

*Année Universitaire 2005 - 2006*

*Master Sciences et Technologie  
mention Informatique, Mathématiques et leurs Applications  
spécialité professionnelle en Génie Informatique*



*Stage de fin d'études*

Migration d'une base  
**Hyper File**  
vers  
**SQL**

*réalisé à SATTI Informatique à La Rochelle*

GAËL DUCERF

Stage de fin d'études

# Migration d'une base Hyper File vers SQL

réalisé à  
SATTI Informatique  
à La Rochelle

Master Sciences et Technologie  
mention Informatique, Mathématiques et leurs Applications  
spécialité professionnelle en Génie Informatique

du 6 février au 24 juin 2006

Année Universitaire 2005 - 2006

# *Remerciements*

Je tiens à adresser mes remerciements aux personnes qui m'ont permis d'effectuer ce stage au sein de la société SATTI Informatique.

*Je remercie particulièrement :*

- Vincent HIDALGO, le PDG de SATTI Informatique, pour son accueil au sein de son entreprise.
- Ludovic GAUDIN, mon maître de stage, qui a participé à mon intégration au sein de l'équipe de développement et a suivi le déroulement de mon stage.
- L'ensemble du service développement, qui s'est également investi dans le bon déroulement de mon stage.
- L'ensemble des autres services pour leur accueil dans l'entreprise.

# *Résumé*

Dans un premier temps, je me suis penché sur les systèmes de gestion de bases de données MySQL, PostgreSQL et Microsoft SQL Server aux niveaux fonctionnalités, support et compatibilité avec WinDev.

Puis je me suis intéressé aux différences de fonctionnement entre les bases Hyper File et les bases SQL afin d'établir les méthodes de migration envisageables.

Et enfin, après la migration d'un échantillon de base de données Hyper File vers MySQL et Microsoft SQL Server, il a paru nécessaire de tester les différents types d'accès aux bases de données fournis par WinDev.

En parallèle, j'ai réalisé divers travaux, dont une étude comparative des progiciels de gestion intégrés spécialisés dans la gestion de production assistée par ordinateur. J'ai également adapté et débogué une classe permettant de créer des plannings de tâches. Puis j'ai créé une classe permettant de dialoguer avec des terminaux de saisie de temps en confection.

# *Summary*

Initially, I compared functionalities, support and compatibility with WinDev of MySQL, PostgreSQL and Microsoft SQL Server databases management systems.

Then I was interested in differences between Hyper File databases and SQL databases in order to establish the possible migration methods.

And finally, after migrating of an Hyper File database sample to MySQL and Microsoft SQL Server, it was necessary to test the various database access methods provided by WinDev.

In parallel, I completed different works. The first was a comparative study of Enterprise Resource Planning specialized in computer-assisted production management. I also adapted and debugged a class allowing to create tasks schedule. Then I created a class allowing to dialogue with time collection terminals in clothes industry.

# Sommaire

Remerciements.....	2
Résumé.....	3
Summary.....	4
Introduction.....	6
Présentation de SATTI Informatique.....	7
Présentation du sujet de stage.....	8
Le Progiciel de Gestion Intégré G.I.S.....	9
Module de Suivi de la Relation Commerciale.....	9
Module de gestion commerciale.....	10
Module de gestion de production.....	10
Module de comptabilité.....	10
Module de paie.....	10
Étude comparative des Systèmes de Gestion de Bases de Données.....	11
Introduction.....	11
Les Systèmes de Gestion de Bases de Données étudiés.....	12
Migration.....	18
Remontée des données.....	22
Résultats.....	28
Conclusion.....	29
Réalisations diverses.....	30
Comparatif des Progiciels de Gestion Intégrés.....	30
Terminaux de saisie de temps en confection.....	31
Planning pour G.I.S. Gestion de production.....	34
Microsoft SharePoint.....	35
Conclusion.....	36
Annexes.....	37
Glossaire.....	37
Types d'accès aux SGBD fournis par WinDev.....	39
Résultats des remontées.....	41

# Introduction

Ce rapport est l'aboutissement de 20 semaines de stage effectuées au sein de l'entreprise SATTI Informatique. Ce stage intervient dans le cadre du Semestre 4 de la formation Master Sciences et Technologie mention Informatique, Mathématiques et leurs Applications spécialité professionnelle en Génie Informatique.

