MODELE CONCEPTUEL (MCD), qui représente le point de vue des UTILISATEURS;
- Le MODELE LOGIQUE (MLD), qui représente le point de vue des CONCEPTEURS;
- Le MODELE PHYSIQUE (MPD), qui est représenté par le CODE OBJET de l'application (langage de requêtes).

5.1.LE MODÈLE CONCEPTUEL:

Dans le modèle relationnel, les données sont représentées sous la forme d'ENTITÉS qui peuvent être reliées entre elles par des ASSOCIATIONS:

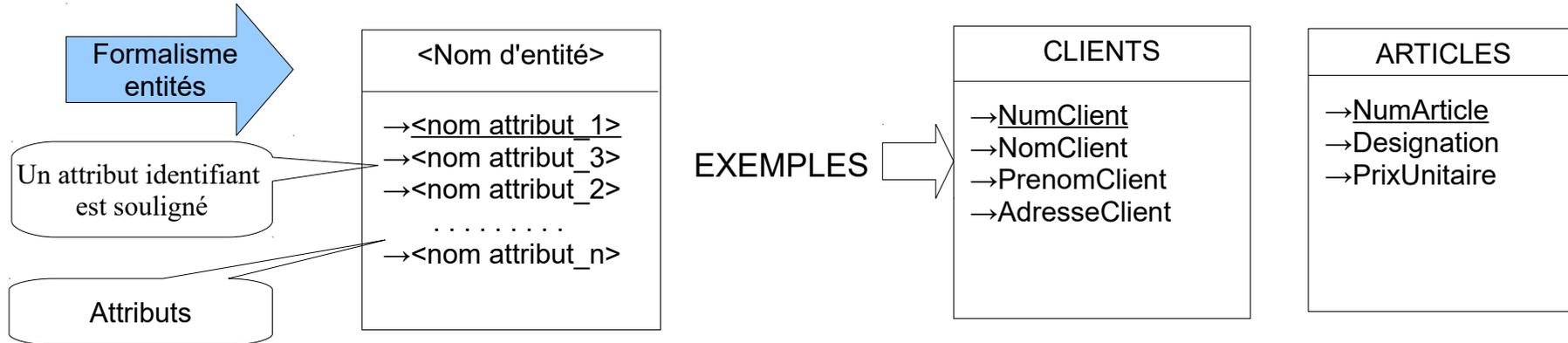
5.1.1.LES ENTITES:

- Une ENTITÉ correspond à une CATÉGORIE d'objets (ici, un OBJET peut désigner une personne ou une chose);
- A une ENTITÉ donnée peuvent être associés des ATTRIBUTS. Chacun des objets représentant d'une entité peut être caractérisé par la valeur de ses attributs. Dans le schéma ci-dessous, chaque élève représentant de l'entité ELEVES peut être caractérisé (distingué des autres) par son nom, ses prénoms et la classe à laquelle il appartient.



5.1.2.FORMALISME DE REPRÉSENTATION DES ENTITÉS:

Le schéma ci-dessous présente le formalisme de la représentation graphique des entités:

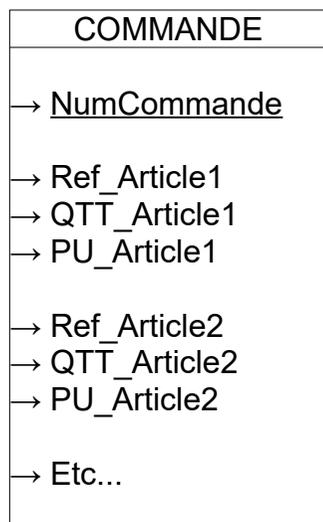


REMARQUES:

- Les noms d'entités doivent être des NOMS COMMUNS (substantifs) au PLURIEL (Clients, Articles, etc.);
- Un IDENTIFIANT est un attribut ou un groupe d'attributs qui permet d'identifier sans ambiguïté chacun des représentants d'une entité. Par exemple, chaque élève d'une entité ELEVES peut, en général, être identifié par le groupe d'attributs {<u>Nom</u>, <u>Prénoms</u>}, ou encore, chaque article de l'entité ARTICLES peut être identifié par son numero d'article (numero de nomenclature). Dans les schémas, les attributs participant à l'identifiant sont soulignés.

5.1.3. REPRÉSENTATION DES ATTRIBUTS RÉPÉTITIFS:

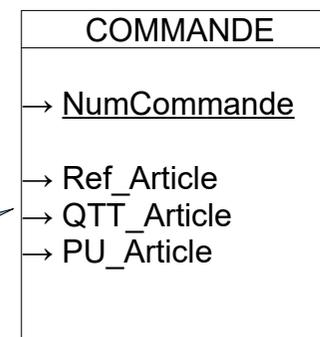
Certaines entités peuvent se voir doter d'attributs ou de groupes d'attributs RÉPÉTITIFS: par exemple, une commande émise par un fournisseur mentionne souvent plusieurs articles, avec, par exemple; pour chaque article sa référence, la quantité commandée et le prix unitaire. On aurait tendance à traduire ce dans le MCD par le schéma de gauche ci-dessous:



Dans ce cas, au niveau du MCD, on se contente de faire figurer un seul représentant de ce groupe répétitif, comme le montre le schéma de droite.

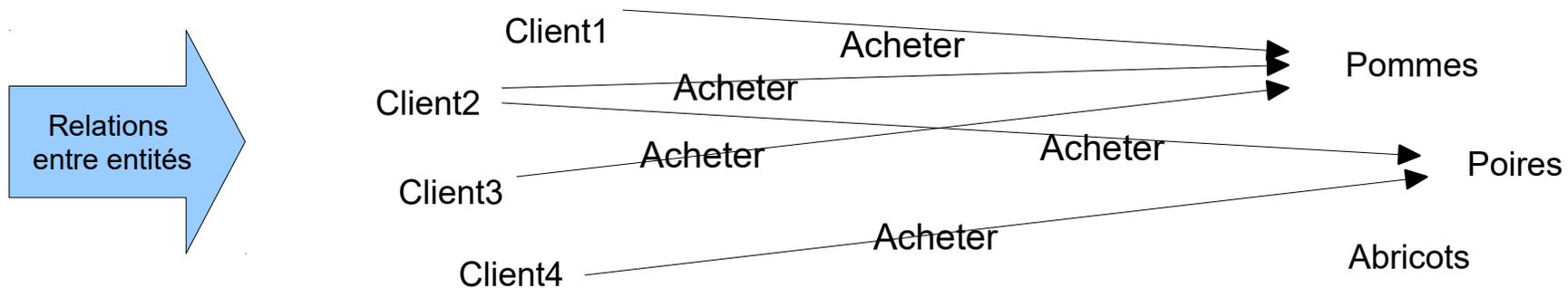
le problème de la répétitivité sera résolu lors de l'élaboration du MLD.

