

Binaire : une association relie 2 entités avec 2 branches
Clef primaire : clé primaire d'une table est une contrainte d'unicité, composée d'une ou plusieurs colonnes, permet d'identifier de manière unique chaque ligne de la table
Clef étrangère : clé étrangère représente un champ (ou des champs) qui pointe vers la clé primaire d’une autre table. L’objectif de la clé étrangère est d’assurer l’intégrité référentielle des données

CREATE TABLE Animal (
id SMALLINT AUTO_INCREMENT,
espece VARCHAR(40) NOT NULL,
sexe CHAR(1),
date_naissance DATETIME NOT NULL,
nom VARCHAR(30),
commentaires TEXT,
PRIMARY KEY (id)
) ENGINE=InnoDB;

CREATE TABLE Orders (
OrderID int NOT NULL,
OrderNumber int NOT NULL,
PersonID int,
PRIMARY KEY (OrderID),
FOREIGN KEY (PersonID) REFERENCES Persons(PersonID)
);

ALTER TABLE Orders
ADD CONSTRAINT FK_PersonOrder
FOREIGN KEY (PersonID) REFERENCES Persons(PersonID);

CLASSE PHP :

```
<?php
class Foo {
    public $aMemberVar = 'aMemberVar Member Variable';
    public $aFuncName = 'aMemberFunc';

    function aMemberFunc() {
        print 'Inside `aMemberFunc()`';
    }
}

$foo = new Foo;
?>
```

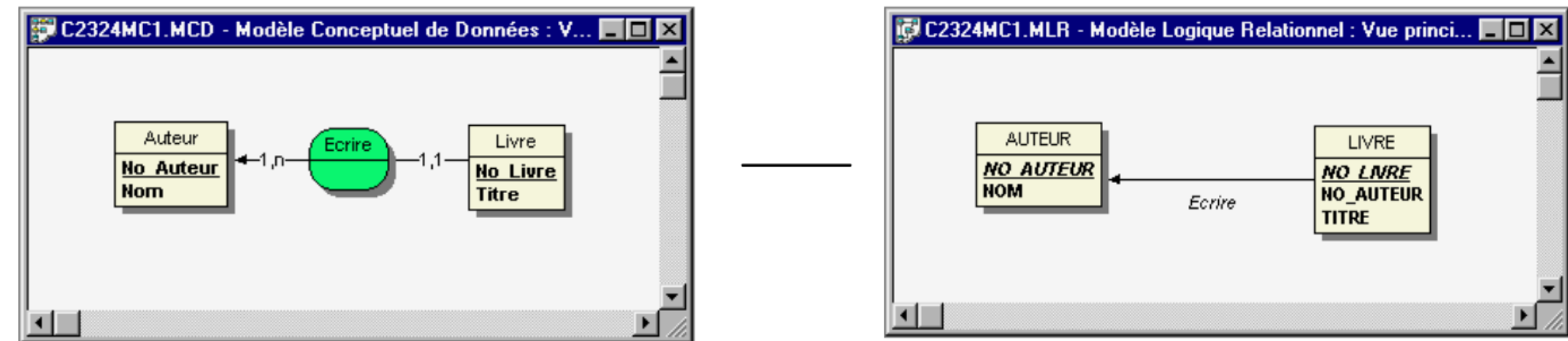
CMS : Content management system
CSS : Cascading style sheet
DAO : Data access object : classe qui permet de manager les bdd
GIT : gestion de la version décentralisé, sauvegarde en ligne et partage en ligne
GUI : Graphical utilisateur interface
Héritage : classe fille qui hérite des attributs et méthode de la classe mère
HTML : Hyper text markup langages (langage de balise)
Index bdd : fichier référence ou nous trouvons les infos (clef forcement indexer)
Instanciation : création d’un objet à partir d’une classe
JS : JavaScript langages serveur et client, employé pour des pages web interactives
Maquettage : représentation graphique
MCD : Modèle conceptuel de données
Merise : méthodes de conception, de développement et de projets informatiques
Méthode agile: sont des groupes de pratiques de pilotage et de réalisation de projets
MLD : Modèle logique de données
Orienté objet : méthode de programmation, paradigme de programmation informatique, elle sert à l’encapsulation : regrouper dans 1 meme objet les méthodes et attributs
Objet : instance de classe
Paradigme : bonne pratique , méthode pas accepté de tout
Pattern : c’est une bonne pratique
PHP : langage de programmation libre , utilisé pour produire des pages web dynamiques via un serveur HTTP
Polymorphisme : plusieurs méthode qui ont le même nom et des signatures différentes
Privilège : GRANT
Produit cartésien : requête sur plusieurs tables sans jointures
Procédure stockée : programme déclenchant plusieurs requête SQL
Produit cartésien : requête sur plusieurs table sans jointure
REGEX : Expression régulière (vérification sur les entrées)
RGPD : règlement général de la protection des données
Surcharge : méthode qui ont la même nom et des signatures différentes
Ternaire : association peut aussi relier 3 ou très rarement 4 entités
Tertiaire :
Trigger : élément déclenchant des événements

