

- Vente de produits par un magasin : un magasin vend un ou plusieurs produits et un produit est vendu dans un ou plusieurs magasins (chaque instance de l'ensemble d'entités **magasin** lui correspond une ou plusieurs instances de l'ensemble d'entités **produit**, chaque instance de l'ensemble d'entités **produit** lui correspond une ou plusieurs instances de l'ensemble d'entités **magasin**).

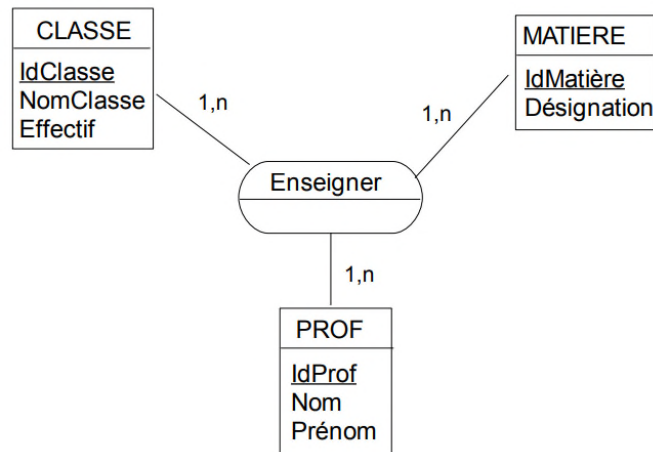
- Achat d'un produit par un client : un client achète un ou plusieurs produits, et un produit peut être acheté par un ou plusieurs clients (chaque instance de l'ensemble d'entités **produit** lui correspond une ou plusieurs instances de l'ensemble d'entités **client**, chaque instance de l'ensemble d'entités **client** lui correspond une ou plusieurs instances de l'ensemble d'entités **produit**).
- Produire un produit par un fournisseur : un produit est produit chez un ou plusieurs fournisseurs et un fournisseur est spécialisé dans la production d'un seul et unique produit (un produit est produit par différent fournisseurs et un fournisseur produit un seul produit). (Chaque **instance** de l'ensemble d'entités **produit** lui correspond une ou plusieurs instances de l'ensemble d'entités **fournisseur**, chaque instance de l'ensemble d'entités **fournisseur** lui correspond une et une seule instance de l'ensemble d'entités **produit**).
- Réalisation d'une commande d'un produit pour un magasin par un fournisseur : Un produit est fourni par un seul fournisseur pour un magasin (pour une instance de l'ensemble d'entités **produit**, et une instance de l'ensemble d'entités **fournisseur** nous avons une ou plusieurs instances l'ensemble d'entités **magasin**, pour une instance de l'ensemble d'entités **produit**, et une instance de l'ensemble d'entités **magasin** nous avons une et une seule instance de l'ensemble d'entités **fournisseur**, pour une instance de l'ensemble d'entités **fournisseur** et une instance l'ensemble d'entités **magasin** nous avons une et une seule instance de l'ensemble d'entités **produit** puisque un fournisseur ne peut produire qu'un seul type de produit).

Remarque :

La dimension d'une association est le nombre d'entités qu'elle relie :

- **Les associations binaires** : ce sont les associations qui relient deux entités. Elles sont donc de dimension 2.
- **Les associations réflexives (ou "unaires")** : ce sont les associations reliant une entité à elle même. Elles sont donc de dimension 1.
- **Les associations N-aire** : un modèle conceptuel peut donc contenir des associations de dimension supérieure à deux (un ensemble d'entités reliée à N autres ensembles entités).

Exemple d'une association N-aire :



Dans la traduction d'un modèle conceptuel en modèle relationnel, les associations N-aires deviennent une relation de même nom, dont les attributs sont les N clefs étrangères des N entités reliées à l'association et dont la clef primaire est formée de ces N attributs.

Cependant, il est préférable et très recommandé d'éviter l'utilisation des associations N-aires, en ayant en tête que toute association N-aire peut s'exprimer par une combinaison d'associations binaires.

Nous allons ainsi introduire un nouvel ensemble d'entités COMMANDE qui va remplacer l'association « Réaliser » (éclater l'association « Réaliser ») et qui sera relié aux trois ensembles d'entités (PRODUIT, MAGASIN, FOURNISSEUR) avec des associations binaires simples. Ainsi, nous introduisons les relations suivantes :

