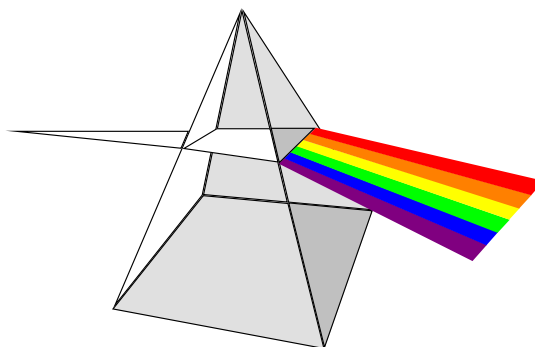


MICROSOFT

# ACCESS

VERSION 2000/2003



## COURS

# PERFECTIONNEMENT

# SOMMAIRE

<i>COURS 1 - Créer une relation entre deux tables</i>	<i>20</i>
<i>COURS 2 – Créer une requête utilisant des données de plusieurs tables</i>	<i>22</i>
<i>COURS 3 – Transformer une requête de sélection en requête paramétrée – Supprimer une requête</i>	<i>23</i>
<i>COURS 4 – Créer une clé primaire composée de plusieurs champs</i>	<i>25</i>
<i>COURS 5 – Modifier les valeurs dans une table à l'aide d'une requête de mise à jour</i>	<i>26</i>
<i>COURS 6 Créer une liste déroulante dans un formulaire</i>	<i>28</i>
<i>COURS 7 - Créer un bouton bascule dans un formulaire</i>	<i>31</i>
<i>COURS 8 – Réaliser des calculs dans une requête – Mettre un champ en monétaire – Ajouter une table</i>	<i>32</i>
<i>COURS 9 – Insérer des sous-formulaires dans un formulaire</i>	<i>34</i>
<i>COURS 10 – Réaliser des calculs dans un formulaire</i>	<i>37</i>
<i>COURS 11 – Réaliser un état multi tables</i>	<i>38</i>
<i>COURS 12 – Réaliser un publipostage dans Word à partir d'une table Access – Exporter des données vers Word</i>	<i>40</i>

## Quelques notions théoriques simplifiées...

Lorsque l'on cherche à stocker avec un système de gestion de base de données (SGBD) les informations utilisées dans une Entreprise, il est important de concevoir d'abord "sur le papier" les différentes tables à réaliser surtout si elles comportent des relations entre-elles. Il faut faire ce que l'on appelle en jargon informatique un **Modèle Conceptuelle de Données** (MCD) qui est un schéma permettant de représenter les tables et leurs relations. Le MCD s'inscrit dans une méthode d'analyse des problèmes informatiques appelée **MERISE**. Les pages 3 à 19 qui suivent expliquent comment réaliser un MCD viable afin de ne pas avoir de mauvaises surprises lors de la mise en place de la base de données avec un SGBD.

**Toutefois cette partie théorique demande un investissement certain.**

**Si vous ne vous en sentez pas le courage vous pouvez passer directement à la page 20.**

### **I) PROBLEMATIQUE DU MODELE CONCEPTUEL DE DONNEES (MCD)**

#### **1) Définition**

Le MCD est un schéma représentant l'ensemble des données mémorisables, sans tenir compte des aspects techniques et économiques du stockage et de l'accès, sans se référer aux conditions d'utilisation par tel ou tel traitement.

#### **2) Objectifs**

Données et traitements apparaissent intimement liés (surtout du point de vue de l'utilisateur). Dans une entreprise, on fait référence à des objets concrets ou abstraits (par exemple pour une compagnie d'assurance : l'assuré, le contrat) et à des associations entre ces objets (le contrat comporte des garanties). L'objectif du modèle conceptuel de données est d'identifier, de décrire par des informations et de modéliser ces objets et associations.

La première étape dans la réalisation d'un MCD est la constitution d'une liste de toutes les informations utilisées.

Cette liste d'information est le résultat d'un recueil d'informations circulant dans l'Entreprise. Elle se présente sans aucune structure de regroupement à priori.

Pour constituer cette liste, le concepteur doit répertorier les informations :

- présentes sur les documents fournis par l'utilisateur,
- mises en évidence lors des entretiens.

Pour chaque information recueillie, le concepteur doit répondre aux questions suivantes avant de l'ajouter à la liste déjà établie :

- La nouvelle information n'a-t-elle pas déjà été répertoriée ? Il est, par exemple, probable que l'information *n°police* apparaisse dans de nombreux documents.
- La nouvelle information n'a-t-elle pas déjà été répertoriée sous une appellation différente ? Le concepteur est en présence d'un synonyme, par exemple *référence dossier* et *n°police*.
- Une appellation identique n'existe-t-elle déjà pas mais associée à une signification différente ? Le concepteur est en présence d'un homonyme, par exemple *date de livraison* (demandée) et *date de livraison* (effective). Il faut impérativement lever l'ambiguïté en modifiant les appellations des informations.

## II) FORMALISATION DU MCD

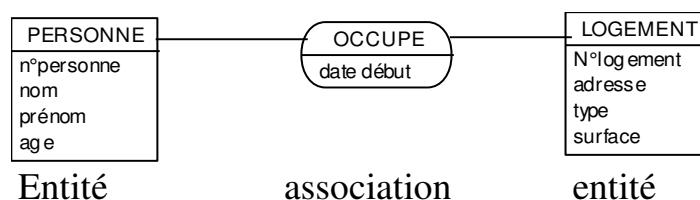
### 1) Concepts de base

Le formalisme utilisé dans MERISE est désigné par individu-relation ou entité/association.

Ce formalisme comporte trois concepts de base :

- l'individu ou entité,
- la relation ou association,
- la propriété ou champ.

#### 11) Représentation graphique du formalisme



<u>Champ :</u>	n°personne nom prénom âge	date début	N°logement adresse type surface
----------------	------------------------------------	------------	--

## 12) Le champ

C'est la modélisation d'une information élémentaire présente dans l'Entreprise. Elle peut prendre des valeurs ; par exemple :

Nom de client : Dupont, Durand...

Date de naissance : 12/02/58, 14/08/53...

Pourquoi distinguer information et champ ? Le champ est une manière de modéliser une information, mais toutes les informations ne seront pas systématiquement traduites par un champ.

Le champ doit vérifier un certain nombre de règles que l'information première n'a pas besoin de respecter.

Le champ est un élément descriptif de l'entité ou de la relation. Un champ est obligatoirement rattaché à une entité ou une relation.

Un champ est unique dans un modèle conceptuel.

Le champ peut être composé ; c'est à dire que sa valeur est obtenue à partir des valeurs d'autres informations à travers une règle de construction, par exemple :

Numéro INSEE : sexe+année+mois+départ+commune+chrono

Le champ composé suit les mêmes règles que tout champ et a sa signification intrinsèque.

## 13) L'entité

L'entité type permet de modéliser un ensemble d'objets de même nature, concrets ou abstrait, perçus d'intérêt.

Quelques règles en régissent la modélisation.

### Règle d'identification

On doit pouvoir faire référence distinctement à chaque occurrence de l'entité. Pour cela l'entité doit être dotée d'un identifiant. Cet identifiant est un champ tel que, à une valeur de l'identifiant, correspond une seule occurrence de l'entité .

Cette correspondance entre l'occurrence de l'entité et la valeur de l'identifiant doit être vérifiée au présent mais également confirmée dans le futur.

Le choix d'un identifiant est un problème délicat. On peut opter pour :

- un champ "naturel" ; par exemple le nom d'un pays pour l'entité pays,
- un champ "artificiel", inventée par le concepteur (numéro, références, numéro d'ordre...),
- un identifiant relatif, par exemple n°d'allocataire+n°d'ordre.

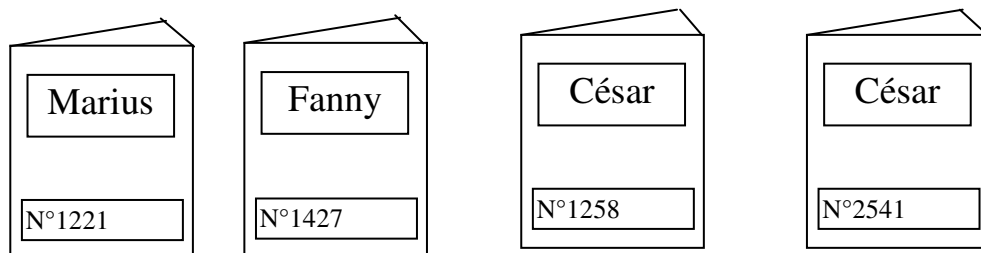
Un identifiant d'une entité doit être :

- *univalué* : à une occurrence correspond une seule valeur pour un identifiant donné ;
- *discriminant* : à une valeur correspond une seule occurrence de l'entité ;
- *stable* : la valeur de l'identifiant d'une occurrence donnée doit être conservée jusqu'à la destruction de l'occurrence ;
- *minimal* : s'agissant d'un identifiant composite, la suppression d'un de ces composants lui ferait perdre son caractère discriminant.

### Règle de distinguabilité

Les occurrences d'une entité doivent être distinguables. Cette distinguabilité induit la compréhension de l'entité et se traduit par le choix de l'identifiant.

Prenons par exemple des livres dans une bibliothèque :



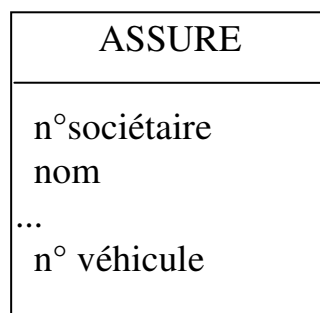
Un lecteur percevra trois occurrences distinctes (les deux César étant le même pour lui), tandis que le bibliothécaire en percevra quatre.

### Règle de vérification

Chaque champ rattaché à l'entité doit impérativement suivre la règle de vérification (ou de non répétitivité) :

A toute occurrence de l'entité, il ne peut y avoir, au plus qu'une valeur du champ.

Si la réponse à cette règle est négative, le champ concerné ne peut appartenir à l'entité. Par exemple dans le cadre d'une compagnie d'assurance :



Si un assuré peut avoir plusieurs véhicules assurés, alors la propriété *n° véhicule* ne peut appartenir à l'entité. Le concepteur devra faire appel à une autre modélisation.

