

Conception de base de données

1. Introduction

Objectif d'une application

L'objectif général d'une application est de permettre la réalisation de *traitements automatisés* ainsi que l'*exploitation d'informations*.

Persistance de l'information

Pour ce faire, les informations doivent être mémorisées *de manière pérenne*.

Exemples d'informations : un client, une facture, un article

Supports de stockage

Cartes et rubans perforés (historique !), magnétiques (bandes, disques), optiques (cd, dvd), semi-conducteurs (Flash, SSD), ...

Formats de stockage

Texte, tableur, XML, JSON, binaire, base de données relationnelle, ...

2. Base de données relationnelle

2.1 Définition

Ensemble d'informations *organisé selon des règles* (modèle relationnel, utilisation d'une méthode de conception) et *exploitable à l'aide du langage SQL* (Structured Query Language)

2.2 Outils de gestion de base de données

Les SGBDR du marché, MySQL, SQL Server, PostgreSQL, SQLite, Oracle, MS Access, ...

2.3 Outils d'exploitation de base de données

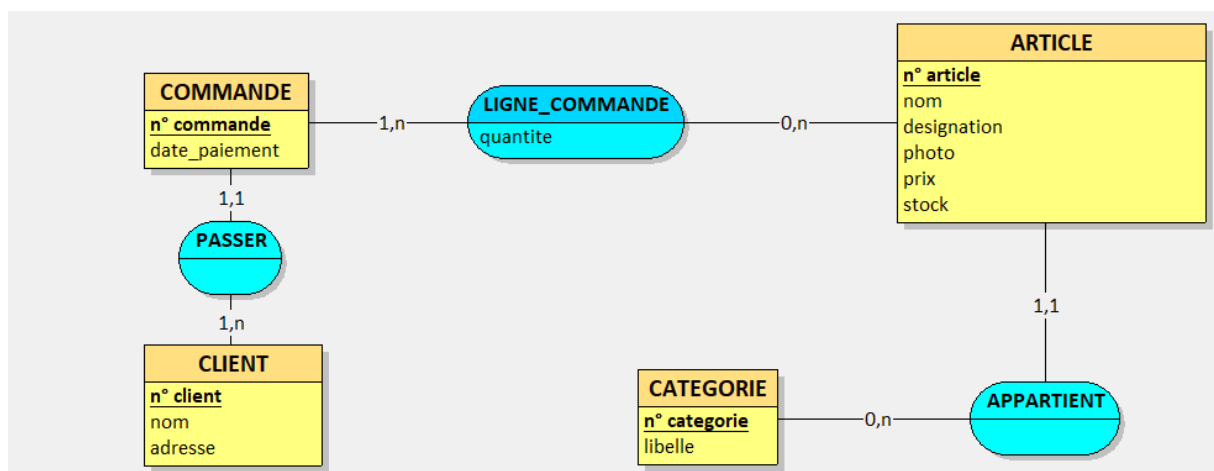
Le langage SQL

- directement dans le SGBDR, soit en ligne de commande soit en mode graphique (MS Access, MySQL Workbench, PHPMyAdmin)
- dans le code de l'application, via un driver

2.4 Exemple de MCD

Représentation graphique de l'organisation optimisée de données regroupées en entités et mettant en évidence les relations entre ces données

Exemple pour une application (site Web) proposant des articles à la vente :



Pour repérer chaque article individuellement, on a besoin d'un numéro qui constitue l'*identifiant* de l'entité (souligné).

Chaque article appartient à une catégorie d'articles. Plutôt que de faire figurer le nom de la catégorie dans l'entité article, on crée une nouvelle entité qui répertoriera les catégories et on établit une association entre cette entité et les articles. Les chiffres indiquent les cardinalités de la relation, c'est-à-dire combien de fois une occurrence d'une entité peut apparaître dans la relation

L'entité COMMANDES permet d'enregistrer les commandes passées par l'application et l'entité CLIENT indique pour chaque commande le client concerné. A nouveau, une association est établie entre les deux entités afin de ne pas placer les noms et coordonnées du client directement dans chaque commande

Le MCD s'appuie sur une méthode de conception et permet d'obtenir facilement un modèle relationnel (MLD ou modèle logique des données) qui lui-même permettra la construction de la base de données dans le SGBDR.

MLD correspondant au MCD de l'exemple ci-dessus :

CATEGORIE(**n_categorie**, libelle)

n_categorie : clé primaire

CLIENT(**n_client**, nom, adresse);

n_client : clé primaire

ARTICLE = (**n_article**, nom, désignation, photo, prix stock, **#n_categorie**);

n_article: clé primaire

#n_categorie : clé étrangère (**CATEGORIES.n_categorie**)

COMMANDE = (**n_commande**, date_paiement, **#n_client**);

n_commande : clé primaire

#n_client : clé étrangère (**CLIENT.n_client**)

LIGNE_COMMANDE = (**#n_article**, **#n_commande**, quantité);

#n_article, **#n_commande** : clé primaire

#n_article : clé étrangère (**ARTICLE.n_article**)

#n_commande : clé étrangère (**COMMANDES.n_commande**)

Chaque bloc de texte est appelé une « relation ». Ces relations permettront de créer les tables dans le SGBDR. Leur expression est très proche de la syntaxe SQL de création de tables (CREATE TABLE)

MCD (Modèle conceptuel) OU Schema Entités / Associations OU Modèle Entités / Associations (MEA)	MLD (Modèle logique) OU Modèle relationnel	MPD (Modèle physique)
Entités / Associations	Relations	Tables

Le modèle relationnel est basé sur le concept de « Dépendance fonctionnelle »

2.5 Dépendance fonctionnelle

Définition : Un attribut B dépend fonctionnellement d'un attribut (**ou groupe d'attributs**) A si à chaque valeur de A correspond une et une seule valeur de B.