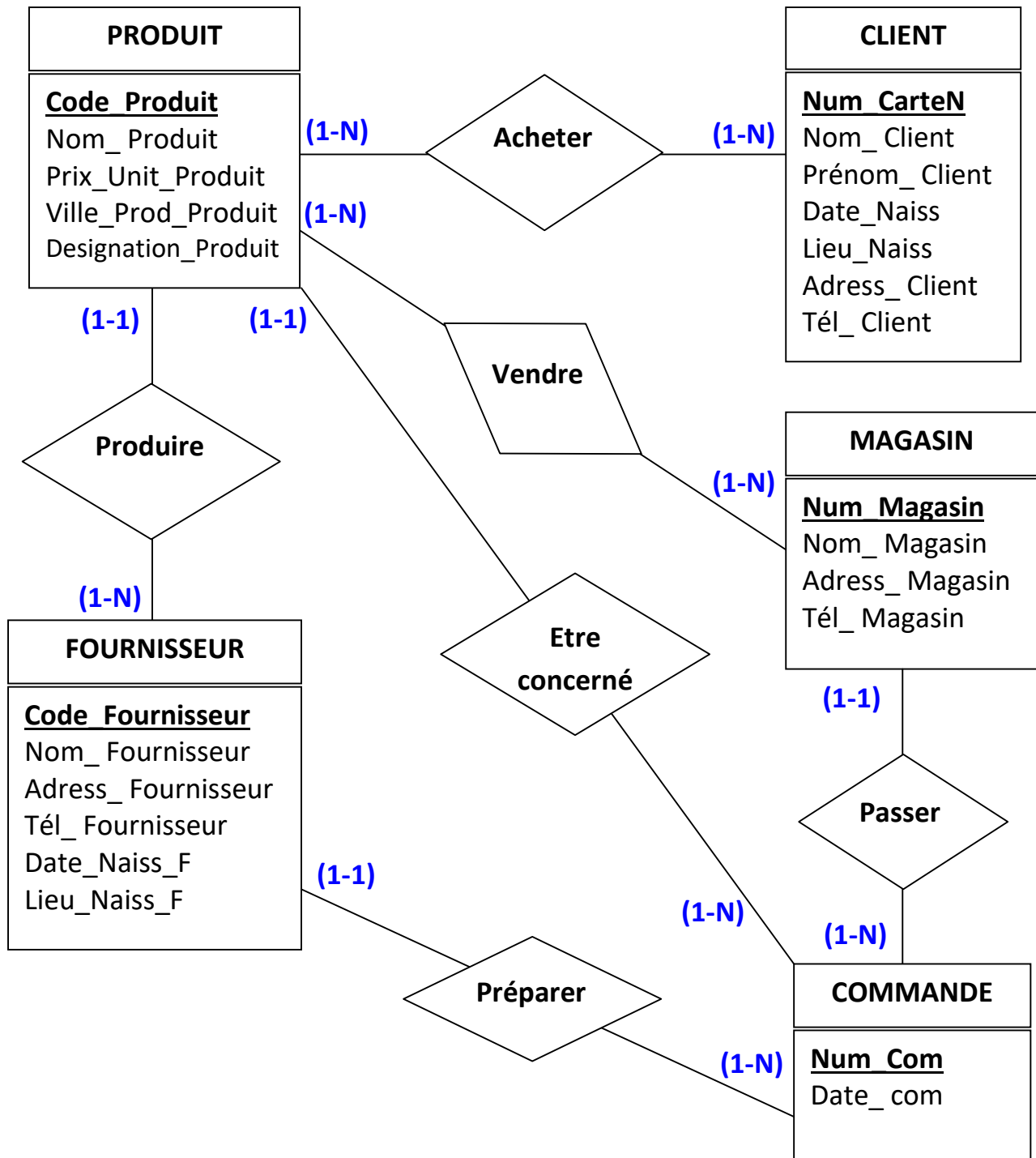
Passer une Commande par un magasin : un magasin passe une ou plusieurs commandes et une commande est liée à un et un seul magasin (chaque instance de l'ensemble d'entités **magasin** lui correspond une ou plusieurs instances de l'ensemble d'entités **commande**, chaque instance de l'ensemble d'entités **commande** lui correspond une seule et unique instance de l'ensemble d'entités **magasin**).

Préparer une Commande par un fournisseur : un fournisseur prépare une ou plusieurs commandes et une commande est préparée par un et un seul fournisseur (chaque instance de l'ensemble d'entités **fournisseur** lui correspond une ou plusieurs instances de l'ensemble d'entités **commande**, chaque instance de l'ensemble d'entités **commande** lui correspond une seule et unique instance de l'ensemble d'entités **fournisseur**).

Etre concerné par une Commande : un produit est concerné par une ou plusieurs commandes et une commande concerne un et un seul produit puisque une commande est préparée par un et un seul fournisseur et un

fournisseur est spécialisé dans la production d'un seul et unique produit (chaque instance de l'ensemble d'entités **produit** lui correspond une ou plusieurs instances l'ensemble d'entités **commande**, chaque instance de l'ensemble d'entités **commande** lui correspond une seule et unique instance de l'ensemble d'entités **produit**).