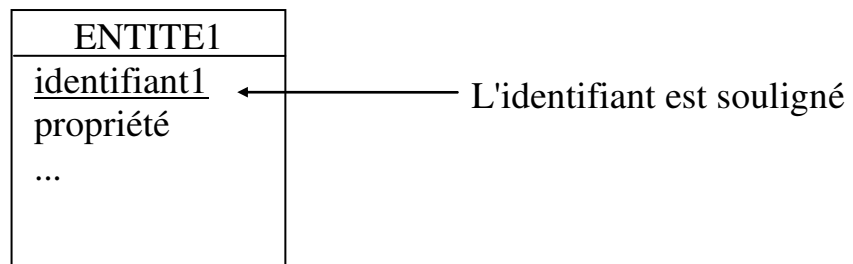
### Règle d'homogénéité

Il est souhaitable que les champs rattachés à une entité aient un sens pour toutes les occurrences de celui-ci. Cette règle invite le concepteur à s'assurer que, dans sa compréhension de l'entité, il n'englobe pas plusieurs populations dont certaines ont des caractères propres exprimés dans la liste des champs.

Le concepteur peut :

- soit confirmer sa modélisation initiale et tolérer que, pour certaines occurrences, des champs ne soient pas pertinentes ;
- soit remodeler sa perception en plusieurs entités.

Représentation schématique d'une entité :

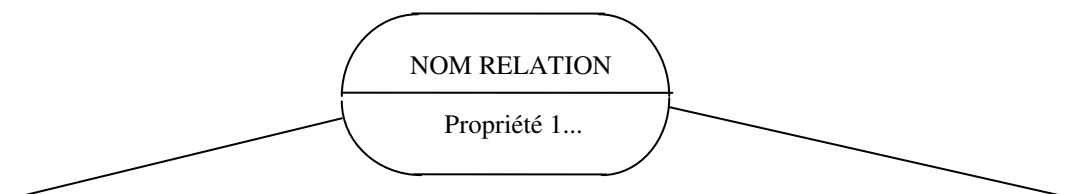


### 14) Relation :

Association entre entités. A l'inverse de l'entité, elle n'a pas d'existence propre, car elle n'a de signification qu'en fonction d'un certain nombre d'objets dont elle assure le lien.

Elle peut posséder des propriétés.

Elle est représentée de la manière suivante :

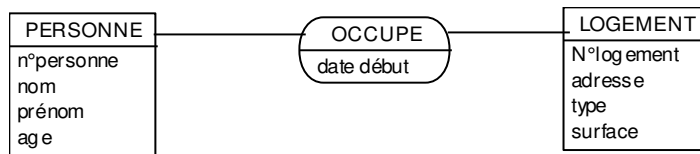


### Dimension et relation

On appelle *dimension* le nombre d'entités composant la relation et *collection* la liste de ces entités. Le nom de la relation est généralement constitué par un verbe.

### Identifiant d'une relation type

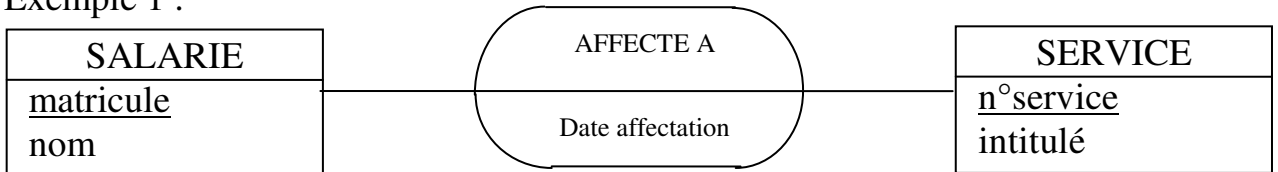
Une relation type n'a pas d'identifiant propre. L'occurrence d'une relation type est déterminée par les occurrences des entités de sa collection.



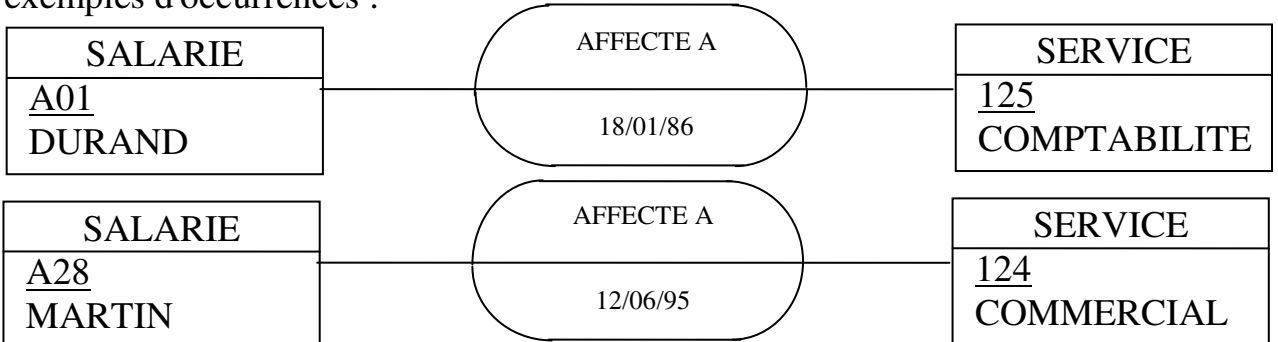
Dans l'exemple ci-dessus, elle est identifiée par la conjonction d'une valeur de n°personne et d'une valeur de n°logement.

### Exemples

Exemple 1 :

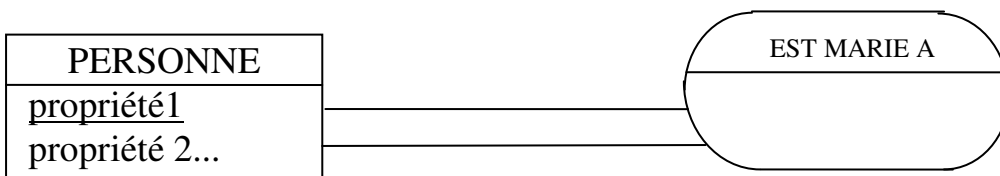


exemples d'occurrences :



Il s'agit d'une relation de dimension 2 (binaire) définie sur la collection SALARIE et SERVICE.

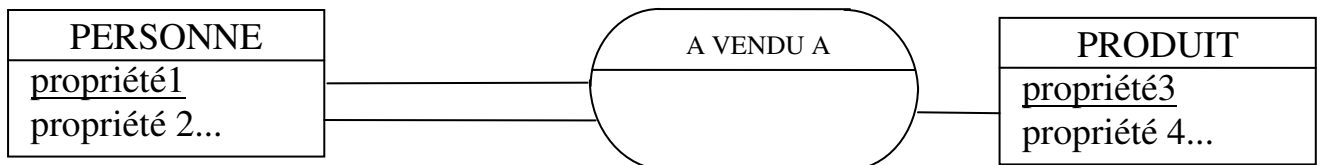
Exemple 2 :



Il faut deux occurrences de l'entité PERSONNE pour une occurrence de la relation EST MARIE A . La relation est binaire.



Exemple 3 :



Par exemple JEAN A VENDU UNE CHEMISE A PAUL (deux occurrences de PERSONNE et une de PRODUIT).

La relation est ternaire.

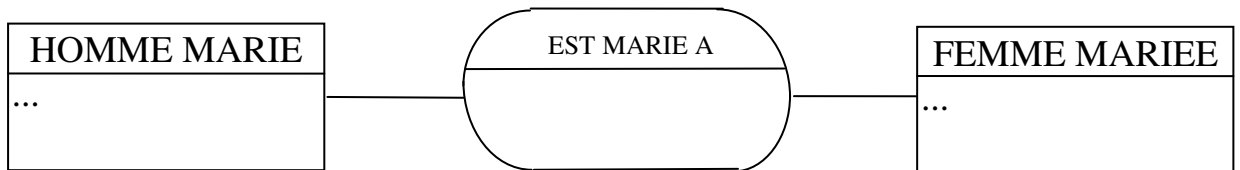
### Fonctionnalité

On définit la fonctionnalité d'une relation par rapport à deux entités X et Y. On distingue les relations :

#### **1 A 1 (1-1)**

A toute occurrence de X ne correspond qu'une seule occurrence de Y et réciproquement.

Exemple :



Un homme n'est marié qu'à une femme et une femme n'est mariée qu'à un homme.

#### **1 A PLUSIEURS (1-n)**

A toute occurrence de X correspond une ou plusieurs occurrences de Y et à toute occurrence de Y une seule de X.

Exemple :

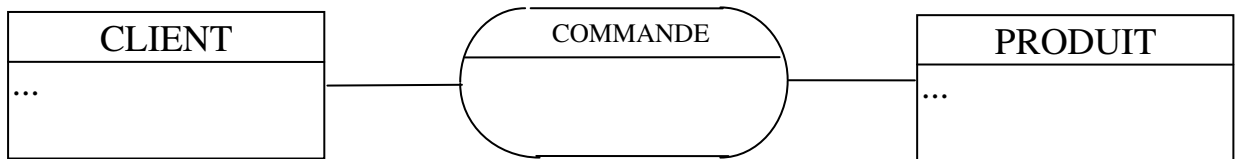


Un auteur a écrit un ou plusieurs livres mais un livre a été écrit par un seul auteur.

## PLUSIEURS A PLUSIEURS (m-n)

A toute occurrence de X correspond une ou plusieurs occurrences de Y et réciproquement

Exemple :



Un client peut commander plusieurs produits et chaque produit peut être commandé par plusieurs clients.

## TOTALITE/PARTIALITE

Une relation mettant en jeu les entités X et Y est dite :

- *Totale* si aucune occurrence de X et aucune occurrence de Y ne peuvent exister sans participer à une occurrence de la relation.
- *Partielle* si certaines occurrences de X ou certaines occurrences de Y peuvent n'être impliquées dans aucune occurrence de la relation.

### Cardinalités

La notion de cardinalité minimum/maximum permet d'exprimer la fonctionnalité et la totalité/partialité d'une relation.

#### **CARDINALITE MINIMUM**

La cardinalité minimum d'une relation est le nombre minimum de fois où chaque occurrence d'une entité participe à la relation.

La cardinalité minimum 0 correspond à une relation partielle.

La cardinalité minimum 1 signifie qu'une occurrence d'entité ne peut exister sans participer à une occurrence de la relation.

La cardinalité minimum n implique que toute occurrence de l'entité participe obligatoirement à n occurrences de la relation.

Les cardinalités minimum non nulles correspondent à des relations totales.

#### **CARDINALITE MAXIMUM**

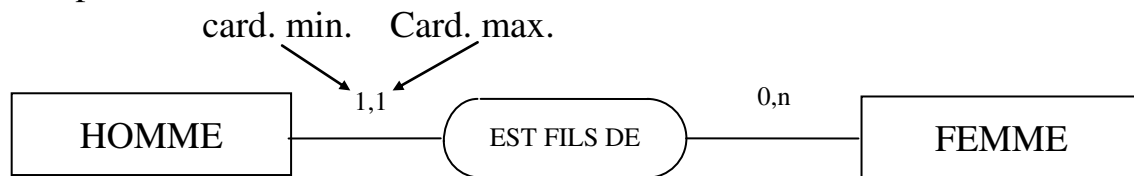
La cardinalité maximum d'une relation est le nombre maximum de fois où chaque occurrence d'une entité peut participer à la relation.