Autrement dit : La connaissance de A permet à coup sûr de connaître B

Notation : $A \rightarrow B$

Exemple : Numéro Insee \rightarrow Nom : A un numéro Insee correspond un et un seul nom, il ne peut pas y avoir plusieurs personnes avec le même n° Insee

Lé réciproque n'est pas vraie : Le même nom peut correspondre à plusieurs n° Insee (homonymes)

2.5.1 Dépendance fonctionnelle élémentaire

La dépendance fonctionnelle $A \rightarrow B$ est dite élémentaire s'il n'existe pas de C appartenant à A tel que $C \rightarrow B$.

Autrement dit : Dans une dépendance fonctionnelle élémentaire, tous les éléments de A doivent être nécessaires pour déterminer B. Il ne doit pas y avoir à gauche d'éléments inutiles pour déterminer B

2.5.2 Dépendance fonctionnelle directe

La dépendance fonctionnelle $A \rightarrow B$ est dite directe s'il n'existe pas de C tel que $(A \rightarrow C)$ et $(C \rightarrow B)$

Autrement dit : Dans une dépendance fonctionnelle directe, on n'a pas besoin de passer par une information intermédiaire pour trouver la valeur de B

Exercice

	DF	Elémentaire	Directe
nom_client → adresse_client	NON		
n°_client → adresse_client	OUI	OUI	OUI
n°_commande, n°_client → quantité_commandée	NON		
n°_commande, n°_article → prix	OUI	NON	OUI
n°_commande, n°_article → quantité_commandée	OUI	OUI	OUI
n°_commande → nom_client	OUI	OUI	NON
n°_commande → nom_article	NON		

Les dépendances fonctionnelles sont à la base de la construction des relations

2.6 Relation

Définition : Une relation est un sous-ensemble du produit cartésien d'une liste d'attributs caractérisé par un nom ... *Ne pas trop chercher !!!*

CLIENT(n_client, nom, adresse);

n_client : clé primaire

Représentation dite en extension (contenu détaillé) :

n_client	nom	adresse
45	Bertrand	Nice
62	Dupont	Marseille
45	Martin	Avignon
...

2.6.1 Clé primaire (identifiant)

C'est l'attribut (ou l'ensemble d'attributs) dont tous les autres attributs de la relation sont en dépendance fonctionnelle. Une relation ne peut comporter qu'une seule clé primaire.

2.6.2 Clé étrangère

Attribut ou ensemble d'attributs qui a un rôle de clé primaire dans une autre relation. Elles permettent d'établir des liens entre les relations

2.6.3 Les formes normales des relations

1^{ère} forme normale (1NF) : La relation possède une clé, donc tous les autres attributs sont en dépendance fonctionnelle avec la clé

2^{ème} forme normale (2NF) : La relation est en 1NF et toutes les dépendances fonctionnelles sont élémentaires

3^{ème} forme normale (3NF) : La relation est en 2NF et toutes les dépendances fonctionnelles sont directes

*Pour optimiser un modèle relationnel et donc une base de données,
il faut que toutes les relations du modèle soient en 3NF*

Exercice

Ecrire les relations en 3NF à partir des attributs suivants :

n° étudiant, n° classe, n° matière, nom étudiant, nom classe, nom matière, coef

Règles de gestion :

- Un étudiant est inscrit dans une et une seule classe
- Un coefficient dépend de la matière et de la classe

Solution

MATIERE(n_matiere, nom_matiere)

n_matiere : clé primaire

CLASSE(n_classe, nom_classe)

n_classe : clé primaire

ETUDIANT(n_etudiant, nom_etudiant, #n_classe)

n_etudiant : clé primaire

#n_classe : clé étrangère

MATIERE_CLASSE(#n_matiere, #n_classe, coef)

#n_matiere, #n_classe : clé primaire

#n_matiere : clé étrangère

#n_classe : clé étrangère

La création d'un modèle relationnel devient de plus en plus difficile avec l'augmentation de la taille du dictionnaire de données. Pour parvenir à établir des modèles relationnels plus importants, on peut s'appuyer sur le modèle conceptuel de données.

3. Modèle conceptuel de données

Le MCD est une méthode qui permet d'organiser l'information simplement et de manière beaucoup plus visuelle et plus facile à construire que le modèle relationnel. Son côté visuel en fait un bon outil de communication entre les intervenants d'un projet.

3.1 Définition

Le MCD est un modèle visuel permettant de schématiser les dépendances fonctionnelles

Il respecte les règles du modèle relationnel (but : dépendances fonctionnelles élémentaires et directes, 3NF) ainsi que des règles visuelles de représentation.

3.2 Entités et associations

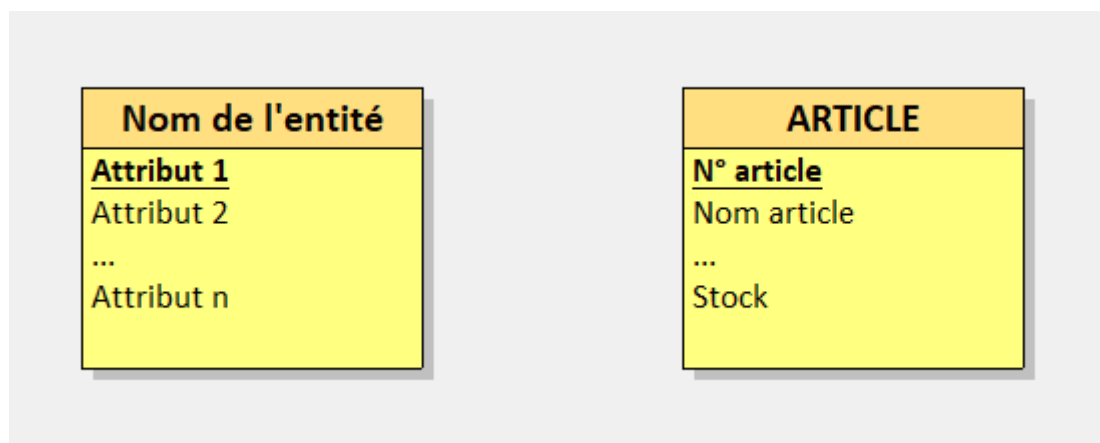
Le MCD est composé de deux types d'éléments :