On obtient ainsi le nouveau modèle conceptuel suivant :



Le passage du modèle entité-association au schéma de base de données relationnelle :

Cas 1 :

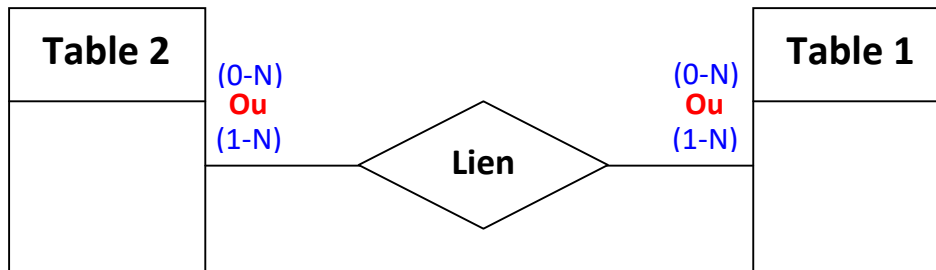


Table 1(Clé Table1, attribut1_1, attribut1_2,, attribut1_N)

Table 2(Clé Table2, attribut2_1, attribut2_2,, attribut2_M)

Lien (Clé Table1, Clé Table2, attribut3_1, attribut3_2,, attribut3_P)

Cas 2 :

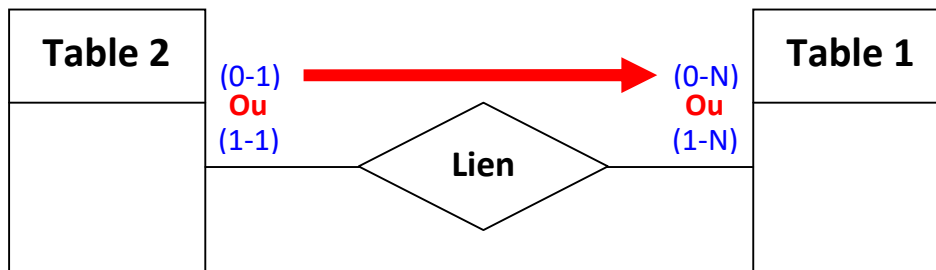


Table 1(Clé Table1, attribut1_1, attribut1_2,, attribut1_N, #Clé_Table2)

(Clé_Table2 est une clé étrangère qui référence la Table 2)

Table 2(Clé Table2, attribut2_1, attribut2_2,, attribut2_M)

Cas 3 :

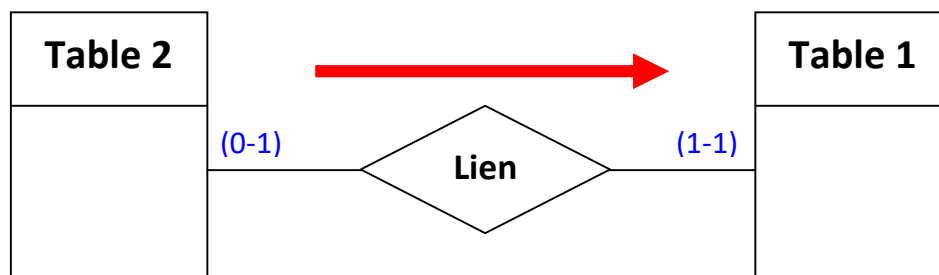


Table 1(Clé Table1, attribut1_1, attribut1_2,, attribut1_N, #Clé_Table2)

(Clé_Table2 est une clé étrangère qui référence la Table 2)

Table 2(Clé Table2, attribut2_1, attribut2_2,, attribut2_M)

(Le sens contraire est aussi correct)

A partir de ces règles citées précédemment et le modèle entité-association on obtient les relations suivantes :

MAGASIN (Num_Magasin, Nom_ Magasin, Adress_ Magasin, Tél_ Magasin)

PRODUIT (Code Produit, Nom_ Produit, Designation_Produit, Prix_Unit_Produit, Ville_Prod_Produit)

FOURNISSEUR (Code Fournisseur, Nom_ Fournisseur, Adress_ Fournisseur, Date_Naiss_F, Lieu_Naiss_F, Tél_ Fournisseur, #Code_Produit)

CLIENT (Num CarteN, Nom_ Client, Prénom_ Client, Date_Naiss, Lieu_Naiss, Adress_ Client, Tél_ Client)

COMMANDE (Num_Com, Date_com, #Num_Magasin, # Code_Produit, #Code_Fournisseur)

Vendre (Num_Magasin, Code Produit, Quatité_Vendue)

Acheter (Num CarteN, Code Produit, Quatité_Achetée)

Nous avons vu dans le **TP1** comment créer une table et lui attribué une clé primaire (le champ Index que l'attribut clé est à **PRIMARY**), nous avons vu également que dans certains cas, une table peut avoir un attribut qui fait référence à une autre table (c'est ce que nous avons appelé une clé étrangère et qui prend la valeur **INDEX** dans le champ Index). Mais nous n'avons jamais vu

le cas d'une table qui comporte une clé composée (composée de plusieurs attributs) et encore moins, quand cette clé composée comporte des attributs qui font référence à d'autres tables, ce qui est très fréquent quand une association est transformée en une relation. C'est le cas de l'association Vendre qui s'est transformée (dans le modèle relationnel) en une relation qui comporte une clé composée des deux attributs « **Num_Magasin** » et « **Code_Produit** ». Ces deux attributs sont également des clés étrangères qui font référence respectivement aux tables « **MAGASIN** » et « **PRODUIT** ».