Function JS :

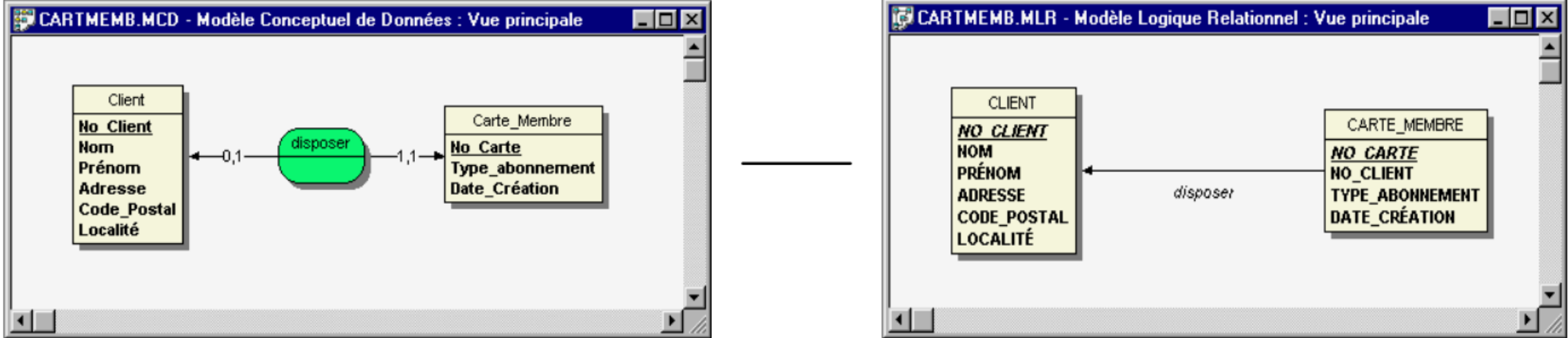
```
function Personne(nom) {
    this.nom = nom;
    this.salutation = function() {
        alert('Bonjour ! Je m\'appelle ' + this.nom + '.');
    };
}
```

UI : interface utilisateur
UML : méthode qui modélisent le projet de manière générale pour visualiser la conception d’1 système
UX : expérience utilisateur permet de savoir le comportement de l’utilisateur par exemple auto complétion
Vues : table virtuel qui joint plusieurs table pour une jointure

