

À quoi servent les relations entre les tables Access et comment les créer ?

Lorsque vous créez une base de données complexe, il vous sera fortement utile d'utiliser les relations et donc d'en comprendre leur fonctionnement. Les relations entre les tables permettent d'éviter les redondances et donc de réécrire les informations plusieurs fois et surtout d'éviter les erreurs.

Par exemple, si j'utilise Access pour créer mes factures, je peux créer une *Table Clients* où je saisis toutes les informations concernant chaque client et je créerai une *Table Factures* où je ne saisis que les informations concernant la facture elle-même comme son *Numéro*, sa *Date d'émission* et sa *Description* (quantité, prix unitaire, montant...) et je n'aurais pas besoin de ressaisir à chaque facture les informations du client grâce à la relation que j'aurais créée entre les deux tables et je n'aurais pas de doublon.

Rappel des différents types de relations sur Access

Pour bien comprendre le fonctionnement des relations, voici un petit rappel.

Il existe 3 types de relations (on peut également appeler ces relations "cardinalité") :

- 1 à 1 (1-1) ;
- 1 à plusieurs (1-N) ;
- Plusieurs à plusieurs (N-N).

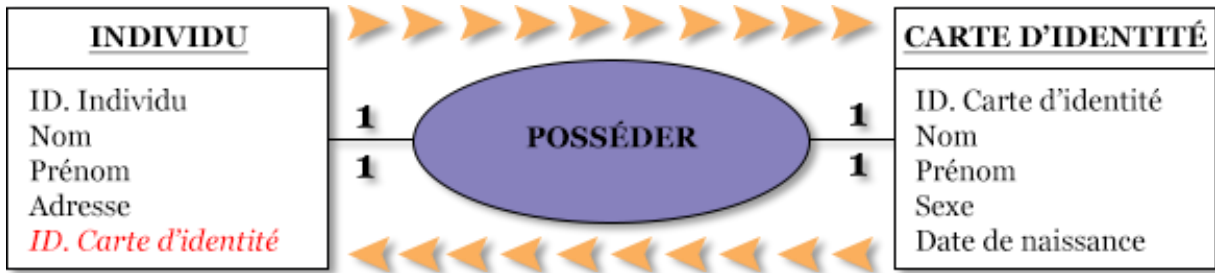
N désigne le *symbole infini* (∞) dans le logiciel Access.

Exemples de relations Access

Nous allons prendre 3 exemples pour mieux comprendre ces 3 types de relations.

À gauche et à droite nous retrouvons les tables avec, en titre, leur nom suivi de leurs champs (dans ces exemples, les champs ont été simplifiés, mais on peut bien sûr en avoir plus).

Au centre, nous utilisons un verbe qui peut être utilisé dans la relation entre les 2 tables.



Dans cet exemple, on peut dire qu'un *Individu* possède une seule *Carte d'identité*. Dans le sens inverse, une *Carte d'identité* ne peut être possédée que par un seul *Individu*.

La relation est donc 1 à 1.



Ici, on peut dire qu'un *Client* peut recevoir plusieurs *Factures*. À l'inverse, une facture ne peut être reçue que par un seul *Client*.

La relation est donc 1 à plusieurs.



Enfin, dans cet exemple, nous pourrions dire qu'un *Étudiant* peut suivre plusieurs *Cours*, mais aussi qu'un *Cours* peut être suivi par plusieurs *Étudiants*.

La relation est donc plusieurs à plusieurs.

À partir du moment où la réponse est différente de 1, je retiens la réponse *plusieurs*.

Maintenant, nous allons mettre en application ce que nous venons de voir pour créer une relation sur Access.

Création d'une relation sur Access

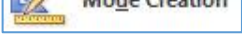
Pour l'exemple, nous allons prendre une *Table Clients* et une *Table Factures* comme dans le schéma n°2 où la relation est *1 à plusieurs*. Vous trouverez une base de données comprenant ces 2 tables, au bas de l'[article](#), afin de mettre en pratique ce tutoriel si vous le souhaitez.


Création de la clé primaire

Pour créer des relations, je dois d'abord **avoir un champ en commun** dans mes 2 tables.

Pour cela, je vais créer un champ *N° client* dans ma *Table Factures* ayant exactement le même nom que le champ *N° client* de la *Table Clients*. Le même nom n'est pas obligatoire, mais le même type oui.

J'ouvre ma *Table Factures* et je clique soit directement sur **Affichage**  , soit sur la

flèche sous le bouton **Affichage** pour passer en **Mode Création**  . Je peux

placer mon nouveau champ où je le souhaite, à la fin ou sous mon *N° de facture* par exemple. Si je souhaite l'ajouter sous mon *N° de facture*, il suffit de cliquer droit sur *Date d'émission* (qui est le champ suivant) et de cliquer sur **Insérer des lignes**  .

