

1. Introduction.....	2
2. Création d'une requête.....	2
3. Définition des critères de sélection.....	5
3.1 Opérateurs	5
3.2 Les Fonctions.....	6
3.3 Plusieurs critères portant sur des champs différents.....	7
3.4 Requête paramétrée	8
4. Les requêtes multitables	9
5. Les fonctions de regroupement.....	10
5.1 Les opérations.....	11
5.2 Quelques exemples.....	12
6. Les requêtes d'analyse croisée	13
7. Les requêtes ACTION.....	14
7.1 Les requêtes Création.....	14
7.2 Les requêtes Ajout	15
7.3 Les requêtes Mise à Jour.....	16
7.4 Les requêtes Suppression.....	17

LES REQUETES

1. Introduction

Les requêtes vont servir à afficher uniquement certaines données contenues dans les tables selon certains critères. Elles peuvent aussi faire des calculs sur vos données, ainsi que créer des sources de données pour les formulaires, les états ou même d'autres requêtes (on peut faire une requête sur le résultat d'une autre requête). Elles servent encore à modifier des tables existantes ou à en créer des nouvelles.

Il existe différents types de requêtes que nous allons détailler après :

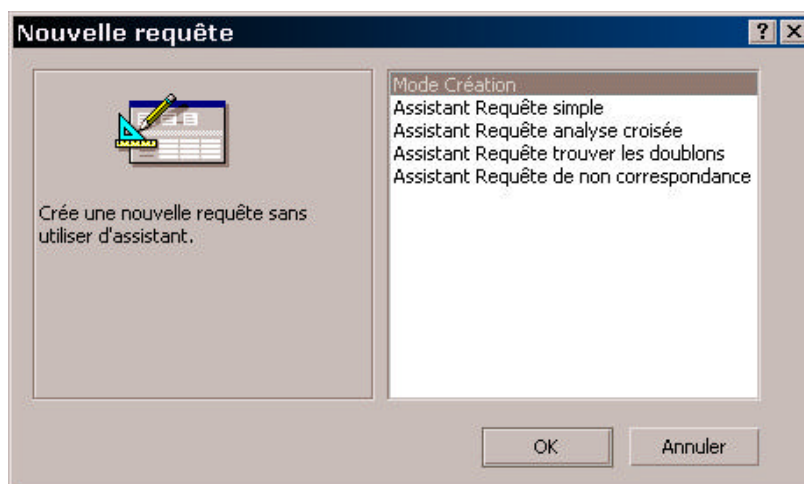
- ?? **La requête sélection** : C'est celle qu'on utilisera le plus souvent. Elle permet de sélectionner des enregistrements, de faire des calculs et des regroupements. Elles ressemblent beaucoup aux filtres, mais permettent, en plus, de travailler sur plusieurs tables simultanément.
- ?? **La requête d'Analyse croisée** : Cette requête présente ses résultats sous forme de tableau (de type Excel). On l'utilisera pour comparer des valeurs, dégager des tendances.
- ?? **La requête de Création de table** : Cette requête crée une table à partir des données qu'elle a extraites dans une ou plusieurs autres tables.
- ?? **La requête Mise à Jour** : Cette requête modifie le contenu d'un ou plusieurs champs d'une ou plusieurs tables. C'est le moyen le plus efficace pour mettre à jour un grand nombre d'enregistrements en une seule opération.
- ?? **La requête Ajout** : Cette requête ajoute les données qu'elle a extraites à la fin d'une table déjà existante.
- ?? **La requête Suppression** : Cette requête supprime un ou plusieurs enregistrements dans une ou plusieurs tables.

Les trois derniers types de requêtes ne seront pas étudiés :

- ?? La requête SQL direct
- ?? La requête de définition de données
- ?? La requête UNION

2. Création d'une requête

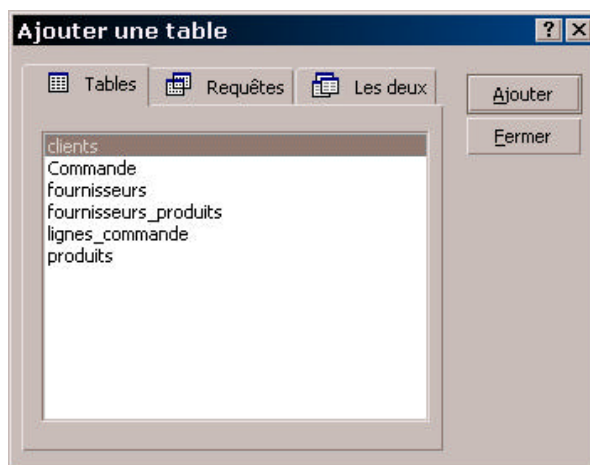
Dans la fenêtre principale d'Access, cliquez sur l'onglet "Requêtes", puis sur le bouton "Nouveau".



