

Secteur Tertiaire Informatique
Filière « Etude et développement »

Séquence « Développer des pages Web »

Initiation aux Bases de Données

Apprentissage

Mise en situation

Evaluation



SOMMAIRE

1/ : Avertissement.....	4
2/ : Définition de l'information et des systèmes d'informations.....	4
3/ : Les données, les traitements et les informations	5
4/ : La représentation informatique des données	5
5/ : Les systèmes de gestion de bases de données	5
5.1/ : Définitions	5
5.2/ : Les tables	6
5.2.1/ : Définition	6
5.2.2/ : Les champs d'une table	7
5.2.3/ : Clé primaire	7
5.2.4/ : Index	8
6/ : La méthode d'analyse MERISE	9
6.1/ : Modèle conceptuel des données (MCD)	11
6.1.1/ : Définition	11
6.1.2/ : Entité	11
6.1.3/ : Propriété	12
6.1.4/ : relation	12
6.1.5/ : contraintes d'intégrité fonctionnelle (CIF)	14
6.2/ : Modèle logique de données (MLD)	14
6.2.1/ : Définition	14
6.2.2/ : Passage du MCD au MLD	14
6.2.2.1/ : Entité	14
6.2.2.2/ : Relation avec cardinalités (x,n) et (x,1) ou x = 0 ou 1	15
6.2.2.3/ : Relation avec cardinalités (x,n) et (x,n) ou x = 0 ou 1	15
6.3/ : Modèle physique de données (MPD)	16
6.3.1/ : Définition	16
6.3.2/ : Passage du MLD au MPD	16

1/ : Avertissement

Ce cours permet de présenter les concepts élémentaires des bases de données. Il donne un aperçu des différentes notions nécessaires pour comprendre la réalisation d'applications informatiques mettant en œuvre des bases de données. On utilisera la méthode d'analyse **MERISE** pour étudier les concepts de réalisation d'une base de données.

Il a été réalisé dans un esprit de vulgarisation pour se mettre à la portée des lecteurs les plus novices, tout en conservant les mêmes principes du secteur professionnel (vocabulaire, méthodologie, etc...).

Il faut indiquer l'importance des bases de données dans le domaine informatique. Il est sans nul doute, en terme de nombre des utilisateurs et du marché des développements, en première place.

Le lecteur de ce cours sera un jour acteur dans la réalisation d'un projet de base de données. Il lui sera alors très utile pour lui permettre d'appréhender plus facilement sa participation.

2/ : Définition de l'information et des systèmes d'informations

Une information est un élément qui permet de compléter notre connaissance sur un objet, un événement, une personne, etc...

Exemple: Le nom d'une personne est une information concernant cette personne.
Le titre d'un film est une information concernant ce film.

Un système d'information est constitué par l'ensemble des informations relatives à un domaine bien défini.

Exemple: Toutes les informations relatives à la gestion d'une librairie constituent le système d'information de cette librairie. Ce système peut couvrir le simple stockage des livres, mais également la gestion des commandes, des ventes et même des clients.

ATTENTION: Un système d'informations ne doit pas nécessairement être informatisé. Bien que la plupart des systèmes actuels se basent sur la technologie de l'informatique, il existe encore des systèmes d'information où l'information est stockée, manipulée et communiquée à l'aide de moyens "traditionnels" tels que armoires, classeurs, calculatrices, fiches sur papier etc...

Un système d'information existe indépendamment des techniques informatiques.

Le système d'information ne doit pas être confondu avec le système informatique qui est constitué par les éléments suivants:

- * Les ordinateurs
- * Les programmes
- * Les structures de données (Fichiers, Bases de données)

Dans ce cours nous allons découvrir une démarche d'informatisation, qui nous permet de modéliser un système d'information et de le représenter à l'aide d'un **système informatique**. Le but de cette démarche est de concevoir des systèmes **stables** et **optimisés** en termes de **performance**, de **fiabilité** et de **convivialité**.