Je nomme mon champ *N° client*, dans ce cas, je ne peux pas choisir le *Numéroauto* comme type de champ puisque qu'Access n'accepte qu'un seul type *Numéroauto* par table et que *N° facture* possède déjà ce type. Je choisis donc *Numérique*. Ce champ est une **clé étrangère**, c'est-à-dire **une clé primaire venant d'une autre table**.

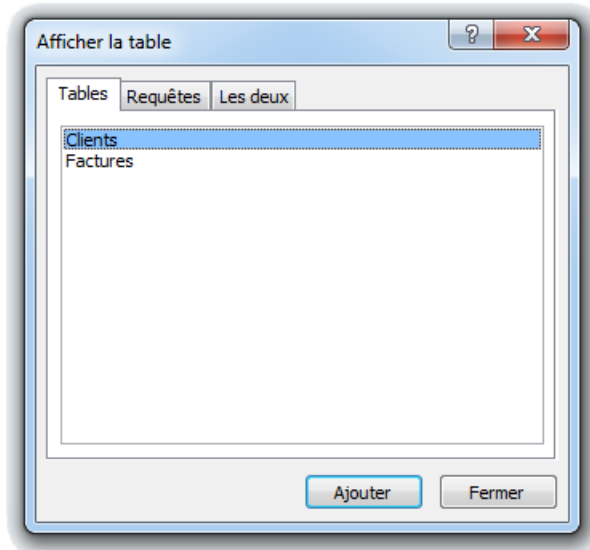
Pour savoir quelle table recevra la clé étrangère, il faut retenir que c'est la clé primaire du côté *1* qui sera reprise dans la table du côté *plusieurs*. Comme j'avais indiqué qu'un client peut recevoir plusieurs factures, ma clé primaire de ma *Table Clients* sera reprise dans ma *Table Factures*.

Pour créer une relation, il est important d'avoir fermé les tables, auquel cas, Access nous le signalera par la suite. Donc, une fois mon nouveau champ créé, je referme ma table par un clic droit sur son nom dans les onglets. On me demande si je souhaite enregistrer, je clique sur **Oui**.

Création de la relation via la fenêtre Relations



Ensuite, je vais dans l'onglet **Outils de base de données** et je clique sur **Relations**.




Dans la boîte de dialogue qui s'affiche, je reste sur l'onglet *Tables* et double-clique sur chacune des 2 tables ou je clique sur chaque table et clique sur **Ajouter**.




La même boîte de dialogue peut apparaître en cliquant sur **Afficher la table**.

Une fois mes 2 tables présentes sur mon écran, je clique sur **Fermer**. Je peux réduire ou agrandir mes tables mais aussi les déplacer sans aucune incidence sur mes relations, cela permet juste une meilleure visualisation. Pour information, il ne faut jamais faire apparaître

2 fois la même table ou requête sur l'espace des relations, car il pourrait y avoir des problèmes dans la base de données par la suite.

Clients	
	N° client
	Civilité
	Prénom
	Nom
	Adresse
	CP
	Ville
	Téléphone
	Email
	Date de démarrage

Factures	
	N° de facture
	N° client
	Date d'émission
	Mode de paiement
	Désignation
	Quantité
	Prix unitaire
	Total HT
	TVA
	Total TTC

Pour créer ma relation, j'ai juste à me placer sur *N° client* de la *Table Clients* et à le glisser vers *N° client* de la *Table Factures*. Mon curseur doit faire apparaître un carré blanc avec une croix pour m'indiquer que je peux lier ces 2 tables.

Dans la boîte de dialogue qui s'affiche, j'ai bien mon champ *N° client* de la *Table Clients* et *N° client* de la *Table Factures* qui vont être liés. Je vais cocher *Appliquer l'intégrité référentielle*.

Principe de l'intégrité référentielle sur Access

L'**intégrité référentielle** permet à Access de garantir que les relations entre les enregistrements dans les tables liées sont valides et que vous n'allez pas effacer ou modifier par erreur des données liées. Ce qui permet de créer une relation *forte* entre ces tables.