Access nous propose 5 façons de créer une requête :

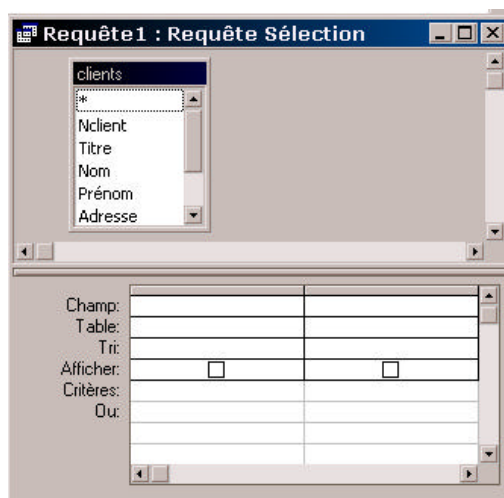
- ?? **Mode création** : nous allons créer la requête de A à Z par nous-mêmes
- ?? **Assistant de Requête simple** : Crée une requête simple sur une ou plusieurs tables, on pourra, dans cette requête simple, faire des calculs, des regroupements.
- ?? **Assistant de Requête d'analyse croisée** : Aide à la création d'une requête d'analyse croisée
- ?? **Assistant de Requête trouver les doublons** : Localise les enregistrements doublons dans une table ou une requête.
- ?? **Assistant de Requête de non-correspondance** : Localise les enregistrements d'une table auxquels ne correspond aucun autre enregistrement d'une autre table. On peut, par exemple, utiliser un tel type de requête pour localiser les clients qui n'ont pas passé de commande.

Comme nous ne reculons pas devant l'effort, nous allons créer les requêtes par nous même, nous cliquons donc sur **Mode Création**, puis, sur OK :

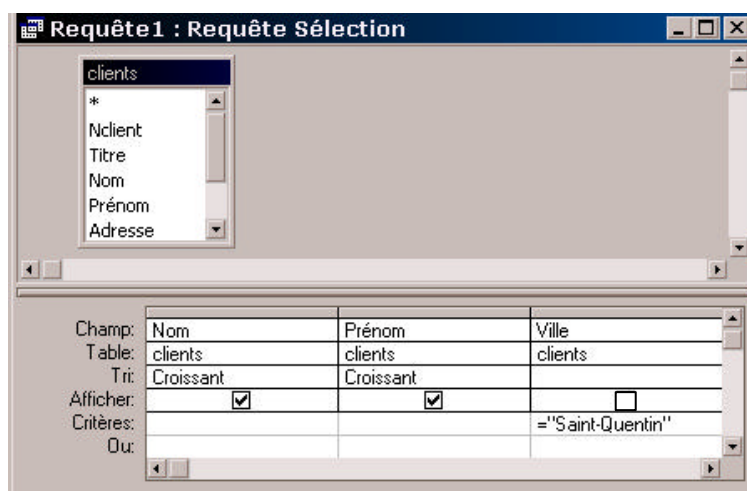


Access nous affiche la liste des tables de la base. Nous choisissons ici les tables sur lesquelles vont porter la requête. Nous allons prendre pour exemple la requête "liste des clients habitant la ville de Paris". La requête va donc porter sur la table "**Clients**", on la sélectionne, et on clique sur "**Ajouter**", comme il n'y a aucune autre table à ajouter, on clique sur "**Fermer**".

Notez qu'on peut aussi faire porter la requête sur le résultat d'une autre requête, dans ce cas, on choisira la requête dans l'onglet "Requête".



Dans la partie supérieure de la fenêtre se trouvent la ou les tables avec la listes de leur champs, c'est sur ces tables que vont porter les requêtes. Dans la partie inférieure, se trouve la description de la requête.



Voici notre requête :

Nous avons un tableau composé de colonnes et de lignes, dans chaque colonne, on indique les champs qui vont apparaître dans le résultat de la requête ou qui vont servir dans la requête, ici on veut la liste des clients habitant Saint-Quentin, on veut donc voir apparaître le champ **Nom** et le champ **Prénom**, le critère de la requête va se faire sur le champ **ville**, on ajoute donc aussi ce champ.

Pour ajouter un champ, on peut, soit le sélectionner dans la table et l'amener avec la souris sur une colonne, soit on clique sur la ligne "**Champ**", la liste de tous les champs s'affiche alors et on en sélectionne un dans la liste. Un des champs proposés s'appelle '*', ce champ signifie "tous les champs de la table", si on choisit ce champ, tous les champs de la table apparaîtront dans le résultat de la requête, en plus des autres champs que vous aurez choisis.

La ligne "**Table**" sert à sélectionner la table à laquelle appartient le champ sélectionné, dans notre cas, il n'y a qu'une table, le choix est vite fait.

La colonne "**Tri**" indique de quelle façon vont être triés les champs dans le résultat de la requête : il y a trois sortes de tri : Croissant (de A à Z), Décroissant (de Z à A) et non trié. On a choisi ici de trier le résultat de la requête par nom et par prénom de façon croissante. Le tri se fait toujours de gauche à droite : le résultat de la requête sera d'abord trié par nom, puis par prénom.

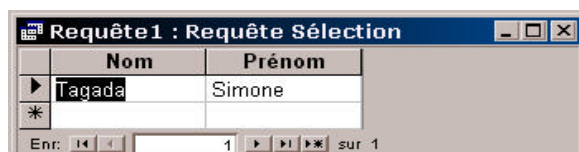
La colonne "**Afficher**" indique si le champ doit être affiché ou non, ici on veut afficher le nom et le prénom de chaque client habitant à Saint-Quentin, mais on a pas besoin d'afficher, pour chaque client, la ville dans laquelle il habite, puisqu'il s'agira toujours de Saint-Quentin, on a donc désélectionné l'affichage du champ "**Ville**".