3/ : Les données, les traitements et les informations

les termes "informations" et "données" sont souvent utilisés comme synonymes, il existe une différence notable entre eux.

exemple: Dans un club vidéo, un client demande à l'employé si le DVD de "SEVEN" est disponible en location. L'employé consulte la base de données à l'aide de son ordinateur et confirme au client que ce DVD est disponible. L'employé a donc donné au client l'*information* que le DVD est disponible. Afin de pouvoir donner cette information, l'employé a dû consulter les *données* qui représentent les DVD du vidéo club. Le fait de consulter le stock de DVD constitue un *traitement* sur les DVD en stock.

Nous pouvons généraliser:

Un système d'information contient les données et les traitements nécessaires pour assimiler et stocker les informations entrantes et produire les informations sortantes.

Dans les systèmes d'information nous retrouvons généralement les **traitements** suivants

- * Consultation des données
- * Calculs sur les données
- * Ajout de données
- * Suppression de données
- * Modification de données
- * Recherche de données

Exemple: Le propriétaire d'une vidéothèque reçoit une livraison de nouveaux DVD. Pour chaque DVD, il **lit** le titre, le producteur et la durée, puis il **sauvegarde** ces informations dans la **base de données** de la vidéothèque. Il a donc utilisé un **traitement** d'ajout de données afin de **transformer** les informations entrantes (titre, producteur, durée) en données.

4/ : La représentation informatique des données

Les données d'un système d'information peuvent être stockées et manipulées à l'aide d'un outil informatique spécialisé dans ce domaine. Actuellement les *Systèmes de Gestion de Bases de Données* (**SGBD**, exemples: ORACLE, **MySQL**, SQL Server, DB2, etc...) constituent le type de logiciel le mieux adapté pour informatiser la plupart des systèmes d'information. Sachant que les **tables** représentent la base de stockage d'une base de données, on peut représenter n'importe quel **système d'information** par un **ensemble de tables** dont chacune contient un certain nombre de **champs de données**.

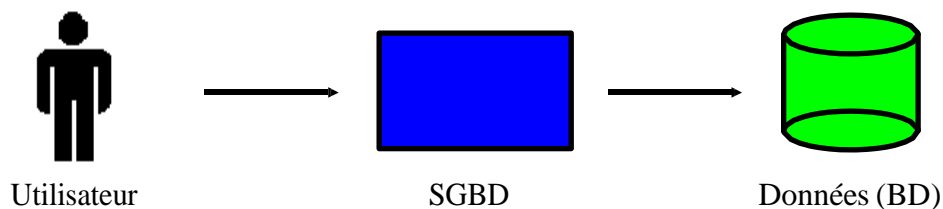
5/ : Les systèmes de gestion de bases de données

5.1 / : Définitions

Une base de données (BD) est un ensemble bien structuré de données relatives à un sujet global. Ces données peuvent être de nature et d'origine différentes.

Exemple: Un établissement scolaire peut avoir une BD qui contient les informations nécessaires à la gestion des élèves (notes, matières, etc...).

Un système de gestion de bases de données (SGBD) est un logiciel qui nous permet de créer, de modifier et d'exploiter des bases de données. Ce système constitue donc notre interface pour accéder aux données.

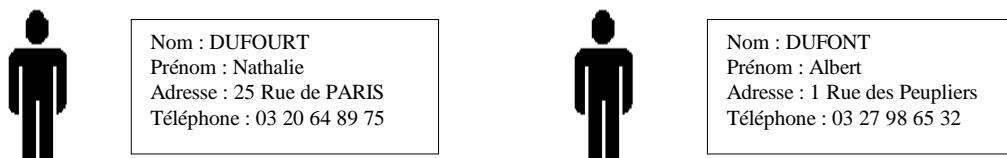


5.2 / : Les tables

5.2.1/ : Définition

Une **table** est une **collection de données** relatives à un **domaine bien défini**, par exemple les élèves d'une école ou les DVD d'une vidéothèque. Elle contient des **enregistrements** dont chacun est composé par les mêmes champs de données. Le nombre d'enregistrements dans une table peut être très grand (million voire milliard).

Voici, à titre d'exemple, quelques élèves d'une école:



Propriétés des tables:

* Les champs de données définissent les types d'information que l'on veut stocker dans la table (par exemple: des informations concernant les élèves d'une école).