- Les **entités** : relation en 3NF dont la clé primaire ne comporte qu'un seul attribut
- Les **associations** : relation en 3NF dont la clé est composée de plusieurs attributs

Tous les autres éléments du MCD sont des dérivés des entités et associations.

3.2.1 Les entités

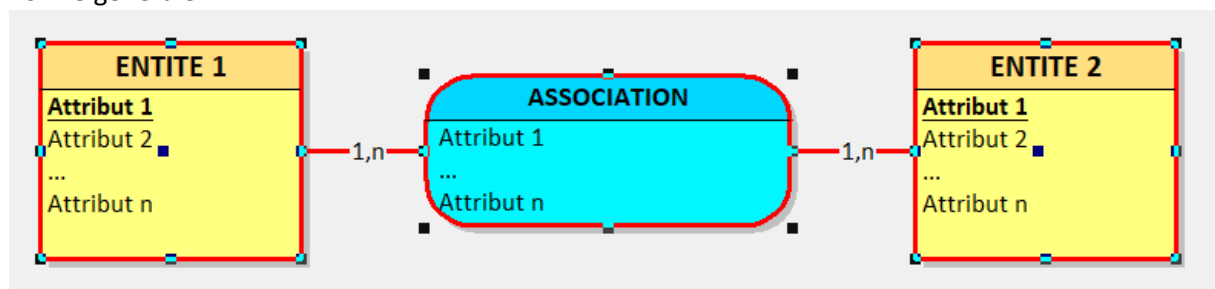
Elles sont représentées par un rectangle qui comporte dans sa partie haute le nom de l'entité et dans sa partie basse la liste des attributs. La clé primaire apparaît en premier dans la liste des attributs et elle est soulignée



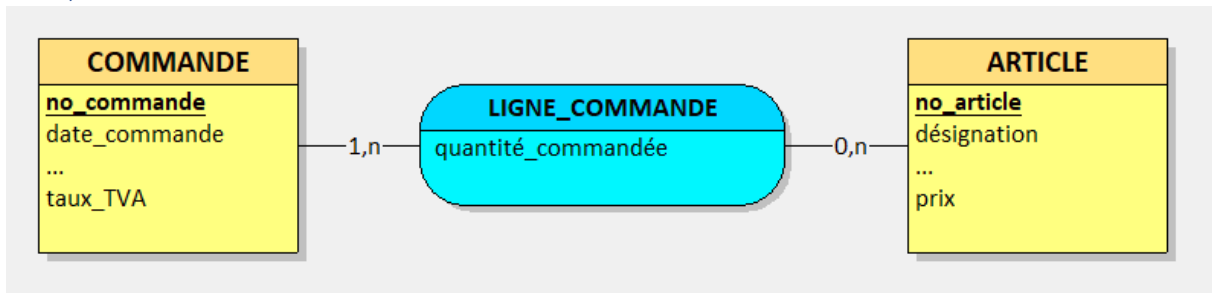
- Une seule clé primaire (soulignée)
- La clé primaire est en première position de la liste des attributs
- 0 à n attributs
- Les clés étrangères ne figurent pas dans l'entité

3.2.2 Les associations

Forme générale :



Exemple :



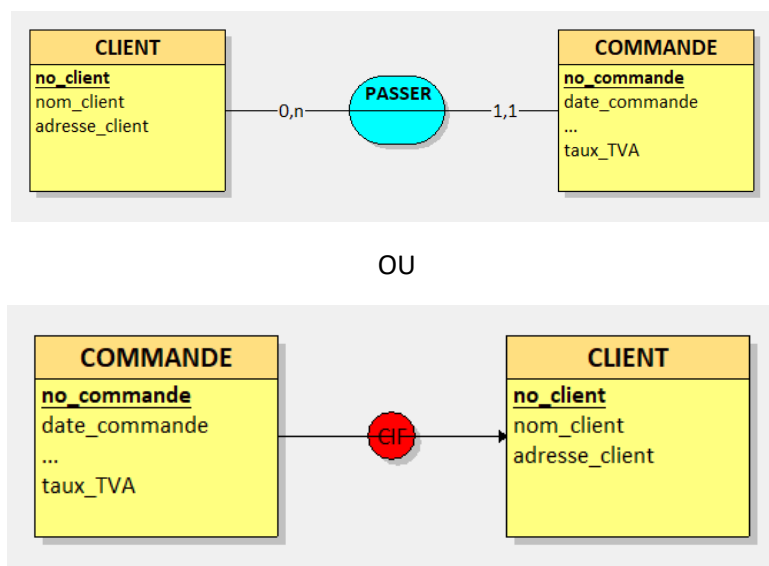
- Obligatoirement plusieurs clés primaires (représentées pas les liaisons)
- Au minimum 2 liens directs
- 0 à n attributs
- Les clés étrangères ne figurent pas dans l'association

3.2.3 Dépendances fonctionnelles entre entités

C'est une dépendance fonctionnelle entre les clés primaires des deux objets (entités ou associations).