La cardinalité maximum 1 signifie que toute occurrence de l'entité ne peut participer qu'à une occurrence de la relation, au plus.

La cardinalité maximum n signifie qu'une occurrence de l'entité peut être impliquée dans un maximum de n occurrences de la relation.

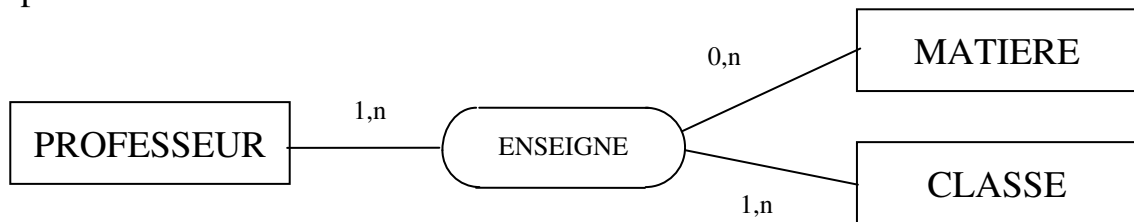
Représentation graphique :

Exemple 1 :



Un homme est fils d'au moins et d'au plus une femme, c'est-à-dire d'une femme et d'une seule. Une femme peut ne pas avoir d'enfant (0 enfant) ou au contraire en avoir plusieurs (n enfants).

Exemple 2 :



Un professeur fait au moins un enseignement. Il peut en faire plusieurs. Une matière peut ne pas être enseignée. Si elle l'est, elle peut l'être plusieurs fois. Une classe a au moins un enseignement et peut en avoir plusieurs.

## 2) Règles de gestion

Les règles de gestion du MCD précisent les contraintes qui doivent être respectées par le modèle.

Exemple :

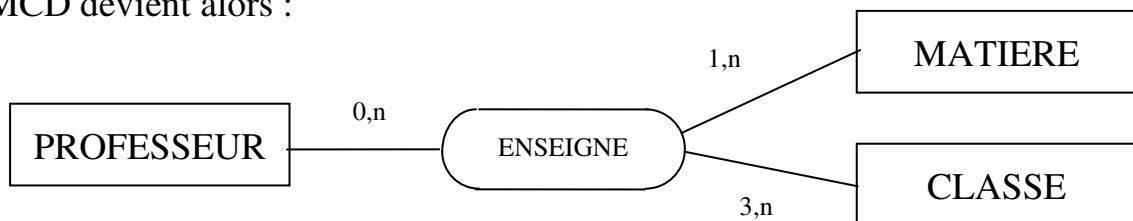
Dans le MCD d'une école les règles de gestion peuvent être les suivantes :

Règle 1 : Tout professeur enseigne en principe au moins une matière, mais certains d'entre eux peuvent être dispensés d'enseignement en raison de travaux de recherche .

Règle 2 : Toute matière est enseignée dans au moins une classe.

Règle 3 : toute classe a au moins trois enseignants.

Le MCD devient alors :



Les règles de gestion expriment les **contraintes d'intégrités** du modèle. Ces contraintes d'intégrités représentent les lois de l'univers réel modélisé dans le système d'information.

On distingue :

### **les contraintes statiques**

Elles peuvent porter :

- sur une propriété (forme, liste de valeurs possibles, fourchette de valeurs admissibles...).

- sur diverses propriétés d'une même relation ou entité. Exemple :

COMMANDE(N°COMMANDE, DATE\_CDE, DATE\_LIVRAISON)

On doit avoir DATE\_CDE<=DATE\_LIVRAISON

- sur des propriétés d'occurrence distinctes d'une relation ou entité. Exemple :

LIGNE\_ECRITURE(N°ECRITURE, LIBELLE, MONTANT, SENS)

La somme des montants des lignes de sens "débit" doit être égale à celle des lignes de sens "crédit".

- sur des propriétés d'entités/rerelations différentes. Exemple :

La somme des CA des produits doit être égale à celle des CA des clients.

- sur les cardinalités.

- sur les dépendances fonctionnelles (cf plus loin).

### **les contraintes dynamiques**

Elles expriment les règles d'évolution et portent directement sur le passage du système d'information d'un état vers un autre.

Exemple : le salaire d'un employé ne doit pas diminuer.

## **3) Dépendances fonctionnelles**

### **Dépendances fonctionnelles entre propriétés**

**Dépendance fonctionnelles** : on dit que deux propriétés a et b sont reliées par une dépendance fonctionnelle

a ————— df —————> b

si la connaissance de la valeur de a détermine une et une seule valeur de b.

Exemple : code\_client ————— df —————> nom\_client

La connaissance du code\_client détermine une et une seule valeur de nom\_client. Autrement dit, si on connaît le code du client, on doit pouvoir connaître son nom et celui-ci sera unique.

La réciproque est fautive. Le nom du client ne permet pas de déterminer son code, car plusieurs clients peuvent avoir le même nom.

La dépendance fonctionnelle peut porter sur la concaténation de plusieurs propriétés.

Exemple :

$N^{\circ}BON\_DE\_CDE+REF\_PRODUIT \xrightarrow{df} QUANT\_CDEE$

La référence seule ne suffit pas pour déterminer la quantité commandée, une même référence pouvant figurer sur plusieurs bons de commande.

Le numéro de bon de commande ne suffit pas non plus puisqu'un même bon peut concerner plusieurs références.

En revanche, si on admet qu'une référence ne peut figurer qu'une seule fois sur un bon, la connaissance du numéro de bon et de la référence du produit détermine la quantité commandée.

**Dépendance fonctionnelle élémentaire** : on dit qu'il y a dépendance fonctionnelle élémentaire entre les propriétés a et b et on note :

$a \xrightarrow{df} b$

si a  $\xrightarrow{df}$  b et si aucune partie de a ne détermine b

Exemple :

$CODE\_CLIENT+NOM\_CLIENT \xrightarrow{df} ADRESSE\_CLIENT$

n'est pas élémentaire puisque la connaissance de CODE\_CLIENT suffit à déterminer l'adresse.

$CODE\_CLIENT \xrightarrow{df} ADRESSE\_CLIENT$

est élémentaire, on peut écrire :

$CODE\_CLIENT \xrightarrow{df} ADRESSE\_CLIENT$

De même

$N^{\circ}BON\_DE\_CDE+REF\_PRODUIT \xrightarrow{df} QUANT\_CDEE$

**Dépendance fonctionnelle élémentaire directe** : on dit que la propriété b dépend fonctionnellement de a par une dépendance fonctionnelle élémentaire directe si cette dépendance est élémentaire  $a \xrightarrow{df} b$  et s'il n'existe pas de propriété c telle que  $a \xrightarrow{df} c$  et  $c \xrightarrow{df} b$ .

Autrement dit, on élimine toute transitivité.

Exemple :

$N^{\circ}PROFESSEUR \xrightarrow{df} CODE\_MATIERE$

$CODE\_MATIERE \xrightarrow{df} NOM\_MATIERE$

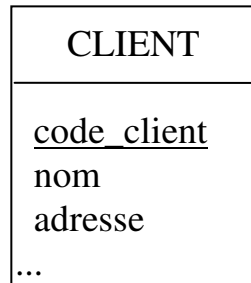
$N^{\circ}PROFESSEUR \xrightarrow{df} NOM\_MATIERE$

Les deux premières dépendances fonctionnelles sont directes, mais la troisième ne l'est pas en raison de la transitivité :

$N^{\circ}PROFESSEUR \xrightarrow{df} CODE\_MATIERE \xrightarrow{df} NOM\_MATIERE$

**Clé (d'identification) d'une entité** : une clé d'une entité est une propriété (ou une concaténation de propriétés) de cette entité telle que toutes les autres propriétés de l'entité dépendent d'elle fonctionnellement et telle que ceci ne soit plus vrai pour aucune de ses parties.

Exemple :  
soit l'entité :



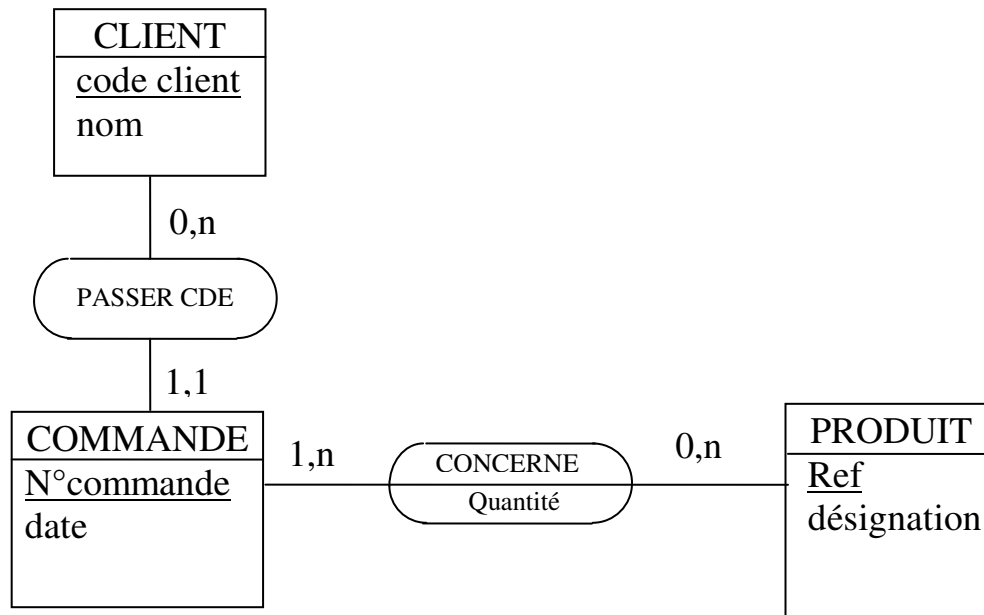
CODE\_CLIENT + NOM n'est pas une clé bien que  
 $\text{CODE\_CLIENT + NOM} \xrightarrow{\text{df}} \text{ADRESSE}$   
 car ceci reste vrai pour la partie CODE\_CLIENT de CODE\_CLIENT + NOM  
 puisque CODE\_CLIENT détermine ADRESSE.

En revanche CODE\_CLIENT est une clé car :  
 $\text{CODE\_CLIENT} \xrightarrow{\text{df}} \text{NOM}$   
 $\text{CODE\_CLIENT} \xrightarrow{\text{df}} \text{ADRESSE}$   
 et car ceci n'est plus vrai pour aucune partie de CODE\_CLIENT.