Durant ce stage, j'ai réalisé une étude comparative des SGBD (*Systèmes de Gestion de Bases de Données*) de type SQL interfaçable avec l'Atelier de Génie Logiciel WinDev en vue de remplacer le SGBD Hyper File actuellement utilisé par SATTI dans le développement de leur progiciel de gestion.

Dans le cadre de mon intégration au sein de l'entreprise, j'ai également été amené à m'investir sur des dossiers n'ayant pas de rapport direct avec mon sujet de stage. Ils sont présentés dans la partie « Réalisations diverses ».

Je vais vous conduire au coeur de mon sujet de stage en passant par une brève présentation de ce dernier.

Avant ceci, je vous propose la lecture d'une présentation de la société SATTI Informatique.

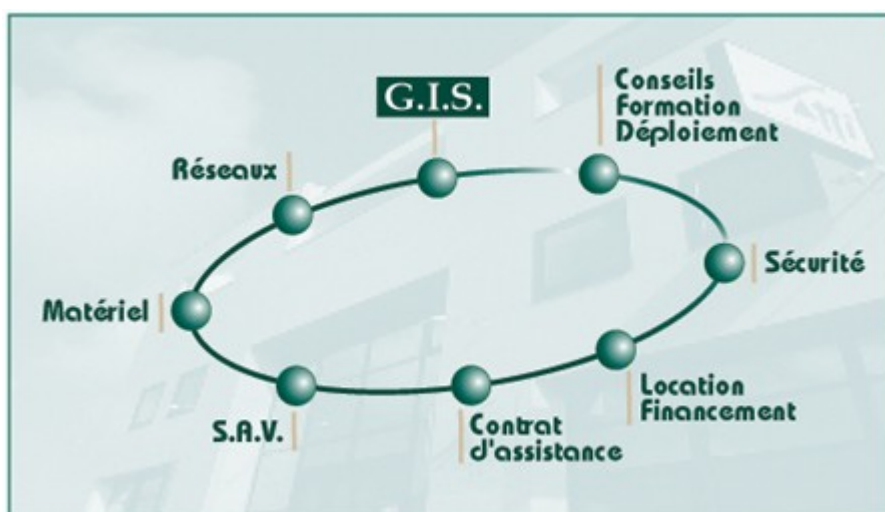


# Présentation de SATTI Informatique

La société de services et de distribution informatique SATTI est née en 1977 à La Rochelle et connaît depuis une croissance régulière. Elle est aujourd'hui dirigée par Vincent HIDALGO et réalise un chiffre d'affaire de 5 M€ pour un effectif de 20 personnes.

*Elle est spécialisée dans deux domaines informatiques :*

- Le développement et la vente de solutions de gestion
- La mise en place et le support informatique de réseaux en partenariat avec les acteurs leaders du marché.



Elle propose ses prestations sur la région auprès de plus de 1200 clients professionnels PME/PMI et collectivités. SATTI assure à la fois le support technique, la veille technologique et l'assistance.

Cette société est associée à divers partenaires, éditeurs de logiciels ou fabricants de matériel informatique. Dans le cadre de ces partenariats, les ingénieurs et consultants SATTI suivent régulièrement des plans de formation et de certification.



# *Présentation du sujet de stage*

Le stage vient s'intégrer dans un projet de l'entreprise qui envisage de changer le Système de Gestion de Bases de Données du progiciel de gestion intégré qu'elle développe et diffuse auprès des PME/PMI de la région.

Le stage consiste donc à réaliser l'étude comparative et les tests de performances de Systèmes de Gestion de Bases de Données Relationnelles du marché, à savoir, MySQL, PostgreSQL et Microsoft SQL Server.

Après cette étude comparative, il s'agit de faire le choix d'un Système de Gestion de Bases de Données Relationnelles interfaçable avec Atelier de Génie Logiciel WinDev.

Ensuite la migration de G.I.S., le progiciel de gestion intégré de SATTI, pourra être planifiée.

Je vous propose ci-après une présentation de G.I.S..