* Chaque enregistrement représente une occurrence de ce qu'on veut stocker (par exemple un élève).

- * Chaque table possède un nom unique (par exemple: tbl_Eleves).
- * Chaque enregistrement correspond à une ligne de la table.
- * Chaque champ correspond à une colonne de la table.
- * Chaque champ peut représenter des données de natures différentes (Nom, prénom, Date de naissance, etc ...).
- * Chaque champ peut représenter des données de types différents (Texte, Nombre, Date, etc...).

Le nom d'une table doit être unique à l'intérieur d'une BD.

5.2.2/ : Les champs d'une table

La quantité des données stockées dans une table est en général représentée à l'aide de plusieurs informations pour chaque enregistrement. Une table avec les données concernant les élèves d'une école contient par exemple pour chaque enregistrement (= chaque élève) les informations suivantes:

- * Nom
- * Prénom
- * Adresse
- * Téléphone

Voici comme exemple une table qui représente les élèves dans une BD:

NOM	PRENOM	ADRESSE	TELEPHONE
DUFOURT	Nathalie	25 rue de PARIS	03 20 64 89 75
DUPONT	Albert	1 rue des peupliers	03 27 98 65 32
.	.	.	.
LECLERC	Jean	56 Blv. PASTEUR	03 21 66 55 44

Afin de pouvoir représenter des données de types différents, les SGBD offrent des types de données standards pour les champs de données. Voici quelques types de données connus par la plupart des SGBD:

<i>Type de données</i>	<i>Description</i>
Date/Heure	Date et heure
Valeur booléenne	Seulement les 2 valeurs Vrai/Faux
Texte	Chaînes de caractères
Numérique	Nombres décimaux
Entier	Nombres entiers
Mémo	Documents (texte long)

Lors de la création d'une **table**, nous devons indiquer au **SGBD**, pour **chaque champ**:

1. Le nom du champ, qui doit être **unique** dans la table
2. Le **type de données** du **champ**

5.2.3/ : Clé primaire

Dans la plupart des cas, on désire pouvoir identifier de manière unique chaque enregistrement de la table. Ceci n'est pas possible pour la table des élèves. Il se peut très bien que dans une école 2 élèves aient le même nom. Il faut donc un moyen qui permette d'adresser sans ambiguïté chaque enregistrement de la table. C'est le rôle de la **clé primaire**.

La **clé primaire**, constituée d'un ou de plusieurs champs, nous permet d'identifier de manière unique chaque enregistrement d'une table. Chaque table ne doit pas nécessairement disposer d'une clé primaire. Toutefois, pour définir des liens entre plusieurs tables la clé primaire est indispensable, donc: **CHAQUE TABLE DOIT AVOIR UNE CLE PRIMAIRE !!!**

Exemples: Le numéro d'élèves pour les élèves d'une école.

Clé primaire



NUM	NOM	PRENOM	ADRESSE	TELEPHONE
1	DUFOURT	Nathalie	25 rue de PARIS	03 20 64 89 75
2	DUPONT	Albert	1 rue des peupliers	03 27 98 65 32
.
100	LECLERC	Jean	56 Blv. PASTEUR	03 21 66 55 44

5.2.4/ : Index

Une des utilisations fréquentes des tables consiste en la **recherche** et le **tri des enregistrements**.

Lorsque les tables contiennent un grand nombre d'enregistrements, la recherche de certains enregistrements ainsi que les tris des enregistrements deviennent parfois très lents en terme d'exécution. Les **index** sont des **structures** qui **accélèrent** les **tris** et **recherches** dans les **tables**.

Propriétés importantes des index:

- * Un **index** est toujours lié à **un** ou **plusieurs champs** d'une **table**.
- * Un **index** peut seulement contenir des champs ayant le type de données **Texte**, **Numérique** ou **Date/Heure**.
- * Un **index** est **automatiquement** mis à jour par le **SGBD** lors d'un **ajout**, d'une **modification** ou d'une **suppression** d'enregistrements dans la **table**. Ceci est transparent pour l'utilisateur de la BD.