Un seul représentant du groupe répétitif est représenté dans le MCD



5.1.4.LES ASSOCIATIONS:

Elles permettent d'établir des correspondances logiques entre les représentants de plusieurs entités. Ainsi, dans le schéma ci-dessous, l'entité CLIENTS a pour représentants CLIENT1, CLIENT2, CLIENT3, CLIENT4. L'entité ARTICLE a pour représentants Pommes, Poires, Abricots. L'association ACHETER peut être représentée par le graphe ci-dessous:



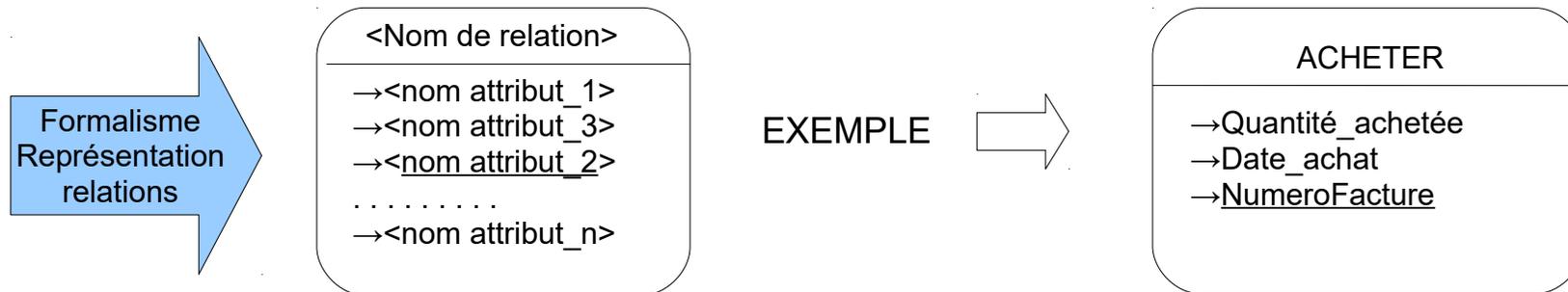
Nous voyons que l'association ACHETER recouvre l'ensemble des ACHATS effectués par les représentants de l'entité CLIENTS et concernant les représentants de l'entité ARTICLES. Une ASSOCIATION regroupe donc, comme une entité, un ensemble de REPRÉSENTANTS. Dans le schéma ci-dessus, chaque flèche schématise un représentant de l'association Acheter (dans le cas présent, ces représentants sont des "actes d'achat").

Comme dans le cas des entités, Il est possible d'associer à une association des ATTRIBUTS qui permettent de caractériser chaque représentant de cette association. Par exemple, dans le cas ci-dessus, on peut associer à la relation ACHETER l'attribut "QuantitéAchetée", qui permet de caractériser chaque achat. On pourrait également définir les attributs "DateAchat" ou "NumeroFacture".

Comme dans le cas des entités, il est possible de définir des ATTRIBUTS IDENTIFIANTS (par exemple, le numéro de facture est un identifiant pour un achat).

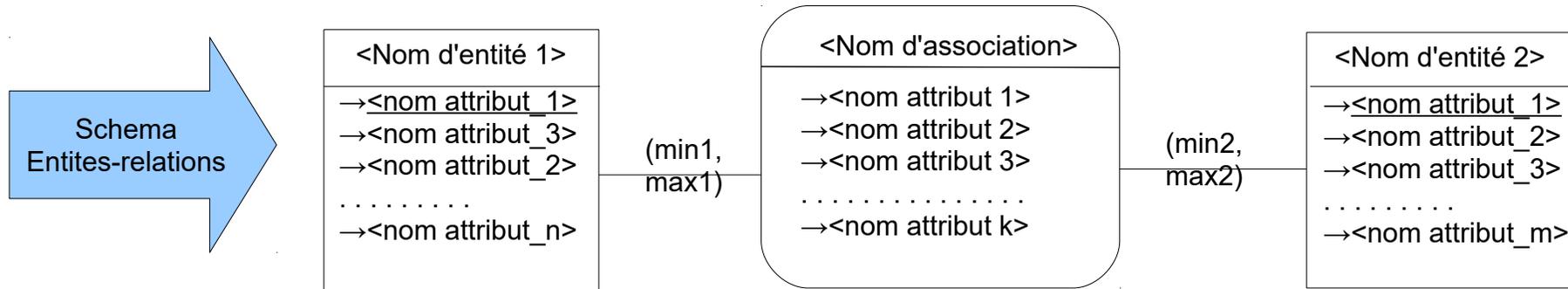
5.1.5.FORMALISME DE REPRÉSENTATION DES ASSOCIATIONS:

Le nom d'une association est un VERBE ou un GROUPE VERBAL. Une association peut, comme une entité, porter des attributs qui la caractérisent:

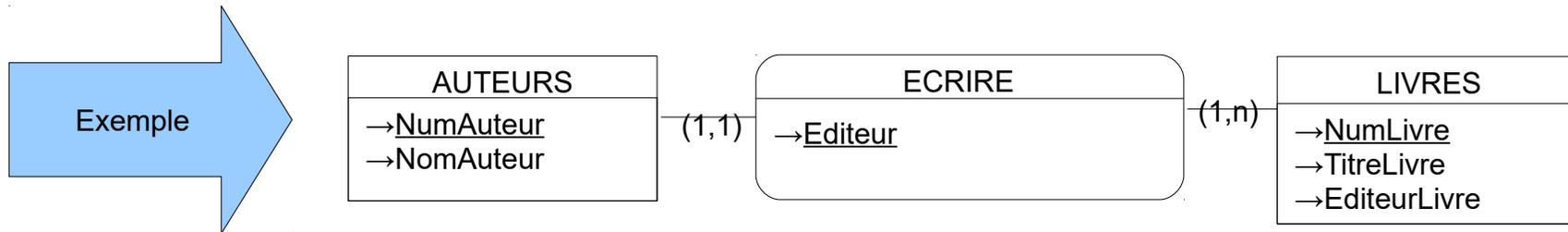


Le formalisme de représentation se distingue de celui des entités par la forme arrondie des angles du cadre.

5.1.6.LES SCHÉMAS ENTITÉS-ASSOCIATIONS:



Le schéma ci-dessus représente une association BINAIRE (qui relie deux entités). Les couples de nombres qui sont associés à chaque lien (nombres entiers positifs ou nuls) représentent la CARDINALITÉ de la relation.



