

# Comprendre les jointures dans Access

Par Maxence HUBICHE

De nombreuses questions ayant été posées sur le forum à propos des jointures (jointure équivalente, jointure externe, jointure interne, jointure gauche,...) et de leur bon usage, il m'a parut bon de faire un petit rappel, ou un petit apprentissage, du fonctionnement interne des requêtes. Pour avoir une claire compréhension de l'incidence du type de jointure sur une requête, nous allons faire une série d'exercices. Il s'agira de requêtes, simples. À chaque fois, des questions seront posées : quel est le résultat attendu ? Quel est le résultat obtenu ? Et que pouvons-nous en déduire ? Ce cours est téléchargeable ici, dans un fichier au format pdf, zippé.

1. DONNEES DE BASE
2. RAPPELS FONDAMENTAUX
  - 2.1. Les Relations
  - 2.2. L'intégrité Référentielle
  - 2.3. Les Jointures
3. EXERCICES
  - 3.1. Cas d'une requête monotable
  - 3.2. Cas d'une requête multitables
  - 3.3. Cas d'une requête sans relations
  - 3.4. Cas d'une requête avec regroupement
  - 3.5. Les types de jointures
  - 3.6. Cas d'une requête avec jointure externe
  - 3.7. Cas d'une requête de non-correspondance
4. EN CONCLUSION
5. REMARQUE
6. AUTO-TEST

## 1. DONNEES DE BASE

Afin de faciliter l'apprentissage de chacun, la base servant de référence dans ces exercices est la base fournie par défaut par Microsoft® lors de l'installation d'ACCESS, à savoir Comptoir.mdb.

## 2. RAPPELS FONDAMENTAUX

### 2.1. Les Relations

ACCESS est un logiciel qui fait partie de la famille des SGBDR, c'est-à-dire des systèmes de gestion de bases de données relationnelles. La notion de relations est donc essentielle. Bien qu'il existe plusieurs types de relations, ACCESS ne gère que les relations de un à un, et les relations de un à plusieurs.

Pour créer une relation de un à un, il faut impérativement lier des champs ayant leur propriété "indexé" à "oui-sans doublon".

Pour créer une relation de un à plusieurs, il suffit de lier les champs entre eux.

D'habitude, la relation se fait d'une clé primaire vers une clé étrangère.

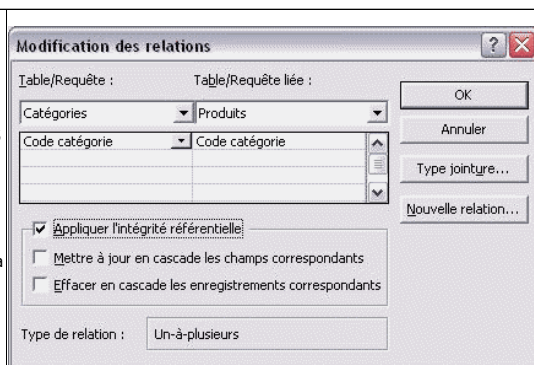
La clé primaire est le champ de la table qui sert d'identifiant à un enregistrement. Il s'agit d'un champ indexé sans doublon.

La clé étrangère est le champ de la table qui servira à matérialiser la relation avec la clé primaire. Il doit être de même taille et de même type de données que la clé primaire à laquelle il est lié.

*NB : Il est possible de créer une clé primaire multi-champs. Cependant, il n'est pas efficace de l'utiliser dans le cadre d'une relation. Une clé primaire multi-champs sert surtout de règle de gestion pour les enregistrements. Elle empêche d'avoir deux enregistrements ayant les mêmes valeurs dans les champs de la clé primaire. On peut obtenir le même résultat avec un index multi-champs, sans qu'il s'agisse d'une clé primaire.*

### 2.2. L'intégrité Référentielle

Il convient, lors de la création de la relation, d'appliquer l'intégrité référentielle. L'intégrité référentielle est un mécanisme de vérification qui s'assure que chaque clé étrangère est en correspondance avec sa clé primaire. Non seulement il vérifie l'intégrité des données écrites dans la clé étrangère, mais encore, il maintient cette intégrité en cas de modification ou de suppression de la clé primaire. En conséquence, lorsque l'intégrité référentielle est appliquée sur une relation, on ne devrait pas trouver, dans la clé étrangère, de données autres que celles de la clé primaire. Dès lors, bien que toutes les données figurant dans la table contenant la clé étrangère soit en relations avec les données de la table contenant la clé primaire, l'inverse n'est pas exact.