Il existe deux types d'index:

1. Index avec doublons (index secondaire unique ou non)
2. Index sans doublons (index primaire)

Voici quelques règles qui aident à déterminer les champs d'une table qui ont besoin d'être indexés:

- * La puissance des index joue uniquement pour des tables qui contiennent beaucoup d'enregistrements
- * Un champ sur lequel on ne fait que rarement ou pas du tout une recherche ou un tri n'a pas besoin d'index.
 - * Les champs référencés fréquemment dans les recherches et tris doivent par contre être indexés.
 - * Pour les index multi-champs, il faut veiller à ce que la combinaison des champs dans l'index corresponde exactement au critère de recherche.
 - * Un index sans doublons sur un champ empêche l'utilisateur d'entrer la même valeur dans ce champ, dans deux enregistrements différents.
 - * Définir trop d'index sur une table ralentit en général les opérations d'ajout, de modification et de suppression, parce que le SGBD doit mettre à jour la table et les index.

6/ : La méthode d'analyse MERISE

MERISE est une méthode d'analyse qui permet d'établir une représentation claire et précise du système d'information. C'est la **méthode d'analyse** la plus utilisée en **France**.

Nous ne développerons qu'une partie de la méthode MERISE permettant l'analyse des systèmes simples, la conception des modèles **conceptuels (MCD)**, **logiques (MLD)**, **physiques (MPD)** afin d'aboutir à l'implémentation de bases de données cohérentes.

MERISE est définie en 3 parties:

1 : Le niveau conceptuel

Il décrit l'ensemble des données du système d'information, sans tenir compte de l'implémentation informatique de ces données.

Il indique la signification des données, représenté par un formalisme qui s'appelle: **Modèle conceptuel des données (MCD)**

2 : Le niveau logique

Il se base sur le modèle conceptuel des données, en prenant en compte l'implémentation du système d'information par un SGBD.

Ce niveau introduit la notion de tables logiques, et établit la première étape vers la constitution des tables sur un SGBD.

Il est représenté par un formalisme qui s'appelle: **Modèle logique des données (MLD)**

3 : Le niveau physique

Il fait référence au modèle logique des données.

C'est à partir de ce niveau que l'on pourra créer la base de données.

Il est représenté par un formalisme qui s'appelle: **Modèle physique des données (MPD)**

Ces **3 étapes** sont nécessaires pour traduire un système d'information en une base de données:



Nous allons étudier une partie de la méthode à l'aide d'un exemple.
Le système d'information contient les propriétés figurant sur les factures:

N° Facture :			Date :	
Nom Client :				
Adresse :				
.....				
Code Produit	Désignation	Quantité	Prix Unitaire	Montant
.....
.....
.....
Total :			