EXPLICATION CARDINALITÉ:

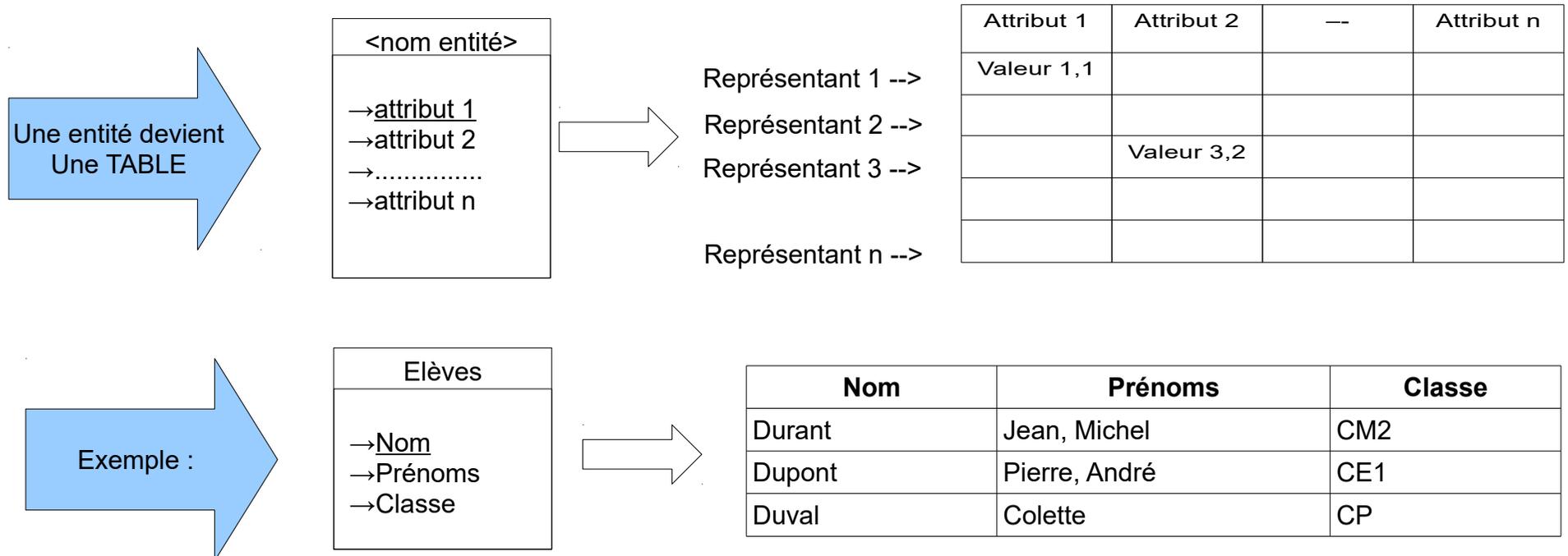
- La cardinalité (1,n) à gauche de l'association écrire signifie: un auteur peut avoir écrit entre 1 et n livres (pour être auteur, il faut au moins avoir écrit un livre);
- La cardinalité (1,1) à droite signifie: un livre est écrit par un auteur et un seul.

5.2.LE MODÈLE LOGIQUE:

Le modèle LOGIQUE a pour objectif de représenter le modèle CONCEPTUEL par des objets ayant un sens du point de vue INFORMATIQUE, de façon à faciliter son écriture sous la forme d'un PROGRAMME informatique. En fait, le modèle logique utilise une seule sorte d'objet informatique: le TABLEAU DE DONNÉES à deux entrées, que l'on appelle souvent RELATION dans le modèle logique

5.2.1.REPRÉSENTATION D'UNE ENTITÉ:

Une entité du MCD devient une TABLE du MLD: les ATTRIBUTS deviennent des COLONNES du tableau (on dit également des CHAMPS), tandis que les REPRÉSENTANTS de l'entité deviennent des LIGNES (ou ENTRÉES) de la table:

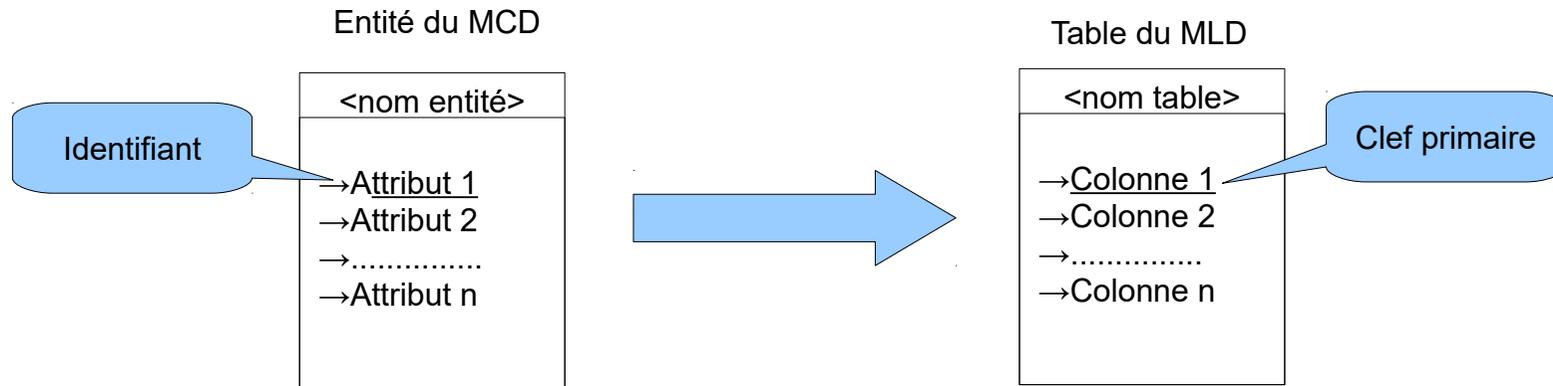


5.2.2. NOTIONS DE CLEF PRIMAIRE:

Dans le modèle logique, la notion de CLEF PRIMAIRE est équivalente à la notion d'IDENTIFIANT dans le MCD. Une CLEF PRIMAIRE est donc représentée par un ou plusieurs colonnes de la table représentant l'entité.