### Dépendances fonctionnelles entre entités

On dit qu'il existe une dépendance fonctionnelle entre entités A et B et on note :  
 $A \longrightarrow B$   
 si toute occurrence de A détermine une et une seule occurrence de B.

Exemple , soit le MCD suivant :



On a : COMMANDE  $\longrightarrow$  CLIENT

Les cardinalités 1,1 de COMMANDE dans cette relation expriment que tout bon de commande détermine un et un seul client. Il s'agit bien entendu d'un client ayant passé commande.

**La cardinalité maximum 1 correspond toujours a une dépendance fonctionnelle.**

Les dépendances fonctionnelles entre entités sont à considérer à propos des relations entre ces entités.

### Propriétés dépendances fonctionnelles

Réflexivité :

$a \xrightarrow{\text{df}} a$   
 exemple : REF  $\xrightarrow{\text{df}}$  REF

Projection :

$a \xrightarrow{\text{df}} b+c \Rightarrow a \xrightarrow{\text{df}} b \text{ et } a \xrightarrow{\text{df}} c$   
 exemple : REF  $\xrightarrow{\text{df}}$  DESIGN+PU  
 $\Rightarrow$  REF  $\xrightarrow{\text{df}}$  DESIGN et REF  $\xrightarrow{\text{df}}$  PU

Augmentation :

$a \xrightarrow{\text{df}} b \Rightarrow \forall c \quad a+c \xrightarrow{\text{df}} b$   
 exemple : REF  $\xrightarrow{\text{df}}$  PU  $\Rightarrow$  REF+DESIGN  $\xrightarrow{\text{df}}$  PU

Additivité :

$a \xrightarrow{df} b$  et  $a \xrightarrow{df} c \Rightarrow a \xrightarrow{df} b+c$

exemple :

$$\left. \begin{array}{l} \text{REF} \xrightarrow{df} \text{DESIGN} \\ \text{REF} \xrightarrow{df} \text{PU} \end{array} \right\} \text{REF} \xrightarrow{df} \text{DESIGN+PU}$$

Transitivité :

$a \xrightarrow{df} b$  et  $a \xrightarrow{df} c \Rightarrow a \xrightarrow{df} c$

exemple :

$$\left. \begin{array}{l} \text{REF} \xrightarrow{df} \text{CODE\_TVA} \\ \text{CODE\_TVA} \xrightarrow{df} \text{TAUX\_TVA} \end{array} \right\} \text{REF} \xrightarrow{df} \text{TAUX\_TVA}$$

Pseudo-transitivité :

$a \xrightarrow{df} b$  et  $b+c \xrightarrow{df} d \Rightarrow a+c \xrightarrow{df} d$

exemple :

$$\left. \begin{array}{l} \text{REF} \xrightarrow{df} \text{CODE\_TVA} \\ \text{CODE\_TVA+PU} \xrightarrow{df} \text{TAUX\_TVA} \end{array} \right\} \text{REF+PU} \xrightarrow{df} \text{TAUX\_TVA}$$

## 4) Règles relatives au MCD

### Normalisation des entités

Les entités du MCD doivent vérifier les règles suivantes :

#### a) Première forme normale

Dans une entité, toutes les propriétés sont élémentaires et il existe au moins une clé caractérisant chaque occurrence de l'objet représenté.

Exemple :

CLIENT
nom
adresse
cp
ville

Cette entité n'est pas en première forme normale car il n'y a pas de clé (plusieurs clients peuvent avoir le même nom).



La clé si elle est unique sera prise comme identifiant. Lorsqu'il y a plusieurs clés, on en choisira une comme identifiant.

Toute entité doit avoir un identifiant.

b) Deuxième forme normale

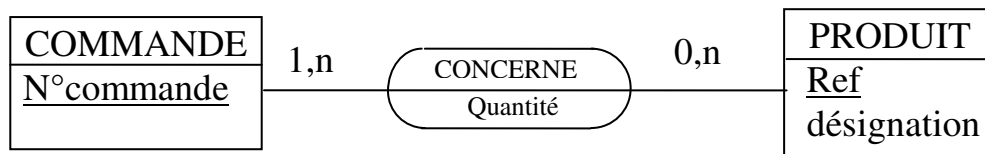
Toute propriété d'une entité doit dépendre de la clé par une dépendance fonctionnelle élémentaire. Autrement dit toute propriété de l'entité doit dépendre de tout l'identifiant.

Exemple :

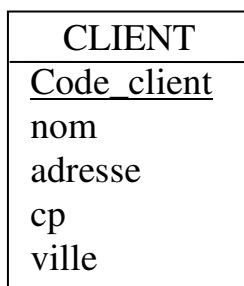


La clé est la concaténation de N°commande + Ref mais la dépendance fonctionnelle  $N^{\circ}commande + Ref \xrightarrow{df} Désignation$  n'est pas élémentaire puisque  $Ref \xrightarrow{df} Désignation$ .

Cette entité n'est pas en 2<sup>ème</sup> forme normale. Le MCD devrait devenir :



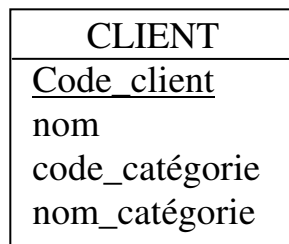
En revanche l'entité :



est en 2<sup>ème</sup> forme normale.

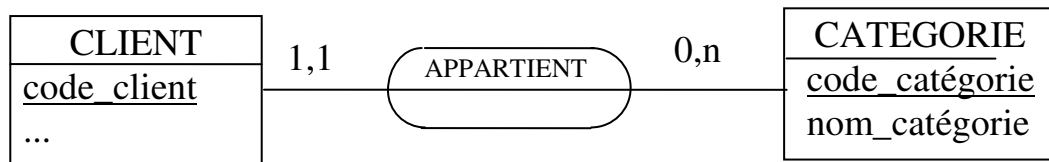
c) Troisième forme normale

Dans une entité toute propriété doit dépendre de la clé par une dépendance fonctionnelle élémentaire directe. Exemple :



n'est pas en 3<sup>ème</sup> FN car la dépendance fonctionnelle  
code\_client → nom\_catégorie  
n'est pas directe du fait de la transitivité.

Le MCD devrait devenir :



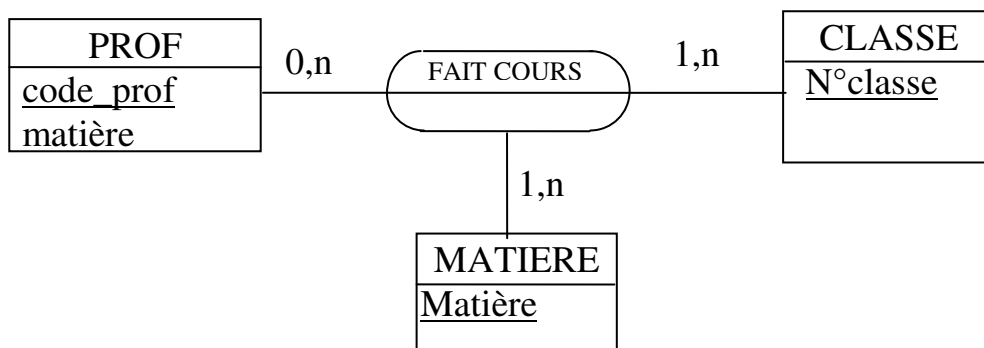
Les normalisations ci-dessus ont pour but d'éliminer les redondances (inutile de répéter la désignation du produit commandé à chaque commande d'un même produit) et d'éliminer les anomalies de mise à jour (si on annule un client on veut sans doute toutefois conserver la catégorie de ce client).

d) Respect des contraintes d'intégrité

Le MCD doit respecter les règles de gestion qui expriment les contraintes d'intégrité.

Exemple :

Le MCD suivant ne respecte pas la règle de gestion : un professeur enseigne une ou plusieurs matières.



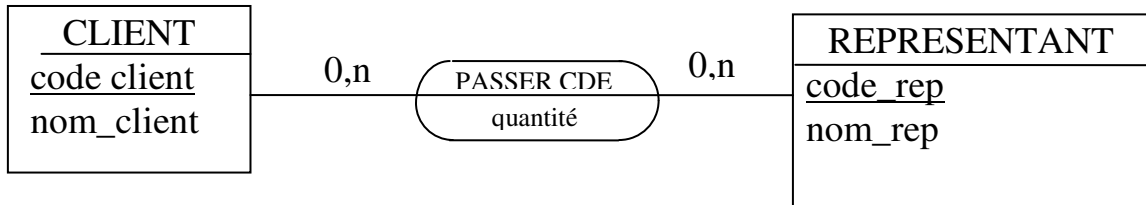
En effet ce MCD admet des professeurs qui n'enseignent pas, ce qui contredit la règle de gestion.

## Vérification

Dans toute occurrence d'entité ou de relation, il ne doit y avoir qu'une seule valeur de chaque propriété (non répétitivité). Pour les entités cette règle résulte de la 1<sup>re</sup> FN. Elle doit rester vraie pour les relations.

Exemple :

Soit le MCD :



La relation PASSER CDE n'est pas vérifiée, car il peut y avoir plusieurs valeurs de la quantité dans une commande passée par un client à un représentant. La quantité ne dépend pas seulement du client et du représentant mais aussi du produit commandé.

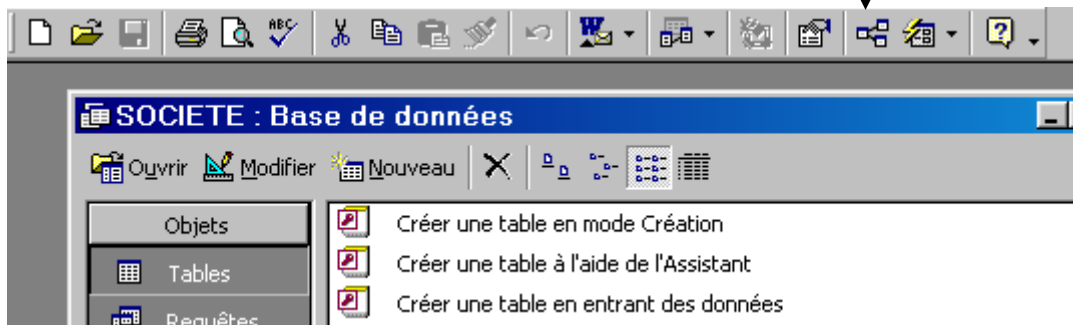
Autrement dit :


dans une relation les propriétés doivent dépendre fonctionnellement de l'ensemble des identifiants des entités concernées par la relation (mais d'aucun sous ensemble de cet ensemble). La concaténation de ces identifiants constitue l'identifiant de la relation.