└─ Ligne de facture

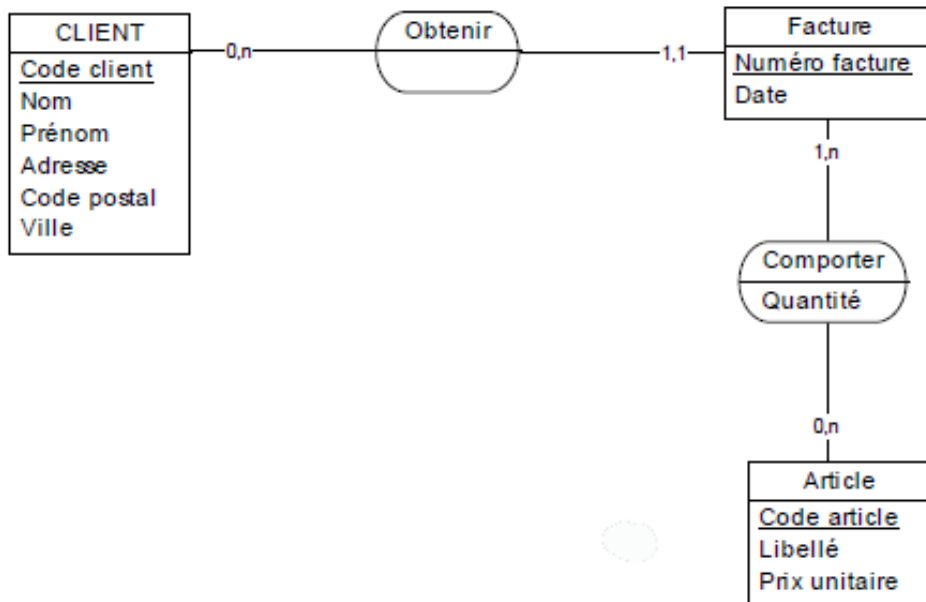
Règles de gestion

- 1: Un client peut obtenir une ou plusieurs factures ou aucune facture.
- 2: Une facture ne peut être affectée qu'à un seul client.
- 3 : Une facture peut concerner un ou plusieurs produits.
- 4 : Un produit peut être affecté à une ou plusieurs factures ou aucune facture.

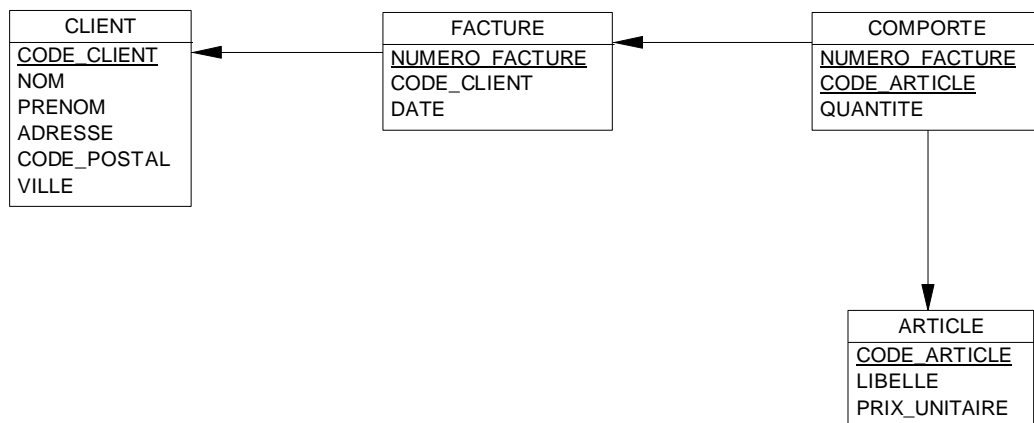
Traitements

- 1: Création, Modification, Suppression d'un client.
- 2: Création, Modification, Suppression d'un produit.
- 3: Etablissement de la facture.
- 4: Mise à jour facture (Solde de la facture ou non).
- 5: Edition liste des factures:
 - * Par client
 - * Par état (Soldées ou non)
 - * Par ordre chronologique.
- 6: Liste des factures par ordre chronologique (Soldées, non soldées ou les deux).

Modèle conceptuel de données



Modèle logique de données



6.1 / : Modèle conceptuel des données (MCD)

6.1.1/ : Définition

Le modèle conceptuel de données (**MCD**) fait référence à tous les objets du système d'information et aux relations entre les objets. Le formalisme utilisé est connu sous le nom de "**Schéma Entité-Relation**". Il se base autour de 3 concepts principaux, les **entités**, les **relations** et les **propriétés**.

6.1.2/ : Entité

Une **entité** permet de modéliser un ensemble d'objets concrets ou abstraits de caractéristiques comparables.

Une entité est caractérisée par son nom et ses propriétés.

Représentation graphique

Nom de l'entité
Propriété 1
Propriété 2
Propriété 3
Propriété 4
Propriété 5

Exemple de l'entité CLIENT

CLIENT
Nom
Prénom
Adresse
Code postal
Ville

Quelques exemples de clients

CLIENT 1	CLIENT 2
DUPONT Bernard 2 Rue de D 59450 SIN LE NO	DURANT Marie 59 Biv SAINT MICHEL 59500 DOUAI

Chacun de ces clients représente une **occurrence** de l'entité «CLIENT».

6.1.3/ : Propriété

Une **propriété** est une **donnée élémentaire** d'une **entité**.

exemple: Nom du client : DUPONT, DURANT
Ville: DOUAI, SIN-LE-NOBLE

Une propriété n'est rattachée qu'à une et une seule entité.