Dans le cas de la relation « Vendre », nous allons tout d'abord créer la table « Vendre » (comme nous avons l'habitude de créer n'importe quelle table) mais en spécifiant deux attributs clés comme montré dans la figure qui suit :

The screenshot shows the MariaDB Table Designer window for a table named 'Vendre'. The 'Structure' tab is active, displaying the following columns:

Nom	Type	Taille/Valeurs*	Valeur par défaut	Interclassement	Attributs	Null	Index
Num_Magasin_V	INT		Aucun(e)			<input type="checkbox"/>	PRIMARY
Code_Produit_V	INT		Aucun(e)			<input type="checkbox"/>	PRIMARY
Quantité_Vendue	INT		Aucun(e)			<input type="checkbox"/>	---

At the bottom, the 'Moteur de stockage' (Storage Engine) is set to 'InnoDB'.

Vous remarquerez que j'ai fait exprès de changer le nom des attributs « **Num_Magasin** » et « **Code_Produit** » à « **Num_Magasin_V** » et « **Code_Produit_V** » dans la table « Vendre ». Le but étant de savoir dans quelle table je suis quand je parle de l'attribut code produit (pour l'attribut **Code_Produit** je suis dans la table PRODUIT et pour l'attribut **Code_Produit_V** je suis dans la table Vendre) et numéro magasin (pour l'attribut **Num_Magasin** je suis dans la table MAGASIN et pour l'attribut **Num_Magasin_V** je suis dans la table Vendre).

Attention à ne pas cocher l'option **A.I** (AUTO_INCREMENT) pour les deux attributs « **Num_Magasin_V** » et « **Code_Produit_V** » (même s'ils sont auto-incrémentale dans les tables de références) car le premier tuple de la table Vendre peut engager, par exemple, le produit de code 214 et non pas forcément le produit de code 1.... !!!! C'est logique).

Faites bien attention à mettre le champ « Moteur de stockage » à InnoDB (et non pas à MyISAM) lors de la création de toutes vos tables (cela vous sera d'une grande utilité lors de l'élaboration des liaisons entre vos tables).

Après la création de la table Vendre, vous allez remarquer que la structure de la table comporte une clé primaire composée de deux colonnes (comme illustré dans la figure qui suit)

The screenshot shows the MariaDB interface for the 'vendre' table. The table structure is as follows:

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	Num_Magasin_V	int(11)			Non	Aucun(e)			Modifier, Supprimer, Plus
2	Code_Produit_V	int(11)			Non	Aucun(e)			Modifier, Supprimer, Plus
3	Quantité_Vendue	int(11)			Non	Aucun(e)			Modifier, Supprimer, Plus

The 'Index' tab is selected, showing the following index:

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
Éditer, Supprimer	PRIMARY	BTREE	Oui	Non	Num_Magasin_V, Code_Produit_V	0	A	Non	

Pour donner aux deux attributs « **Num_Magasin_V** » et « **Code_Produit_V** » de la table « Vendre » la caractéristique d'être en plus des clés étrangères qui font référence à d'autres tables, nous allons, pour chacun des deux attributs, sélectionner l'option « plus » et cliquer sur « Index » puis confirmer cette manipulation par « Ok »

The screenshot shows the MariaDB interface for the 'vendre' table. The 'Index' tab is selected, showing the following index:

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
Éditer, Supprimer	PRIMARY	BTREE	Oui	Non	Num_Magasin_V, Code_Produit_V	0	A	Non	

The 'Index' tab is selected, and the 'Plus' button is clicked for the 'Num_Magasin_V' column. The 'Index' option is selected from the dropdown menu.

Confirmer

Faut-il vraiment exécuter « ALTER TABLE `vendre` ADD INDEX(`Num_Magasin_V`); » ?

OK Annuler

Après validation, vous allez remarquer que devant chacun des attributs « **Num_Magasin_V** » et « **Code_Produit_V** » de la table « Vendre », il y'a deux clés (la clé jaune pour dire que l'attribut est une clé primaire et une clé grise pour dire que l'attribut est une clé étrangère). Vous allez remarquer également que le volet index comporte trois lignes, une ligne pour la clé primaire (composée de deux attributs) et deux lignes pour les deux clés étrangères.

#	Nom	Type	Interclassement	Attributs	Null	Valeur par défaut	Commentaires	Extra	Action
1	Num_Magasin_V	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
2	Code_Produit_V	int(11)			Non	Aucun(e)			Modifier Supprimer Plus
3	Quantité_Vendue	int(11)			Non	Aucun(e)			Modifier Supprimer Plus

Action	Nom de l'index	Type	Unique	Compressé	Colonne	Cardinalité	Interclassement	Null	Commentaire
Éditer Supprimer	PRIMARY	BTREE	Oui	Non	Num_Magasin_V 0 Code_Produit_V 0	A		Non	
Éditer Supprimer	Num_Magasin_V	BTREE	Non	Non	Num_Magasin_V 0	A		Non	
Éditer Supprimer	Code_Produit_V	BTREE	Non	Non	Code_Produit_V 0	A		Non	

Pour le moment, nous n'avons établi aucune relation entre la table Vendre et d'autres tables, nous avons simplement deux attributs qui vont nous permettre de relier la table Vendre à d'autres tables.

