

Formation

Requêtes SQL

Proposée par :

ALGIDE

Siège : L'ACHARD, 38 880 AUTRANS (Grenoble)
Tél. : 04.38.49.90.15 Fax : 04.38.49.90.16
Email : info@algide.fr Web : www.algide.fr

Sommaire

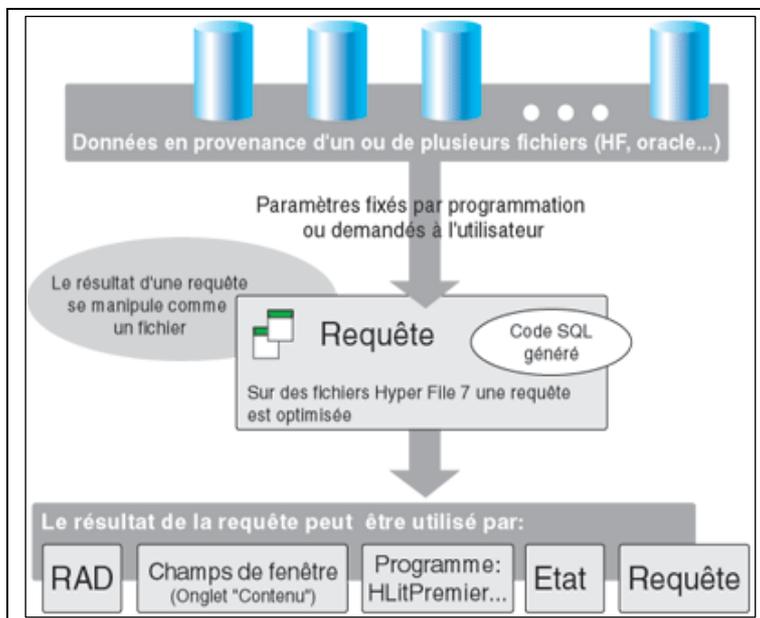
Introduction	4
Qu'est ce qu'une requête ?	4
Les requêtes SQL	6
Syntaxe	6
Exemples de requêtes	8
1) Sélection sans condition.....	8
2) Tri du résultat – ORDER BY.....	8
3) Sélection avec une condition simple - WHERE	10
5) Sélection avec plusieurs conditions portant sur plusieurs fichiers.....	17
6) Extension pour les calculs.....	17
7) Sélection avec partitions et calcul – GROUP BY.....	19
8) Requêtes composées	21
9) Sous-requêtes	21
Dans GEPMI...	31
Les « jumelles ».....	31
1) Avec l'assistant – onglet « Recherche par conditions »	31
2) Directement en SQL – onglet « Requête SQL ».....	33
3) Requêtes paramétrées.....	35
« Impression/Requêtes »	39
Requête au format WDR.....	39
Requête SQL	39
Alors « Jumelles » ou « Impression/requêtes » ?	43
WDEtat 9 Editeur de requêtes.....	47

Introduction

Qu'est ce qu'une requête ?

Une requête signifie dans le langage courant une demande. En informatique, une requête est une demande de traitement.

Une requête sert à interroger une base de données pour visualiser, insérer ou supprimer des données. En effet, une requête est une opération qui consiste à extraire de fichiers (d'une base de données) tout ou partie des éléments qui s'y trouvent et éventuellement à les modifier.



Nous aborderons ici uniquement les requêtes de sélection.

Les requêtes SQL

Syntaxe

La structure de la requête définit les données manipulées.

Voici la syntaxe générale d'une commande SELECT :

⇒ **SELECT** [**DISTINCT** ou **ALL**] * ou liste de rubriques

FROM noms des fichiers

[**WHERE** prédicats]

[**GROUP BY** ordre des groupes]

[**HAVING** condition]

[**ORDER BY**] liste de colonnes

- dans cette syntaxe, les mots clef du SQL sont en gras, les paramètres en minuscule et entre crochets on trouve les parties optionnelles

Les rubriques sélectionnées sont affichées dans l'ordre où elles ont été spécifiées.

En fait l'ordre SQL SELECT est composé de 6 clauses dont 4 sont optionnelles.

Clauses de l'ordre SELECT :

SELECT	Spécification des rubriques du résultat
FROM	Spécification des fichiers sur lesquels porte l'ordre
[WHERE]	Filtre portant sur les données (conditions à remplir pour que les lignes soient présentes dans le résultat)
[GROUP BY]	Définition d'un groupe (sous ensemble)
[HAVING]	Filtre portant sur les résultats (conditions de regroupement des lignes)
[ORDER BY]	Tri des données du résultat

Exemples de requêtes

1) Sélection sans condition

Le caractère * (étoile) récupère toutes les rubriques des fichiers précisés dans la clause FROM de la requête. Juste après le mot clef SELECT, on précise les rubriques des fichiers qui doivent être présentées dans la réponse.

L'utilisation du caractère étoile ramène toutes les rubriques des fichiers dans la réponse. Dans le cas contraire il faut expressément nommer chacune des rubriques et les séparer par des virgules.

Sélection de toutes les rubriques de la table FACTURE

⇒ **SELECT ***
FROM FACTURE

Sélection des champs RF_FACTURE et RF_CLIENT de la table FACTURE

⇒ **SELECT RF_FACTURE, RF_CLIENT**
FROM FACTURE

2) Tri du résultat – ORDER BY

Sélection de toutes les références des clients pour lesquels on a des factures

⇒ **SELECT RF_CLIENT**
FROM FACTURE

- dans le résultat de cette requête, les références des clients vont apparaître par ordre des enregistrements des factures (ordre aléatoire).

- si un client a plusieurs factures, sa référence apparaîtra plusieurs fois

Sélection de toutes les références des clients pour lesquels on a des factures, chaque référence ne devant apparaître qu'une seule fois :

⇒ **SELECT DISTINCT RF_CLIENT**
FROM FACTURE

Lorsque le moteur construit la réponse, il rapatrie toutes les lignes correspondantes, généralement dans l'ordre où il les trouve, même si ces dernières sont en double, c'est à dire qu'il récupère toutes les lignes (ALL par défaut). C'est pourquoi il est souvent nécessaire d'utiliser le mot clef **DISTINCT** qui permet d'éliminer les doublons dans la réponse.

Sélection de toutes les références des clients pour lesquels on a des factures, chaque référence ne devant apparaître qu'une seule fois et par ordre alphabétique :

```
⇒ SELECT DISTINCT RF_CLIENT  
FROM FACTURE  
ORDER BY RF_CLIENT DESC
```

- la colonne sur laquelle porte le tri doit apparaître dans la liste des colonnes sélectionnées.

La clause **ORDER BY** permet de définir le tri des colonnes de la réponse, soit en précisant le nom littéral de la rubrique.

ASC spécifie l'ordre ascendant et **DESC** l'ordre descendant du tri. **ASC** ou **DESC** peut être omis, dans ce cas c'est l'ordre ascendant qui est utilisé par défaut.

Bien que la clause **ORDER BY** ne soit pas nécessaire, il est souvent utile de trier la réponse en fonction des colonnes. En revanche le temps de réponse s'en ressent souvent.

Attention : - le tri est un tri interne, il ne faut donc placer dans cette clause que les noms des colonnes présentées dans la clause **SELECT**.

- la clause **ORDER BY** est la dernière clause de tout ordre SQL et ne doit figurer qu'une seule fois dans le **SELECT**.

3) Sélection avec une condition simple - WHERE

Le prédicat doit contenir n'importe quelle expression logique renvoyant une valeur vrai.