Pour faire référence à chaque occurrence d'une entité, l'entité doit être munie d'un **identifiant**. L'identifiant est composé d'une ou de plusieurs propriétés de l'entité. A chaque valeur de l'identifiant doit correspondre une et une seule occurrence de l'entité.

Exemple: Il est nécessaire d'ajouter un code client dans l'entité «CLIENT» car les propriétés

existantes ne sont pas nécessairement uniques (ex Nom du client). Le code client permet donc l'unicité du client.

La ou les propriétés qui constituent l'identifiant d'une entité sont soulignées.

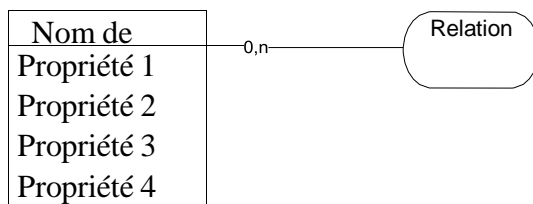
CLIENT
<u>Code client</u>
Nom
Prénom
Adresse
Code postal
Ville

6.1.4/ : relation

Une **relation** décrit un lien entre **deux** ou **plusieurs entités**. Chaque relation possède un nom, qui est généralement constitué par un **verbe à l'infinitif**.

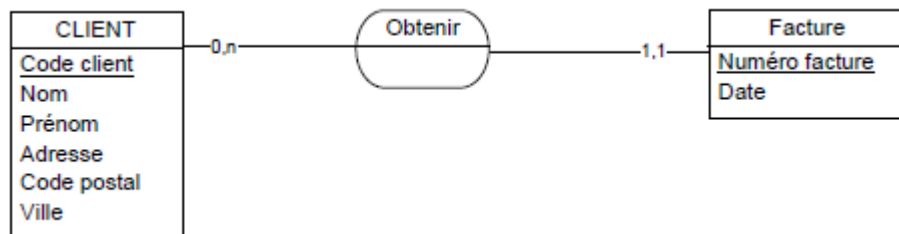
Chaque relation a obligatoirement un identifiant, qui est composé par les identifiants des entités auxquelles elle est liée. L'occurrence d'une relation est représentée par les occurrences des entités liées à la relation.

Exemple de relation



Une relation est liée à chacune de ses entités par un lien sur lequel on indique les **cardinalités**. Les **cardinalités** représentent la **participation de l'entité** concernée à la relation. Le **premier nombre** indique la **cardinalité minimale**, le **deuxième** la **cardinalité maximale**.

Exemple de relation entre un client et une commande



Entité «CLIENT» et relation «OBTENIR»

- * Cardinalité minimale = 0: chaque client obtient aucune facture.
- * Cardinalité maximale = n: chaque client peut obtenir plusieurs (n) factures.

Entité «FACTURE» et relation «OBTENIR»

- * Cardinalité minimale = 1: chaque facture est obtenue par au moins un client.
- * Cardinalité maximale = 1: chaque facture est obtenue au maximum par un seul client.