Il reste maintenant à relier la table « Vendre » aux deux tables « MAGASIN » et « PRODUIT » en faisant référence, à partir de la table Vendre, à :

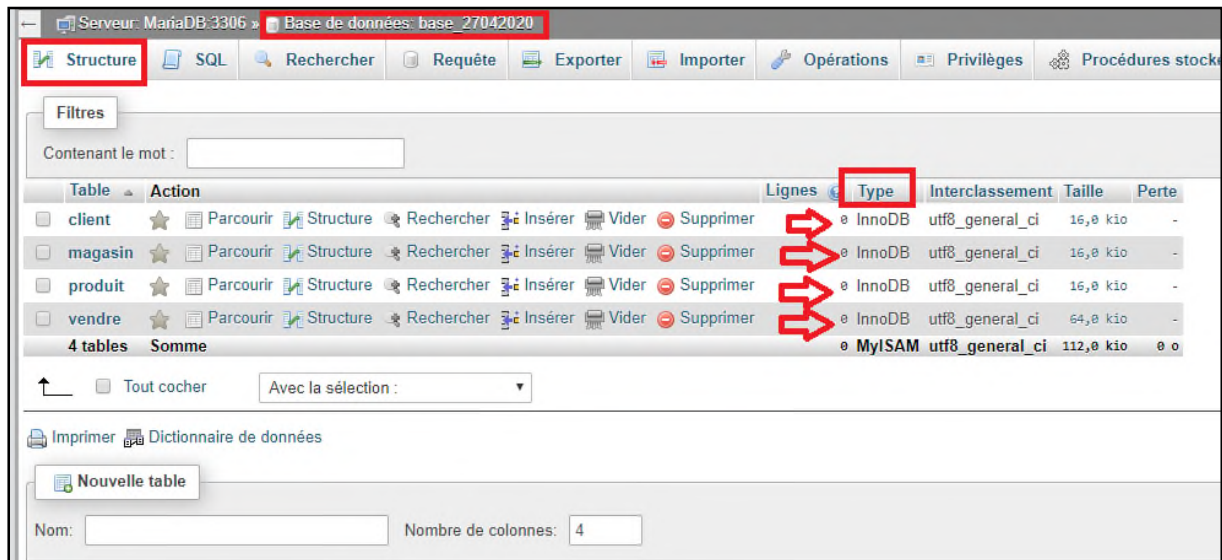
- la table « MAGASIN » par usage de la clé étrangère « **Num_Magasin_V** ».
- la table « PRODUIT » par usage de la clé étrangère « **Code_Produit_V** ».

Nous avons déjà vu comment relier deux tables à travers une clé étrangère dans la première séance de TP, mais pour ceux qui ne savent pas le faire, voici les étapes par lesquelles passées (et c'est valable pour toute les liaisons entre tables):

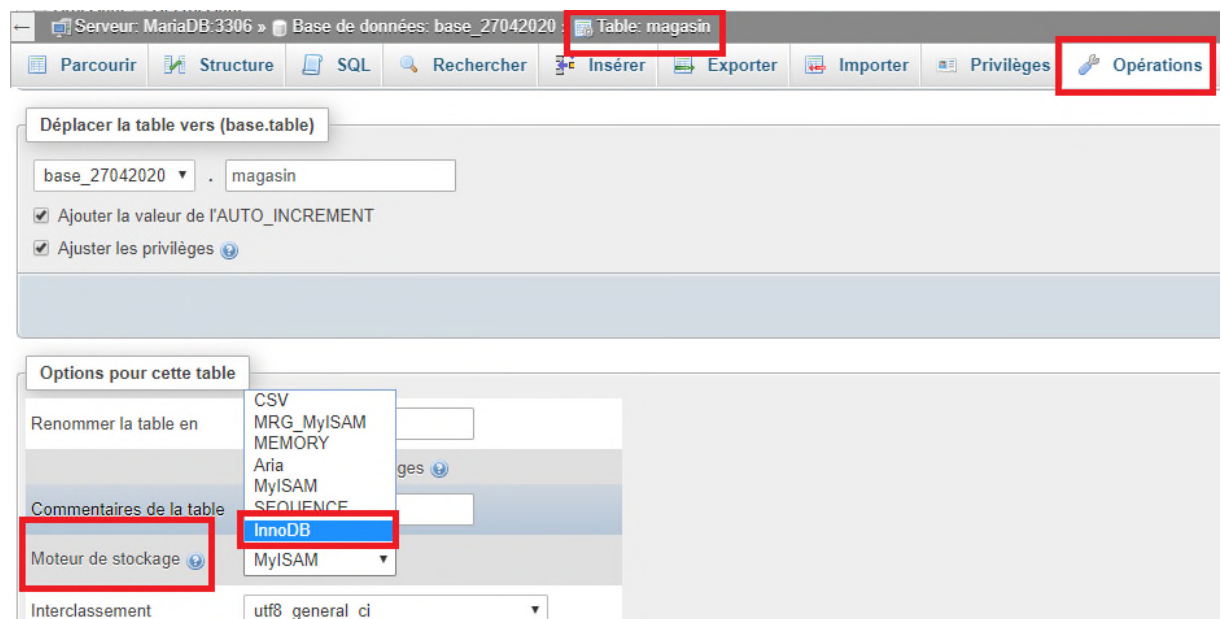
Nous voulons relier la table « Vendre » à la table « MAGASIN » et faire référence à la table « MAGASIN » par l'attribut « **Num_Magasin_V** » (qui se trouve dans la table Vendre), pour se faire, allez au champ **Structure** de la table **Vendre** et plus exactement le champ « **Vue relationnelle** ».