Enfin la colonne "**Critères**" va indiquer le critère de la requête, on veut la liste des clients habitant Saint-Quentin, le critère est donc : le champ ville doit être égal à Saint-Quentin, d'où le critère = "Saint-Quentin".

Pour exécuter la requête, on clique sur l'icône :



Access affiche le résultat :



Le résultat de la requête s'affiche sous la forme d'une table que l'on peut modifier comme si il s'agissait d'une table normale, on peut ajouter des enregistrements, les modifier, faire des filtres, des tris, etc...

Attention : Les résultats des requêtes, même si ils sont présentés sous forme de tables ne sont pas de véritables tables qu'aurait généré la requête. Ils ne sont qu'une "vue" faite à partir des tables qui ont servi à faire la requête. Autrement dit, si vous modifiez quelque chose dans le résultat de la requête, la modification se repercutera dans la table qui a servi à faire la requête, si on change ici le nom du client, la modification sera reportée dans la table Clients, si on ajoute un client au résultat de la requête, un nouvel enregistrement va être créé dans la table Clients, et en plus, cet enregistrement sera incomplète car seuls deux champs auront pu être saisis dans le résultat de la requête (les champs nom & prénom).

3. Définition des critères de sélection

3.1 Opérateurs

On peut utiliser dans les requêtes les opérateurs suivants :

Opérateur	Signification
=	Egal
<>	Différent
<	Inférieur
>	Supérieur
<=	Inférieur ou égal
>=	Supérieur ou égal

Access met à notre disposition d'autres opérateurs :

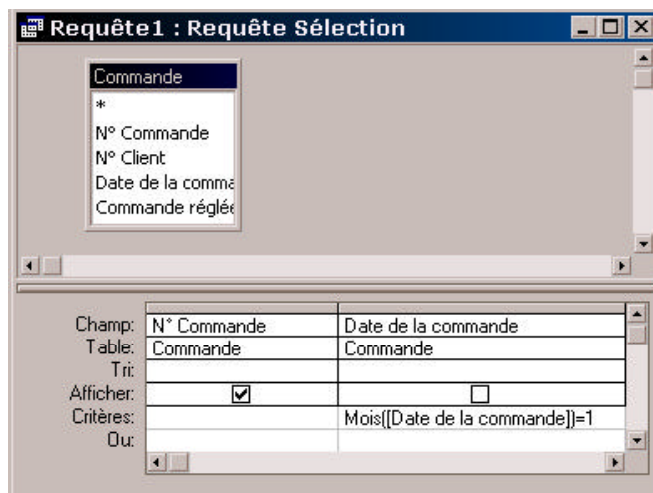
Opérateur	Signification	Exemple
Entre	Sélectionne les enregistrements pour lesquels la valeur d'un champ est comprise dans un intervalle de valeurs.	Entre "A" et "C" Entre 10 et 20 Entre #01/01/99# et #31/12/99#
Dans	Sélectionne les enregistrements pour lesquels la valeur d'un champ est comprise dans une liste.	Dans ("Paris";"Saint-Quentin")
Est	Sélectionne les enregistrements pour lesquels un champ est vide ou non	Est NULL Est pas NULL
Comme	Sélectionne les enregistrements contenant une donnée approximative.	Comme "rue*"
Pas	Sélectionne les enregistrements ne correspondant pas au critère	Pas Entre "A" et "C"

3.2 Les Fonctions

On peut intégrer des fonctions dans les critères de sélection. Access met à notre disposition un très grand nombre de fonctions (pour en avoir la liste complète, consultez l'aide intégrée à Access). Ce sont les mêmes fonctions que celles qui sont utilisés dans les contrôles des formulaires (et d'ailleurs dans tous les logiciels de la gamme Microsoft Office).

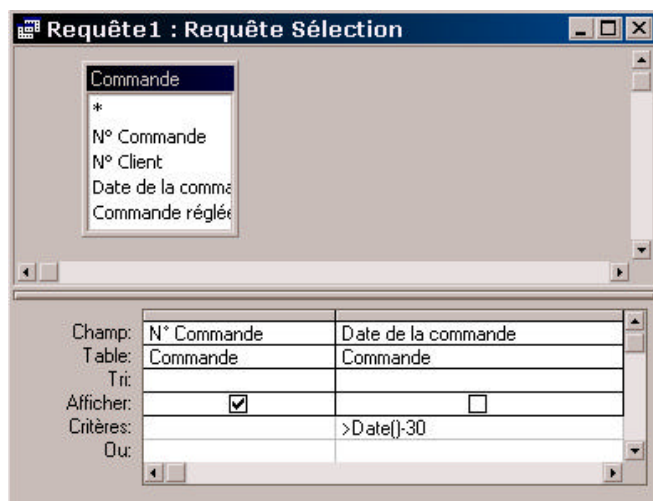
Par exemple, parmi ces fonctions, on trouve la fonction Mois (), à qui on passe une date en paramètre, cette fonction renvoie le mois de cette date, par exemple Mois (#10/2/98#) renvoie 2. Une autre fonction, Date () renvoie la date du jour. On peut utiliser ces deux fonctions dans des critères de sélection :

Par exemple, pour obtenir la liste des commandes du mois d'octobre :



Notez que pour faire référence au champ "Date de la commande" dans la fonction Mois (), on a écrit le champ entre crochets []. On utilise les crochets lorsqu'on travaille avec un champ dont le nom comporte des espaces. Par extension et par mesure de précaution, on les utilisera avec tout type de champ.