Sélection des FATURE du client 'DURAND'

```
⇒ SELECT *  
FROM FACTURE  
WHERE RF_CLIENT='DURAND'
```

- Outre l'opérateur = vous pouvez utiliser :

- < (inférieur à),
- <= (inférieur ou égal à),
- > (supérieur à),
- >= (Supérieur ou égal à)

- Si la condition porte sur un champ de type chaîne il faut entourer la valeur de ''

Sélection de toutes les factures dont la référence du client commence par 'DU' :

```
⇒ SELECT *  
FROM FACTURE  
WHERE RF_CLIENT LIKE 'DU%'
```

- Le caractère % remplace un ou plusieurs caractères. Il peut être placé au début de la chaîne pour signifier « finit par » ou de chaque côté pour « contient ».

- Le mot réservé **LIKE** remplace l'égalité quand la comparaison n'est pas fixe.

L'opérateur LIKE permet d'effectuer une comparaison partielle. Il est surtout employé avec les colonnes contenant des données de type alpha. Il utilise les jokers % et _ ('pour cent' et 'blanc souligné'). Le joker % remplace n'importe quelle chaîne de caractères, y compris la chaîne vide. Le blanc souligné remplace un et un seul caractère.

Attention : L'opérateur LIKE effectue une recherche en tenant compte de la différence entre lettres majuscules et minuscules.

T rubriques transformation des chaînes en majuscules :

```
⇒ SELECT UPPER(RF_CLIENT)
FROM FACTURE
⇒ SELECT RF_CLIENT
FROM FACTURE
WHERE UPPER(RF_CLIENT)=UPPER('Durand')
```

- LOWER permet de passer la chaîne en minuscules,

Les opérateurs **LOWER** et **UPPER** permettent de mettre en majuscules ou en minuscules des chaînes de caractères dans les requêtes.

Pour effectuer une recherche en ne tenant aucunement compte de la différence entre majuscules et minuscules, il faut utiliser l'opérateur UPPER (ou LOWER)

Attention : à la transformation des accents !

Sélection des factures de certains clients :

```
⇒ SELECT *
FROM FACTURE
WHERE RF_CLIENT IN ('DURAND', 'DUPOND', 'DUMOULIN')
```

- IN (DANS en français), on recherche les éléments qui sont dans l'ensemble de valeurs compris dans les parenthèses.

Sélection des factures des clients qui ne font pas parti de l'ensemble précédent :

```
⇒ SELECT *
FROM FACTURE
WHERE RF_CLIENT NOT IN ('DURAND', 'DUPOND', 'DUMOULIN')
```

- NOT peut être appliqué à toutes les conditions vues précédemment.

L'opérateur **IN** permet de rechercher si une valeur se trouve dans un ensemble donné, quel que soit le type des valeurs de référence spécifiées (alpha, numérique, date...). Bien entendu, il est possible d'inverser le fonctionnement de l'opérateur IN en lui adjoignant l'opérateur NOT.

4) Sélection avec plusieurs conditions simples

Sélection de toutes les factures dont du client commence par 'DU' et dont le montant est supérieur à 18 000 :

```
⇒ SELECT *
FROM FACTURE
WHERE RF_CLIENT LIKE 'DU%' AND PX_HT_EU >= 18000
```

- AND (ET en Français), on recherche les éléments vérifiant la condition 1 portant sur la référence du client et la condition 2 portant sur le montant HT.

Sélection de toutes les factures dont la référence des clients DURAND et DUPOND :

```
⇒ SELECT *
FROM FACTURE
WHERE RF_CLIENT = 'DURAND' OR RF_CLIENT='DUPOND'
```

- OR (OU en Français), dans ce cas les résultats contiennent à la fois toutes les factures dont le client est DURAND et toutes les factures dont le client est DUPOND

Sélection de toutes les factures pour l'Isère et le Rhône :

```
⇒ SELECT *
FROM FACTURE
WHERE (CD_POSTAL >= 38000 AND CD_POSTAL <= 39000) OR
(CD_POSTAL >= 69000 AND CD_POSTAL <= 70000)
```

La similitude entre le ET et la multiplication

opérateur ET	FAUX	VRAI
FAUX	FAUX	FAUX
VRAI	FAUX	VRAI

multiplication	0	1
0	0	0
1	0	1, <> 0

La similitude entre le OU et l'addition

opérateur OU	FAUX	VRAI
FAUX	FAUX	VRAI
VRAI	VRAI	VRAI

addition	0	1
0	0	1, <> 0
1	1, <> 0	2, <> 0

Sélection de toutes les factures pour l'Isère et le Rhône plus simplement :

```
⇒ SELECT *
FROM FACTURE
WHERE (CD_POSTAL BETWEEN 38000 AND 38999) OR
(CD_POSTAL BETWEEN 69000 AND 69999)
```

- BETWEEN équivaut à un >= et <=.

L'opérateur BETWEEN permet de rechercher si une valeur se trouve dans un intervalle donné, quel que soit le type des valeurs de référence spécifiées (alpha, numérique, date...).

Remarque : plus la requête aura de conditions, plus le résultat sera précis. Attention toutefois à ne pas écrire de conditions incompatibles :

```
⇒ SELECT *  
FROM FACTURE  
WHERE MT_HT_EU>18000 AND MT_HT_EU<15000
```

5) Sélection avec plusieurs conditions portant sur plusieurs fichiers

Sélection des factures dont la commande a été livrée :

```
⇒ SELECT *  
FROM FACTURE, ET_CMD, FACCOM  
WHERE FACTURE.RF_FACTURE=FACCOM.RF_FACTURE AND  
FACCOM.RF_COMMAND=ET_CMD.RF_COMMAND AND  
ET_CMD.F_LIVREE=1
```

- les 2 premières conditions permettent de faire une jointure sur les fichiers, c'est-à-dire relier les fichiers entre eux :

la requête se construit en plusieurs étapes :

1 - pour chaque FACTURE, FACTURE.RF_FACTURE=FACCOM.RF_FACTURE permet de rechercher les FACCOM liés à cette FACTURE

2- pour chaque enregistrement de FACCOM précédemment sélectionné, FACCOM.RF_COMMAND=ET_CMD.RF_COMMAND permet de rechercher les ET_CMD liés à ces FACCOM

3- enfin, on sélectionne les ET_CMD, parmi les ET_CMD retenus précédemment, ceux dont le flag F_LIVREE est à 1.

- attention, dans la « partie » FROM, entre chaque nom de fichier, l'espace après la virgule est important

- dans une requête portant sur plusieurs fichiers, il faut préfixer les rubriques sur lesquelles portent les conditions du fichier auquel elles appartiennent : FACTURE.RF_FACTURE

6) Extension pour les calculs

Il est possible de réaliser des comptages statistiques sur les colonnes, à l'aide des opérateurs AVG (moyenne), MAX (maximum), MIN (minimum), SUM (total), COUNT (nombre). On les appellent aussi **fonctions d'agrégations**.