### 2.3. Les Jointures

Sous ACCESS, comme d'ailleurs dans la plupart des SGBDR, il y a plusieurs types de jointures.

Le type de jointure par défaut est la jointure équivalente, qu'on qualifie également de jointure interne.

Il existe cependant deux autres types de jointure, la jointure gauche, et la jointure droite, qu'on qualifie également de jointures externes.

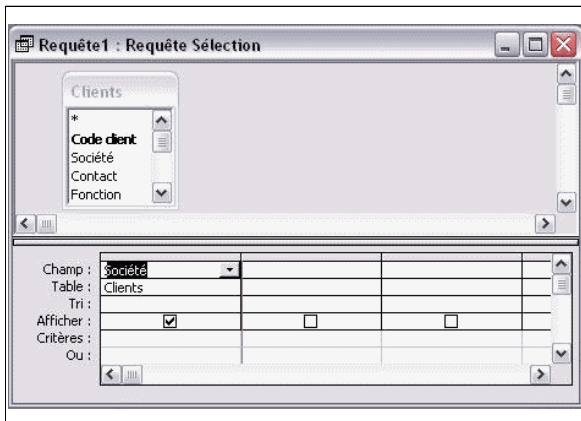
Examinons ensemble quelles en sont les influences à travers une série de requêtes simples.

## 3. EXERCICES

### 3.1. Cas d'une requête monotable

Créer une requête basée sur la table client

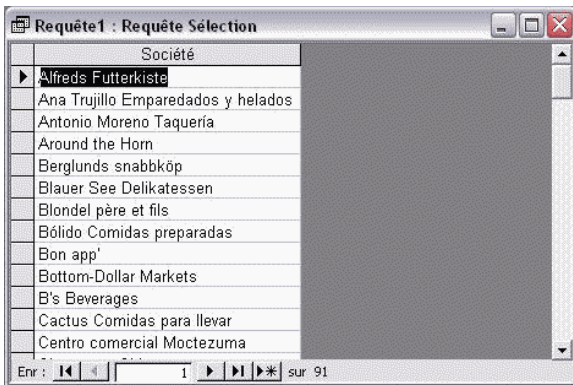
Afficher le champ [Société]



**Quel est le résultat attendu ?**

Aucun traitement particulier n'est effectué sur cette requête. Il devrait donc s'agir de l'affichage brut de la liste des champs [société] de la table des clients. Le nombre de lignes du résultat de la requête devrait donc correspondre au nombre de lignes dans la table.

**Quel est le résultat obtenu ?**



Nous obtenons ici quatre-vingt-onze lignes.

**Que pouvons-nous en déduire ?**

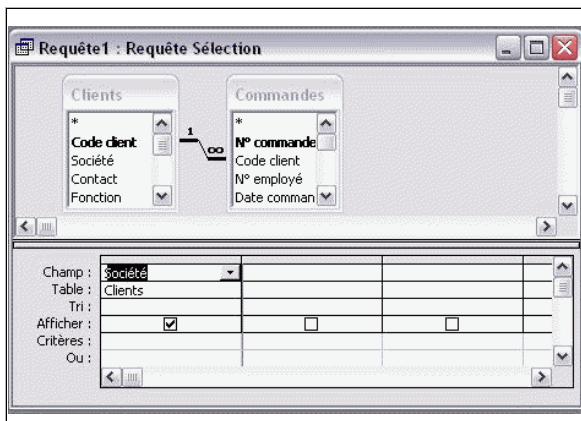
Qu'il y a 91 clients. Pour s'en assurer il suffit de regarder le contenu de la table Clients.