Si le champ « **Vue relationnelle** » du volet « **Structure** » de la table « Vendre » n'apparaît pas dans votre interface, cela signifie que le moteur de base de données définie pour la table « Vendre » n'est pas mis à « InnoDB » (il s'agit d'un réglage important à ne pas oublier lors de la création d'une table, surtout lorsqu'il s'agit de mettre en relation des tables. En effet, les relations entre

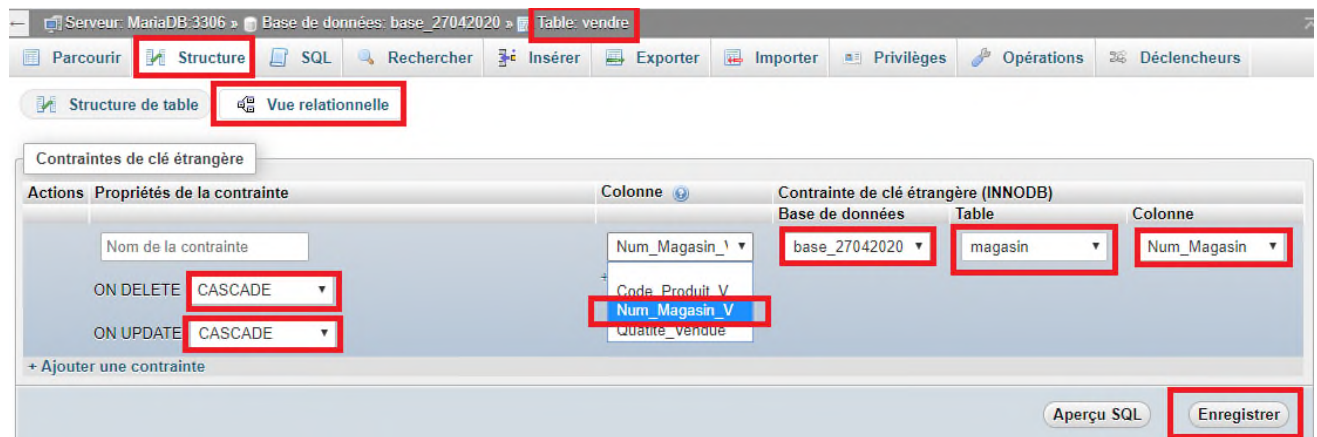
tables ne sont définies que si le moteur de stockage est à InnoDB. La gestion et la manipulation de ces liaisons ne peuvent pas être réalisées par le moteur de base de données MyISAM). Pour être sûr que toutes les tables créées ont le moteur de base de données InnoDB, allez sur la vue global de la base de donnée et vérifiez que la colonne type qui désigne le moteur de base de données de chacune des tables est à InnoDB.



Dans le cas où le moteur de base de données d'une table est mis à MyISAM, allez à la vue global de la table, volet « opérations », section « Moteur de stockage » et mettez la valeur du moteur de stockage à InnoDB, puis exécutez pour valider votre action. Vous devez sortir de phpMyAdmin puis ré-accéder de nouveau pour mettre à jour le « Moteur de stockage » utilisé.



Une fois ces prérequis terminés, nous pouvons poursuivre notre travail qui consiste à relier la table Vendre à la table « MAGASIN», et nous étions sur le champ « **Vue relationnelle** » du volet « **Structure** » de la table « Vendre » qui permet d'accéder à l'interface de gestion des relations de la table « Vendre » :