Sélection de la quantité maximum fabriquée pour un O.F. sur l'atelier 'Montagne' :

⇒ **SELECT** RF_ORDREFA, RF_ARTICLE, **MAX**(QT_REAL)
FROM ORDREFAB
WHERE RF_ATELIER='Montage'

- idem pour MIN

Calcul de la somme des quantités fabriquées sur l'atelier 'Montage' :

⇒ **SELECT** **SUM**(QT_REAL)
FROM ORDREFAB
WHERE RF_ATELIER='Montage'

Calcul de la moyenne des quantités fabriquées sur l'atelier 'Montage' :

⇒ **SELECT** **AVG**(QT_REAL)
FROM ORDREFAB
WHERE RF_ATELIER='Montage'

Calcul du nombre d'O.F. en attente pour l'atelier 'Montage' :

⇒ **SELECT** **COUNT**(*)
FROM ORDREFAB
WHERE RF_ATELIER='Montage' **AND** F_ATTENTE='A'

7) Sélection avec partitions et calcul – GROUP BY

La clause **GROUP BY** permet de classer une sélection d'enregistrements par groupe. Dans une instruction **SELECT**, la clause **GROUP BY** vient après la clause et précède la clause **ORDER BY**.

Sélection des factures pour chaque client :

⇒ **SELECT** RF_FACTURE, RF_CLIENT
FROM FACTURE
GROUP BY RF_CLIENT, RF_FACTURE

La clause **HAVING** permet de spécifier une ou plusieurs conditions sur des groupes d'enregistrements générés par la clause **GROUP BY**. Dans une instruction **SELECT**, la clause **HAVING** vient après la clause **GROUP BY** et précède la clause **ORDER BY**.

Sélection des clients dont le CA est supérieur à 5000 :

```
⇒ SELECT CLIENT.RF_CLIENT, SUM(PX_HT_EU) AS MT_HT  
FROM FACTURE, CLIENT  
WHERE FACTURE.RF_CLIENT=CLIENT.RF_CLIENT  
GROUP BY RF_CLIENT  
HAVING MT_HT>5000
```

- AS permet de créer un alias (une variable). Elle peut être utilisée sur des rubriques ou sur des fichiers.

8) Requêtes composées

Les requêtes composées permettent de combiner des requêtes de sélection (instruction SELECT). Chaque requête de sélection doit porter sur des fichiers de structure identique.

Sélection de tous les articles qui entrent dans la composition d'ensembles ou de produits :

```
⇒ SELECT RF_ARTICLE  
FROM ARTENS  
UNION  
SELECT RF_ARTICLE  
FROM PROART
```

9) Sous-requêtes

Une sous-requête est une requête dont le résultat est utilisé par une autre requête. Elle peut être utilisée :

- soit en tant que condition de sélection.
- soit en tant que fichier de la requête principale.

Limites : L'instruction ORDER BY est interdite dans les sous requêtes.

Sélection de tous les articles qui n'entrent pas dans la composition d'un ensemble ou d'un produit :

```
⇒ SELECT RF_ARTICLE, LIBELLE
FROM ARTICLE
WHERE RF_ARTICLE NOT IN (SELECT RF_ARTICLE
FROM ARTENS
UNION
SELECT RF_ARTICLE
FROM PROART)
```



```
⇒ SELECT RF_ARTICLE
FROM (SELECT RF_ARTICLE
FROM ARTENS
UNION
SELECT RF_ARTICLE
FROM PROART)
```


En résumé des commandes SQL

Les "commandes SQL" (instructions, fonctions, clauses, ...) utilisables dans une requête SQL vérifiée par le moteur Hyper File sont les suivantes :

*		Utilisé avec l'instruction SELECT
	<i>p8</i>	Sélectionne toutes les rubriques
%		Utilisé avec l'instruction WHERE
	<i>p10</i>	Remplace des caractères d'un paramètre de type chaîne.
ALL		Utilisé avec l'instruction SELECT Sélectionne tous les enregistrements Utilisé avec l'opérateur UNION Sélectionne les enregistrements communs Utilisé dans les sous-requêtes
AS	<i>P20</i>	Crée un alias pour chaque rubrique et chaque fichier utilisé
ASC	<i>P10</i>	Utilisé avec la clause ORDER BY Définit l'ordre ascendant
ASCII		Renvoie le code ASCII : soit d'un caractère. soit du première caractère d'une chaîne.
AVG	<i>P18</i>	Calcule la moyenne d'un ensemble de valeurs
AND	<i>P14</i>	Combine plusieurs conditions de sélection (l'une et l'autre)
ANY		Utilisé dans les sous-requêtes
BETWEEN	<i>P14</i>	Sélectionne des enregistrements dont la valeur d'une rubrique appartient à une plage de valeurs
BIN		Renvoie la chaîne de caractères représentant "entier" en binaire
BOTTOM		Renvoie uniquement les n derniers enregistrements du résultat d'une requête
CEILING		Renvoie la valeur arrondie par excès d'un nombre
CHAR_LENGTH		Renvoie la taille (le nombre de caractères) d'une expression
CHARACTER_LENGTH		Renvoie la taille (le nombre de caractères) d'une expression
CONCAT		Concatène plusieurs chaînes entre elles
COUNT	<i>P18</i>	Renvoie le nombre d'enregistrements présents dans un fichier ou dans un groupement d'enregistrements
DELETE		Supprime des enregistrements d'un fichier
DESC	<i>P10</i>	Utilisé avec la clause ORDER BY Définit l'ordre descendant
DISTINCT	<i>P8</i>	Utilisé avec l'instruction SELECT Supprime les enregistrements dupliqués (doublons) du résultat de la requête
EXISTS		Utilisé dans les sous-requêtes

FULL OUTER JOIN	Réalise une jointure externe
FROM	Utilisé avec l'instruction SELECT Définit les fichiers utilisés Utilisé avec l'instruction DELETE <i>p8</i> Définit les fichiers utilisés
GROUP BY	<i>P18</i> Classe une sélection d'enregistrement par groupe
HAVING	<i>P20</i> Permet de spécifier une ou plusieurs conditions sur des groupes d'enregistrements générés par la clause GROUP BY
HEX	Renvoie en hexadécimal un entier (exprimé en base 10).
IN	Sélectionne des enregistrements dont les valeurs correspondent à une liste de valeurs spécifiée <i>P12</i> Utilisé dans les sous-requêtes
INNER JOIN	Réalise une jointure interne
INSERT	Ajoute un enregistrement dans un fichier de données
INSTR	Renvoie la position d'une chaîne de caractères
INTO	Utilisé avec l'instruction INSERT Définit le fichier utilisé
IS NULL	Sélectionne des enregistrements pour lesquels la valeur d'une rubrique est nulle
LEFT	Extrait les premiers caractères du contenu d'une expression
LEFT OUTER JOIN	Réalise une jointure externe
LEN	Renvoie la taille (le nombre de caractères) d'une expression
LENGTH	Renvoie la taille (le nombre de caractères) d'une expression
LIKE	<i>P10</i> Sélectionne des enregistrements pour lesquels la valeur d'une rubrique correspond à une valeur spécifiée (avec des caractères génériques)
LOWER	<i>P12</i> Convertit en minuscules le contenu d'une expression
LPAD	Renvoie une chaîne de taille déterminée. Pour atteindre la taille requise, cette chaîne est complétée à gauche : soit par des espaces. soit par un caractère ou une chaîne donnée.
LTRIM	Renvoie une chaîne de caractères : soit sans les espaces situés à gauche. soit sans une liste de caractères.
MAX	<i>18</i> Renvoie la plus grande valeur d'une rubrique pour tous les enregistrements sélectionnés dans le fichier
MID	Extrait une sous-chaîne, à partir d'une position donnée
MIN	<i>P18</i> Renvoie la plus petite valeur d'une rubrique pour tous les enregistrements sélectionnés dans le fichier
NOT	<i>P12</i> Inverse la signification de l'opérateur logique utilisé Utilisé dans les sous-requêtes
OCT	Renvoie en octal un entier (exprimé en base 10)
OCTET LENGTH	Renvoie la taille (le nombre de caractères) d'une expression