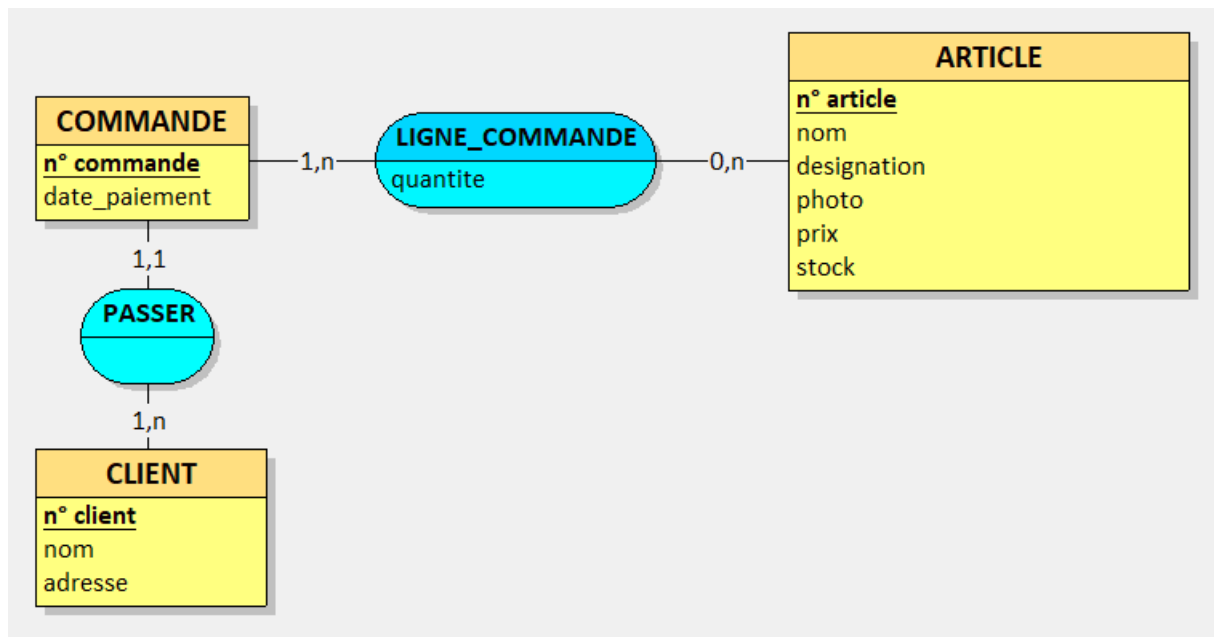
Le terme « dépendance fonctionnelle » a ici exactement le même sens qu'entre deux attributs d'une entité, c'est-à-dire que la connaissance d'une clé permet de connaître l'autre clé sans aucune ambiguïté.

On parle de Contrainte d'Intégrité Fonctionnelle (CIF)



3.2.4 Cardinalités

Permet d'exprimer le nombre minimum et maximum d'occurrences d'une entité par rapport à l'autre (dans le cadre d'une association)

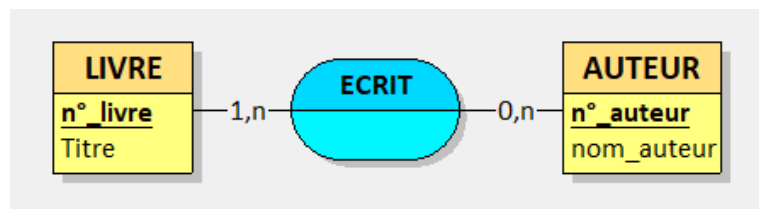


- Une commande doit être passée par un et un seul client (1,1)
- Un client peut passer de 1 à n commande (ou 0 à n, si on autorise de créer un client avant même qu'il ne passe sa première commande) → 1,n ou 0,n
- Une commande peut contenir de 1 à n articles (1,n)
- Un article peut se trouver dans 0 à n commande (0,n)

3.2.5 Associations vides

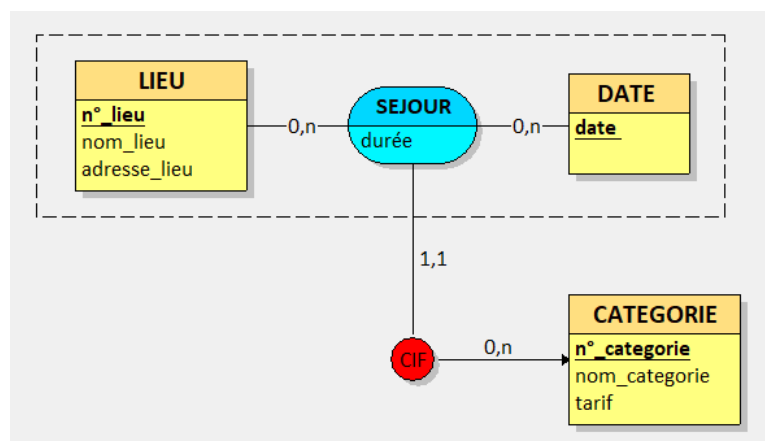
Une CIM (Contrainte d'Intégrité Multiple) est une association qui ne contient aucun attribut.