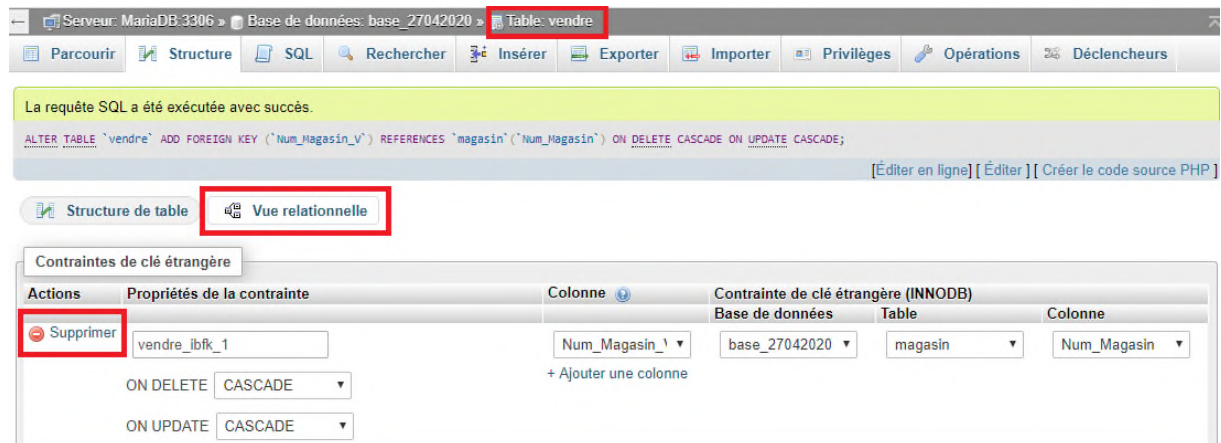


Dans la première liste déroulante « colonne » nous allons désigner le champ (ou l'attribut) qui appartient à la table « Vendre » et qui permet de référencer la table « MAGASIN», ce champ est la clé étrangère « **Num_Magasin_V** » ; dans le champ « base de données », nous allons désigner la base de données ou les tables ont été créées (en d'autres termes désigner la base sur laquelle on travaille), puis nous allons désigner la table référencée avec laquelle nous souhaitons établir cette relation (il s'agit de la table « MAGASIN » qu'on veut lier à la table Vendre). La seconde liste déroulante « colonne » désigne la clé primaire de la table « MAGASIN » qui permet de concrétiser cette liaison (et qui a été attribuée à la table Vendre comme clé étrangère pour permettre cette liaison).

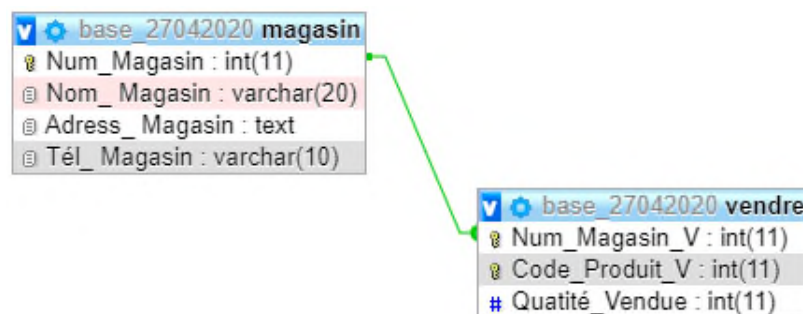
Les deux options ON DELETE/ ON UPDATE permettent de définir le comportement des tables pour la relation. Ces deux options (ON DELETE/ ON UPDATE) lorsqu'elles sont à CASCAD, elles permettent d'établir (ou de provoquer) des réactions en chaîne pour un changement établi sur l'une des deux tables liées. Pour comprendre la valeur CASCAD, on donne l'exemple de la suppression d'un magasin de la liste des magasins, cette suppression engendre automatiquement la suppression de toutes les ventes liées à ce magasin (c'est ce que nous appelons une suppression en cascade), de même, les traces de la suppression d'un produit de la liste de produit (suppression opérée dans la table produit) doivent se répercuter sur toutes les ventes liées à ce produit (malgré que j'ai donné cet exemple avant de faire la liaison entre la table PRODUIT et Vendre, mais pas grave dans la suite je vais faire cette liaison). Ainsi, pour déclencher des réactions en chaîne, nous mettons ces deux options de correspondances de données à CASCAD.

Une fois ces champs remplis, il ne reste qu'à sauvegarder cette nouvelle liaison et la concrétiser comme relation existante dans le volet de relation de la table Vendre.

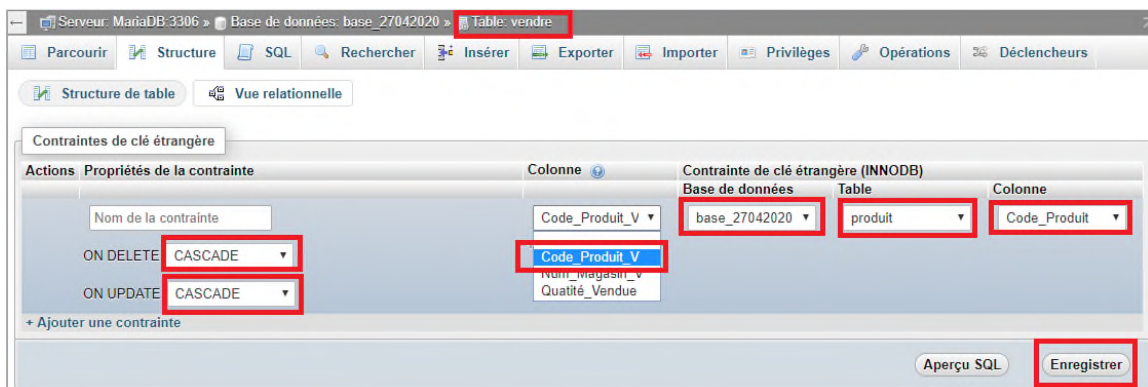
Dans le cas où vous voulez supprimer cette liaison, il suffit de faire supprimer (comme le montre la figure suivante) :