ON

<u>ON</u>		Utilisé lors de la création d'une jointure
OR	<i>P14</i>	Combine plusieurs conditions de sélection (l'une ou l'autre)
<u>ORDER BY</u>	<i>p10</i>	Trie les enregistrements sélectionnés
<u>PATINDEX</u>		Renvoie la position de la première occurrence d'une chaîne de caractères
<u>POSITION</u>		Renvoie la position d'une chaîne de caractères dans une expression
<u>REPLACE</u>		Renvoie une chaîne de caractères : soit en remplaçant toutes les occurrences d'un mot présent dans une chaîne par un autre mot. soit en remplaçant toutes les occurrences d'un mot présent dans une chaîne.
<u>RIGHT</u>		Extrait les derniers caractères du contenu d'une expression
RIGHT OUTE R JOIN		Réalise une jointure externe
<u>RPAD</u>		Renvoie une chaîne de taille déterminée. Pour atteindre la taille requise, cette chaîne est complétée à gauche : soit par des espaces. soit par un caractère ou une chaîne donnée.
<u>RTRIM</u>		Renvoie une chaîne de caractères : soit sans les espaces situés à droite. soit sans une liste de caractères
SELECT	<i>p10</i>	Recherche des enregistrements dans un ou plusieurs fichiers
SET		Utilisé avec l'instruction UPDATE Définit les noms des rubriques utilisées
SOME		Utilisé dans les sous-requêtes
<u>SUBSTR</u>		Extrait une sous-chaîne, à partir d'une position donnée
<u>SUBSTRING</u>		Extrait une sous-chaîne, à partir d'une position donnée
<u>SUM</u>	<i>P18</i>	Renvoie la somme totale des valeurs d'une rubrique pour tous les enregistrements sélectionnés dans le fichier
<u>TOP</u>		Renvoie uniquement les n premiers enregistrements du résultat d'une requête
<u>TRANSLATE</u>		Renvoie une chaîne de caractères en remplaçant tous les caractères indiqués par d'autres caractères
<u>TRIM</u>		Renvoie une chaîne de caractères : soit sans les espaces situés à gauche et à droite. soit sans une chaîne de caractères située en début et en fin de chaîne. soit sans une chaîne de caractères située en début de chaîne. soit sans une chaîne de caractères située en fin de chaîne.
UNION	<i>P20</i>	Réalise des requêtes composées
UPDATE		Met à jour des enregistrements d'un fichier
<u>UPPER</u>	<i>P12</i>	Convertit en majuscules le contenu d'une expression
VALUES		Utilisé avec l'instruction INSERT Définit les valeurs ajoutées
WHERE	<i>p10</i>	Permet de spécifier la ou les condition(s) de sélection d'une requête

Vos notes :

Dans GEPMI...

Dans GEPMI, il y a 2 possibilités pour créer et exécuter des requêtes :

- sur chaque liste, par les « jumelles »
- dans le menu « Autres », par « impression/requête »

Les « jumelles »

Dans GEPMI, sur chaque liste il est possible d'appliquer des requêtes SQL pour sélectionner des enregistrements. Pour cela, il suffit de cliquer sur l'icône :  à gauche de la barre de menu de la liste.

A partir de là, 2 possibilités pour écrire la requête :

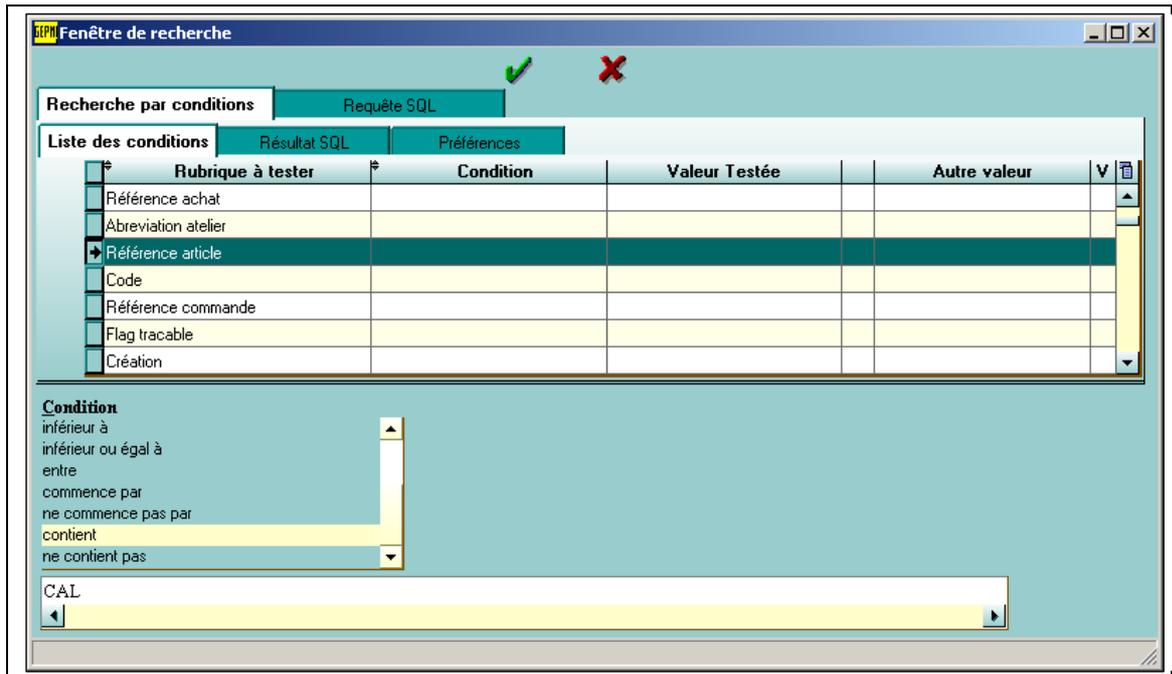
- avec l'assistant,
- directement en SQL.

1) Avec l'assistant – onglet « Recherche par conditions »

Pour écrire une requêtes SQL avec l'assistant :

1. double cliquez sur la rubrique sur laquelle va porter la condition dans le 1^{er} tableau
2. sélectionnez le type de condition dont il s'agit
3. saisissez la valeur attendue
4. validez en cliquant sur l'icône : 

Sélection des articles contenant 'CAL' dans leur référence :



Une fois la requête validée, il est possible de mémoriser la requête grâce au 2^{ème} tableau. Pour cela :

1. cliquez sur l'icône modifier : 
2. saisissez le nom de la requête
3. saisissez son code parmi ceux qui ont préalablement été renseignés dans la table des paramètres
4. cliquez sur l'icône :  pour passer les conditions déjà saisies dans la nouvelle requête