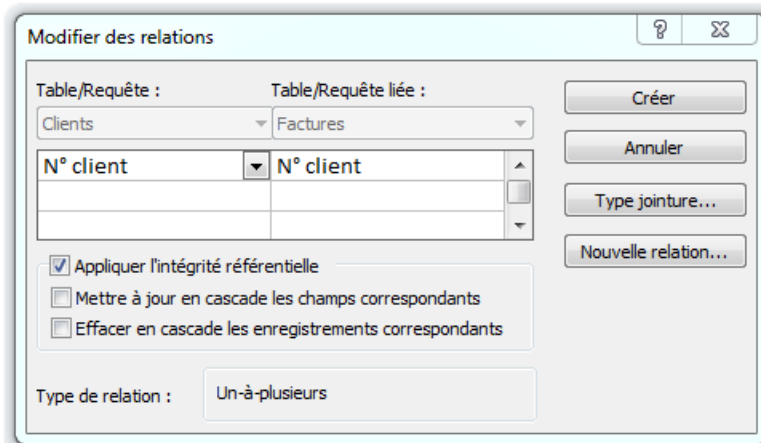
Pour appliquer cette intégrité référentielle :

- Les tables doivent posséder une clé primaire ou l'option d'index sans doublons ;
- Les tables ne doivent pas contenir de doublons (par exemple si plusieurs clients possèdent le même numéro de client, l'intégrité référentielle est impossible) ;
- Le champ en commun ne doit pas être vide ;
- Les tables doivent contenir les mêmes équivalences (si vous avez 5 numéros de produits dans votre *Table Factures* alors que vous n'en avez que 4 dans votre *Table Produits*, l'intégrité référentielle n'est pas possible) ;
- Les tables que nous souhaitons relier ne doivent pas être ouvertes dans les onglets.

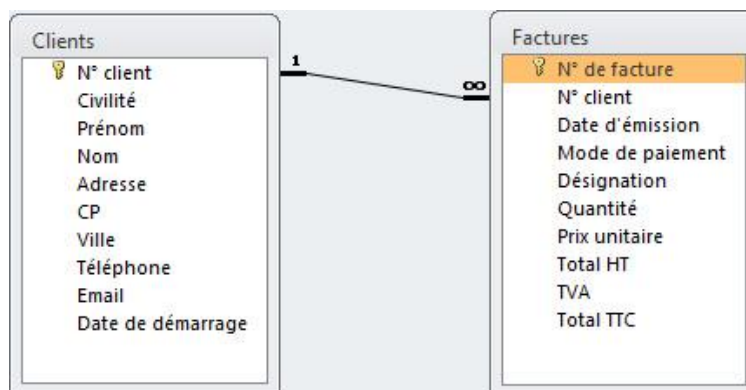
Le concept de l'intégrité référentielle est un peu complexe, mais, pour simplifier, c'est une bonne chose de l'appliquer, car elle permet de **synchroniser les informations** et **d'éviter les erreurs de gestion d'informations**.


Cependant, Access peut ne pas l'accepter pour les raisons évoquées précédemment.

Je ne coche pas *Mettre à jour en cascade les champs correspondants* et *Effacer en cascade les enregistrements correspondants*, mais cela permettrait de mettre à jour les tables si l'on modifie ou supprime un enregistrement dans les tables liées.



Je clique sur **Créer**. Je constate que ma relation est créée et que j'ai bien une relation Un-à-plusieurs.



Je peux maintenant fermer l'espace de création des relations en cliquant sur **Fermer**  et j'accepte d'enregistrer.

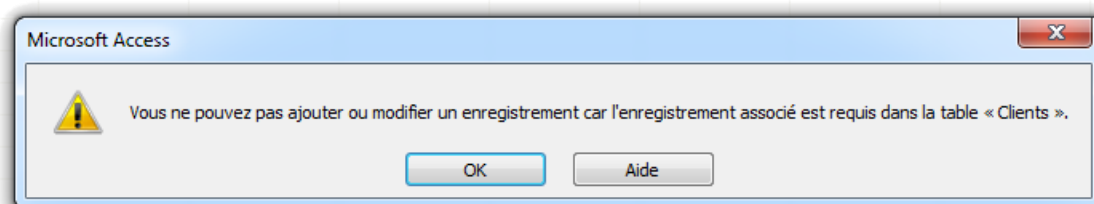
Il me suffit maintenant de renseigner le N° client dans ma *Table Factures* (pour faire correspondre chaque client et facture).