Ou pour obtenir la liste des commandes passées il y a moins d'un mois :



Il existe un très grand nombre de fonctions, pour en avoir la liste et la syntaxe, consultez l'aide intégrée à Access.

3.3 Plusieurs critères portant sur des champs différents

On peut avoir plusieurs critères de sélection, ces critères étant séparés entre eux par des OU ou des ET, par exemple: "liste des clients habitant à Saint-Quentin OU à Paris", "Liste des clients s'appelant Dupont ET vivant à Paris".

1- Liste des clients s'appelant Dupont et vivant à Saint-Quentin

Champ:	Nom	Prénom	Ville
Table:	clients	clients	clients
Tri:			
Afficher:	☒	☒	☐
Critères:	= "Dupont"		= "Paris"
Ou:			

La requête ressemble à la précédente, sauf que là, il y a deux critères : le premier Nom = "Dupont" ET le deuxième Ville = "Paris". Vous noterez au passage qu'il n'est pas nécessaire d'afficher le contenu du champ "ville" puisqu'on sait que ce sera toujours Paris. Vous me direz que c'est la même chose avec "Nom", certes, mais c'est mieux d'avoir une liste de nom + prénom qu'une seule liste de prénom.

2- Liste des clients habitant Saint-Quentin OU Paris

Champ:	Nom	Prénom	Ville	Ville
Table:	clients	clients	clients	clients
Tri:				
Afficher:	☒	☒	☒	☒
Critères:			= "Paris"	
Ou:				= "Saint-Quentin"

Quelle est la différence ?

Les deux critères ne sont plus sur la même ligne. Ils sont sur deux lignes séparées. Le premier est sur la ligne "Critères", le second sur la ligne "Ou". Si on avait voulu ajouter un troisième critère (Liste des clients habitant Saint-Quentin OU Paris OU Lille), on aurait ajouté le critère = "Lille" sur une troisième ligne et ainsi de suite.

En règle générale :

- ?? Si deux critères sont séparés par des ET, on les place sur la même ligne.
- ?? Si deux critères sont séparés par des OU, on les place sur des lignes différentes.

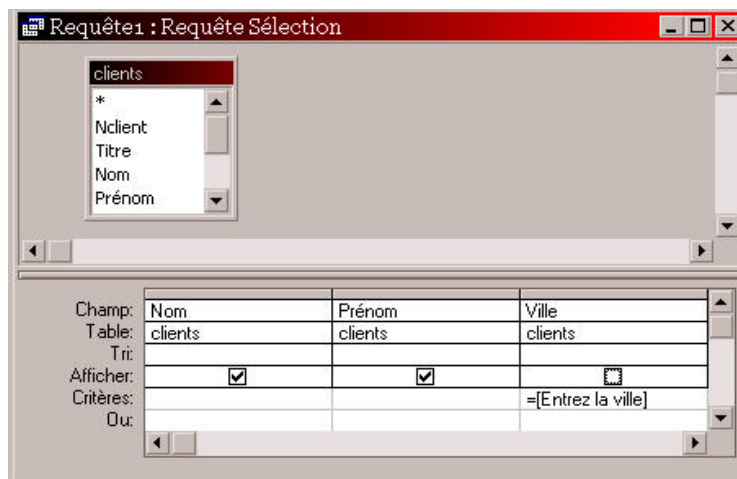
3.4 Requête paramétrée

Pour l'instant, les critères de chaque requête que nous avons fait étaient indiqués explicitement dans la requête. Supposons maintenant que nous voulions faire une requête du type :

« Liste de tous les clients qui habitent dans une ville » mais qu'on veuille entrer nous même la ville.

On ne connaît pas par avance la ville en question, et on ne va pas préparer autant de requêtes qu'il peut y avoir de villes (on n'est pas sorti de l'auberge), la solution la plus simple est alors de demander à l'utilisateur (l'utilisateur, c'est celui qui va utiliser votre requête plus tard) d'entrer la ville, et de faire la requête en fonction de ce qu'il a entré.

Pour faire ça, on procède ainsi :



Au lieu d'indiquer une ville, on a mis =[Entrez la ville]. Que va-t-il se passer lorsqu'on va exécuter la requête?