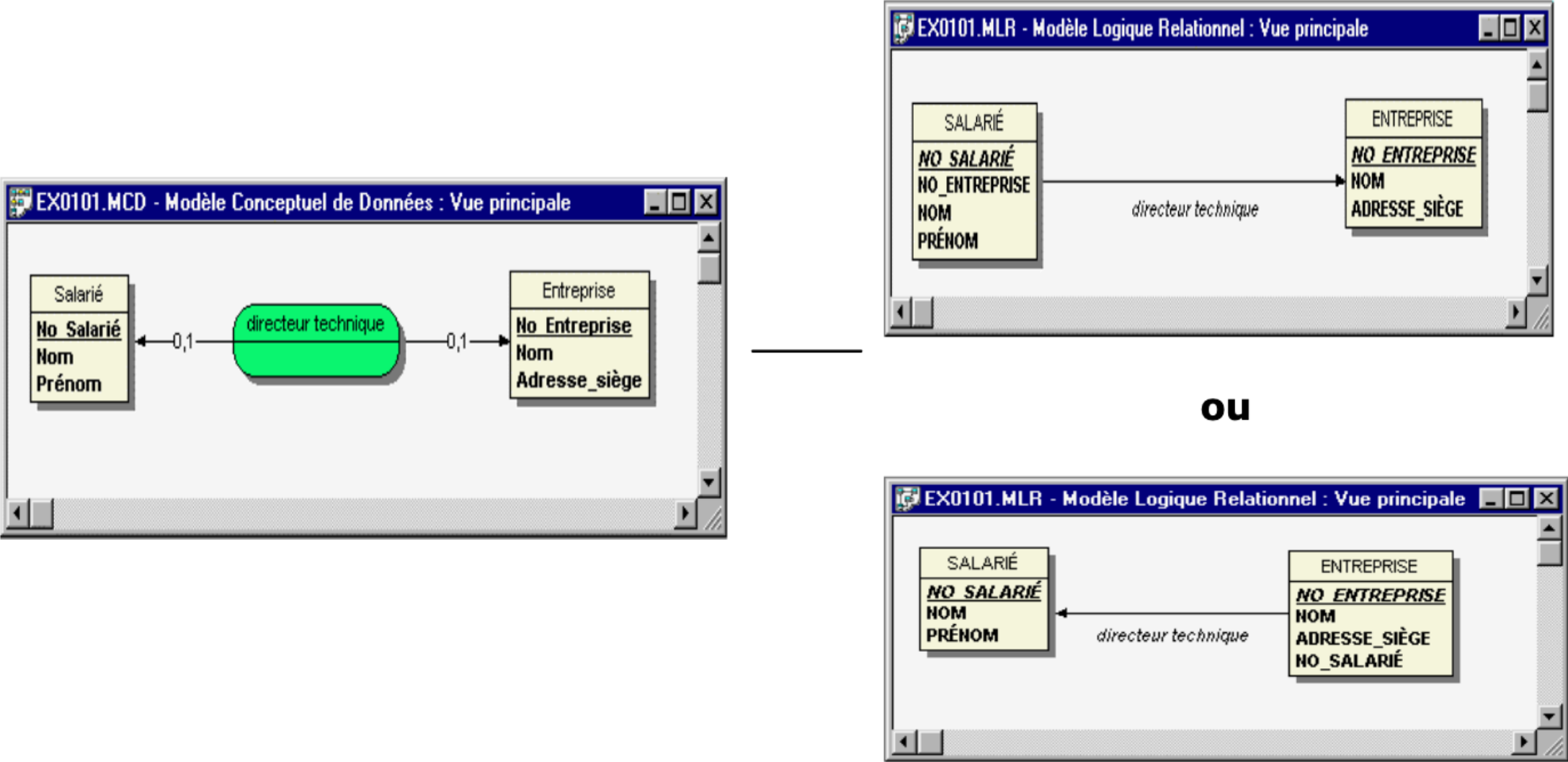
MCD -> MLD



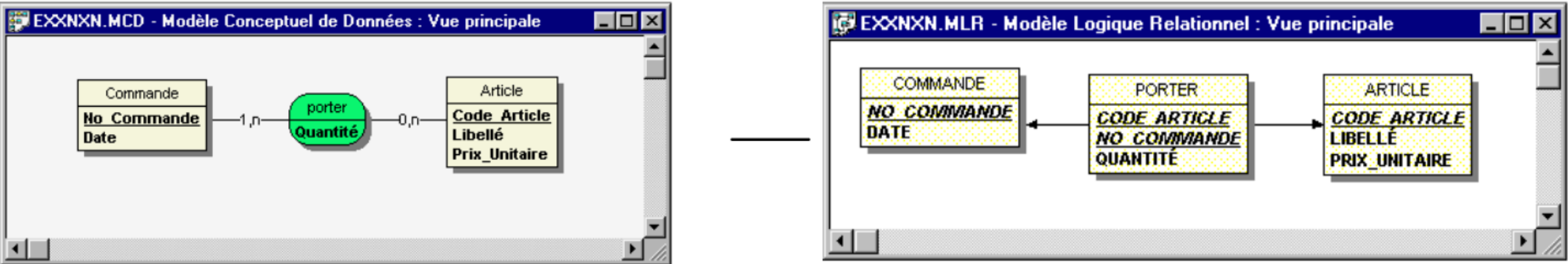
L'attribut *No_Auteur* qui est clé primaire de la table *Auteur*, devient clé étrangère dans la table *Livre*.



Le *No_Client*, qui est clé primaire de la table *Client*, devient clé étrangère dans la table *Carte_Membre*.



ou

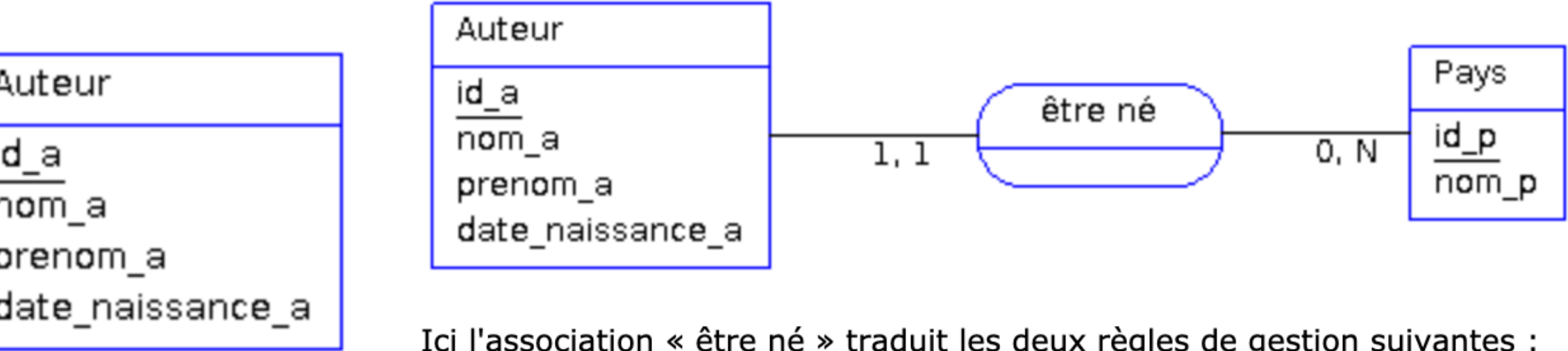


On crée une table *Porter*, qui contient comme clé primaire une clé composée de *No-Commande* et *Code_Article*. Elle contient également la propriété *Quantité* issue de la relation *Porter*.