Par exemple, un livre peut avoir plusieurs auteurs :



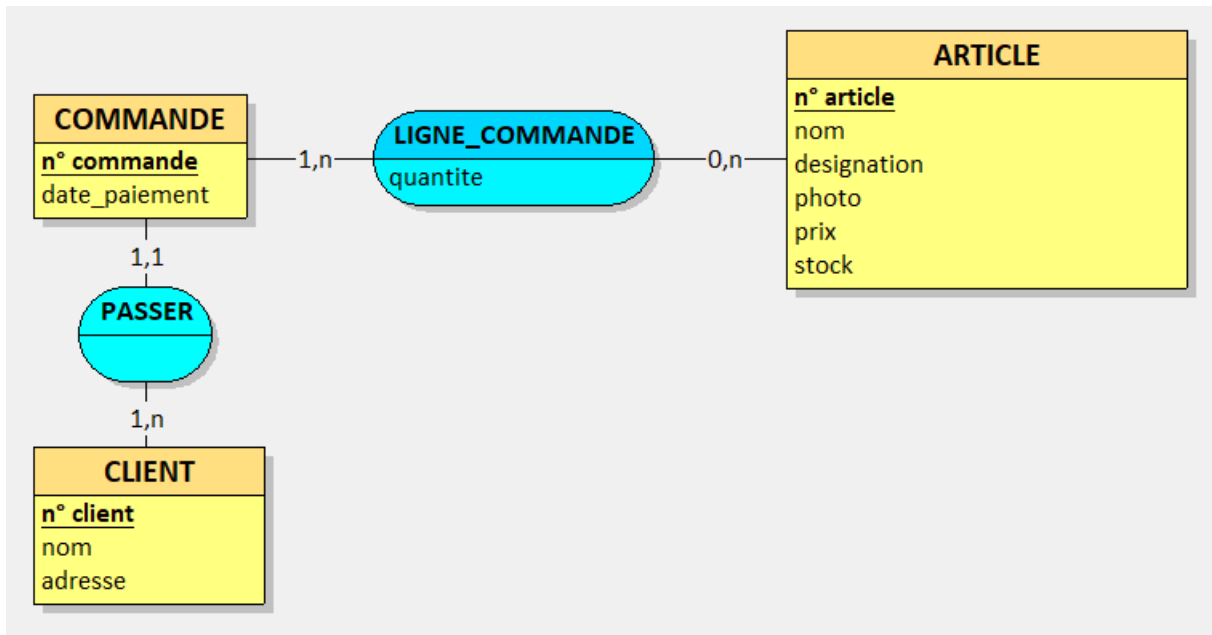
3.2.6 Les CIF sur associations

Elles traduisent une dépendance fonctionnelle entre la clé primaire d'une association et la clé primaire d'une entité.



3.2.7 Intégrité fonctionnelle

L'intégrité fonctionnelle traduit une dépendance forte entre des objets. Elle permet d'éviter l'apparition de données parasites



Dans cet exemple, on pose que

- Une commande doit être passée par un client **existant**
- La suppression d'une commande doit **entraîner la suppression** des lignes de la commande
- Un client qui est associé à une ou plusieurs commandes **ne peut pas être supprimé**
- Un article qui est associé à une ou plusieurs commandes **ne peut pas être supprimé**

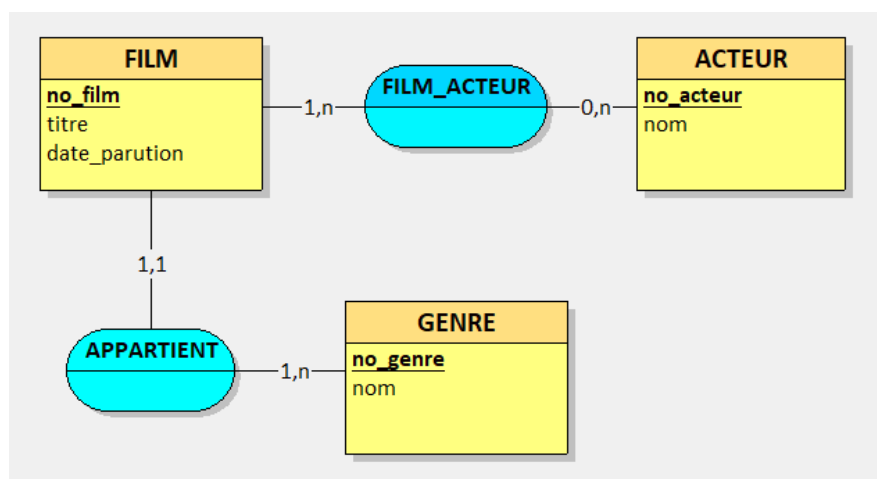
Ces comportements peuvent être dévolus au SGBD grâce à l'intégrité fonctionnelle

Exercice

Tracer le MCD correspondant aux règles suivantes :

- Un film est repéré par un numéro
- Il possède un titre, un genre, un ou plusieurs acteurs et une date de parution
- On veut classer les films par genre ou par acteur

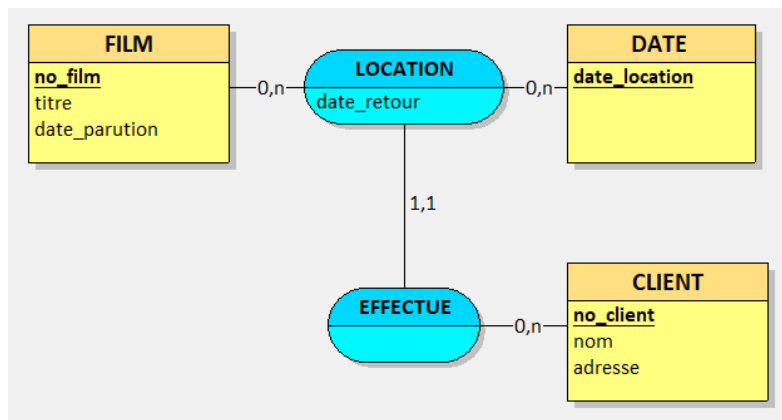
Solution



3.3 Cas d'école

3.3.1 Historique

Enregistrer les locations de films avec la date de location, la date de retour et l'identité du loueur