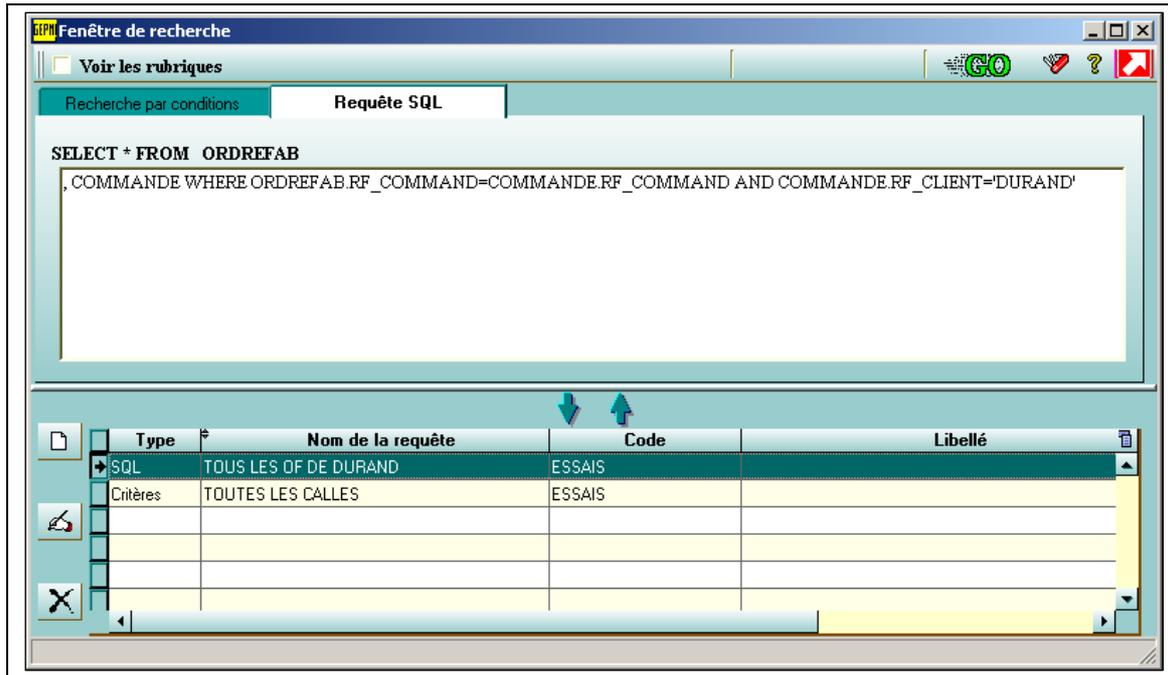
2) Directement en SQL – onglet « Requête SQL »

Pour écrire une requêtes SQL avec l'assistant :

1. double cliquez sur la rubrique sur laquelle va porter la condition dans le 1^{er} tableau
2. sélectionnez le type de condition dont il s'agit
3. saisissez la valeur attendue
4. validez en cliquant sur l'icône : 

Vos notes :

Sélection de tous les OF des commandes de DURAND



Que ce soit avec l'assistant ou en la saisissant directement, pour exécuter un requête, cliquez sur l'icône !

3) Requêtes paramétrées

Les requêtes paramétrées servent à effectuer des requêtes SQL dynamiques. Le contenu de la requête peut donc varier.

Par exemple, pour rechercher tous les OF client par client, il est possible de créer une requête par client. Il sera néanmoins plus rapide de créer une requête sélectionnant les OF d'un client et de demander à l'utilisateur de quel client il s'agit.

Par l'assistant ou directement par SQL, pour créer une requête paramétrée, il suffit de saisir le paramètre attendu entre accolades.

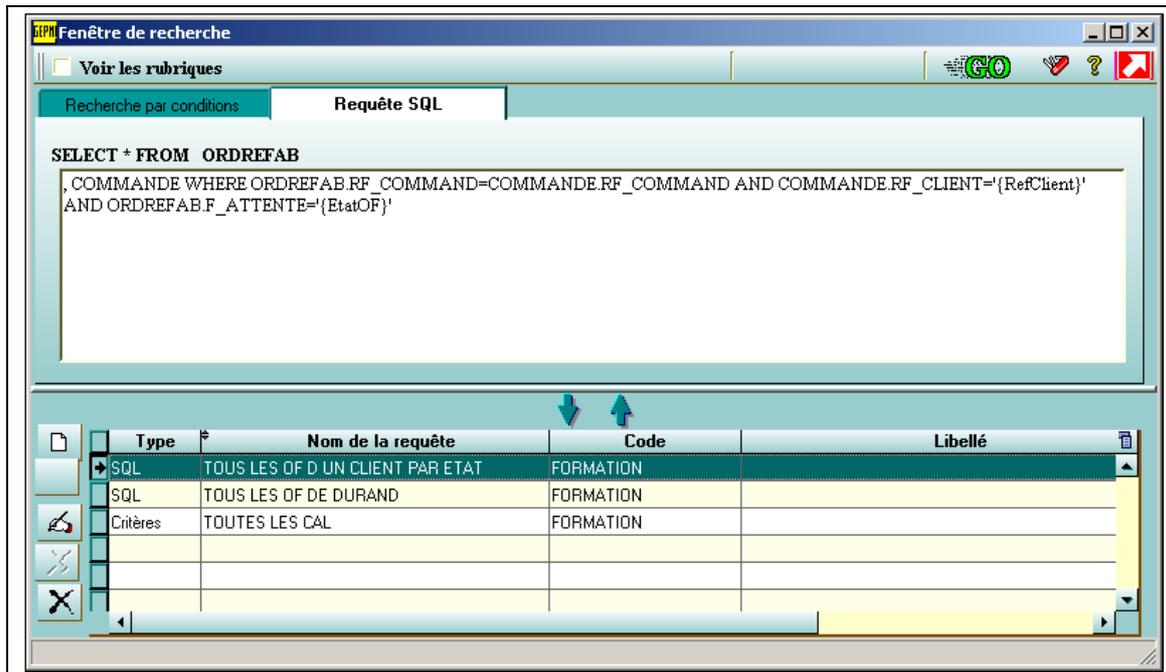
Sélection des lots pour un atelier jour par jour client :

⇒ **SELECT ***
FROM ORDREFAB, COMMANDE
WHERE COMMAND.RF_CLIENT='{RefClient}'

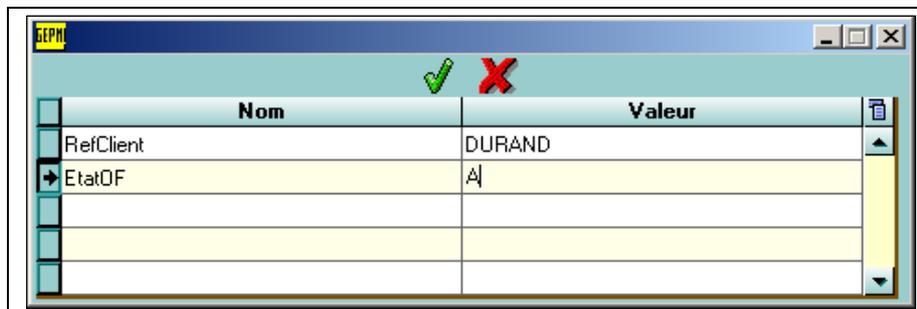
- si la rubrique sur laquelle porte la condition est une chaîne, il faut mettre les '' de par et d'autre des accolades.

Il peut y avoir plusieurs paramètres dans une même requête.

Sélection de tous les OF d'un client par état :



Dans GEPMI, que ce soit avec ou sans l'assistant, l'exécution d'une requête paramétrée ouvre une fenêtre permettant de saisir la valeur des paramètres. Le nom du paramètre sera le texte saisi entre les accolades.



Une fois les valeurs des paramètres saisies, il suffit de valider pour exécuter la requête.