Habituellement, pour Access, tout ce qui est indiqué entre crochets est le nom d'un champ, si on avait mis par exemple = [Nom], il aurait cherché les clients qui habitent une ville qui s'appelle comme leur nom. Or ici, [Entrez la ville] n'est pas un nom de champ, Access ne sait donc pas ce que c'est, et il demande à l'utilisateur d'entrer la valeur de ce champ inconnu :

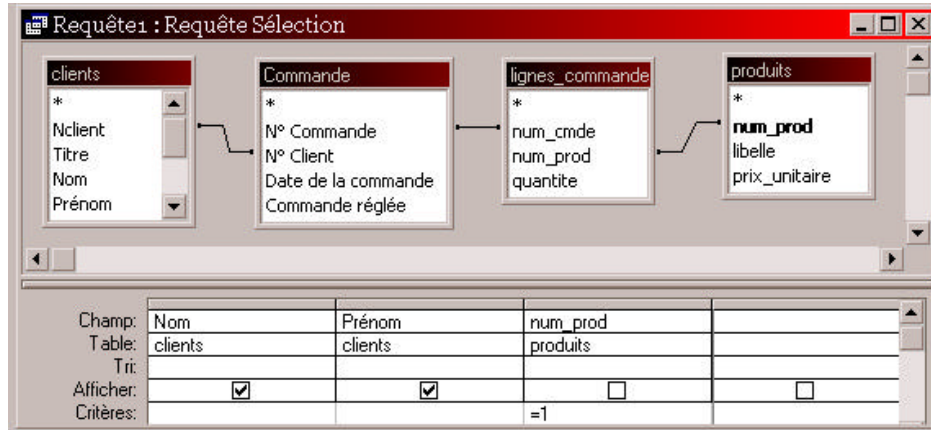


L'utilisateur va entrer ici la ville (notez que le message affiché est ce qui était indiqué entre crochets dans la requête). Maintenant, à chaque fois que Access va rencontrer dans la requête [Entrez la ville], il va le remplacer par ce qu'à saisi l'utilisateur. Si l'utilisateur a saisi Paris, Access va remplacer dans son critère =[Entrez la ville] par = "Paris".

4. Les requêtes multitables

Pour l'instant, nos requêtes ne portaient que sur une seule table, il est possible de faire des requêtes qui portent sur plusieurs tables simultanément. Dans ce cas, les requêtes peuvent être très complexes.

Par exemple, si nous voulons la liste des clients ayant commandé le produit n° 1 :



Avec le bouton : , on affiche la liste des tables de la base.

On va choisir ici toutes les tables entrant en jeu pour pouvoir faire la requête. Ici on veut la liste des clients, il faut donc la table **clients**. "ayant commandé", il faut donc la table **commandes**, "le produit n°1", il faut donc la table **produits**. Il faut aussi faire intervenir la table "**Lignes-commandes**".

Pourquoi ? parce que c'est elle qui fait la liaison entre la table commande et la table produits. De façon générale, lorsqu'on fait une requête portant sur plusieurs tables, il faut respecter deux règles :

- ?? Toutes les tables intervenant dans la requêtes doivent être reliées entre elles, il ne doit pas y avoir de tables isolées, sinon, Access va essayer de trouver lui-même les relations entre ces tables isolées et parfois le résultat peut être folklorique.
- ?? Il ne doit pas y avoir de tables n'ayant rien à faire dans la requête, sinon, Access va se baser sur les relations entre ces tables n'ayant rien à voir avec la choucroute et celles ayant à voir et va donner des résultats erronés.

En résumé : toutes les tables nécessaires, mais pas plus.

Vous pouvez noter que, une fois les bonnes tables installées avec les bonnes relations entre elles, la requête est fort simple, il suffit d'indiquer produits=1, et Access, grâce aux relations, va retrouver la liste des clients ayant commandé ce produit.

5. Les fonctions de regroupement

Jusqu'à présent, nos requêtes nous permettaient de répondre à des questions du type : "Liste des clients habitant Paris", "Liste des produits commandés par le client 1". Grâce aux fonctions de regroupement, nous allons pouvoir répondre à des questions du type : "Combien de clients habitent Paris ?" ou "Pour combien à commandé chaque client ?".

Pour cela, cliquez sur le bouton :



Une nouvelle ligne '**Opération**' apparaît dans la requête, c'est grâce à elle que nous allons faire nos opérations.

1- Combien de clients habitent Paris

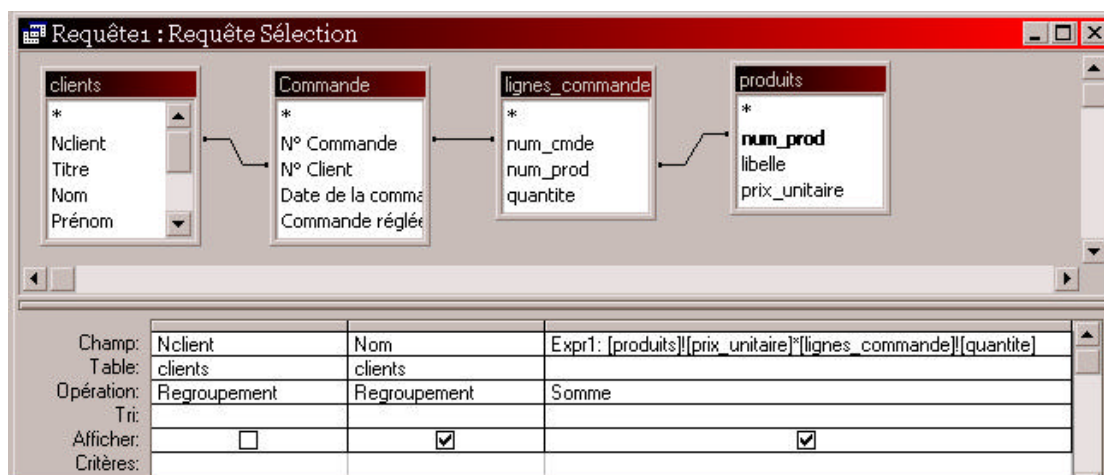


Qu'y-a-t-il de différent ?