# Le Progiciel de Gestion Intégré G.I.S.

G.I.S. (*Gestion Intégrée SATTI*) est une suite progiciels modulaire et évolutive, qui intègre toutes les principales fonctions et spécificités nécessaires à la gestion d'une entreprise. Elle centralise toutes les informations de l'entreprise. Cette suite est divisée en plusieurs modules présentés ci-dessous.

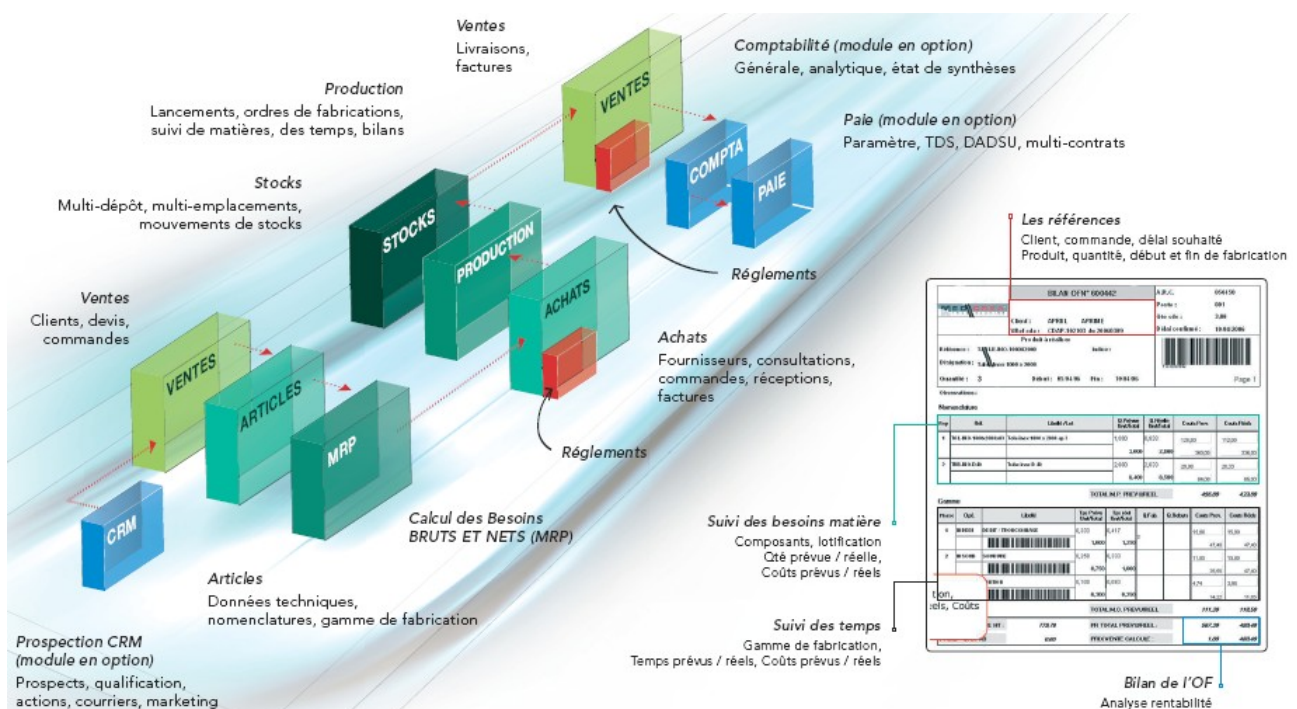


Illustration 1: Suivi de la chaîne de fabrication

## Module de Suivi de la Relation Commerciale

G.I.S Suivi de la Relation Commerciale est un outil de suivi des actions commerciales. Il permet de consulter toutes les tâches programmées, de la relance téléphonique à la prise de rendez-vous en passant par l'envoi de documents. Il permet d'accéder à la fiche société, de consulter toutes les informations enregistrées : le chiffre d'affaire, le potentiel, les interlocuteurs, l'historique des contacts commerciaux, les projets identifiés.

Lors d'une opération commerciale, G.I.S Suivi de la relation Commerciale permet de définir des critères de sélection, d'associer des actions à mener et d'affecter ces tâches à des collaborateurs.