« Impression/Requêtes »

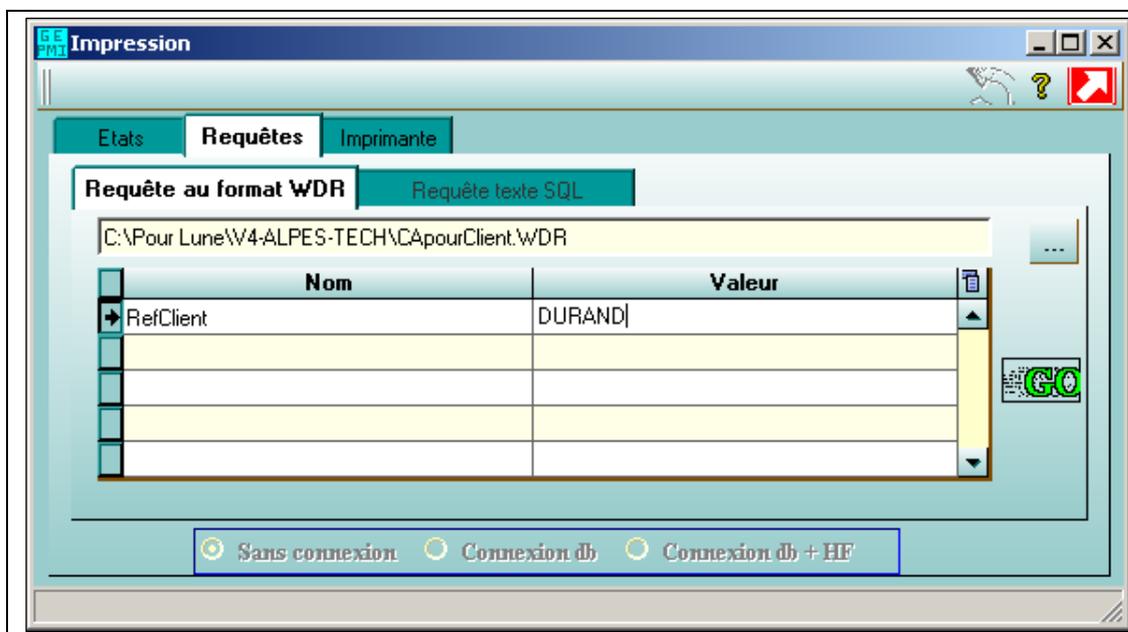
Vous avez la possibilité d'utiliser les requêtes issues de l'éditeur de requête (WDR) ou les requêtes en langage SQL.

Requête au format WDR

Pour utiliser exécuter requête qui a été créée avec l'éditeur de requête :

1. cliquez sur le bouton [...],
2. si cette requête attend des paramètres, alors la liste des paramètres est affichée dans le tableau inférieur. Il suffit alors de saisir les valeurs des différents paramètres.
3. cliquez sur l'icône  pour exécuter la requête.

Calcul du chiffre d'affaire pour un client :



Requête SQL

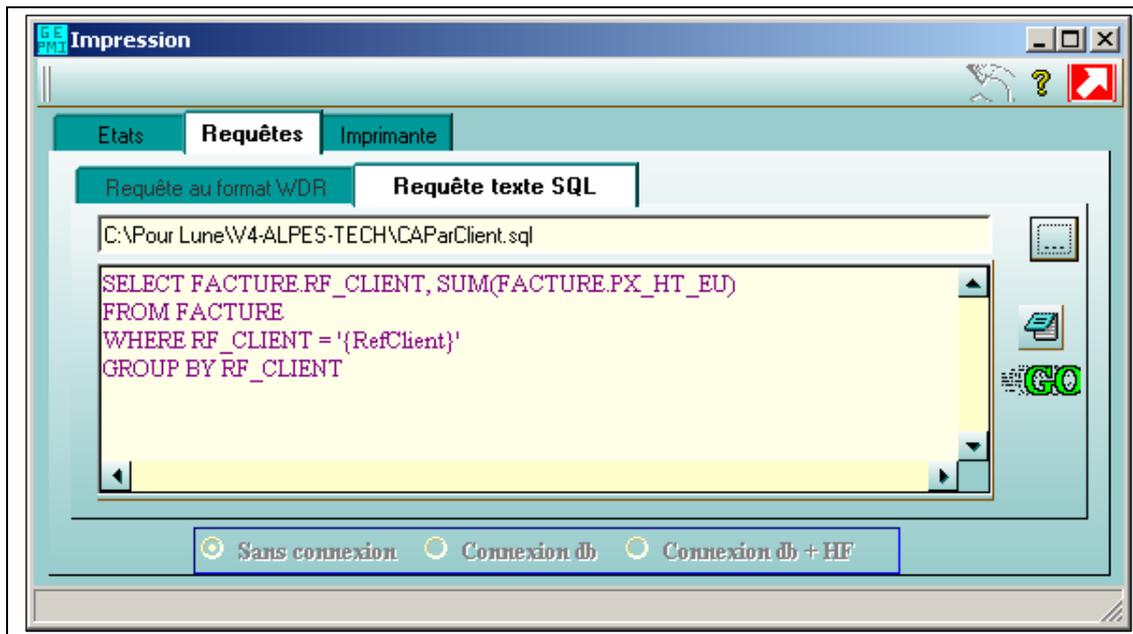
La requête peut être :

- soit issue d'un fichier, ayant l'extension '.sql', en cliquant sur le bouton [...],
- soit directement saisie dans la zone de texte ou grâce au bloc note 

Pour que GEPMI, demande les paramètres de la requête lié à l'état, il faut que celle ci porte le nom de l'état suffixé par '_requete' :

NomRequête=NomEtat_requete

Calcul du chiffre d'affaire pour un client :



Si cette requête attend au moins un paramètre, l'exécution d'une requête paramétrée ouvre une fenêtre permettant de saisir la valeur des paramètres.



Alors « Jumelles » ou « Impression/requêtes » ?

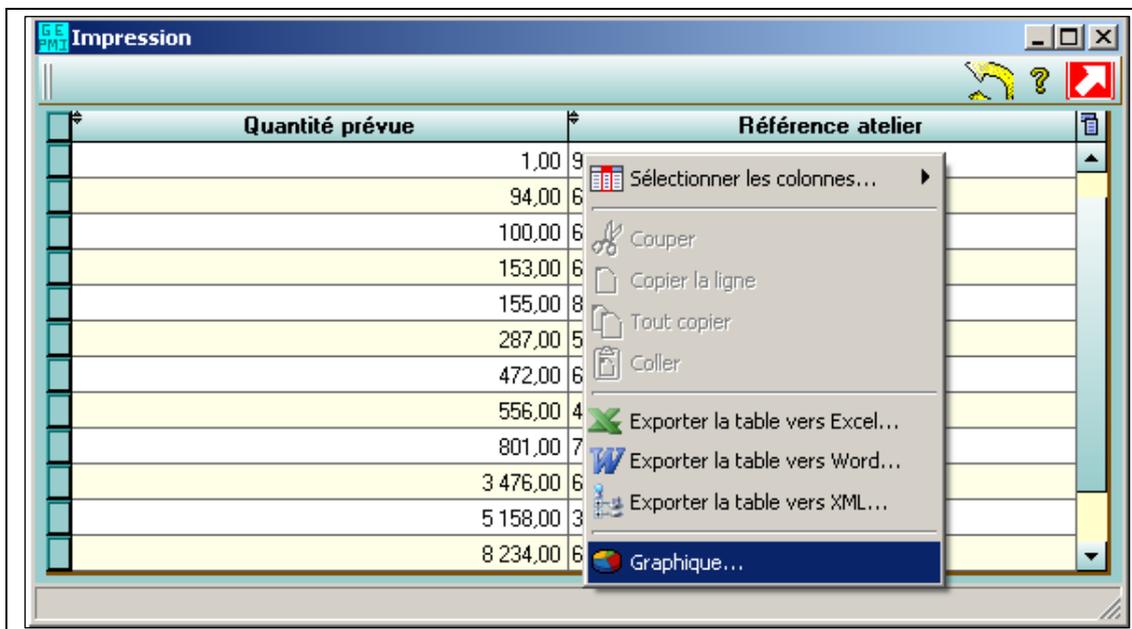
Les « jumelles » permettent de sélectionner la fiche sur laquelle on va travailler. Ainsi, il n'y a pas le choix des rubriques à sélectionner puisqu'elles sont toutes sélectionnées (par le caractère *). De plus, les requêtes devront obligatoirement porter sur le fichier correspondant à la liste par laquelle on accède à la fenêtre de recherche, éventuellement en liaison à d'autres fichiers.