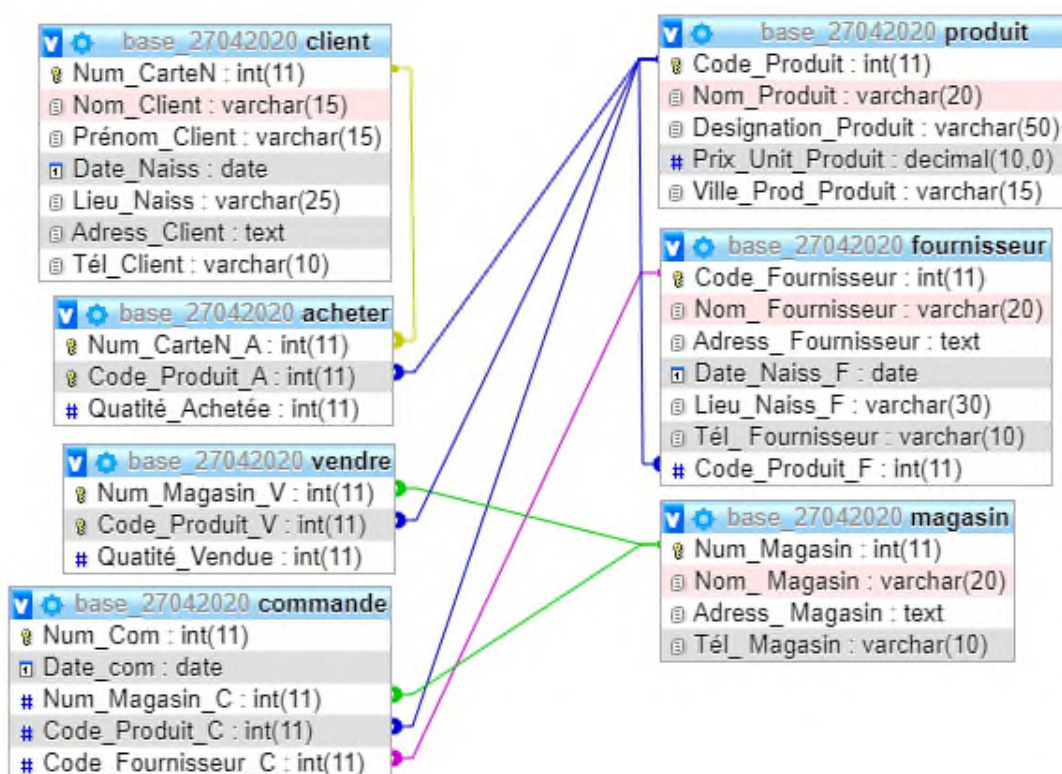
Pour vérifier si la liaison a été établie entre les deux tables « Vendre » et « MAGASIN », allez à la vue globale de la base de données, option « plus », puis concepteur, et vous allez voir que réellement il existe un lien entre ces deux tables (de ma part je ne fais affiché que ces deux tables, juste pour montrer le lien qui existe entre ces deux tables « Vendre » et « MAGASIN »)



Revenons maintenant à la vue de la table Vendre, champ **Structure** de la table **Vendre** et plus exactement le champ « **Vue relationnelle** », nous allons maintenant relier la table « Vendre » à la table « PRODUIT » et faire référence à la table « PRODUIT » par l'attribut « **Code_Produit_V** » (qui se trouve dans la table Vendre) :



Voilà, pour les autres liaisons, c'est la même chose. Le schéma final est le suivant :



Une dernière remarque pour conclure, dans le cas où une table TAB1 comporte une clé primaire composée (une clé comportant plusieurs attributs) et que cette table TAB1 est référencé par une table TAB2 (la clé de TAB1 est une clé étrangère dans TAB2), alors la clé étrangère dans TAB2 est une clé qui comporte plusieurs attributs, dans ce cas de figure, déclarez l'ensemble des attributs comportant la clé étrangère dans la table TAB2, puis mettre à chaque fois le champ Index à la valeur **INDEX** et au moment où vous aurez un message qui vous demande si vous avez un Index sur une colonne ou sur plusieurs colonnes, sélectionnez l'option index sur plusieurs colonnes (dans le cas contraire, il va considéré que chaque attribut est un index indépendant des autres, alors que la clé étrangère comportent plusieurs colonnes à la fois, donc plusieurs attributs réunis dans la même la clé étrangère).