# COURS 1 - Créer une relation entre deux tables

## 1) Créer une relation entre deux tables

\* cliquer sur

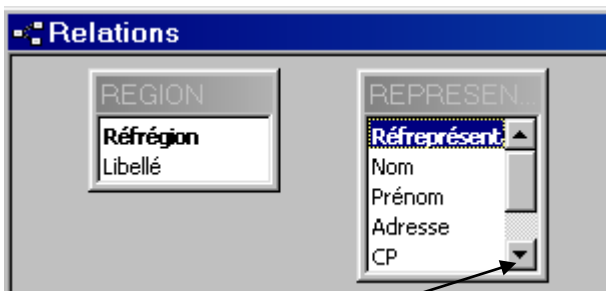


\* On obtient une fenêtre permettant de choisir les tables à mettre en relation (si cette fenêtre n'apparaît pas il faut cliquer sur  ) :



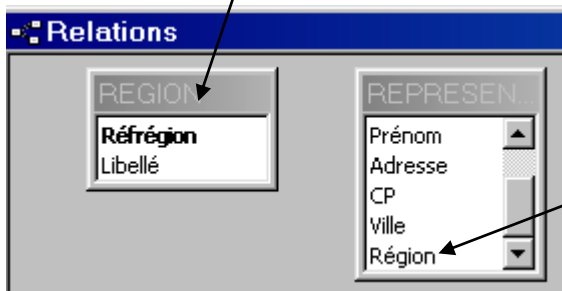
\* Cliquer sur la table à choisir puis sur **Ajouter**. Recommencer pour les autres tables. Pour arrêter cliquer sur **Fermer**.

\* On obtient :



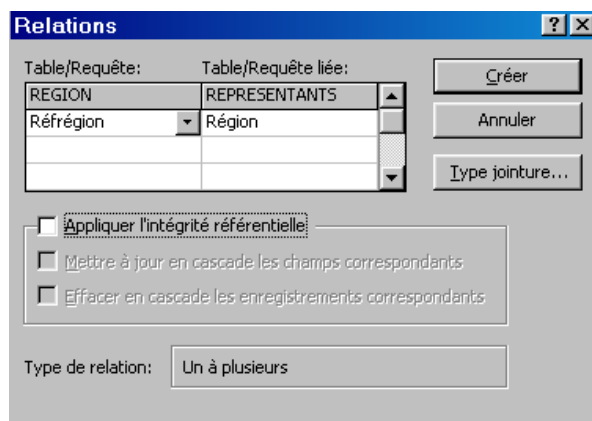
\* Faire défiler pour voir le champ en relation (**Région** par exemple)

\* Cliquer sur le champ entrant dans la relation et le faire glisser dans l'autre table vers son homologue.



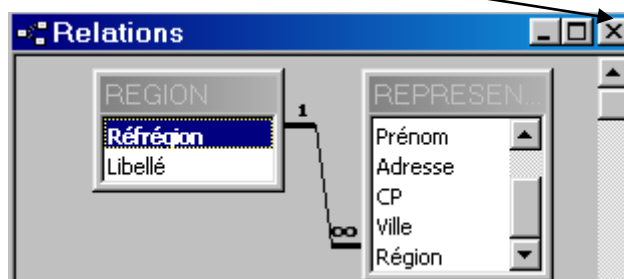
Remarque importante : le sens du déplacement est très important, il est toujours de "un vers plusieurs". En effet : une région peut avoir plusieurs représentants. On part donc de la table REGIONS vers la table REPRESENTANTS (on part en fait de la clé primaire vers un autre champ).

\* On obtient :



\* Si on veut qu'Access vérifie la cohérence des données lors de la saisie il faut cocher la case **Appliquer l'intégrité référentielle**. Par exemple on ne pourra pas entrer dans REPRESENTANTS une région qui n'existe pas dans la table REGIONS. C'est une sécurité pour la saisie. On peut aussi **Mettre à jours ou/et effacer en cascade les champs correspondants**. Par exemple si on modifie/efface un numéro de région dans la table REGION il est également modifié/effacé dans la table REPRESENTANT

\* Cliquer sur **Créer** puis fermer la fenêtre



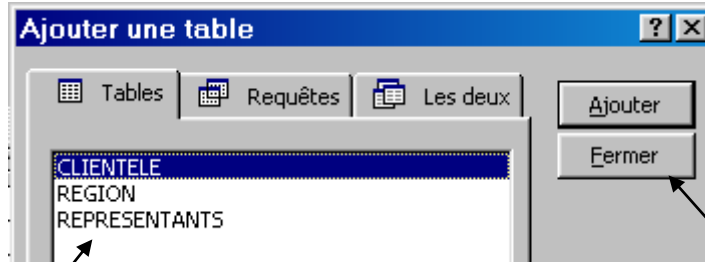
1 et ∞ représentent les cardinalités respectivement 1 et plusieurs (signe infini).

\* Enregistrer ou pas les relations.

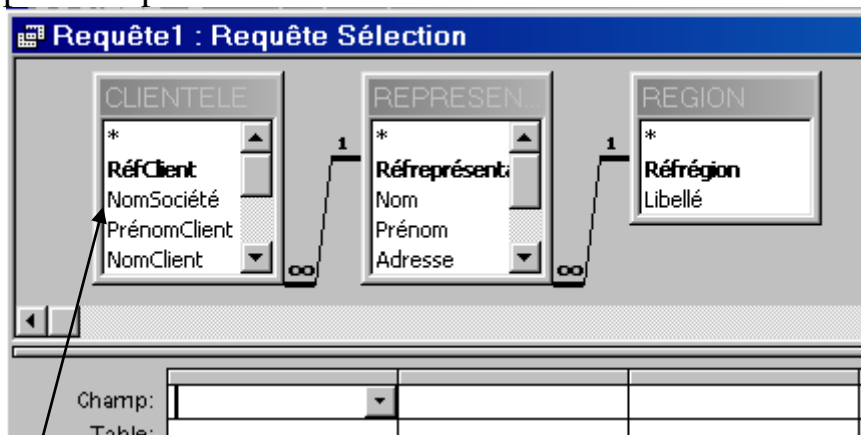
## COURS 2 – Créer une requête utilisant des données de plusieurs tables

### 1) Créer une requête utilisant des données de plusieurs tables

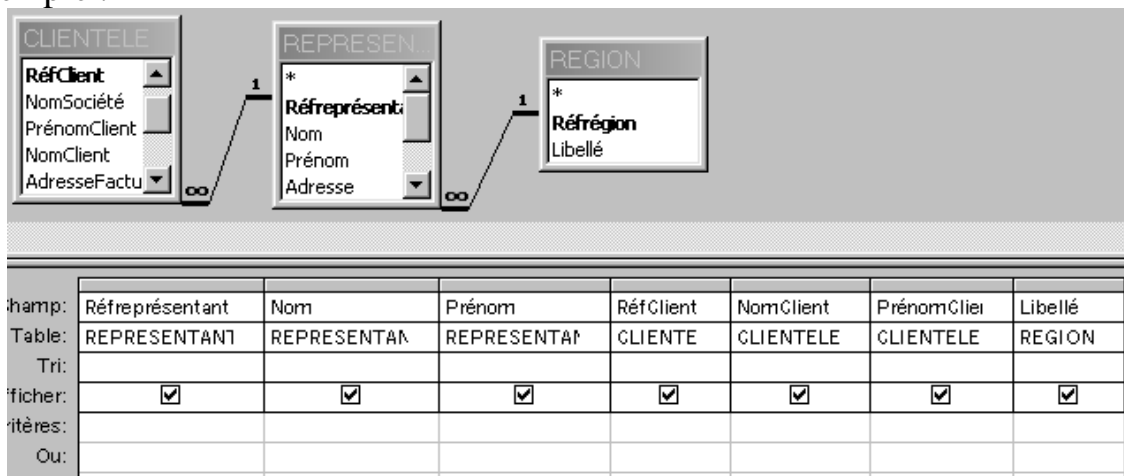
\* Cliquer sur l'onglet **Requêtes** puis double clic sur **Créer une requête en mode création**, on obtient :



\* Faire des doubles clics sur les tables à mettre dans la requête puis **Fermer**. On obtient par exemple :



\* Faire des doubles clics sur les champs à insérer dans la requête. On obtient par exemple :

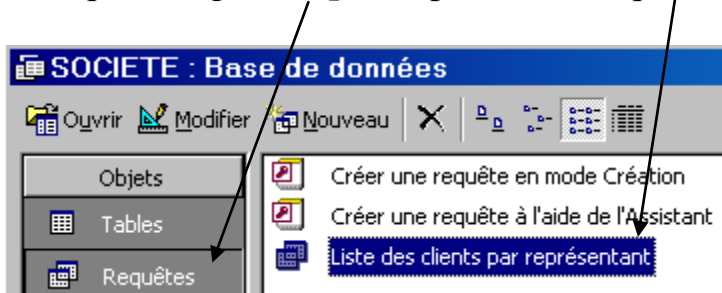


puis cliquer sur pour l'exécuter puis sur pour fermer. Enregistrer ou pas.

## COURS 3 – Transformer une requête de sélection en requête paramétrée – Supprimer une requête

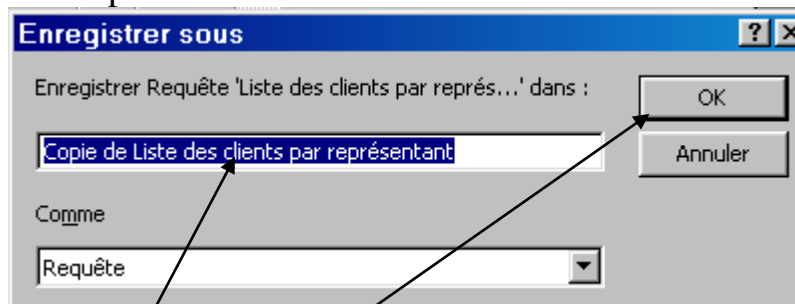
### 1) Faire une copie de requête

\* Cliquer l'onglet **Requêtes** puis sur la requête



\* Cliquer sur le menu **Fichier** puis **Enregistrer sous...**

On obtient par exemple :

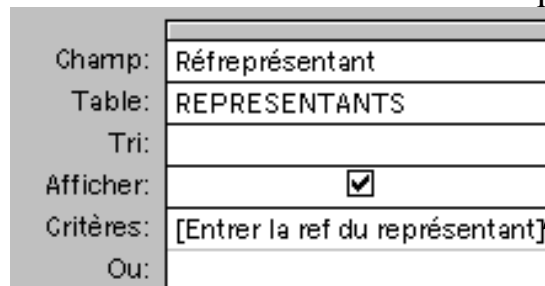



\* Entrer le nouveau nom puis **OK**

### 2) Transformer une requête de sélection en requête paramétrée

\* Cliquer sur l'onglet **Requêtes** puis sur la requête à modifier puis sur **Modifier** puis **OK**

\* Sur la ligne de critères dans la colonne du champ écrire un message entre crochet. Par exemple si on veut demander de choisir le représentant on aura :

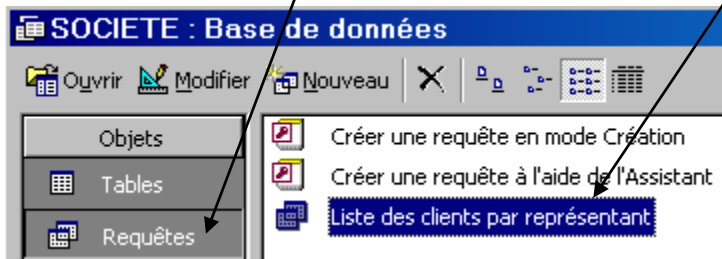


\* Cliquer sur  pour l'exécuter, le message apparaît invitant à entrer la valeur cherchée.