### 3.2. Cas d'une requête multitable

Dans la requête précédente ajoutez la table commande. Nous allons maintenant examiner ce qui se passe lorsqu'on met dans une requête des tables qui ne servent à rien pour obtenir le résultat attendu.

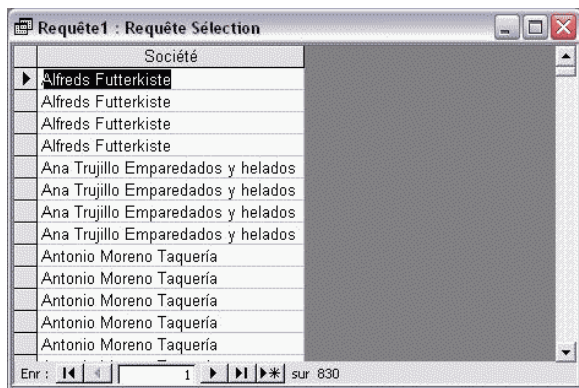
Notez au passage la description automatique de la relation existante entre la table des clients et celle des commandes. ACCESS reprend ici l'ensemble des informations définies préalablement dans la fenêtre des relations. Il s'agit d'une relation un à plusieurs, telle que un client peut avoir plusieurs commandes, chaque commande n'ayant qu'un seul client.

**Quel est le résultat attendu ?**



Puisqu'on continue à n'afficher que le champ [société], on s'attend logiquement à obtenir la liste des sociétés ayant des commandes, soit nos 91 lignes trouvées précédemment.

### Quel est le résultat obtenu ?



Nous obtenons maintenant 830 lignes.

### Que pouvons-nous en déduire ?

La réponse qui nous vient à l'esprit immédiatement est : les 91 clients ont effectué 830 commandes.

Cependant, cette affirmation n'est pas obligatoirement exacte. Pourquoi ? Parce que la relation entre les 2 tables est qualifiée de jointure équivalente. C'est-à-dire que cette requête travaille uniquement sur l'ensemble des associations possibles entre le champ [code client] de la table des clients et le champ [code client] de la table des commandes. Mais se pourrait-il qu'il y ait des commandes n'ayant pas de clients référencés dans la base ? N'oublions pas la présence de l'intégrité référentielle dans cette relation. Comme nous l'avons vu précédemment, elle permet de s'assurer que pour une clé étrangère il y a toujours une clé primaire qui lui corresponde. On peut donc dire qu'il y a effectivement 830 commandes car nous sommes certains qu'il n'y a que des commandes liées à des clients.

En regardant attentivement le résultat de la requête, nous notons que le nom de la société apparaît à plusieurs reprises. Cela montre le nombre de relations possibles entre le client et la commande. Le nom de la société apparaît autant de fois que cette société a passé des commandes.

#### Attention ! Notez bien ceci :

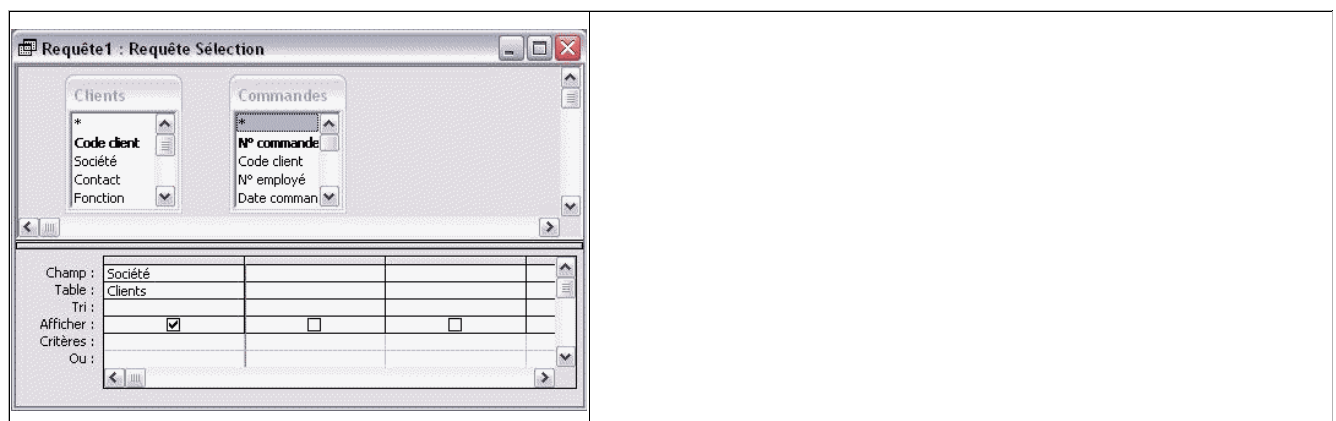
Si l'intégrité référentielle n'avait pas été activée sur cette relation, ce nombre de 830 lignes ne pourraient pas signifier autre chose que 830 associations client-commandes. Il serait en effet possible qu'il y ait, dans la table des commandes, des commandes ayant des codes client n'existant pas dans la table des clients.