Nous avons deux champs : nous allons compter le nombre de clients, chaque client a un numéro, on insère donc dans la requête le numéro de client, c'est le nombre de ces numéros que nous allons compter, il faut qu'on compte les clients habitant Paris, d'où le deuxième champ Ville.

La différence est sur la ligne Opération :

Elle indique comment va se faire l'opération, si nous voulons compter, on va choisir l'opération "**compte**". Dans tous les champs où il y a un critère de sélection (ici Ville = "Paris"), on choisira l'opération où. Le résultat de la requête sera une table avec un unique champ, non modifiable, qui indiquera le nombre de clients habitant Paris.

2- Montant commandé par chaque client


Champ:	Nclient	Nom	Expr1: [produits].[prix_unitaire]*[lignes_commande].[quantite]
Table:	clients	clients	
Opération:	Regroupement	Regroupement	Somme
Tri:			
Afficher:	☐	☒	☒
Critères:			

Ca se corse !

Ici, on veut, par client, la somme de ses commandes. A partir du moment où on veut un résultat par catégorie (une somme par client, un nombre de clients par ville, etc...), il y a regroupement, ici on veut la somme des commandes regroupées par clients, c'est pour ça qu'on a choisi comme opération pour numéro de client "Regroupement".

Pourquoi at-on ajouté le champ "nom" ? Simplement pour ne pas avoir une liste de numéros avec un montant correspondant. Pourquoi dans ce cas, n'a-t-on pas regroupé les clients par leur nom au lieu de le faire par numéro et nom ? parce que plusieurs clients peuvent avoir le même nom et qu'on ne veut pas cumuler le montant des commandes par clients homonymes, on regroupe donc les clients par numéro et nom.

Enfin, le troisième champ : on veut la somme du montant des commandes : qu'est-ce qu'une commande ? c'est une liste de prix unitaires * une quantité. A la place d'un champ, on indique donc qu'on veut les prix unitaires * les quantités. La syntaxe est **[nom de la table].[champ de la table]**, une commande, c'est donc une liste de [produits].[prix unitaire] * [lignes_commandes].[quantité]. Et on veut la somme de toutes ces commandes, on choisi donc l'opération "Somme".

5.1 Les opérations

En plus de la somme et de compte, il existe d'autres opérations possibles :

Opération	Signification
Compte	Compte le nombre de valeurs
Dernier	Valeur du dernier enregistrement
Ecartype	Ecart type
Max	Valeur la plus élevée
Min	Valeur la plus faible
Moyenne	Moyenne
Premier	Valeur du premier enregistrement
Somme	Total
Var	Variance

5.2 Quelques exemples

Voici quelques exemples de requêtes :

Nombre de clients nommés "Dupont" par ville

Requête1 : Requête Sélection

clients

*

Nclient

Titre

Nom

Prénom

Champ:	Ville	Nclient	Nom
Table:	clients	clients	clients
Opération:	Regroupement	Compte	Où
Tri:			
Afficher:	☒	☒	☐
Critères:			= "Dupont"

Liste des clients ayant commandé plus de 1000 F du produit 1

Requête1 : Requête Sélection

clients

*

Nclient

Titre

Nom

Prénom

Commande

*

N° Commande

N° Client

Date de la commande

Commande réglée

lignes_commande

*

num_cmde

num_prod

quantite

produits

*

num_prod

libelle

prix_unitaire

Champ:	Nclient	Nom	num_prod	Expr1: [produits]![prix_unitaire]*[Commande]!
Table:	clients	clients	produits	
Opération:	Regroupement	Regroupement	Regroupement	Somme
Tri:				
Afficher:	☒	☒	☐	☒
Critères:			=1	>1000

6. Les requêtes d'analyse croisée

Les requêtes d'analyse croisée permettent de répondre à des questions du type "qui à commandé combien de quoi ?". Elles retournent le résultat sous forme d'un tableau comportant des champs en abscisse et en ordonnée, avec, dans chaque case la réponse à notre question.

Exemple : Qui a commandé combien de quoi ?

	Qui	Qui
Quoi	Combien	Combien
Quoi	Combien	combien

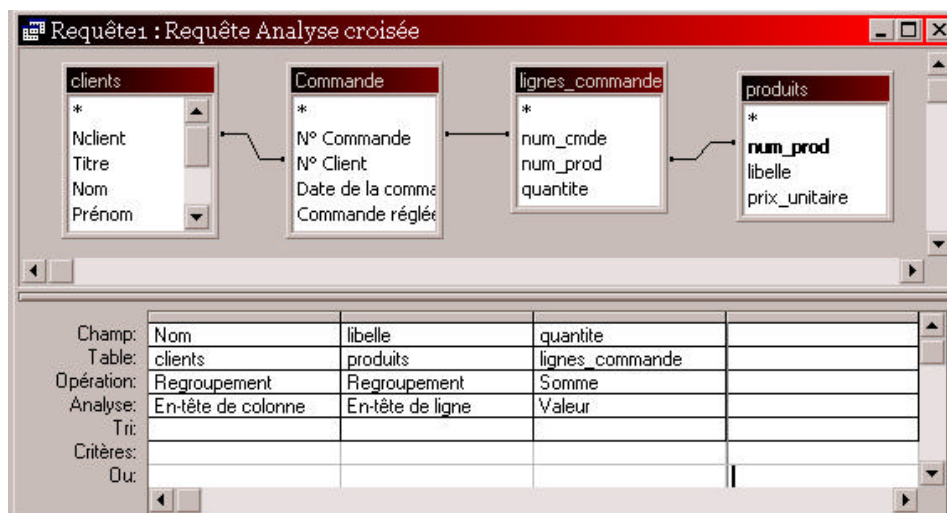
L'en-tête "Qui" va contenir le nom des clients