Grâce à un module de Couplage Téléphonique Informatique, on peut lancer un appel téléphonique par un simple clic sur la fiche prospect, client ou fournisseur. Et lorsque l'un d'entre eux appelle, les informations essentielles de sa fiche apparaissent automatiquement à l'écran.

## *Module de gestion commerciale*

Ce module offre la possibilité de gérer toutes les données commerciales de l'entreprise. Il centralise les informations d'un établissement. Il gère toutes les transactions commerciales que ce soit les ventes, les achats ou les factures à partir du premier contact jusqu'au règlement final.

Il permet de gérer les stocks en visualisant les réservations et la disponibilité des articles, le seuil de sécurité, et leur rotation. Les délais de livraison sont contrôlés en suivant les reliquats clients et fournisseurs. La traçabilité est assurée par les numéros de lots et de série. Il permet de définir les tarifs et les conditions commerciales.

Ce module offre des fonctionnalités pratiques comme l'envoi de documents par télécopie et messagerie, ou leur export vers les logiciels courants du marché.

Il permet de dresser un état des lieux rapide de l'activité de l'entreprise via des tableaux de bord et des états statistiques paramétrables.

## *Module de gestion de production*

G.I.S Suivi de Production est un module de gestion de production destiné aux PMI. A partir des nomenclatures et des gammes de fabrication, G.I.S Production permet d'établir et de suivre les prix de revient, le plan de fabrication, les stocks prévisionnels et le plan de charge.

Les commandes clients sont lancées en fabrication, et les ordres de fabrication générés permettent un suivi des temps passés depuis l'atelier.

Le calcul des besoins offre les analyses sur les stocks de matières premières.

Une fois le travail effectué, la préparation du bon de livraison est effectuée, les reliquats sont suivis, les informations peuvent être données en temps réel aux clients et exploitées par la gestion commerciale.

## *Module de comptabilité*

G.I.S comptabilité est un outil personnalisable et adaptable au type de l'entreprise. On y trouve les fonctionnalités de base, de la création du plan comptable au bilan, en passant par le rapprochement bancaire, la saisie des règlements, le multi-exercices, le suivi des budgets et la gestion des portefeuilles d'effets.

## *Module de paie*

Ce module permet de paramétrer la paie pour l'adapter à tous les profils d'une société et à tous les changements sociaux. Il calcule et édite les bulletins, génère les déclarations obligatoires, suit la gestion des congés et les comptes épargne temps, calcule les provisions, ainsi que le prix de revient horaire par salarié. Il dispose d'une gestion du personnel (fiche signalétique, évolution des rémunérations...)

# *Étude comparative des Systèmes de Gestion de Bases de Données*

## *Introduction*

### *But de l'étude*

Dans le cadre de l'évolution du progiciel de gestion intégré G.I.S. développé par SATTI, l'équipe de développement s'interroge sur la nécessité de changer de SGBD (*Système de Gestion de Base de Données*). Leur progiciel repose actuellement sur une base de données Hyper File. Quelques clients semblent être intéressés par une migration vers un SGBD de type SQL. Un système SQL pourrait apporter d'avantage de fiabilité dans le stockage des données, accroître la rapidité d'accès à ces données, mais aussi offrir une meilleure interopérabilité de G.I.S. avec les autres logiciels présents dans les entreprises.

D'une part, l'équipe de développement G.I.S. souhaite connaître l'étendu des modifications à apporter à G.I.S. dans le cadre d'une migration vers une base de données de type SQL. Et d'autre part, elle cherche à connaître les performances des SGBD SQL quand on les utilise via des applications développées sous WinDev.

### *Principe de l'étude*

Le progiciel de gestion G.I.S. est développé grâce à l'Atelier de Génie Logiciel WinDev. Il faut donc sélectionner des SGBD compatible avec WinDev. Ensuite il s'agit d'étudier le fonctionnement du SGBD Hyper File utilisé par G.I.S., puis d'évaluer les modifications à apporter dans le fonctionnement de G.I.S. pour qu'il puisse utiliser un SGBD de type SQL. Des tests de remontée de données ont ensuite été prévus pour déterminer le SGBD qui offre les meilleures performances. Une décision sera ensuite prise quant à la nécessité d'utiliser un SGBD de type SQL. Si l'utilisation d'un tel type de SGBD se révèle nécessaire, il s'en suivra le choix du SGBD SQL pour finalement lancer la migration de G.I.S..