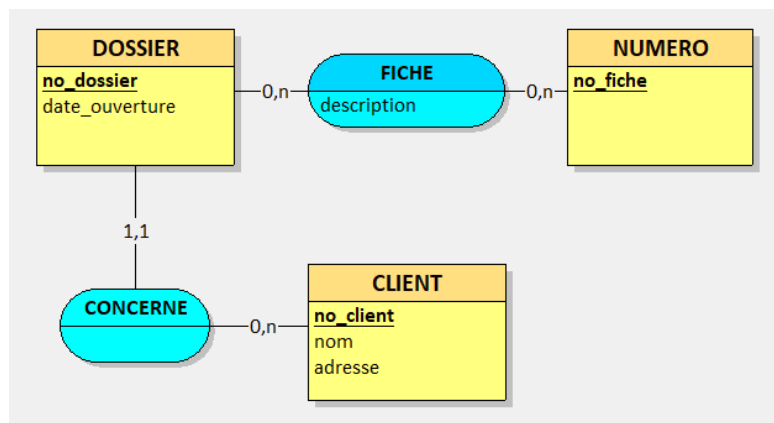


L'entité temporelle DATE n'existe que pour l'association. Elle ne donnera pas lieu à une relation dans le MLD

3.3.2 Numérotation

Certaines informations nécessitent une numérotation en fonction d'un autre

Un dossier concerne un client et est constitué de fiches. Ces dernières sont numérotées à partir de 1 dans chaque dossier

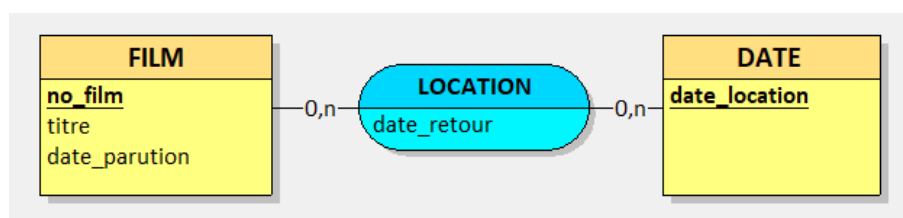


L'entité NUMERO n'existe que pour l'association. Elle ne donnera pas lieu à une relation dans le MLD

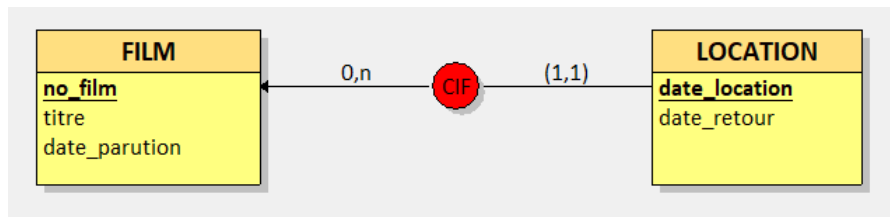
3.3.3 Lien identifiant

Les liens identifiants permettent d'éliminer les entités temporelles et de numérotation ; Il ont été créés dans la version Merise/2.

Exemple sans lien identifiant



Exemple avec lien identifiant



La version Merise/2 permet d'éviter l'entité temporelle : On crée une entité et on indique les cardinalités 1,1 entre parenthèses pour indiquer que l'identifiant date_location est relatif à l'identifiant no_film de l'entité FILM. Ainsi, la clé primaire de LOCATION sera composée de no_film et date_location

Il ne s'agit que d'une simplification de représentation. Les relations générées par le modèle sont exactement les mêmes

Exercice

Voici un ensemble de relations :

DOSSIER(id_dossier, date_ouverture, #id_client)

id_dossier : clé primaire

#id_client : clé étrangère

CLIENT(id_client, nom, adresse, tel)

id_client : clé primaire

FICHE_DOSSIER(id_dossier, num_fiche, date_fiche, description)

id_dossier, num_fiche : clé primaire

id_dossier : clé étrangère

Questions :

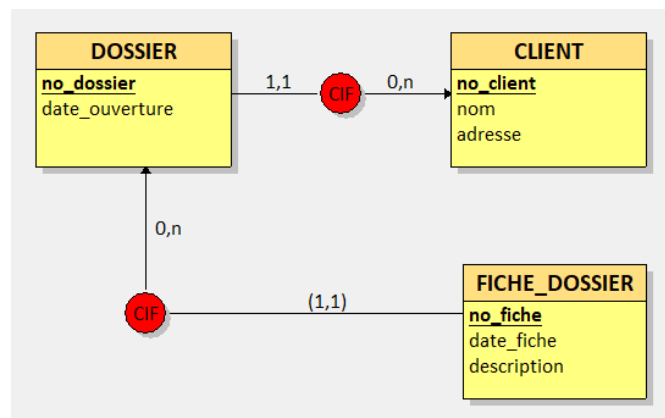
Pourquoi num_fiche n'est pas aussi clé étrangère dans la relation FICHE_DOSSIER ?

→ num_fiche n'est pas clé étrangère car il devrait pour cela être clé primaire d'une autre relation

Comment appelle-t-on le lien qui unit FICHE_DOSSIER à DOSSIER ?