\* Cliquer sur  pour fermer.

### 3) Supprimer une requête

\* Cliquer l'onglet **Requêtes** puis sur la requête à supprimer



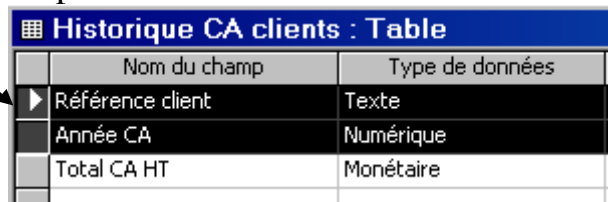
\* Appuyer sur la touche **Suppr** confirmer ou pas la suppression.



## COURS 4 – Créer une clé primaire composée de plusieurs champs

### 1) Créer une clé primaire composée de plusieurs champs

- \* Se mettre en mode création/modification de la table
- \* Sélectionner les champs en faisant un cliquer glissé devant le nom de champ



	Nom du champ	Type de données
▶	Référence client	Texte
	Année CA	Numérique
	Total CA HT	Monétaire

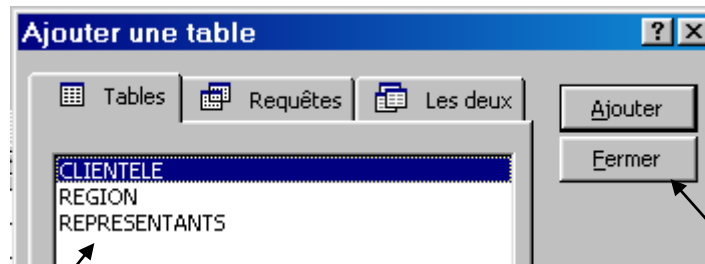
Remarque : il est possible de sélectionner des champs non contigus en laissant appuyer la touche **CTRL**.

- \* cliquer sur 

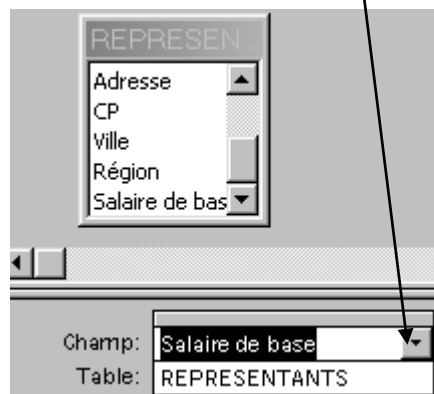
## COURS 5 – Modifier les valeurs dans une table à l'aide d'une requête de mise à jour

### 1) Créer une requête de mise à jour

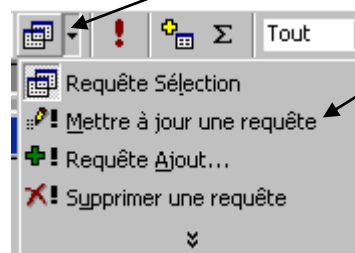
\* Cliquer sur l'onglet **Requêtes** puis **Nouveau** puis **Mode Création** puis **OK**, on obtient :



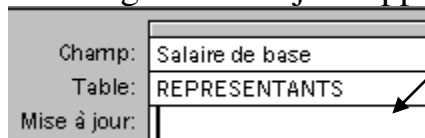
\* Faire un double clic sur la table à mettre dans la requête puis **Fermer**. Puis insérer les champs nécessaires dans la requête. On obtient par exemple :



\* Transformer la requête en requête de mise à jour en cliquant sur puis sur



\* une ligne Mise à jour apparaît alors :




\* Ecrire sur cette ligne la nouvelle valeur du champ ou la formule de calcul s'il y a un calcul à faire (sachant que les champs s'écrivent entre crochets). Pour augmenter le salaire de 5%, la formule sera par exemple :

[salaire de base]\*1,05

Champ:	Salaire de base
Table:	REPRESENTANTS
Mise à jour:	[salaire de base]*1,05
Critères:	

Il est possible de faire la mise à jour que pour certains enregistrements en ajoutant un critères. Par exemple on peut augmenter de 5% les salaires supérieur à 10000. La requête serait alors :

Champ:	Salaire de base
Table:	REPRESENTANTS
Mise à jour:	[salaire de base]*1,05
Critères:	>10000

\* Exécuter la requête en cliquant sur  confirmer ou pas la mise à jour. Fermer et enregistrer ou pas la requête.

#### Remarques très importantes :

- la mise à jour de la table s'effectue à chaque exécution ou ouverture de la requête (si vous ouvrez ou exécutez 2 fois la requête précédente, les salaires seront augmentés deux fois de 5%)
- la remarque précédente est vraie même si on n'enregistre pas la requête de mise à jour.

**Donc faire très attention lors de l'utilisation de ce type de requête.**


# COURS 6 Créer une liste déroulante dans un formulaire

## 1) Créer une liste déroulante dans un formulaire

\* Faire un formulaire instantané


On obtient par exemple :

Réprésentant	10
Nom	Duponchel
Prénom	Eric
Adresse	1, rue Rose
CP	13000
Ville	Marseille
Région	1
Salaire de base	6 830,25 F

\* Cliquer sur  pour passer en mode création/modification

\* Supprimer le contrôle qui aura la liste déroulante

( par ex)

\* Cliquer au besoin sur  pour faire apparaître cette barre d'outils :



\* Cliquer sur puis sur. Le pointeur de la souris se transforme, faire alors un cliqué glissé à l'emplacement de la liste.

On obtient :

### Assistant Zone de liste

Cet Assistant crée une zone de liste, qui affiche une série de valeurs que vous pouvez sélectionner.

Comment souhaitez-vous que votre zone de liste obtienne ces valeurs ?

Je veux que la zone de liste recherche les valeurs dans une table ou requête.

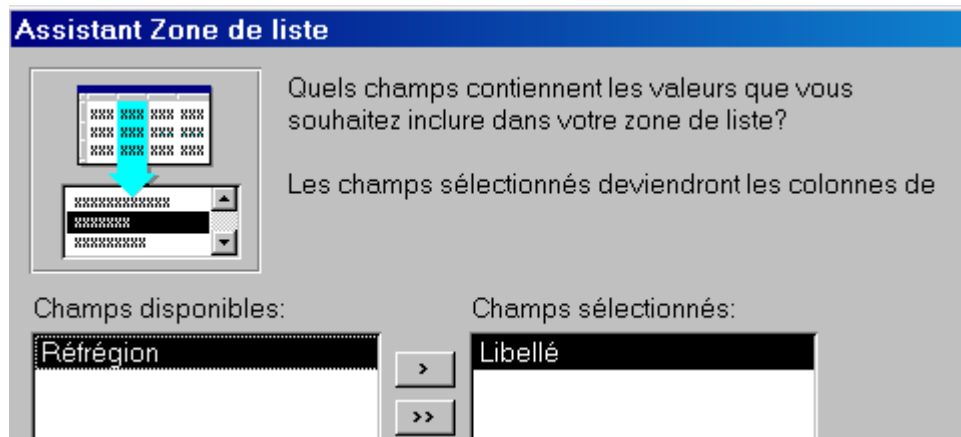
Je taperai les valeurs souhaitées.

Rechercher un enregistrement dans mon formulaire basé sur la valeur sélectionnée dans zone de liste

\* Faire le choix voulu (ici le premier car les valeurs vont être cherchées dans la table **REGION**). Puis cliquer sur **Suivant**.

\* Choisir la table (ici **REGION**) puis **Suivant**.

\* Double clic sur le champ figurant dans la liste déroulante (ici libellé). On obtient :



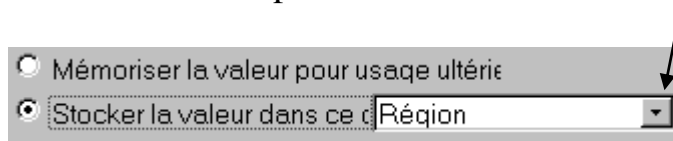
puis **Suivant**

\* Elargir éventuellement la colonne :



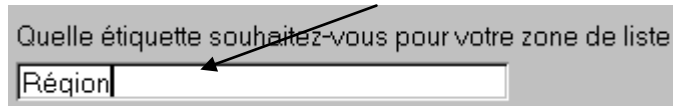
puis **Suivant**

\* Choisir le champ stockant la valeur de la liste (ici Région)



puis **Suivant**

\* Donner un nom au contrôle :




puis **Terminer**



On a :

Détail	
Répreprésentant	Répreprésentant
Nom	Nom
Prénom	Prénom
Adresse	Adresse
CP	CP
Ville	Ville
Région	Région
Salaire de base	Salaire de base

\* On peut ensuite déplacer et dimensionner des contrôles par exemple :



\* Passer ensuite en mode formulaire pour commencer la saisie 

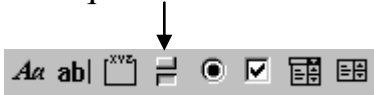
\* Cliquer sur  pour enregistrer le formulaire. Donner un nom puis **OK** puis fermer 

# COURS 7 - Créer un bouton bascule dans un formulaire

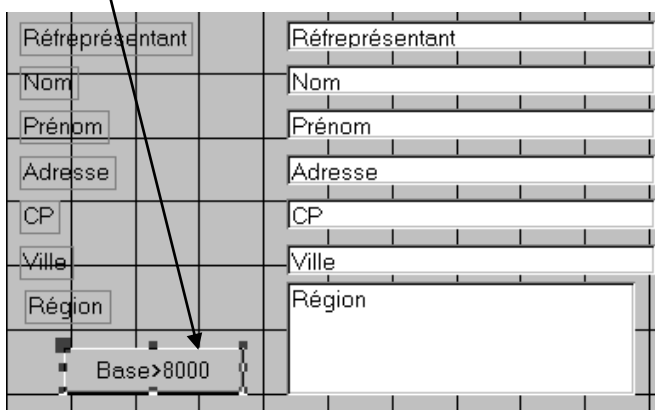
## 1) Créer un bouton bascule dans un formulaire

\* Se mettre en mode création/modification du formulaire.