# *Les Systèmes de Gestion de Bases de Données étudiés*

L'étude commence par la recherche des SGBD présents sur le marché compatibles avec WinDev. On me demande de me pencher plus particulièrement sur les SGBD MySQL, PostgreSQL et Microsoft SQL Server.

## *MySQL<sup>1</sup>*

MySQL est un gestionnaire de bases de données libre. Il est très utilisé dans les projets libres mais également dans le milieu industriel.

MySQL est un serveur de bases de données relationnelles SQL. Il est multi-thread et multi-utilisateurs. C'est un logiciel libre développé sous double licence en fonction de l'utilisation, commerciale ou non, qui en est faite. Si c'est le cas, la licence est payante, sinon elle est libre. Le prix de ce système de gestion de bases de données est d'environ 500 €<sup>HT</sup> par an et par serveur<sup>2</sup> pour une utilisation commerciale. Ce prix inclut également un support technique par MySQL AB France.

MySQL fait partie du quatuor LAMP : Linux, Apache, MySQL, PHP. Le couple PHP/MySQL est très utilisé sur internet et proposé par la majorité des hébergeurs. L'utilisation de MySQL en entreprise représente 15,8 %<sup>3</sup> des serveurs de bases de données en production.

## *Historique*

La première version de MySQL est apparue le 23 Mai 1995. Il a été créé pour un usage personnel à partir de mSQL que les concepteurs trouvaient trop lent et trop rigide. Ils ont donc créé une nouvelle interface SQL en gardant la même API que mSQL.

En 2003, MySQL AB et SAP concluent un accord. Le résultat s'appelle MaxDB. Il est issu de l'intégration du système SAP DB dans MySQL.

La dernière version de MySQL qui est sortie est la 5.0, la version 5.1 est en cours de préparation.

## *Systèmes supportés*

MySQL fonctionne sur beaucoup de plateformes différentes, incluant AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetBSD, OpenBSD, OS/2 Warp, SGI Irix, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows 95, 98, NT, 2000 et XP.

Les bases de données sont accessibles en utilisant les langages de programmation C, C++, Delphi / Kylix, Eiffel, Java, Perl, PHP, Python, Ruby, Tcl et W-Langage; une interface spécifique est disponible pour chacun d'entre eux. Une interface ODBC\* appelée MyODBC est aussi disponible.

---

1 Voir le site français de MySQL : <http://www.mysql.fr>

2 Voir MySQL Online Shop. Disponible sur : <https://shop.mysql.com>

3 Référence : Alain CLAPAUD. SGBD : l'open source stimule le marché. 01 Réseaux. 25/01/2006. Disponible sur : <http://www.01net.com/article/302634.html>

\* Voir glossaire en annexe

## *Moteurs de base de données inclus*

*MySQL propose plusieurs moteurs pour gérer les bases de données :*

- MyISAM : Moteur par défaut de MySQL. Il est le plus simple à utiliser et mettre en oeuvre. Il utilise plusieurs fichiers qui grandissent au fur et à mesure que la base grossit. Il ne supporte ni les transactions ni les clefs étrangères.
- InnoDB : Moteur créé et maintenu par InnoDB récemment racheté par Oracle. Il gère les transactions et les clefs étrangères. En contrepartie, les bases qui l'utilisent occupent bien plus d'espace sur le disque.
- BerkeleyDB
- MERGE : Moteur fait pour fusionner plusieurs tables qui doivent être identiques.
- ARCHIVE : Moteur adapté pour l'archivage de données. Les lignes sont compressées au fur et à mesure qu'elles sont insérées.
- HEAP : Moteur où les tables sont stockées uniquement en mémoire.
- CSV : Moteur utilisant des fichiers textes (au format CSV) comme stockage.
- BLACKHOLE : Moteur réceptionnant les données , les transférant mais ne les stockant pas. Il peut être utilisé comme répéteur ou comme filtre de données.
- ISAM : Moteur d'origine de MySQL, maintenant obsolète et remplacé par MyISAM. Il est resté pour des raisons de compatibilité.
- NDB (uniquement dans la version MaxDB) : Network DataBase. Moteur de base de données gérant le clustering.