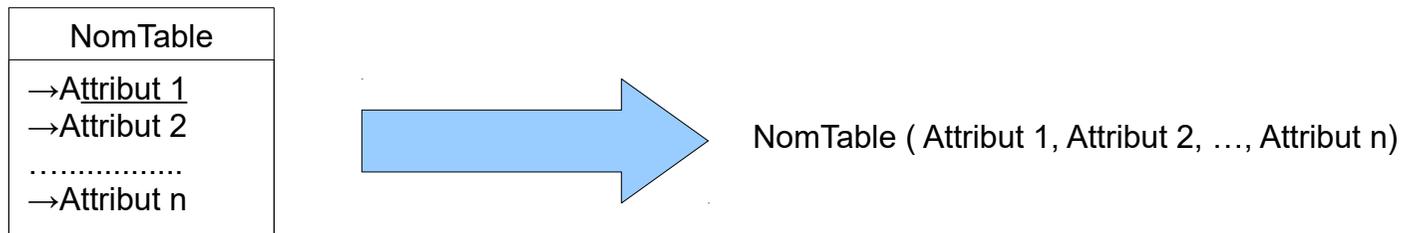
5.2.3. REPRÉSENTATION GRAPHIQUE D'UNE TABLE DANS LE MLD:

Le formalisme de représentation d'une TABLE du MLD est pratiquement identique à celui d'une ENTITÉ du MCD:



La colonne 1 de la table porte le nom de l'attribut 1 de l'entité correspondants, et ainsi de suite...

5.2.4. REPRÉSENTATION ALGÈBRIQUE D'UNE TABLE:

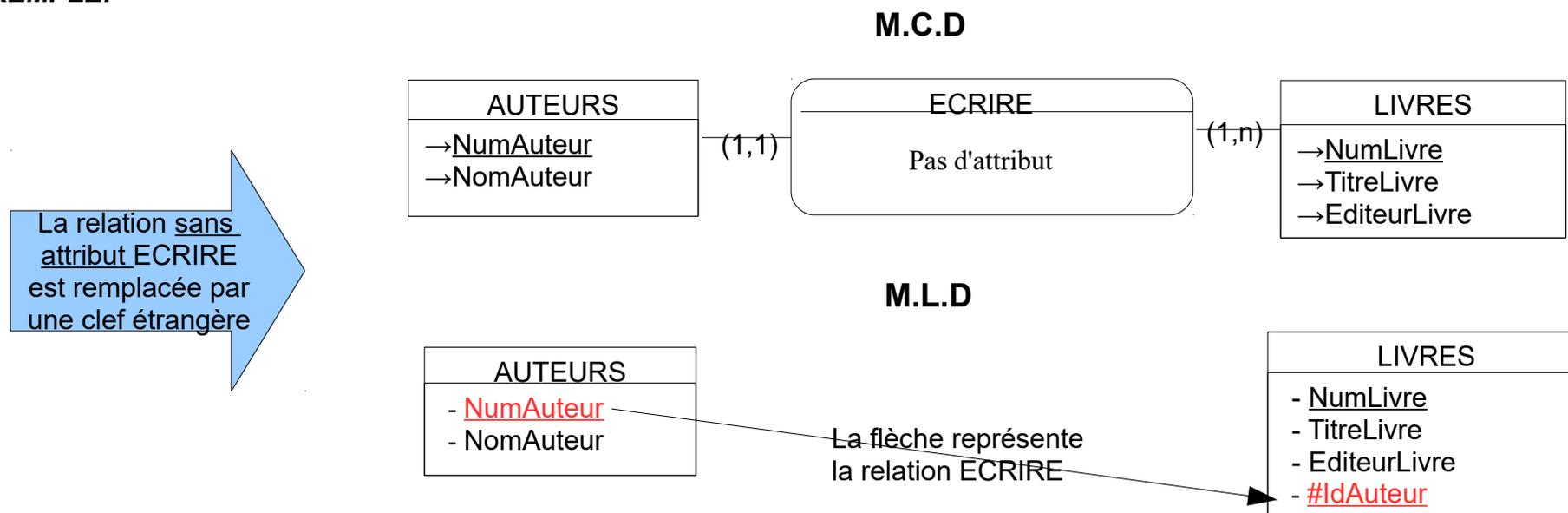


5.2.5.REPRÉSENTATION D'UNE ASSOCIATION DANS LE MLD:

CAS D'UNE ASSOCIATION NON PORTEUSE D'ATTRIBUTS ET DONT UNE SEULE DES CARDINALITÉS MAXIMALE EST > 1:

Ces associations sont dites de **TYPE 1,n**. Dans ce cas, l'association peut être simplement remplacée par l'ajout d'une CLEF ÉTRANGÈRE dans la table de droite. Une CLEF ÉTRANGÈRE est une COLONNE (ou une ensemble de colonnes) qui est ajoutée dans une table pour référencer les entrées d'une autre table.

EXEMPLE:

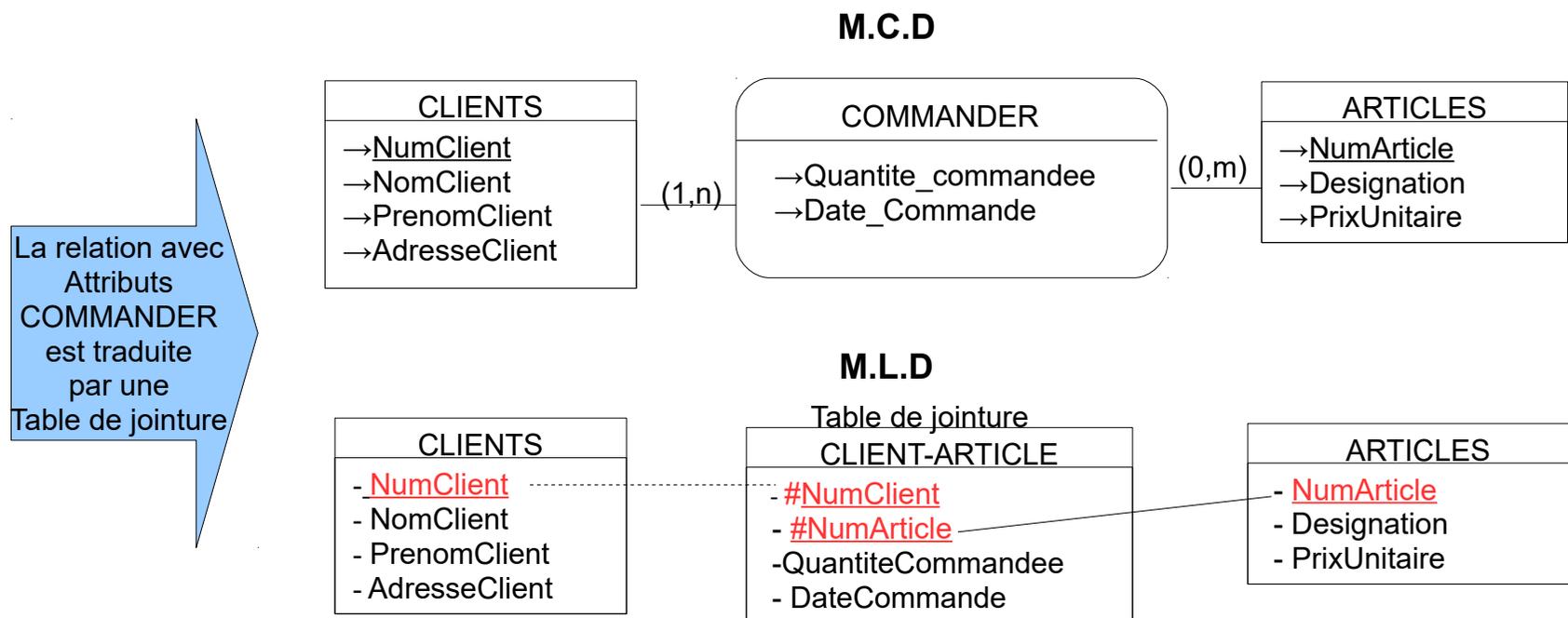


Dans la table LIVRES, on ajoute la colonne #IdAuteur dont les valeurs correspondent à celles de la colonne NumAuteurs de la table AUTEURS. Les valeurs des champs de la clef étrangère #IdAuteur référencent donc les entrées de la table AUTEURS, ce qui permet de déterminer quel AUTEUR a écrit tel LIVRE. Noter le préfixe # du nom de la clef étrangère.

SYNOPTIQUE SUR LES BASES DE DONNEES

AUTRES CAS: ASSOCIATION PORTEUSE D'ATTRIBUTS OU DONT LES 2 CARDINALITES MAXIMALES SONT > 1:

Ces associations sont dites de **TYPE m,n**. Dans ce cas, l'association doit être représentée par une table appelée TABLE DE JOINTURE. Outre les attributs de l'association, cette table sera munie de deux CLEFS ÉTRANGÈRES faisant référence aux CLEFS PRIMAIRES des tables mises en relation:



Le couple de clefs étrangères (#NumClient, #NumArticle) permet de mettre en correspondance un article avec le client qui l'a commandé. Il matérialise donc l'association COMMANDER par la table de jointure CLIENT-ARTICLE.

5.2.6.LES FORMES NORMALES:

RAPPEL PRÉLIMINAIRE:

Le terme RELATION est équivalent au terme TABLE dans l'algèbre relationnelle.

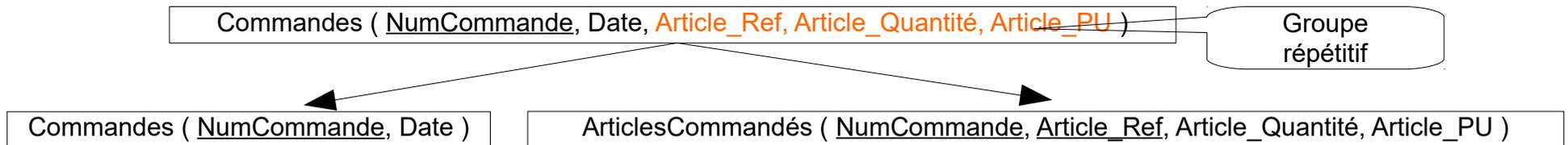
PREMIÈRE FORME NORMALE:

Définition:

Une RELATION est en PREMIÈRE FORME NORMALE (1NF) si elle ne contient aucun groupe d'attributs RÉPÉTITIF.

Transformation d'une table contenant des groupes répétitifs en tables 1NF:

Chaque groupe répétitif est traduit par une nouvelle table dont la clef primaire est celle de la table d'origine:



La clef NumCommande de la table Articles permet de référencer pour chaque commande, les différentes lignes de commandes qui la composent. Les deux tables résultantes sont en 1NF.

SYNOPTIQUE SUR LES BASES DE DONNEES

DEUXIÈME FORME NORMALE:

Définition:

Une relation est en DEUXIÈME FORME NORMALE (2NF) si et seulement si:

- Elle ne contient aucun groupe d'attributs RÉPÉTITIF (Donc, elle est en 1NF);
- Tous ses attributs qui ne sont pas des clefs primaires **dépendent de l'ensemble des attributs** qui forment cette clef primaire.

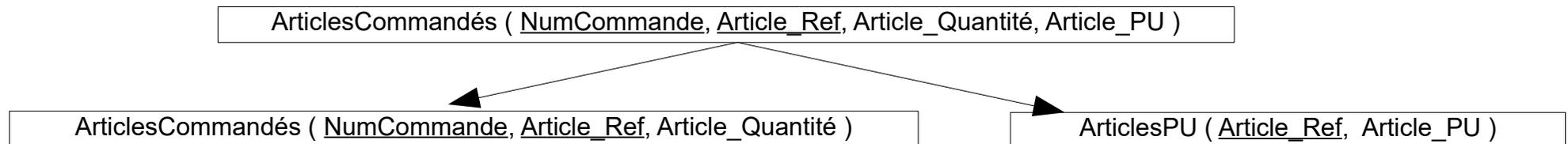
Exemple:

La table ArticlesCommandés (NumCommande, Article_Ref, Article_Quantité, Article_PU), obtenue précédemment est en 1NF. En revanche, on peut constater que l'attribut Article_PU (prix unitaire de l'article), s'il dépend bien de Article_Ref, ne dépend pas de NumCommande (à priori, le prix unitaire d'un article ne dépend pas de la commande). La table n'est donc pas en 2NF.

Remarque: une relation qui est en 1NF et dont la clef est SIMPLE (une seule colonne) est forcément en 2NF.

Transformation d'une table 1NF en 2NF:

Il suffit de transférer les attributs qui ne dépendent que d'une partie de la clef dans une nouvelle table:



Les deux tables résultantes sont alors en 2NF.

SYNOPTIQUE SUR LES BASES DE DONNEES

TROISIÈME FORME NORMALE 3NF:

Définition:

Une RELATION est en TROISIEME FORME NORMALE (2NF) si et seulement si:

- Elle ne contient aucun groupe d'attributs RÉPÉTITIF (Donc, elle est en 1NF) ;
- Tous ses attributs qui ne sont pas des clefs primaires **dépendent de l'ensemble des attributs** qui forment cette clef primaire (donc, elle est en 2NF);
- Aucun de ses attributs n'est DÉPENDANT TRANSITIVEMENT d'un autre de ses attributs.