Définition

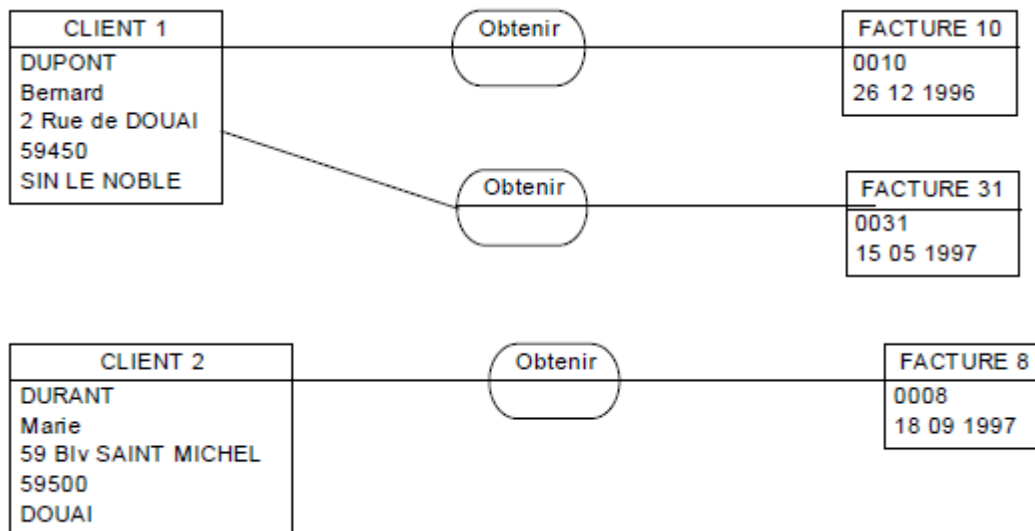
La cardinalité minimale exprime le nombre de fois minimum où une occurrence d'une entité participe à une relation. Cette cardinalité est généralement 0 ou 1.

- * **Cardinalité minimale = 0:** Certaines occurrences de l'entité ne participent pas à la relation
- * **Cardinalité minimale = 1:** Chaque occurrence de l'entité participe au moins une fois à la relation

La cardinalité maximale exprime le nombre de fois maximum où une occurrence d'une entité participe à une relation. Cette cardinalité vaut souvent 1 ou n, avec n indiquant une valeur >1 mais pas connue.

- * **Cardinalité maximale = 1:** Chaque occurrence de l'entité participe au plus une seule fois à la relation.
- * **Cardinalité maximale = n:** Chaque occurrence de l'entité peut participer plusieurs fois à la relation.

Exemple de quelques occurrences de la relation «OBTENIR»



6.1.5/ : contraintes d'intégrité fonctionnelle (CIF)

Une **CIF** indique que l'une des entités est totalement déterminée par la connaissance de l'autre.

Exemple: connaissant une facture bien précise, on connaît avec certitude le client correspondant.

6.2 / : Modèle logique de données (MLD)

6.2.1/ : Définition

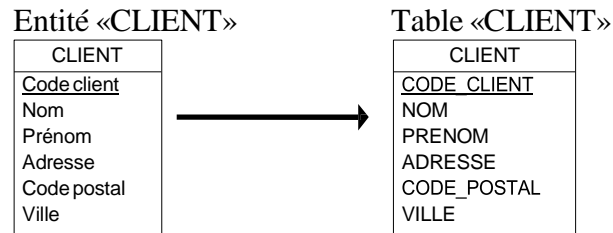
Le modèle logique de données (MLD) se base sur un **modèle conceptuel des données**. Il est composé de **tables logiques** reliées entre elles par des flèches.

6.2.2/ : Passage du MCD au MLD

6.2.2.1/ : Entité

L'**entité** est transformée en **table**. Les propriétés de l'entité deviennent les attributs de la table. L'identifiant de l'entité devient la clé primaire de la table.

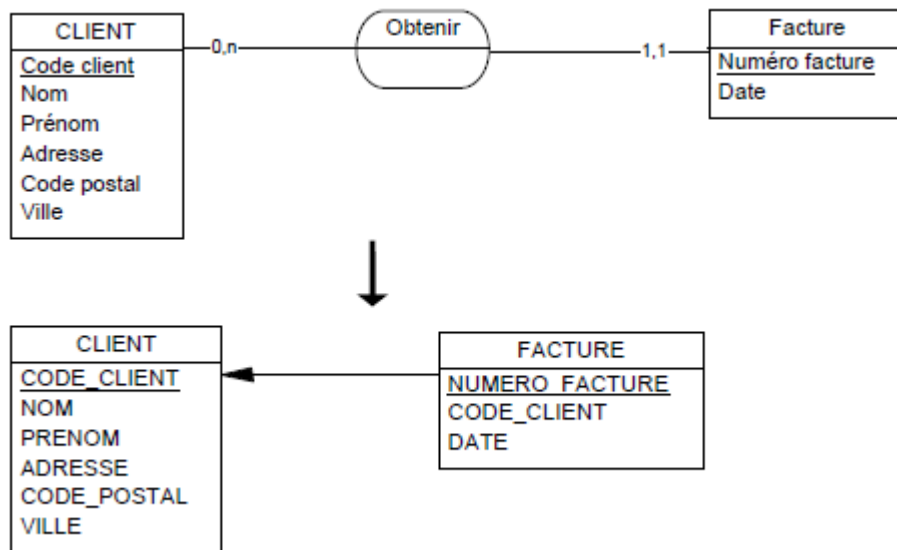
Exemple: Entité «CLIENT»