\* Cliquer sur



\* le pointeur de la souris se transforme faire un cliqué glissé pour délimiter le bouton

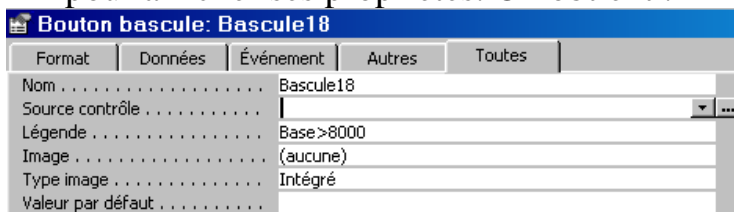


\* Ecrire dedans son libellé.(base>8000 par exemple ou à 1200)

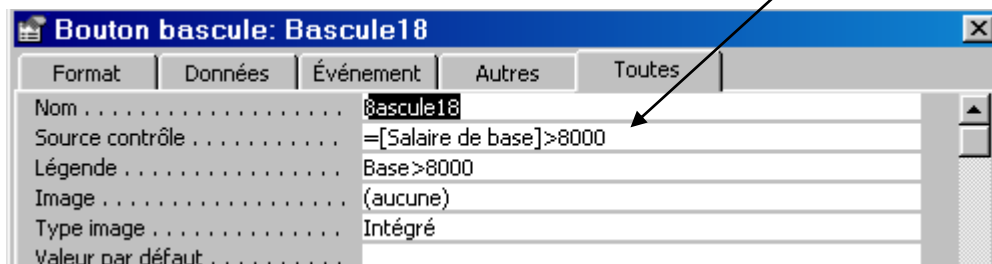
\* Sélectionner ce bouton (il apparaît entouré de petits carrés) puis cliquer sur



pour afficher ses propriétés. On obtient :



\* Ecrire la formule pour laquelle le bouton apparaît enfoncé



\* Fermer la fenêtre des propriétés et passer en mode formulaire  pour vérifier que le bouton fonctionne. Fermer le formulaire puis enregistrer ou pas.


# COURS 8 – Réaliser des calculs dans une requête – Mettre un champ en monétaire – Ajouter une table

## 1) Réaliser des calculs

- \* Se mettre en mode création/modification de requête
- \* Taper le libellé du champ suivi de deux points, taper ensuite la formule voulue sachant que les champs de la table s'écrivent entre crochets.

Exemple de formule permettant de trouver le TTC à partir du HT:


Champ:	Total CA TTC: [Total CA HT]*1,196
Table:	
Tri:	
Afficher:	<input checked="" type="checkbox"/>
Critères:	
Ou:	

- \* Exécuter la requête 
- \* Fermer puis enregistrer ou pas

## 2) Afficher un champ en monétaire

- \* Se mettre en mode création/modification de requête (si on ne l'est pas déjà)
- \* Sélectionner le champ en cliquant ici (le pointeur prend l'aspect d'une flèche noire dirigée vers le bas)

Champ:	Total CA TTC: [Total CA HT]*1,196
Table:	
Tri:	
Afficher:	<input checked="" type="checkbox"/>
Critères:	
Ou:	

- \* cliquer sur  pour afficher les propriétés du champ. On obtient :

Propriétés du champ	
Général	Liste de choix
Description . . . . .	
Format . . . . .	
Décimales . . . . .	
Masque de saisie . . . . .	
Légende . . . . .	


- \* Cliquer ici puis sur **Monétaire** puis sur **Euro**



\* Fermer la fenêtre

### **3) Ajouter une table dans un requête existante**

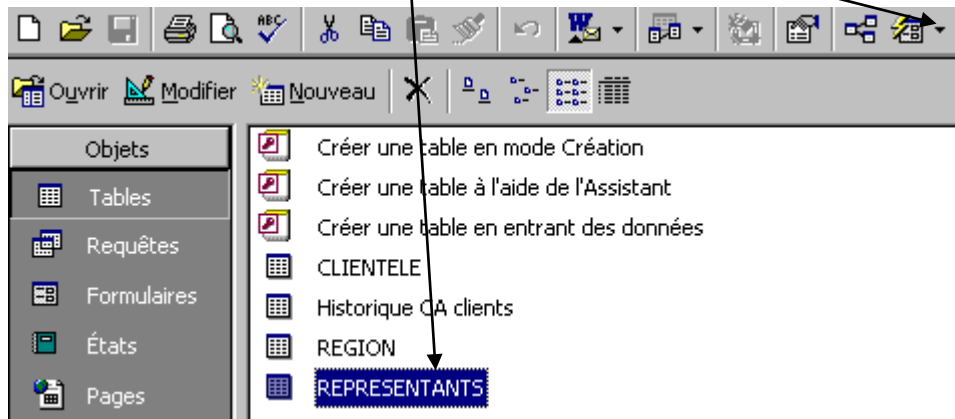
\* Se mettre en mode création/modification de requête

\* Cliquer sur  puis double clic sur la table à ajouter puis cliquer sur **Fermer**.

# COURS 9 – Insérer des sous-formulaires dans un formulaire

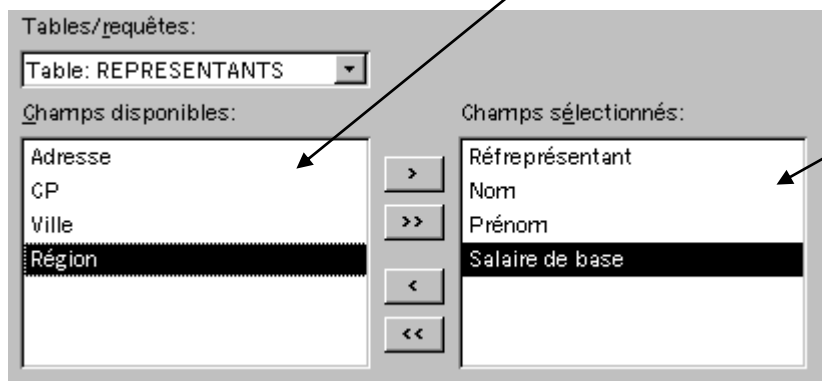
## 1) Insérer des sous-formulaires

\* Cliquer sur la table principale puis sur

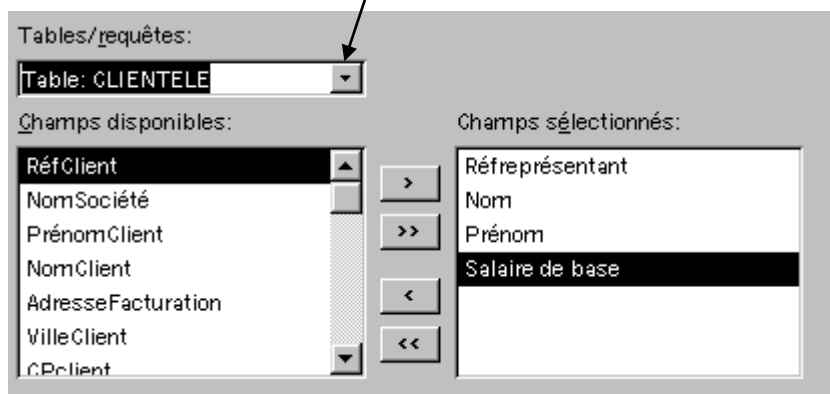


\* Cliquer ensuite sur **Formulaire** puis sur **Assistant Formulaire** puis sur **OK**

\* Choisir les champs de la première table et les faire passer dans la fenêtre de droite



\* Choisir la deuxième table



\* Choisir les champs de la deuxième table et les faire passer dans la fenêtre de droite

Tables/requêtes:  
Table: CLIENTELE

Champs disponibles:  
NomSociété  
AdresseFacturation  
VilleClient  
CPClient  
NuméroTél  
RéfCommercial

Champs sélectionnés:  
Nom  
Prénom  
Salaire de base  
RéfClient  
NomClient  
PrénomClient

\* Choisir la troisième table puis faire passer les champs dans la fenêtre de droite

Tables/requêtes:  
Table: Historique CA clients

Champs disponibles:  
Référence client

Champs sélectionnés:  
Salaire de base  
RéfClient  
NomClient  
PrénomClient  
Année CA  
Total CA HT

\* Cliquer sur **Suivant**. On obtient :

**Assistant Formulaire**

Comment souhaitez-vous afficher vos données ?

par CLIENTELE  
par Historique CA clients  
par REPRESENTANTS

Réfreprésentant, Nom, Prénom, Salaire de base

RéfClient, NomClient, PrénomClient

Année CA, Total CA HT

Fomulaire avec sous-formulaire(s)  Fomulaires attaché

\* Cliquer trois fois sur **Suivant**. On obtient :

Assistant Formulaire

Quels titres souhaitez-vous pour vos formulaires ?

Formulaire: REPRESENTANTS

Sous-formulaire: CLIENTELE Sous-formulaire1

Sous-formulaire: Historique CA clients Sous-formulaire1

Ce sont toutes les réponses dont l'Assistant a besoin pour créer votre formulaire.

Souhaitez-vous ouvrir le formulaire ou en modifier la structure ?

Ouvrir le formulaire pour afficher ou entrer des informations.

Modifier la structure du formulaire.

Afficher l'Aide sur l'emploi des formulaires

Annuler < Précédent Suivant > Terminer

- \* Entrer éventuellement les titres puis cliquer sur **Terminer**
- \* Faire défiler les enregistrements dans le formulaire et les sous formulaires.
- \* Fermer

## COURS 10 – Réaliser des calculs dans un formulaire

### 1) Réaliser des calculs dans un formulaire

- \* Cliquer sur l'onglet **Formulaires** puis sur le formulaire à modifier
  - \* Cliquer sur **Modifier**
  - \* Cliquer sur **abl** puis cliquer dans le formulaire à l'emplacement de la formule.
- On a :



Ecrire le message à gauche et la formule à droite. On a par exemple :

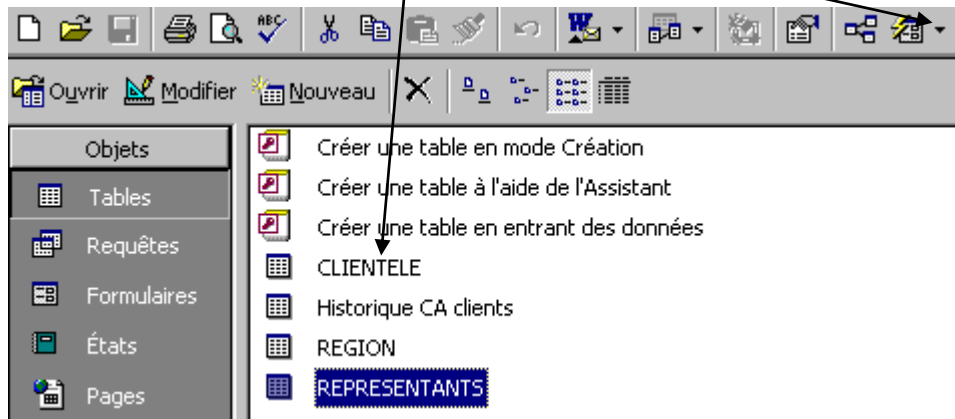
En-tête de formulaire			
Détail			
Réprésentant	Réprésentant	Salaire de base	Salaire de base
Nom	Nom	Frais déplacements	= salaire de base*15/100
Prénom	Prénom	Total:	= salaire de base*1,15
CLIENTELE			
CLIENTELE Sous-formulaire1			
Historique CA clients			
Historique CA clients Sous-formulaire1			

- \* Enregistrer puis Fermer.