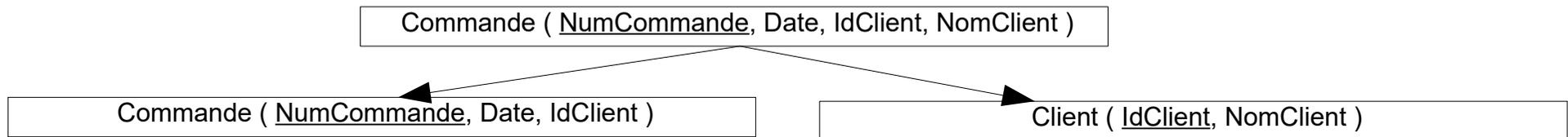
Exemple:

La table Commande (NumCommande, Date, IdClient, NomClient) est en 2NF car elle ne contient aucun groupe répétitif et sa clef primaire est simple. Cependant, on peut remarquer que la valeur de l'attribut NomClient dépend de la valeur de l'attribut IdClient. Or, dans une table en 3NF, un attribut ne peut dépendre que de la clef (et de toute la clef). Il y a donc **dépendance transitive** de NomClient:

NumCommande ---- IdClient ---- NomClient

Transformation d'une table en 2NF en tables en 3NF:

Il suffit de transférer dans une autre table les attributs qui dépendent transitivement de la clef:

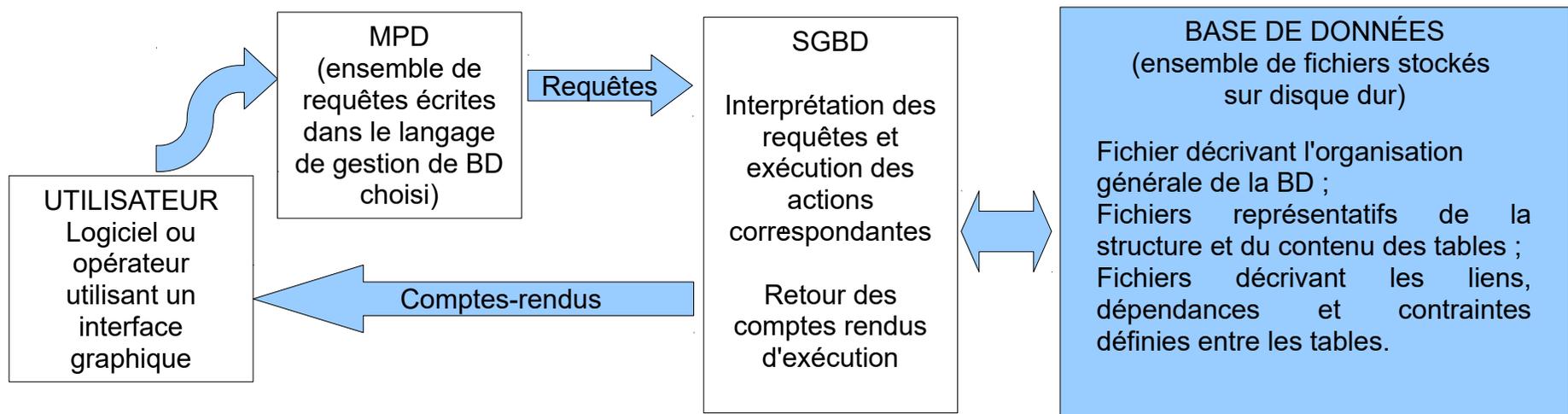


Les deux tables résultantes sont alors en 3NF.

5.3.LE MODÈLE PHYSIQUE (M.P.D):

Le M.P.D est la traduction du MLD) dans le LANGAGE DE REQUÊTES accepté par le SGBD. C'est l'ensemble des REQUÊTES qu'il faut envoyer au SGBD pour que celui-ci crée la base de données définie par le modèle logique sur le support de données choisi:

- Données d'organisation générale de la BD (données d'accès, charsets utilisés par défaut, etc.);
- Données contenues dans chacune des tables définies par le MLD;
- Données décrivant l'organisation interne de ces tables (formats de données, moteurs de BD employés, définition de clefs primaires, etc.);
- Données décrivant les liens, dépendances et contraintes définies entre les tables (exemple: clefs étrangères).



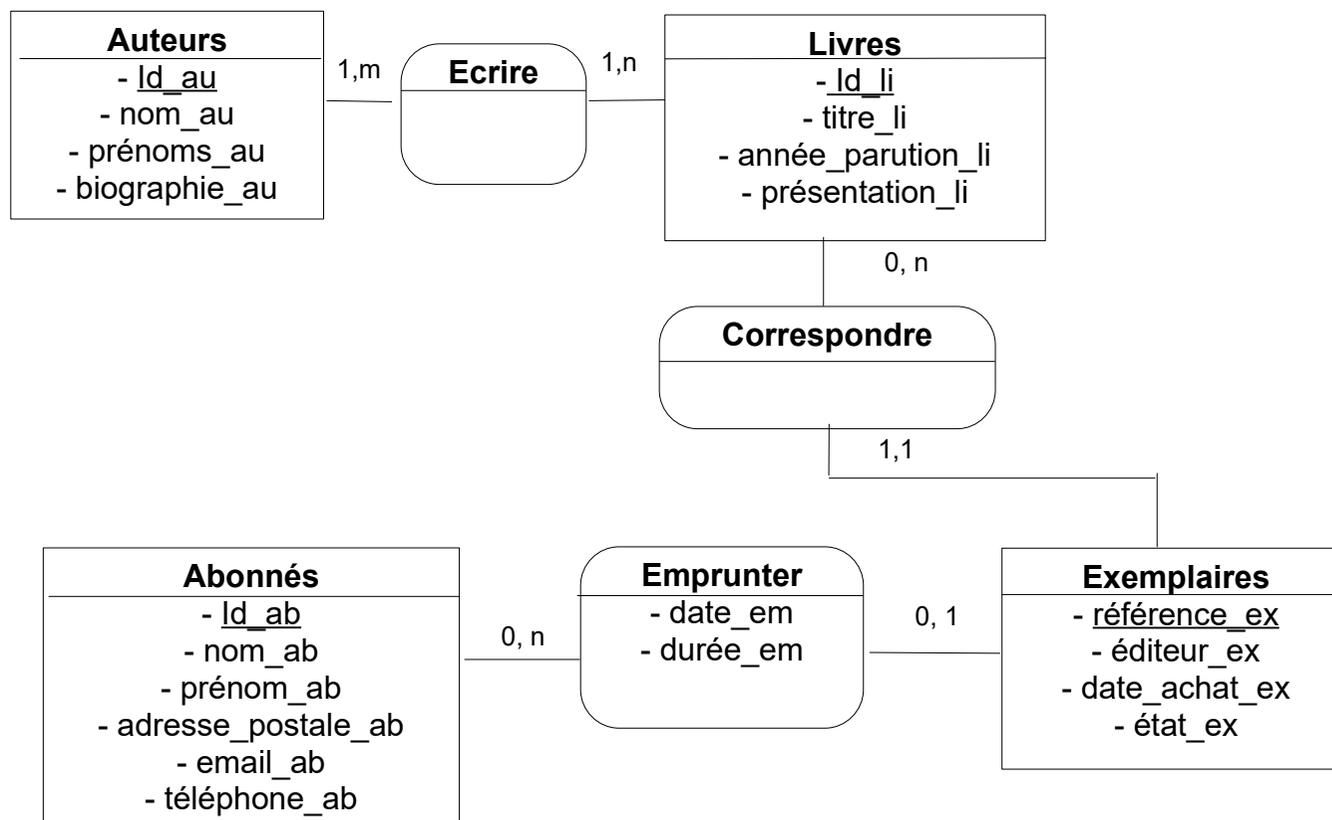
6.EXEMPLES:

6.1.EXEMPLE DE MCD: Ce M.C.D décrit l'organisation (Système d'Information) d'une bibliothèque.