### 3.3. Cas d'une requête sans relations

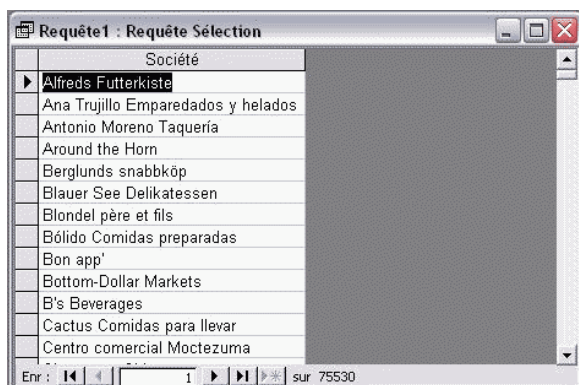
Admettons que nous cherchions à obtenir à nouveau les quatre-vingt-onze lignes de l'exercice numéro un, il semble que ce soit la relation qui nous empêche d'aboutir. La logique immédiate voudrait qu'on supprime cette relation. Alors supprimons-la et voyons ce qui se passe.

### Quel est le résultat attendu ?

Nous voudrions récupérer les quatre-vingt-onze lignes du départ. En effet, la grille indique que nous n'affichons que le champ [société] de la table des "clients".



### Quel est le résultat obtenu ?



Il y a cette fois 75 530 lignes.

### Que pouvons-nous en déduire ?

Que, sauf cas exceptionnel, il faudra toujours veiller à ce qu'il y ait des relations entre les différentes sources d'une requête. L'absence de l'une d'entre elles

que, dans les exceptions, il faudra toujours venir à la question des relations entre les différentes sources d'une requête. L'absence de l'une d'entre elles suffirait à réaliser un produit cartésien, c'est-à-dire l'association de chaque enregistrement d'une table à tous les enregistrements de l'autre table.

Ici ACCESS, n'ayant aucune indication des relations qu'il doit faire entre les clients et les commandes, va associer chacun des 91 clients avec les 830 commandes. Nous obtenons ainsi  $91 \times 830 = 75\,530$  lignes.

Cela ne correspond pas aux résultats attendus. Nous allons donc restaurer notre requête dans l'état précédent en supprimant la table commande et en la rajoutant à la requête. Nous nous retrouvons donc ici exactement dans le même état qu'à la fin de l'exercice numéro deux.