Visualisation des données reliées

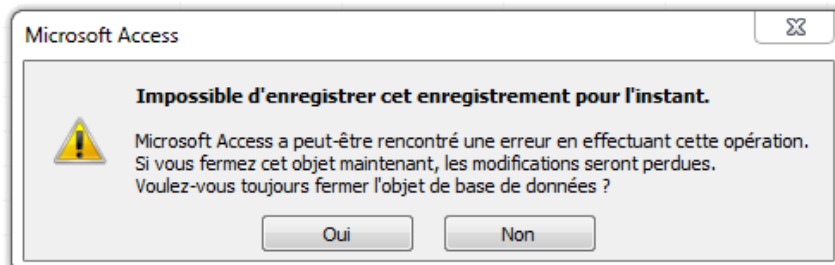
Grâce à cette relation, lorsque je retourne dans ma *Table Clients*, j'ai une nouvelle colonne devant chaque enregistrement avec des signes plus (+). Lorsque je clique dessus, je peux voir les factures concernant chaque client.

N° client	Civilité	Prénom	Nom	Adresse	CP	Ville	Téléphone	Email	Date de démarrage
1	Mme	Emma	MARTIN	2, rue de l'Église	95690	NESLES-LA-VALLÉE	01 25 68 97 45	emma.martin@yahoo.fr	12/01/2012
2	Mlle	Lola	DUBOIS	12, rue du Moulin	75001	PARIS	01 36 98 42 30	lola-dubois@gmail.com	23/01/2012
N° de factur	Date d'émis	Mode de pa	Désignation	Quantité	Prix unitaire	Total HT	TVA	Total TTC	Cliquer pour
1	02/10/2012	Virement	Produit B	2	12,50 €	25,00 €	4,90 €	29,90 €	
12	22/02/2013	Espèces	Produit C	2	11,70 €	23,40 €	4,59 €	27,99 €	
*	(Nouv.)								
3	M.	Nathan	COLLET	5, grande Rue	93000	BOBIGNY	01 73 29 84 16	nathan_collet@msn.com	02/02/2012

Grâce à l'intégrité référentielle, je ne peux plus créer de factures avec un N° client qui n'existe pas, car, si dans ma *Table Factures*, j'ajoute un enregistrement ayant pour N° client 25 (qui n'existe pas dans ma *Table Clients*) et que je souhaite enregistrer ou fermer ma table, Access m'affiche ce message "**Vous ne pouvez pas ajouter ou modifier un enregistrement car l'enregistrement associé est requis dans la table « Clients ».**" :

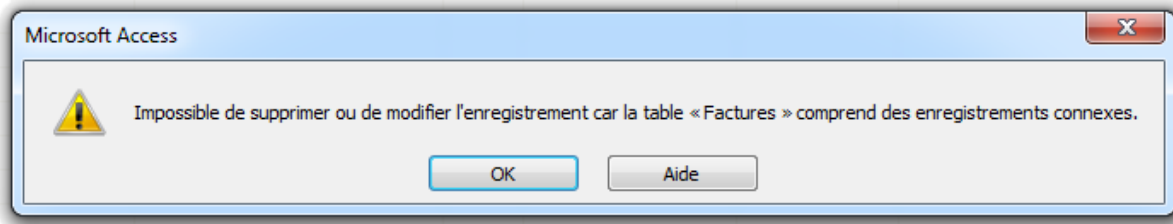


Puis, après avoir cliqué sur **Ok**, Access vous affichera le message "**Impossible d'enregistrer cet enregistrement pour l'instant.**" :

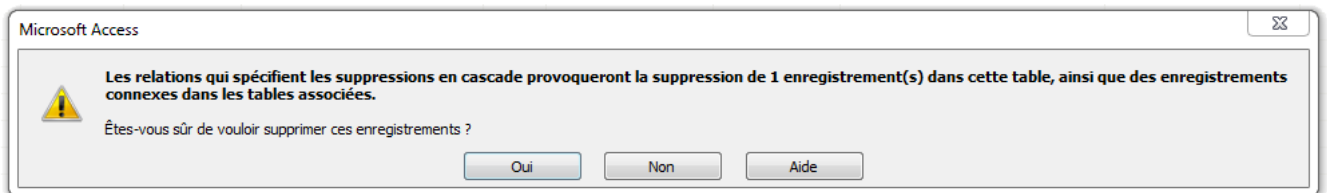


De même, je ne peux plus supprimer un client de la *Table Clients* si une facture le concernant apparaît dans la *Table Factures*. Auquel cas, j'aurais ce message "**Impossible de supprimer**

ou de modifier l'enregistrement car la table « Factures » comprend des enregistrements connexes" :



Par contre, si vous avez coché *Mettre à jour en cascade les champs correspondants* et *Effacer en cascade les enregistrements correspondants* lors de la création de la relation, vous aurez ce message "Les relations qui spécifient les suppressions en cascade provoqueront la suppression de 1 enregistrement(s) dans cette table, ainsi que des enregistrements connexes dans les tables associées." :



Tutoriel réalisé avec Access 2013

[Voir la version vidéo de cet article](#)