Soit on migre la clé primaire de la table *Entreprise* dans la table *Salarie*, soit on fait l'inverse.

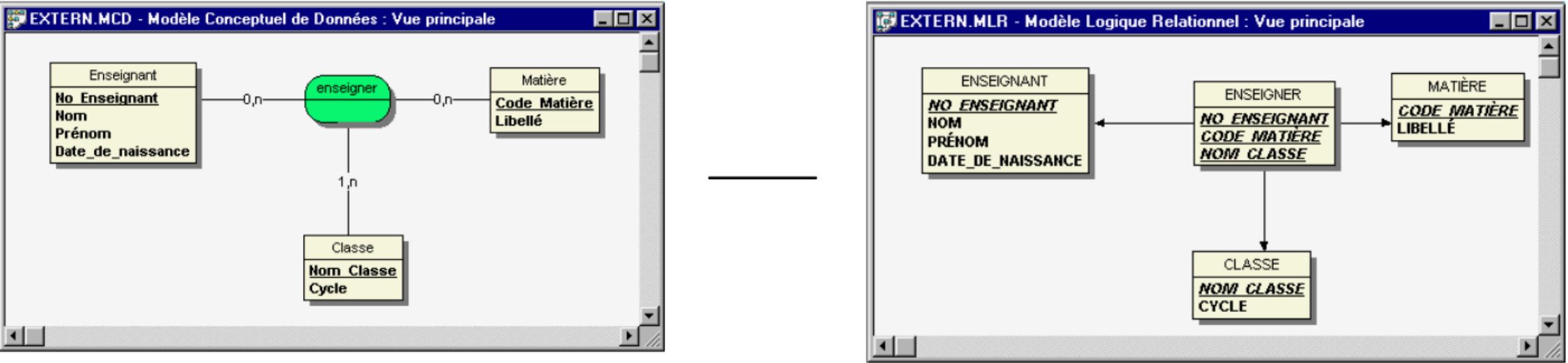
MCD :

Associations :



Ici l'association « être né » traduit les deux règles de gestion suivantes :

- un auteur est né dans **un et un seul** pays ;
- dans un pays, sont nés **aucun, un ou plusieurs** auteurs.



La table *Enseigner* contient une clé composée de *No_Enseignant*, *Code_Matiere* et *Nom_Classe*.

MLD -> SQL

```
CREATE TABLE Pays (  
    id_p INT NOT NULL,  
    nom_p VARCHAR(50),  
    PRIMARY KEY (id_p)  
);
```

```
CREATE TABLE Auteur (  
    id_a INT NOT NULL,  
    nom_a VARCHAR (30),  
    prenom_a VARCHAR (30),  
    date_naissance_a DATE,  
    id_p INT NOT NULL,  
    FOREIGN KEY (id_p) REFERENCES Pays(id_p),  
    PRIMARY KEY (id_a)  
);
```

REQUETE SQL :

SELECT :

SELECT * FROM table WHERE condition GROUP BY expression
HAVING condition { UNION | INTERSECT | EXCEPT } ORDER BY
expression LIMIT count OFFSET start

```
SELECT prenom, nom FROM client
```

INSERT INTO :

```
INSERT INTO client (prenom, nom, ville, age) VALUES  
('Rébecca', 'Armand', 'Saint-Didier-des-Bois', 24),  
('Hilaire', 'Savary', 'Conie-Molitard', 58);
```

UPDATE :

```
1 ligne : UPDATE client  
SET rue = '49 Rue Ameline',  
    ville = 'Saint-Eustache-la-Forêt',  
    code_postal = '76210'  
WHERE id = 2
```

Toutes lignes :

```
UPDATE client  
SET pays = 'FRANCE'
```

DELETE :

```
1 ligne :  
DELETE FROM `utilisateur`  
WHERE `id` = 1
```

Plusieurs lignes :

```
DELETE FROM `utilisateur`  
WHERE `date_inscription` < '2012-04-10'
```

DROP TABLE :

Pour supprimer une table “nom_table” il suffit simplement d’utiliser la syntaxe suivante :

```
DROP TABLE nom_table
```