L'en-tête "Quoi" va contenir la liste des produits

Et combien va donner, pour chaque client, le nombre de produits qu'il a commandé.

Comment procède-t-on ?

1. D'abord on crée une requête standard : pour cette requête nous avons besoin du nom dans la table clients, du libellé du produit dans la table produit et de la quantité commandé dans la table lignes-commandes.
2. On transforme la requête en requête d'analyse croisée en allant dans le menu "Requête" et en choisissant "Analyse croisée".
3. Une nouvelle ligne apparaît dans la requête : la ligne "Analyse"
4. Dans cette ligne, on va indiquer si le champ qu'on a choisi va être l'en-tête des colonnes, l'en-tête des lignes ou la valeur contenue dans les cases du tableau.
5. L'Opération pour les en-têtes est toujours "Regroupement"
6. L'Opération pour les valeurs des cases dépend de ce qu'on cherche, ici on cherche le nombre de produits acheté, l'opération est donc "Somme"



7. Les requêtes ACTION

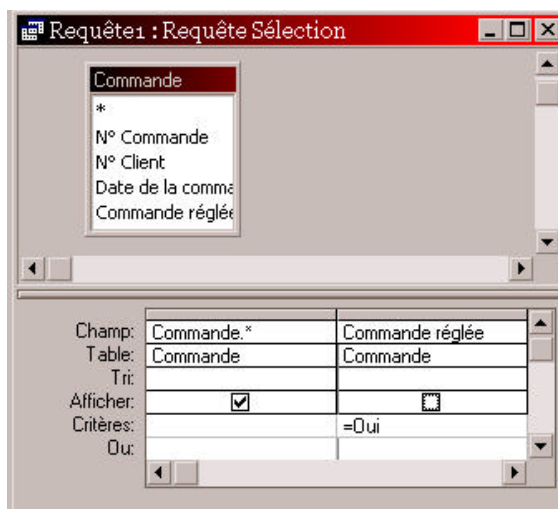
Les requêtes faites jusqu'à présent se contentent de retourner le résultat d'une sélection sous forme de table, cette table, étant, selon le type de requête, modifiable. Il existe cependant d'autres types de requêtes, les requêtes action, ces requêtes vont permettre de :

- ?? Créer une table à partir du résultat d'une requête
- ?? Ajout des enregistrements à une table à partir des résultats de la requête
- ?? Mettre à Jour une table en fonction de certains critères
- ?? Supprimer des enregistrements répondant aux critères de la requête

7.1 Les requêtes Création

Une requête création crée une table à partir des résultats qu'elle produit à partir d'une table existante. Supposons que nous voulions créer une table "Commandes réglées" qui contiendrait la liste des commandes déjà réglées.

1. Créer la requête normalement : nous voulons la liste des commandes réglées :



2. On la transforme en requête Création (Menu Requête / Requête Création de table)
3. Access nous demande le nom de la table à créer :



Attention, si vous sélectionnez une table existante, la table va être écrasée par cette opération

- Exécutez la requête avec l'icône point d'exclamation : la table va être créée avec le résultat de la requête. Si vous voulez vérifier avant de créer la table quel sera le résultat de la requête, cliquez sur l'icône:



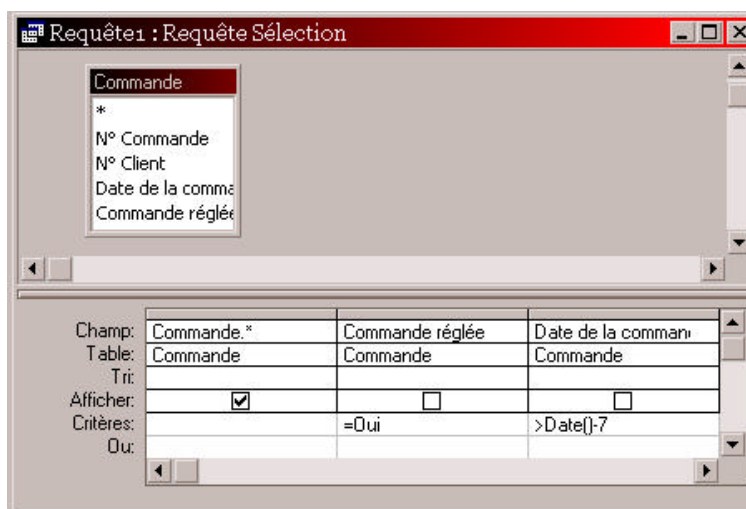
Une fois que le résultat sera conforme à vos attentes, vous pourrez cliquer sur l'icône point d'exclamation.