REMARQUE :

Pour doter les entités Auteurs, Livres et Abonnés, des attributs supplémentaires *Id* (numérotation des occurrences) ont dû être ajoutés aux entités.

Ce n'est pas le cas pour l'entité Exemplaires car la référence d'un exemplaire, attribuée au moment de l'achat, est choisie pour être unique (ex : 2212201507 = date d'achat (22/12/2015)+ n° de l'achat dans la journée (07)).



SYNOPTIQUE SUR LES BASES DE DONNEES

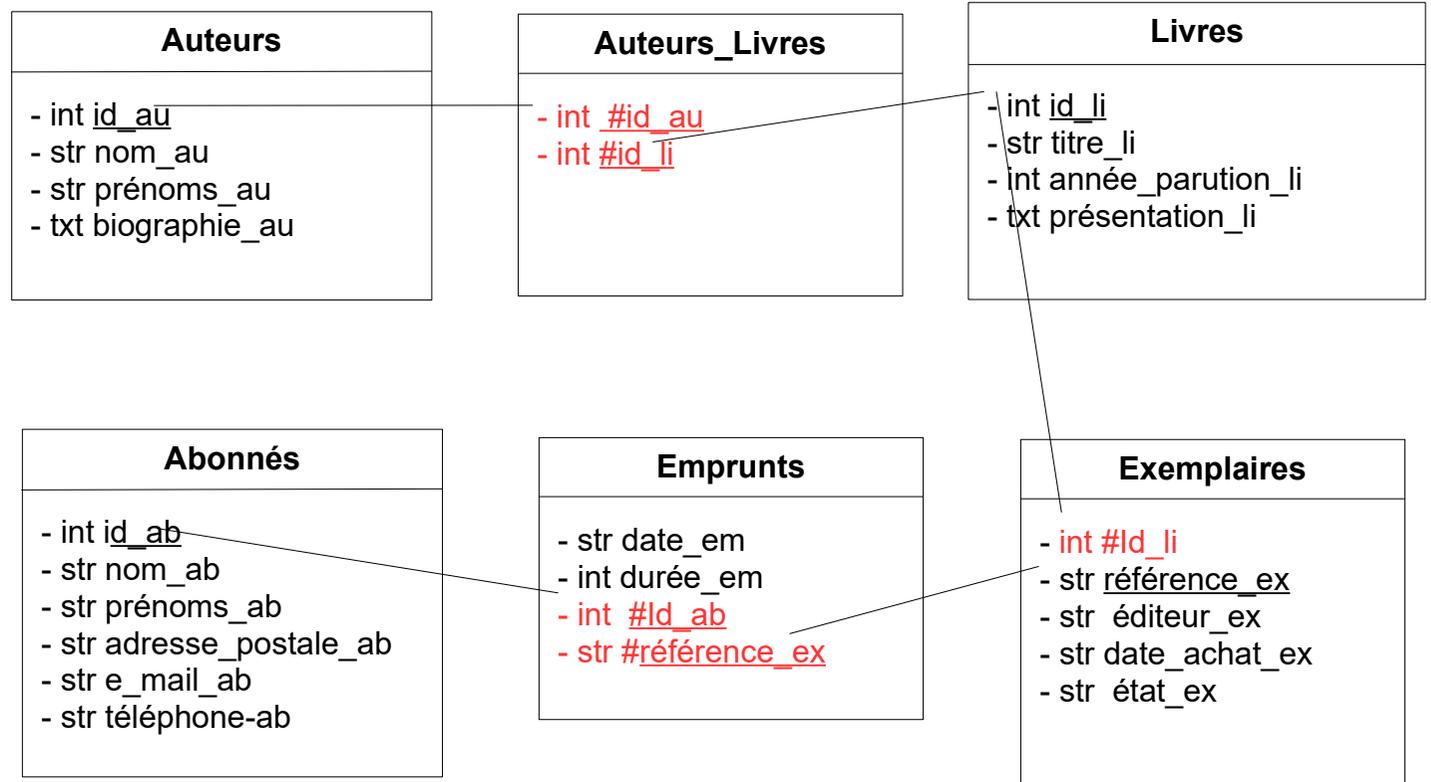
6.2.EXEMPLE DE M.L.D:

Le M.C.D ci-dessous traduit le M.C.D du paragraphe précédent:

→ L'association ECRIRE ne possède pas d'attribut, mais elle est de type m,n. On la traduit donc par la table de jonction Auteurs_Livres. Les clefs primaires des 2 entités mises en relation deviennent clefs étrangères de la table de jonction.

→ L'association Correspondre ne possède pas d'attribut, et une seule des cardinalités supérieure est > 1. Elle est donc de type 1,n. De ce fait, il suffit d'exporter la clef primaire de l'entité du côté 1 (Livres) en tant que clef étrangère de l'entité Exemplaires.

→ L'association EMPRUNTER possède des attributs. Nous la représenterons par la table de jonction Emprunts qui contiendra les attributs d'EMPRUNTER et les clefs primaires des entités Abonnés et Exemplaires, exportées en tant que clefs étrangères.