# COURS 11 – Réaliser un état multi tables

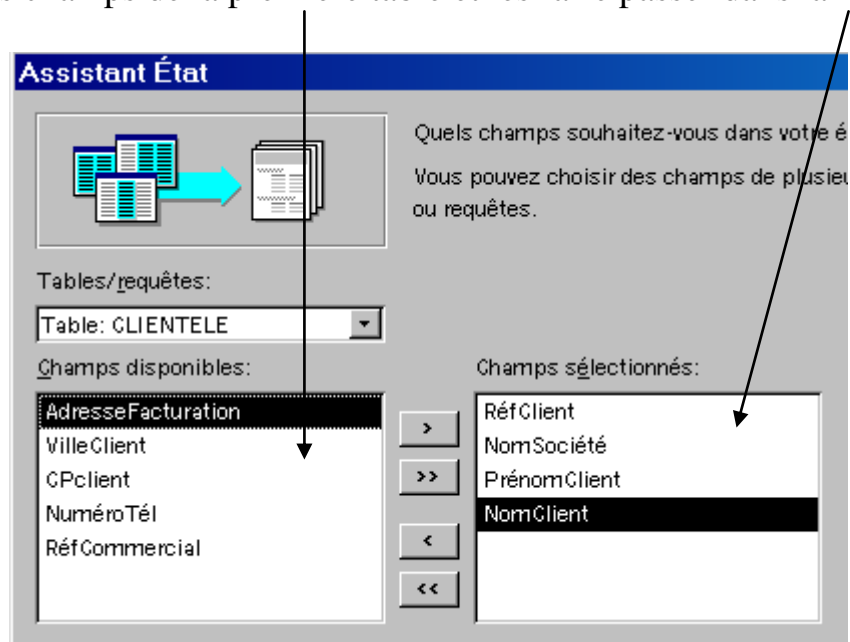
## 1) Réaliser un état multi tables

\* Cliquer sur la table principale puis sur

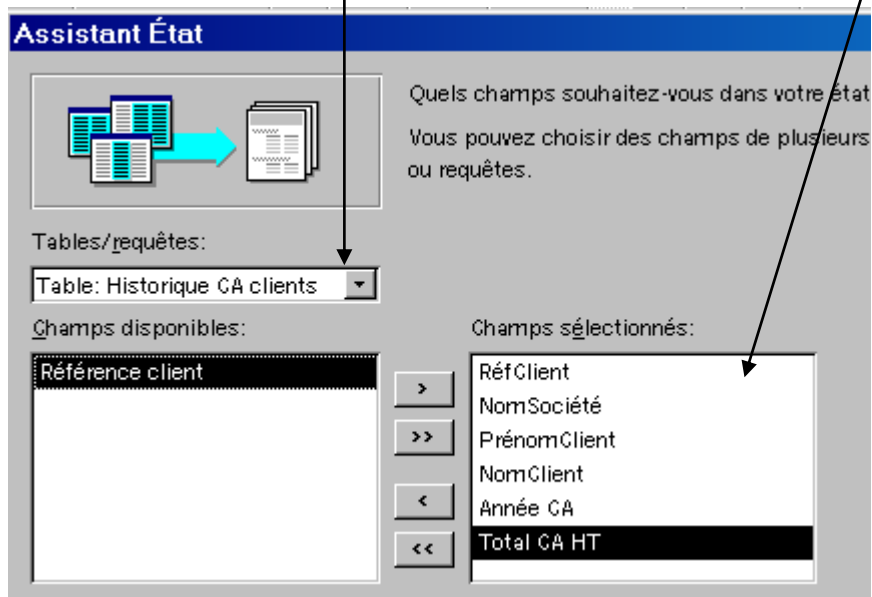


\* Cliquer ensuite sur **Etat** puis sur **Assistant état** puis sur **OK**

\* Choisir les champs de la première table et les faire passer dans la fenêtre de droite



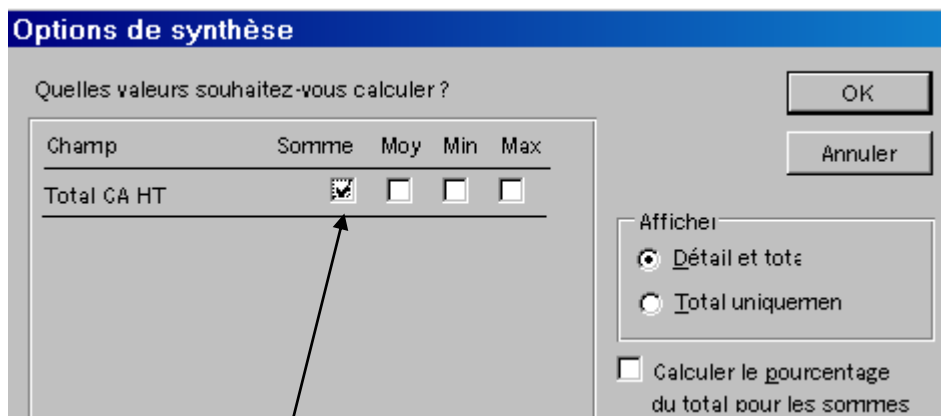
\* Choisir la deuxième table puis faire passer les champs à droite



\* Cliquer trois fois sur **Suivant**

\* Cliquer sur **Options de synthèse...**

On obtient :



\* Choisir les calculs puis **OK**

\* Cliquer sur **Suivant**

\* Choisir la présentation puis **Suivant**

\* Choisir la présentation puis **Suivant**

\* Entrer le titre de l'état puis **Terminer**

\* Fermer l'aperçu

\* Fermer l'état

## COURS 12 – Réaliser un publipostage dans Word à partir d'une table Access – Exporter des données vers Word

### 1) Réaliser un publipostage dans Word à partir d'une table Access

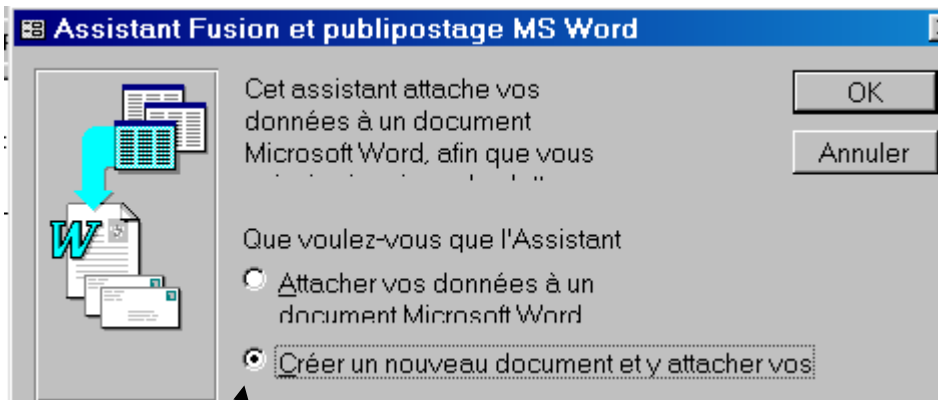
\* Cliquer sur la table ou la requête servant de base au publipostage

\* Cliquer sur



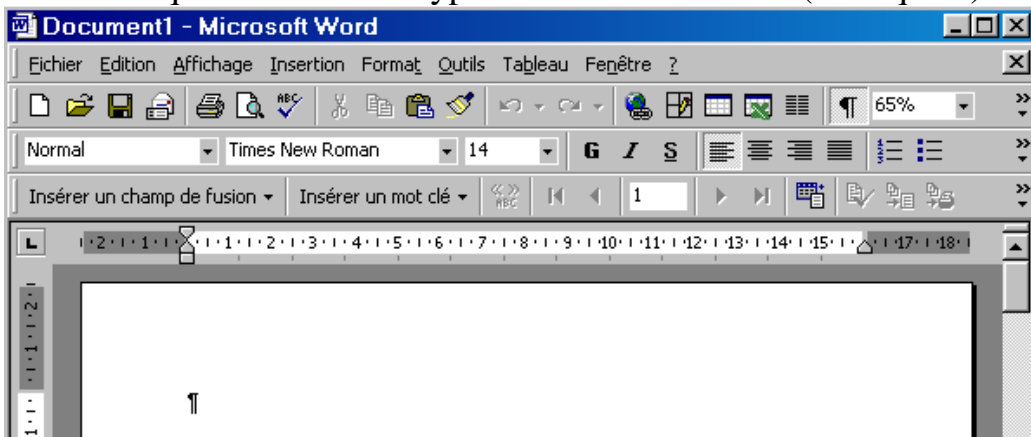
\* Cliquer sur **Fusionner avec MS Word**

On obtient :



\* Faire votre choix puis **OK**

\* La fenêtre de Word apparaît (l'agrandir éventuellement) avec un nouveau document qui est une lettre type associée à une table (ou requête) Access :



\* Réaliser le document dans Word (insérer le destinataire en cliquant sur **Insérer champ de fusion**)



## 2) Exporter des données vers Word

### 21) Méthode 1 :

\* Cliquer sur la table ou la requête servant de base au publipostage


\* Cliquer sur



\* Cliquer sur **Exporter vers MS Word**

\* Word se lance avec un document contenant le contenu de la table (ou requête) sous forme de tableau.

### 22) Méthode 2 :

\* Ouvrir la table (ou la requête), sélectionner les données à copier 

\* Ouvrir Word (si Word n'est pas déjà ouvert, sinon cliquer dessus sur la barre des tâches en bas de l'écran)

\* Ouvrir le document (s'il n'est pas déjà ouvert) où doit s'effectuer le collage

puis coller au bon endroit 