7.2 Les requêtes Ajout

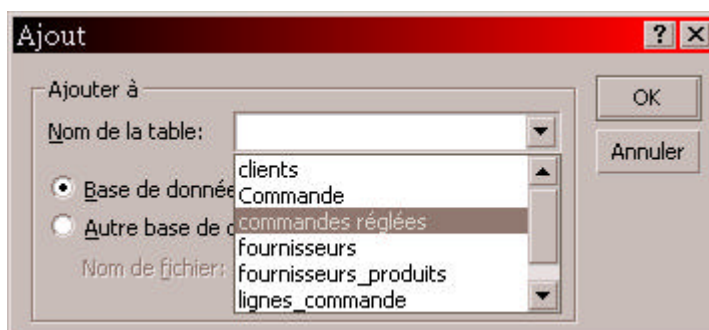
Une requête Ajout copie tout ou partie des enregistrements d'une table (la table source) à la fin d'une autre table (la table cible).

Toujours dans notre table "Commandes réglées", nous voulons ajouter les commandes qui ont été réglées depuis la semaine dernière :

- Création de la requête normale :



- On la transforme en requête Ajout (Menu Requête / Requête Ajout)
- Access nous demande le nom de la table à laquelle il faut ajouter le résultat de la requête :



- Comme précédemment, avec les icônes Affichage et point d'exclamation, vérifiez et validez votre requête.

Attention :

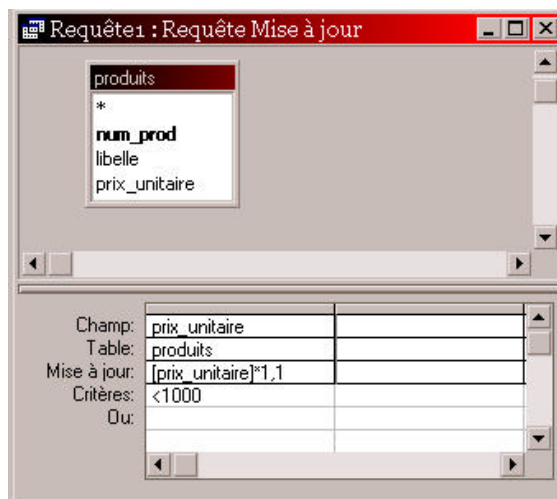
- ?? Les enregistrements sont COPIES de la table source vers la table cible et non pas déplacés.
- ?? Les deux tables doivent avoir des noms de champs identiques (et les mêmes types de données), les deux structures ne doivent pas nécessairement être identiques.
- ?? Si la table source comporte plus de champs que la table cible, les champs supplémentaires sont ignorés
- ?? Si la table source comporte moins de champs que la table cible, les champs dont les noms sont identiques sont copiés, les autres sont laissés vides.
- ?? Access ne copie QUE les champs que vous avez déclarés dans la requête (d'ou le champ *)

7.3 Les requêtes Mise à Jour

Les requêtes mise à jour permettent de modifier rapidement tous les enregistrements d'une table ou un groupe d'entre eux :

Supposons que l'on veuille augmenter de 10% le prix des produits dont le prix actuel est inférieur à 1000 F.

1. On crée une requête, et dans le menu Requête, on clique sur "Requête Mise à Jour"
2. Un champ "Mise à jour" apparaît dans la requête, c'est là qu'on va indiquer la modification qui va avoir lieu :

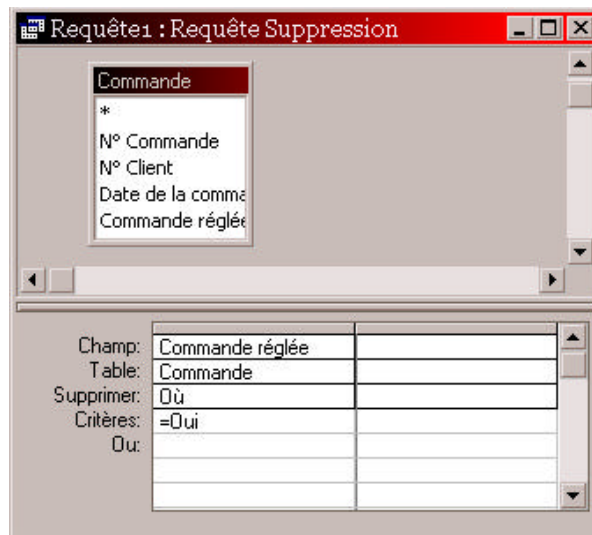


Dans la case Mise à Jour, on a indiqué, qu'à la place du prix unitaire, nous voulions [prix_unitaire]*1.1 (soit une augmentation de 10%), ceci pour les produits dont le prix est inférieur à 1000 (critère).

7.4 Les requêtes Suppression

Les requêtes suppression vous permettent de supprimer un groupe d'enregistrements qui répondent à un critère donné. Supposons que l'on veuille supprimer de la table commande toutes les commandes réglées :

1. On crée une requête standard, puis dans le menu Requête, on clique sur Requête Suppression :



2. On exécute la requête ou on vérifie avant que le résultat est conforme à ce que l'on attend.

Attention :

Vérifiez bien le résultat avant d'exécuter la requête, il n'est pas possible de revenir en arrière après avoir effacé des enregistrements.