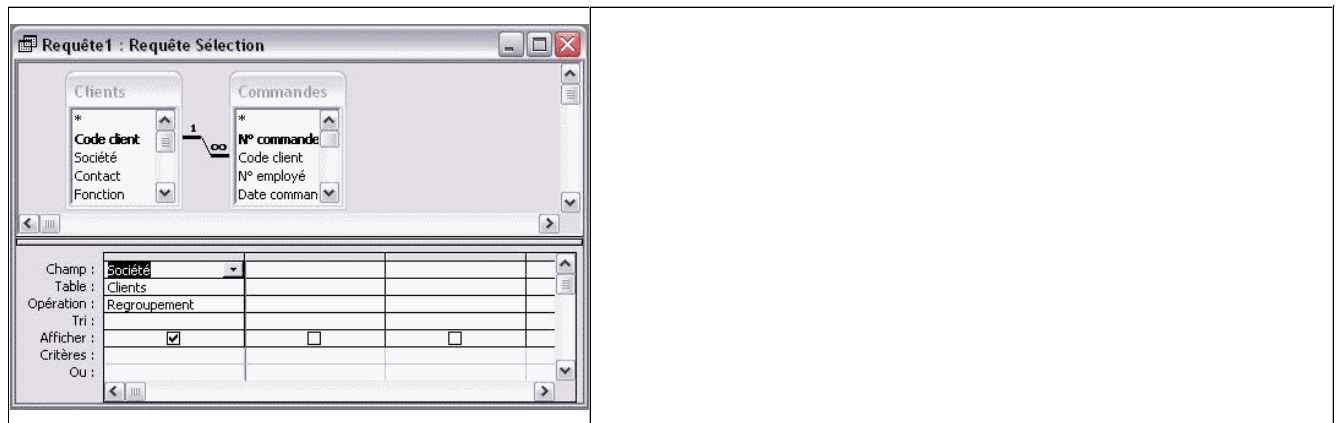
### 3.4. Cas d'une requête avec regroupement

Nous n'avons toujours pas réussi à récupérer les quatre-vingt-onze lignes du départ. En observant attentivement le résultat, comme nous l'avons dit précédemment, nous observons que le nom de la société apparaît plusieurs fois. Si nous parvenions à regrouper en une seule ligne toutes les occurrences d'un même nom de société, nous obtiendrions vraisemblablement les quatre-vingt-onze lignes attendues.

Faisons donc un clic sur le bouton sigma de notre barre d'outils. Cela fait apparaître une nouvelle ligne dans la grille. Il s'agit de la ligne opération.

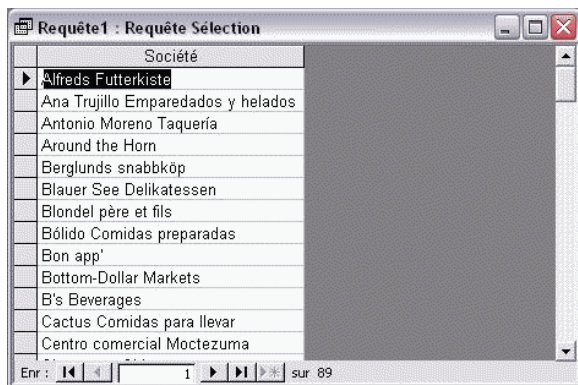
Nous observons que l'opération sélectionnée est l'opération de regroupement.

**Quel est le résultat attendu ?**



ACCESS devrait essayer de regrouper toutes les sociétés identiques pour n'en faire qu'une seule ligne, et nous devrions ainsi récupérer nos quatre-vingt-onze lignes du départ.

**Quel est le résultat obtenu ?**



Nous obtenons cette fois seulement 89 lignes.

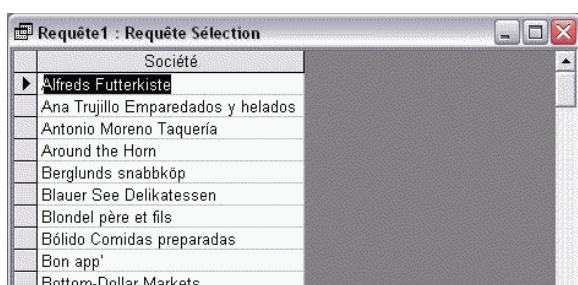
**Que pouvons-nous en déduire ?**

Premièrement, ayant regroupé sur les noms de société, il n'est pas impossible que nous ayons récupéré en une seule ligne au moins deux sociétés différentes de même nom. Il s'agit donc de s'assurer que chaque client apparaît bien dans sa ligne. Pour cela, rien ne vaut le regroupement sur une valeur unique et facilement identifiable, l'identifiant ou clé primaire

Rajoutons donc dans notre requête le champ [code client] de la table clients. L'opération est encore une fois l'opération de regroupement. Il s'agit en effet de l'opération par défaut. Décochons la case afficher, car, si le regroupement est utile, l'affichage du code ne nous est pas indispensable pour cet exercice.