Les requêtes par « impression/requêtes » permettent de traiter des données auxquelles on a pas accès directement par les fiches dans GEPMI. Elles ont plus pour but de faire ressortir certaines analyses. Par exemple, le chiffre d'affaire par client, la charge pas atelier...

Comme dans n'importe quelle table de GEPMI, le résultat des « jumelles » ou d' « impression/requête » peut être exporter sous Excel, Word, XML ou en graphique en courbe, histogramme, secteur... d'un simple clic droit.

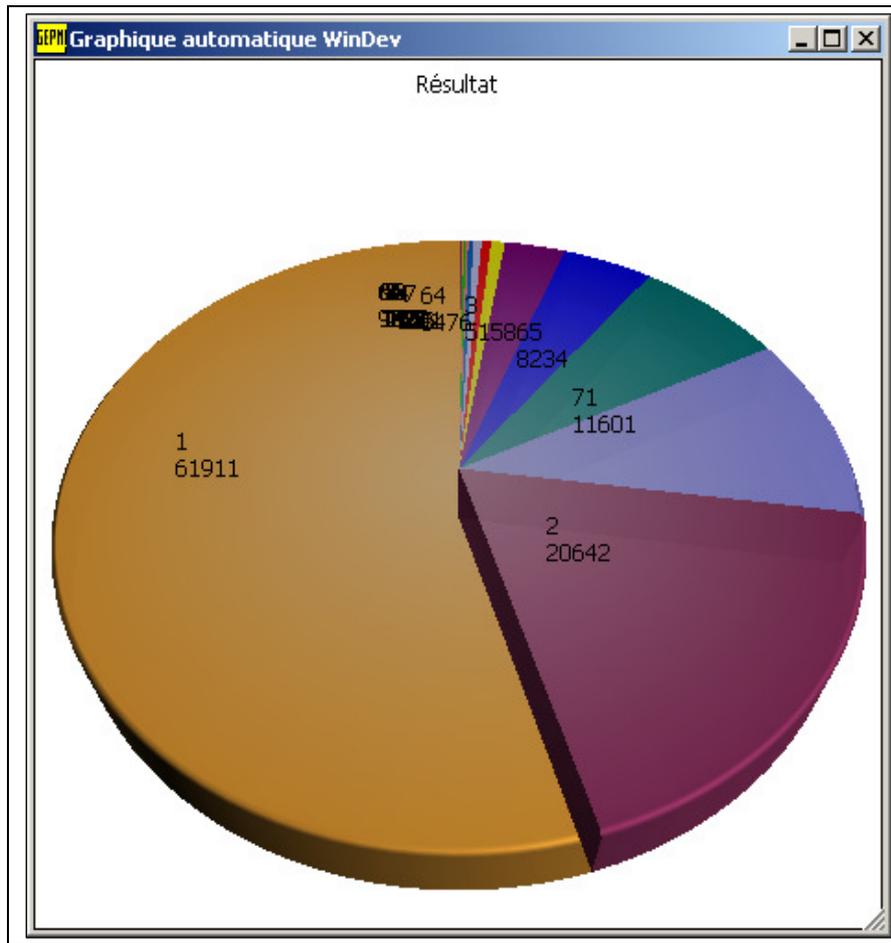
Sélection de la charge (en quantité) prévue par atelier dans « impression/requêtes » :

```
⇒ SELECT SUM(QT_PREVUE), RF_ATELIER
FROM ORDREFAB
WHERE F_ATTENTE IN ('A','N')
GROUP BY RF_ATELIER
```



Quantité prévue	Référence atelier
1,00	9
94,00	6
100,00	6
153,00	6
155,00	8
287,00	5
472,00	6
556,00	4
801,00	7
3 476,00	6
5 158,00	3
8 234,00	6

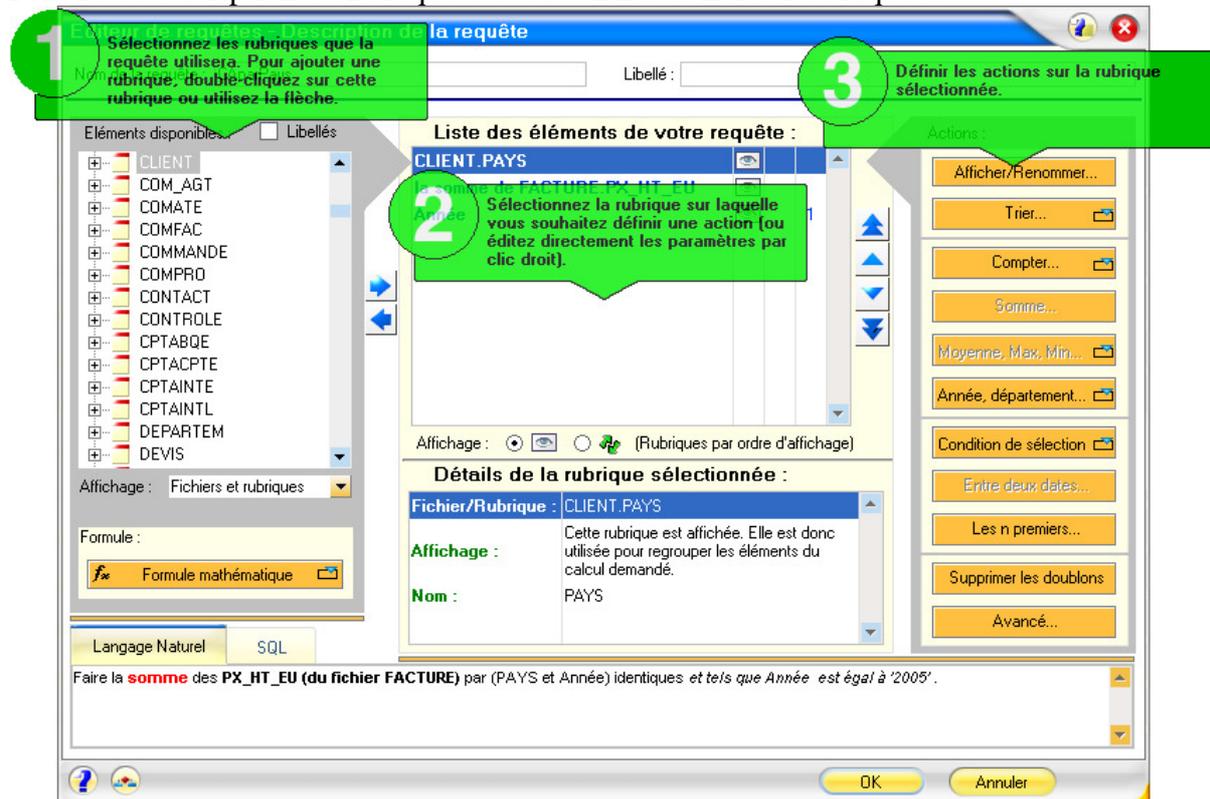
Il est ainsi possible de visionner graphiquement la répartition de la charge prévue par atelier d'un simple clic droit :



WDEtat 9 Editeur de requêtes

L'éditeur de requêtes de Windev 9 permet de créer automatiquement des requêtes sur des fichiers de données. Les requêtes générées peuvent être testée à partir de GEMI avec la « impression/requête ».