→ Le lien FICHE_DOSSIER – DOSSIER est un lien identifiant

Dessiner le Schéma conceptuel de données correspondant à ces relations



Dire si les affirmations suivantes sont vraies :

Une fiche n'appartient qu'à un seul dossier

VRAI : La cardinalité (1,1) indique bien qu'une fiche n'est rattachée qu'à un seul dossier

Un dossier peut comporter plusieurs fiches

VRAI : Du côté DOSSIER, la cardinalité est 0,n

Une fiche peut concerner plusieurs clients

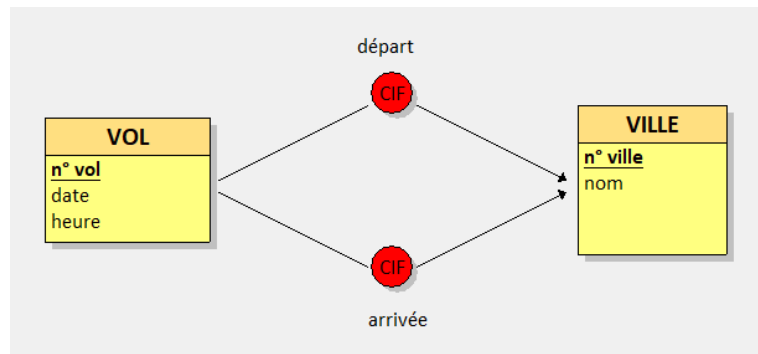
FAUX : Une fiche n'appartient qu'à un seul dossier, qui lui-même n'appartient qu'à un seul client

Un client peut avoir plusieurs fiches

VRAI : Un client peut avoir plusieurs dossiers qui peuvent avoir plusieurs fiches

3.3.4 Double CIF

Dans certaines situations, il est nécessaire d'obtenir 2 informations de la même entité :



Au niveau relationnel, il faut faire attention aux noms :

VILLE(no_ville, nom)

no_ville : clé primaire

VOL(no_vol, date, heure, #ville_depart, #ville_arrivee)

no_vol : clé primaire

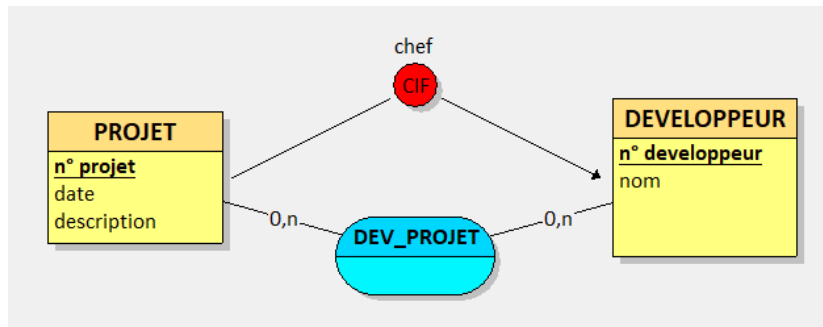
#ville_depart : clé étrangère

#ville_arrivee : clé étrangère

3.3.5 CIF + association

Il faut parfois faire ressortir un cas particulier d'une liste

Un projet est géré par des développeurs dont l'un est chef de projet



Modèle relationnel :

DEVELOPPEUR(no_developpeur, nom)

no_developpeur : clé primaire

PROJET(no_projet, date, description, #no_chef)

no_projet: clé primaire

#no_chef : clé étrangère

DEV_PROJET(#no_projet, #no_dev)

#no_projet, #no_dev : clé primaire

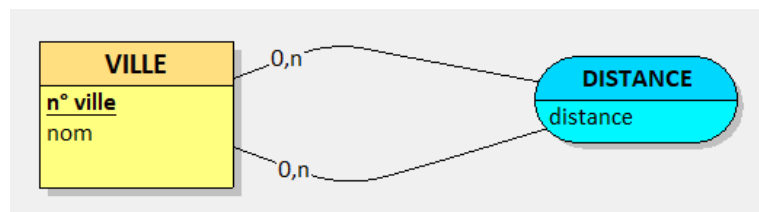
#no_projet : clé étrangère

#no_dev : clé étrangère

3.3.6 Association réflexive

Permet de lier une information avec elle-même

Exemple : enregistrer les distances entre des villes



Modèle relationnel (Attention aux noms d'attributs) :

VILLE(no_ville, nom)

no_ville : clé primaire

DISTANCE(#no_ville_depart, #no_ville_arrivee, distance)

#no_ville_depart, #no_ville_arrivee : clé primaire

#no_ville_depart : clé étrangère

#no_ville_arrivee : clé étrangère

Exercice

Construire le MCD à partir du MLD suivant :

AFFAIRE(no_affaire, date_debut, date_cloture, lieu, description)

no_affaire : clé primaire

PERSONNE(no_personne, nom, adresse, informations)

no_personne : clé primaire

PERSONNE_AFFAIRE(#no_personne, #no_affaire)

#no_personne, #no_affaire : clé primaire

#no_personne : clé étrangère

#no_affaire : clé étrangère

RAISON(no_raison, libelle)

no_raison : clé primaire

LIEN_AFFAIRE(#no_affaire1, #no_affaire2, #raison)

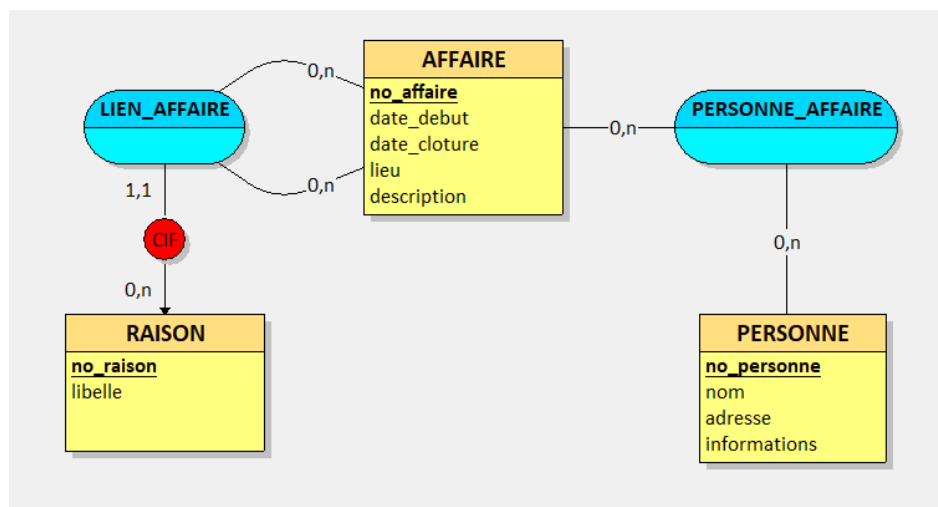
#no_affaire1, #no_affaire2 : clé primaire