6.3.EXEMPLE DE MODÈLE PHYSIQUE (SQL):

Ce qui suit représente le programme informatique en langage de requêtes S.Q.L. (Structures Query Language) qui permet de créer la base de données Bibliothèque:

```
--
-- Création de la base de données 'bibliothèque'(si elle n'existe pas), avec comme option l'utilisation
-- par défaut du jeu de caractères latin 1 (ISO-8859-1) et du type d'interclassement alphabétique
-- latin1_swedish_ci (ordre alphabétique utilisé en France).
--
CREATE DATABASE IF NOT EXISTS `bibliotheque` DEFAULT CHARACTER SET latin1 COLLATE latin1_swedish_ci;

--
-- Spécifie que les requêtes qui suivent s'adressent à la base de données bibliothèque
--
USE `bibliotheque`;

--
-- Première ligne: création de la table 'abonnes'
-- Lignes suivantes: définition de ses champs 'id_ab', 'nom_ab', 'prenoms_ab', 'adresse_postale-ab',
-- 'e_mail_ab' et 'telephone_ab' (champs. Pour chacun des champs, son type de donnée est défini (int(8),
-- varchar(100), etc.
-- id_ab est défini comme une clef primaire (mot clef PRIMARY KEY) autoincrémentée (mot clef
-- AUTOINCREMENT)
-- La dernière ligne définit : le moteur de BD à utiliser (ici : innoBD), le jeu de caractère par défaut
-- (latin 1) et la valeur d'auto-incrémentation à appliquer pour la table (1).
--
CREATE TABLE IF NOT EXISTS `abonnes` (
  `id_ab` int(8) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Id. des abonnés',
  `nom_ab` varchar(100) NOT NULL COMMENT 'nom de l''abonné',
  `prenoms_ab` varchar(100) NOT NULL COMMENT 'prénoms de l''abonné, séparés par des virgules',
  `adresse_postale_ab` varchar(255) NOT NULL COMMENT 'adresse postale de l''abonné',
  `e_mail_ab` varchar(30) NOT NULL COMMENT 'adresse e_mail de l''abonné',
  `telephone_ab` int(30) NOT NULL COMMENT 'numéro de téléphone de l''abonné',
  PRIMARY KEY (`id_ab`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Table des abonnee' AUTO_INCREMENT=1 ;
```

SYNOPTIQUE SUR LES BASES DE DONNEES

```
--  
-- Les 3 requêtes suivantes définissent la structure des tables 'auteurs', 'auteurs_livres' et  
-- 'emprunts'. Les mots clefs KEY utilisés pour les deux dernières permettent d'associer à certains  
-- champs des INDEX : dans ce cas, cela est nécessaire pour que ces champs puissent être définis plus  
-- tard comme des CLEFS ETRANGERES  
--  
CREATE TABLE IF NOT EXISTS `auteurs` (  
  `id_au` int(8) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Clef auteurs',  
  `nom_au` varchar(100) NOT NULL COMMENT 'Nom de l''auteur',  
  `prenoms_au` varchar(100) NOT NULL COMMENT 'prénoms de l''auteur, séparés par des virgules',  
  `biographie_au` text NOT NULL COMMENT 'biographie de l''auteur',  
  PRIMARY KEY (`id_au`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Table des auteurs' AUTO_INCREMENT=1 ;  
  
CREATE TABLE IF NOT EXISTS `auteurs_livres` (  
  `fk_id_au` int(8) unsigned NOT NULL,  
  `fk_id_li` int(8) unsigned NOT NULL,  
  KEY `fk_id_au` (`fk_id_au`),  
  KEY `fk_id_li` (`fk_id_li`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Table de jonction auteurs-livres (association "ecrire")';  
  
CREATE TABLE IF NOT EXISTS `emprunts` (  
  `date_em` varchar(30) NOT NULL COMMENT 'Date d''emprunt',  
  `duree-em` int(2) NOT NULL COMMENT 'durée en semaines de l''emprunt',  
  `fk_id_ab` int(8) unsigned NOT NULL COMMENT 'Clef étrangère (id. de l''abonné emprunteur)',  
  `fk_reference_ex` varchar(30) NOT NULL COMMENT 'clef étrangère (référence de l''exemplaire emprunté)',  
  KEY `fk_id_ab` (`fk_id_ab`),  
  KEY `fk_reference_ex` (`fk_reference_ex`)  
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Table des emprunts en cours ';
```

SYNOPTIQUE SUR LES BASES DE DONNEES

```
--
-- Les 2 requêtes suivantes définissent la structure des tables 'exemplaires', et 'livres' et
-- 'emprunts'. Les mots clefs KEY utilisés pour les deux dernières permettent d'associer à certains
-- champs des INDEX : dans ce cas, cela est nécessaire pour que ces champs puissent être définis plus
-- tard comme des CLEFS ETRANGERES
--
CREATE TABLE IF NOT EXISTS `exemplaires` (
  `reference_ex` varchar(30) NOT NULL COMMENT 'réf érence de l''exemplaire: <nom livre>/<date acquisition>/<numéro
acquisition>',
  `editeur_ex` varchar(255) NOT NULL COMMENT 'éditeur de l''exemplaire',
  `date_achat_ex` varchar(30) NOT NULL COMMENT 'date d''achat',
  `etat_ex` varchar(20) NOT NULL COMMENT 'état (neuf, bon, moyen, médiocre, a changer',
  `fk_id_li` int(8) unsigned NOT NULL COMMENT 'Clef étrangère (Id du livre correspondant à
l''exemplaire)',
  PRIMARY KEY (`reference_ex`),
  KEY `fk_id_li` (`fk_id_li`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Table des exemplaires de livres en circulation';

CREATE TABLE IF NOT EXISTS `livres` (
  `id_li` int(8) unsigned NOT NULL AUTO_INCREMENT COMMENT 'Id du livre',
  `titre_li` varchar(255) NOT NULL COMMENT 'titre du livre',
  `annee_parution_li` varchar(20) NOT NULL COMMENT 'année de première parution',
  `presentation_li` text NOT NULL COMMENT 'présentation sommaire du livre',
  PRIMARY KEY (`id_li`)
) ENGINE=InnoDB DEFAULT CHARSET=latin1 COMMENT='Table des livres' AUTO_INCREMENT=1 ;

--
-- Enfin, les directives qui suivent permettent de définir dans les tables 'auteurs_livres', 'emprunts'
-- et 'exemplaires' les champs qui sont des clefs étrangères (mots clefs 'ADD CONSTRAINT' et
-- 'FOREIGN KEY'.
--
ALTER TABLE `auteurs_livres`
  ADD CONSTRAINT `auteurs_livres_ibfk_2` FOREIGN KEY (`fk_id_li`) REFERENCES `livres` (`id_li`),
  ADD CONSTRAINT `auteurs_livres_ibfk_1` FOREIGN KEY (`fk_id_au`) REFERENCES `auteurs` (`id_au`);
```

SYNOPTIQUE SUR LES BASES DE DONNEES

```
ALTER TABLE `emprunts`  
  ADD CONSTRAINT `emprunts_ibfk_2` FOREIGN KEY (`fk_reference_ex`) REFERENCES `exemplaires` (`reference_ex`),  
  ADD CONSTRAINT `emprunts_ibfk_1` FOREIGN KEY (`fk_id_ab`) REFERENCES `abonnes` (`id_ab`);  
  
ALTER TABLE `exemplaires`  
  ADD CONSTRAINT `exemplaires_ibfk_1` FOREIGN KEY (`fk_id_li`) REFERENCES `livres` (`id_li`);
```