**Quel est le résultat obtenu ?**

Cette fois, ACCESS devrait essayer de regrouper les identifiant des clients ainsi que leurs sociétés, de manière à créer des lignes dont l'association code client-société soit unique. Par conséquent, si plusieurs sociétés avaient le même nom, on devrait voir apparaître autant de lignes qu'il y a des clients différents, qu'ils aient ou n'aient pas le même nom de société.



B's Beverages	
Cactus Comidas para llevar	
Centro comercial Moctezuma	
Enr : 1 sur 89	

Nous obtenons toujours 89 lignes.

### Que pouvons-nous en déduire ?

Cette fois nous sommes absolument certains qu'il s'agit bien de 89 clients différents.

Comme nous l'avons décrit à l'exercice numéro deux, le résultat de la jointure permet d'obtenir l'ensemble des associations possibles entre la table des clients et celle des commandes. Nous avons également déterminé avec certitude qu'il ne pouvait pas y avoir de commandes sans client. Cependant il est tout à fait probable qu'il existe des clients n'ayant passé aucune commande. Conscient de cela, et au vu des points examinés précédemment, nous en arrivons à la conclusion logique que nous avons 2 clients n'ayant pas passé de commandes.

Étant donné le résultat que nous avons obtenu à l'heure actuelle, comment faire pour obtenir les quatre-vingt-onze lignes que nous avions au départ ? Le problème est ici le type de jointure qui est une jointure équivalente. Les associations ne sont que les associations permettant de joindre deux champs identiques.

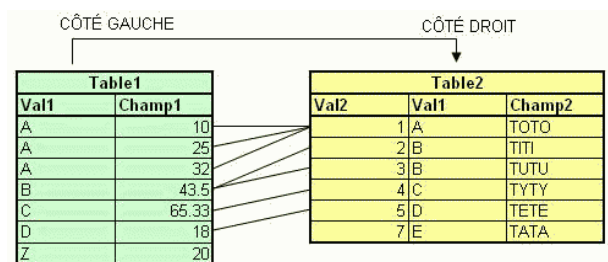
### 3.5. Les types de jointures

Il existe plusieurs types de jointure sous ACCESS :

1. La jointure équivalente
2. La jointure Gauche
3. La jointure Droite

Examinons la réaction de chacune d'entre elles.

Imaginons deux tables dans la plus simple expression : quelques champs, pas de clé primaire, pas d'index. Le champ de liaison est un champ de type texte un seul caractère. Traçons une relation du champ de liaison de la table un vers le champ de liaison de la table deux. Nous dirons ainsi que la table de gauche est la table un, et que la table droite est la table deux. En effet, le côté gauche est le point d'origine de la relation, et le côté droit, de son point d'arrivée. (Cette notion de gauche et droite n'a aucun rapport avec la position des tables dans la requête. Où seraient la gauche et la droite si les deux tables étaient l'une sur l'autre ? Tout dépend du tracé de la relation. Le côté gauche de la jointure est le point d'origine de la relation ; le côté droit son point d'arrivée.) Nous obtenons le schéma ci-dessous :



Lorsque la requête va s'exécuter, elle cherchera à travailler sur le tableau résultant de cette relation à savoir l'association de tous les champs équivalents. Cela donnera la table suivante :

Table de Résultats				
Val1	Champ1	Val2	Val1	Champ2
A	10	1	A	TOTO
A	25	1	A	TOTO
A	32	1	A	TOTO
B	43.5	2	B	TITI
B	43.5	3	B	TUTU
C	65.33	4	C	TYTY
D	18	5	D	TETE

Nous voyons bien que certains points de ces deux tables ne sont pas pris en compte dans la table de résultats. Par exemple comment faire pour qu'apparaissent dans la table de résultats tous les champs de la table un, donc de la table gauche. Il faut changer le type de jointure et la transformer en une jointure gauche. La table résultant d'une requête avec une jointure gauche sera semblable à la table suivante :

Table de Résultats				
Val1	Champ1	Val2	Val1	Champ2
A	10	1	A	TOTO
A	25	1	A	TOTO
A	32	1	A	TOTO
B	43.5	2	B	TITI
B	43.5	3	B	TUTU
C	65.33	4	C	TYTY
D	18	5	D	TETE
Z	20	NULL	NULL	NULL

Dans ce type de jointure nous obtenons une jointure équivalente plus tous les enregistrements orphelins (qui n'ont pu être liés) de la table de gauche, associés à des champs NULL.