#no_affaire1 : clé étrangère

#no_affaire2 : clé étrangère

#no_raison : clé étrangère

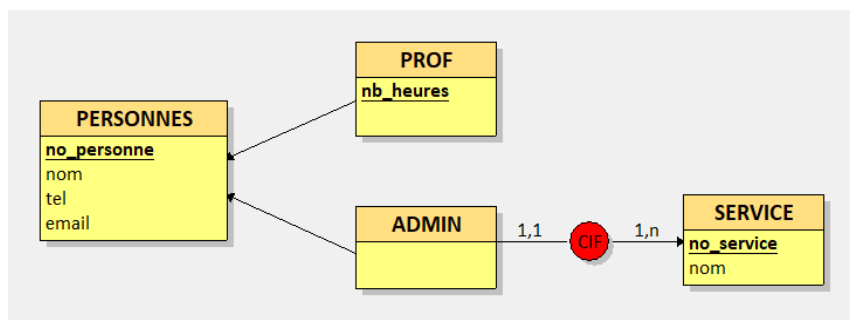
Solution



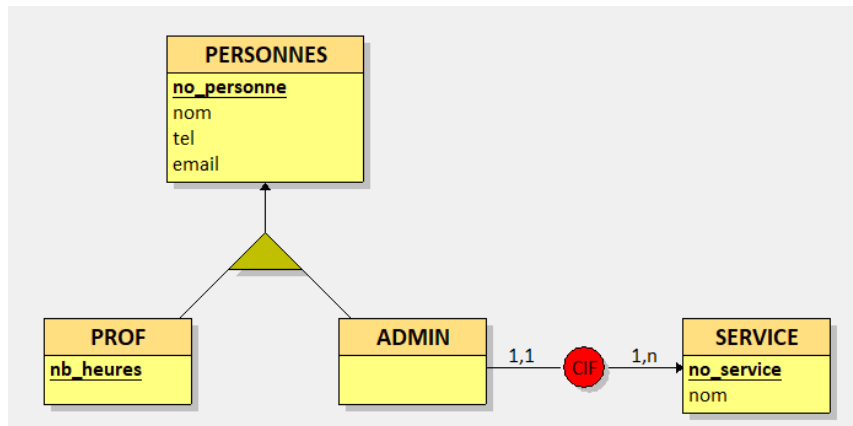
3.3.7 Héritage

Permet de distinguer les attributs communs des attributs spécifiques

Exemple : gérer des personnes avec un nombre d'heures pour les enseignants et un service pour les administratifs



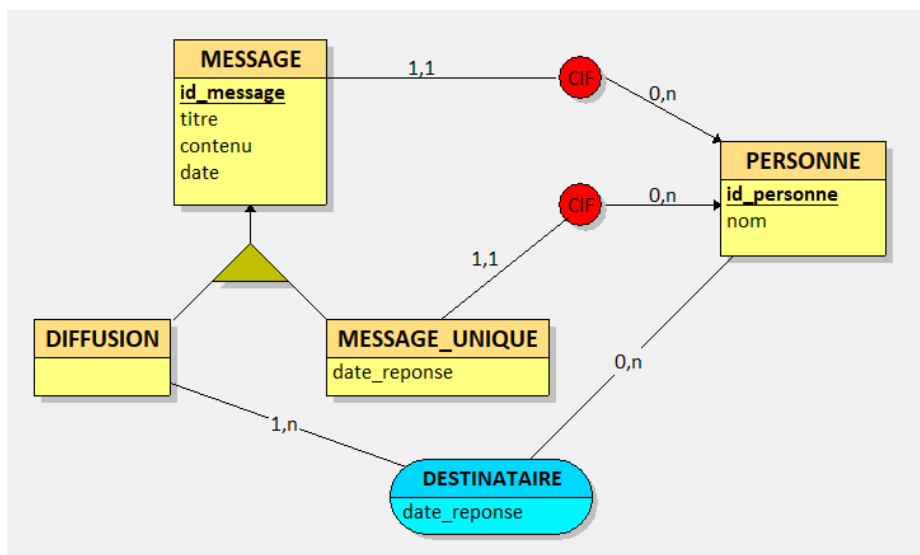
OU



L'héritage va permettre d'éviter d'avoir des champ systématiquement vides (nb_heures pour admin et service pour prof)

Exercice

Ecrire les relations correspondantes :



MESSAGE(id_message, titre, contenu, date, #id_expediteur)

id_message : clé primaire

#id_expediteur : clé étrangère

MESSAGE_UNIQUE(id_message_unique, date_reponse, #id_destinataire)

id_message_unique : clé primaire

id_message_unique : clé étrangère

#id_destinataire : clé étrangère

DIFFUSION(id_diffusion)

id_diffusion : clé primaire

id_diffusion : clé étrangère

PERSONNE(id_personne, nom)

id_personne : clé primaire

DESTINATAIRE(id_diffusion, #id_destinataire, date_reponse)

id_diffusion : clé primaire

id_diffusion : clé étrangère
id_destinataire : clé étrangère