6.2.2.2/ : Relation avec cardinalités (x,n) et (x,1) ou x = 0 ou 1

On copie la clé primaire de la table dont la cardinalité est **(x,n)** dans la table dont la cardinalité est **(x,1)**. L'attribut est appelé clé étrangère. Les deux tables sont liées par une flèche qui pointe de la table à clé étrangère vers la table qui contient la clé primaire correspondante.

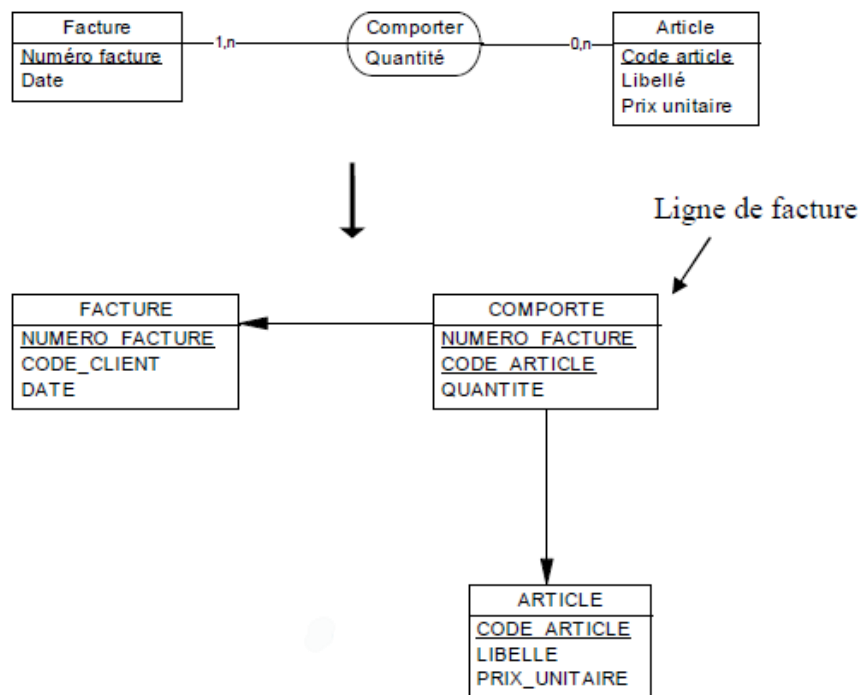
Exemple sur la relation «CLIENT», «FACTURE»



6.2.2.3/ : Relation avec cardinalités (x,n) et (x,n) ou x = 0 ou 1

Il y a **création** d'une **table supplémentaire** ayant comme clé primaire une clé composée des identifiants des 2 entités. Lorsque la relation contient des propriétés, celles-ci deviennent attributs de la table supplémentaire.

Exemple sur la relation «FACTURE», «ARTICLE»



6.3 / : Modèle physique de données (MPD)

6.3.1/ : Définition

Le modèle physique de données (**MPD**) se base sur un **modèle logique de données**. Il est composé de **tables physiques**. Chaque table logique du **MLD** correspondra à une table physique du **MPD**.

6.3.2/ : Passage du MLD au MPD

Les attributs de la table logique deviennent les champs de la table physique. Pour chaque champ on indiquera son nom, son type et sa longueur.

Exemple: table «CLIENT»

CHAMPS	TYPE	LONGUEUR	CLE
CODE_CLIENT	N	5	*
NOM	A	25	
PRENOM	A	25	
ADRESSE	A	45	
CODE_POSTAL	N	5	
VILLE	A	25	

- N : Numérique
- A : Alphanumérique