Fenêtre de description d'une requête avec WDEtat 9 Editeur de requêtes :

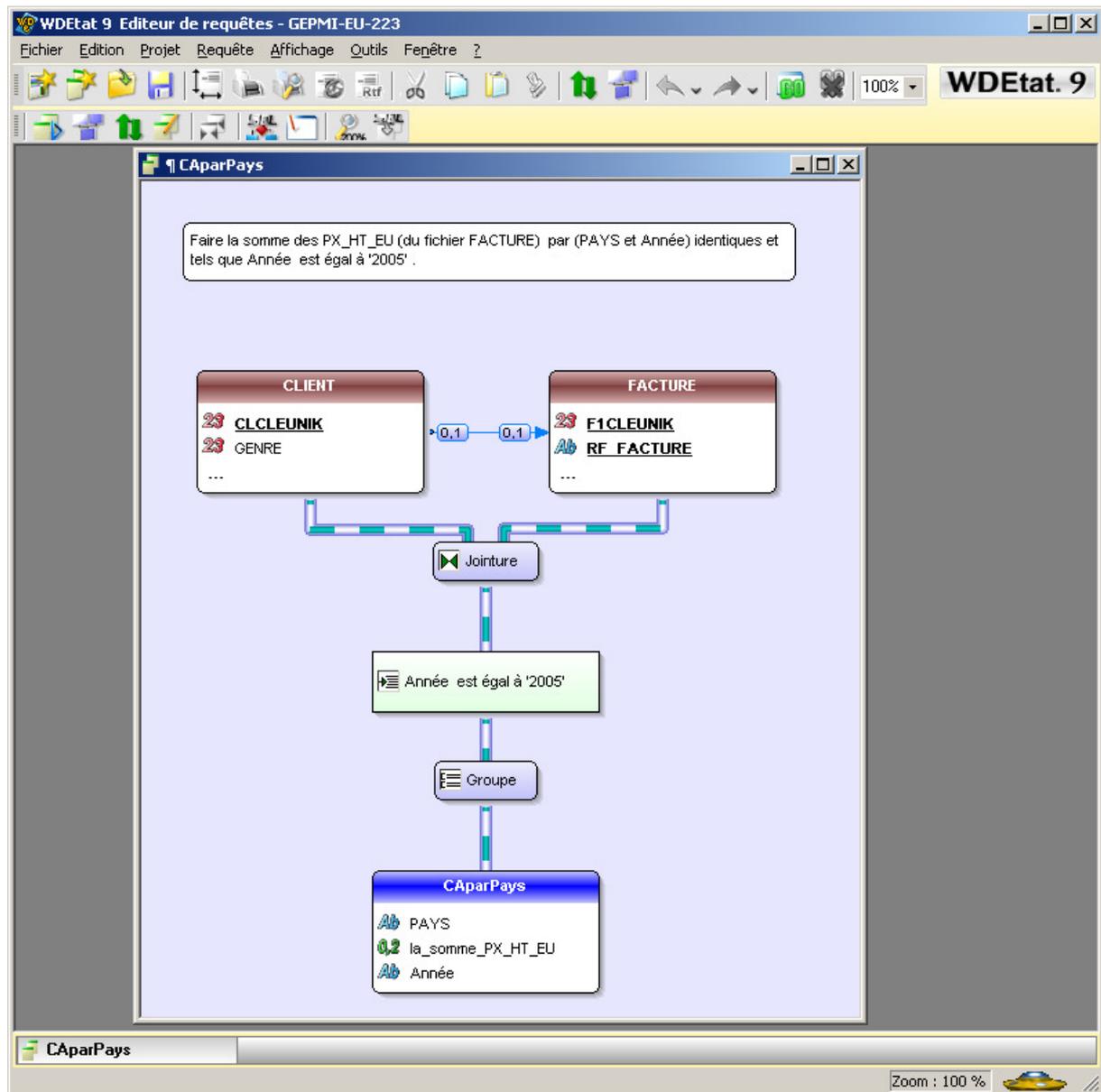


Exemple à partir du calcul du CA par pays pour l'année 2005 :

1. **SELECT** CLIENT.PAYS AS PAYS, SUM(FACTURE.PX_HT_EU) AS SOMME,
LEFT(FACTURE.DT_EFFECTIVE) AS ANNEE
FROM CLIENT, FACTURE
WHERE CLIENT.RF_CLIENT=FACTURE.RF_CLIENT **AND** ANNEE='2005'
GROUP BY CLIENT.PAYS, ANNEE

Pour créer un requête de sélection en utilisant un assistant simplifié :

1. sélectionnez les rubriques à visualiser dans les fichiers concernés,
2. sélectionnez l'ordre dans lequel les rubriques vont être visualisées,
3. choisissez un tri éventuel,
4. donnez un nom à la requête,
5. visualisez graphiquement la requête.



Attention : par défaut, c'est l'assistant qui définit les rubriques sur lesquelles se font les jointures entre fichiers. Il est possible de modifier les jointures dans le menu :
 « Requête / Editer les liaisons/jointures... »

A partir de la fenêtre de description des jointures, il est possible d'ajouter, de supprimer des rubriques de liaisons :



A tous moments, il est possible de modifier la requête à partir du menu :
 « Requête / Description de la requête... »

Éditeur de requêtes - Description de la requête

Nom de la requête : CÀparPays Libellé :

Éléments disponibles : Libellés

- ABREV
- NOM_RESP
- PRENOM
- ADRESSE1
- ADRESSE2
- ADRESSE3
- CD_POSTAL
- VILLE
- PAYS
- NO_TEL
- NO_TEL2
- NO_TEL3
- NO_FAX

Affichage : Fichiers et rubriques

Formule : Formule mathématique

Langage Naturel SQL

Liste des éléments de votre requête :

CLIENT.PAYS	<input type="checkbox"/>		
la somme de FACTURE.PX_HT_EU	<input type="checkbox"/>		
Année	<input type="checkbox"/>	1	

Affichage : (Rubriques par ordre d'affichage)

Détails de la rubrique sélectionnée :

Fichier/Rubrique :	CLIENT.PAYS
Affichage :	Cette rubrique est affichée. Elle est donc utilisée pour regrouper les éléments du calcul demandé.
Nom :	PAYS

Actions :

Afficher/Renommer...

Trier...

Compter...

Somme...

Moyenne, Max, Min...

Année, département...

Condition de sélection

Entre deux dates...

Les n premiers...

Supprimer les doublons

Avancé...

Faire la somme des PX_HT_EU (du fichier FACTURE) par (PAYS et Année) identiques et tels que Année est égal à '2005'.

OK Annuler