Qu'en est-il de la jointure droite ? Il s'agit de l'inverse de la jointure gauche. Si la jointure que nous avons mise en oeuvre était une jointure droite la table résultant d'une telle requête serait la suivante :

Table de Résultats				
Val1	Champ1	Val2	Val1	Champ2
A	10	1	A	TOTO
A	25	1	A	TOTO
A	32	1	A	TOTO
B	43.5	2	B	TITI
B	43.5	3	B	TUTU
C	65.33	4	C	TYTY



D	18	5	D	TETE
NULL	NULL	7	E	TATA

Cette fois, ce sont des champs de la table gauche que nous ne voyons plus. La jointure droite correspond à une jointure équivalente à laquelle on ajoutera tous les champs orphelins de la table droite, liés à des champs NULL.

### 3.6. Cas d'une requête avec jointure externe

Modifions le type de jointure.

Pour cela, il suffit de faire un double-clic sur la relation. Apparaît alors la fenêtre suivante :

Les 3 points mentionnés sont dans l'ordre de l'exposé ci-dessus :

1. 1 = Jointure Equivalente
2. 2 = Jointure Gauche
3. 3 = Jointure Droite

Choisissons le point N°2 (la jointure gauche) de manière à pouvoir avoir l'ensemble des clients, avec et sans commandes. Dans la grille d'Access, le résultat sera le suivant :

Dans notre cas si nous voulons visualiser tous les clients, même ceux n'ayant pas passé de commandes, il faudra prendre la jointure gauche. En effet, la relation ayant été tracée depuis la table clients vers la table commandes, la table clients est la table de gauche. Puisque nous voulons tous les clients, il faudra faire une jointure gauche (Rappel : pour modifier le type de jointure, il suffit de faire un double-clic au milieu de la relation).

**Quel est le résultat obtenu ?**

Nous obtenons cette fois les quatre-vingt-onze lignes attendues.

**Que pouvons-nous en déduire ?**

Que nous avons fini par récupérer les enregistrements de la table clients qui ne sont pas liés à la table commandes, tout en conservant chaque occurrence des clients ayant des commandes.

### 3.7. Cas d'une requête de non-correspondance

Maintenant que nous avons réussi à récupérer les 91 clients du départ la question qui se pose est : "peut-on ne récupérer que les clients n'ayant pas passé de commandes ?".

Dans notre cas nous allons descendre dans notre requête, le Code Client qui est la clé étrangère de la table commande.

**Quel est le résultat obtenu ?**

Requête1 : Requête Sélection

	Société
▶	FISSA Fabrica Inter. Salchichas S.A.
	Paris spécialités
*	

Enr.: 1 sur 2

Nous venons de créer une requête de non correspondance.

Cette petite série d'exercices vous aura permis certainement de toucher du doigt la subtilité des types de jointure, et l'importance des relations dans une requête.

1. Lorsqu'une requête est faite sur une seule table, il faut faire attention aux champs sur lesquels on demande un regroupement;
2. Notez que pour identifier un enregistrement il est toujours préférable de faire apparaître la clé primaire;
3. Lorsqu'une requête est faite sur plusieurs tables, il faut, sauf cas exceptionnel, veiller à ce que toutes les tables soient liées entre elles. En cas contraire nous aurions un produit cartésien;
4. Par défaut les jointures sont équivalentes. Cela signifie que certains enregistrements peuvent "disparaître". Il faut changer le type de jointure pour qu'ils puissent réapparaître;
5. Sur ACCESS, les types de jointure sont : équivalente (par défaut), gauche ou droite. On ne peut pas, en une seule requête, récupérer les champs non joints des deux tables.