## *Outils associés à MySQL :*

*Il existe quelques outils permettant de concevoir, modifier, ou administrer les bases de données MySQL :*

- PhpMyAdmin, un outil d'administration de bases de données MySQL écrit en PHP.
- MySQL Administrator, un logiciel d'administration de bases de données développé par MySQL AB.
- MySQL Query Browser, une interface graphique pour maintenir la base et y faire des requêtes développé par MySQL AB.
- MySQL Workbench est un logiciel de conception et de modélisation de base de données développé par MySQL AB.
- XAMPP est une solution web complète (PHP, MySQL, Apache, Mercury) sous licence GNU General Public License aussi bien pour Linux que pour Windows.
- eSKUeL est une interface web PHP d'administration d'une base MySQL programmée en PHP.

## *Utilisation avec WinDev*

WinDev possède un accès natif qui lui permet d'interagir avec MySQL grâce à des ordres W-Langage. Par défaut, WinDev crée des bases de données utilisant le moteur InnoDB. Si ce moteur

n'est pas disponible sur la version de MySQL installée, il utilise alors le moteur MyISAM.

## *PostgreSQL<sup>4</sup>*

PostgreSQL est un système de gestion de base de données relationnelle et objet. C'est un outil libre disponible selon les termes d'une licence de type BSD.

PostgreSQL représente 4 %<sup>5</sup> des SGBD en production. Sun Microsystems Inc. propose depuis peu un support 24h/24 7j/7 pour ce SGBD mais seulement sur leur système d'exploitation Solaris.

PostgreSQL est totalement libre, et ne nécessite pas d'acheter une licence d'utilisation.

### *Historique*

L'histoire de PostgreSQL remonte à la base de données Ingres, développée à Berkeley par Michael Stonebraker. Lorsque ce dernier décida en 1985 de recommencer le développement de zéro, il nomma le logiciel Postgres, comme raccourci de Post-Ingres. Lors de l'ajout des fonctionnalités SQL en 1995, Postgres fut renommé Postgres95. Ce nom fut changé à la fin de 1996 en PostgreSQL.

La dernière version de PostgreSQL qui est sortie est la 8.1.

### *Fonctionnalités*

PostgreSQL peut stocker plus de types de données que les types traditionnels entier, caractères, etc. L'utilisateur peut créer des types, des fonctions, utiliser l'héritage de type, etc.

PostgreSQL est pratiquement conforme (de plus en plus conforme) aux normes ANSI SQL 89, SQL 92 (SQL 2) et SQL 99 (SQL 3). Il fonctionne sur diverses plates-formes matérielles sous différents systèmes d'exploitation.

PostgreSQL fonctionne sur Solaris, SunOS, Mac OS X, HP-UX, AIX, Linux, Irix, Digital Unix, BSD, NetBSD, FreeBSD, OpenBSD, SCO unix, NeXTSTEP, UnixWare et toutes sortes d'Unix. Depuis la version 8.0, PostgreSQL fonctionne également nativement sur Windows.

### *Outils associés à PostgreSQL*

*Il existe quelques outils permettant de concevoir, modifier, ou administrer les base de données PostgreSQL :*

- DBLink est un module permettant de travailler sur deux bases différentes
- ECPG est un outil de SQL embarqué.
- En Java, PostgreSQL peut être utilisé de façon transparente avec le standard JDO.
- Le logiciel pgAdmin, actuellement en version 3, permet de piloter les base de données PostgreSQL via une interface graphique. Ce programme est disponible sur les plateformes Windows et Unix.
- phpPgAdmin est une application Web écrite en PHP pour gérer des bases PostgreSQL.

---

4 Voir le site français de PostgreSQL : <http://www.postgresqlfr.org>

5 Référence : Alain CLAPAUD. SGBD : l'open source stimule le marché. 01 Réseaux. 25/01/2006. Disponible sur : <http://www.01net.com/article/302634.html>

- PostGIS est le module spatial qui confère à PostgreSQL le statut de SGDBR spatial.

### *Utilisation avec WinDev*

Windev ne supporte pas PostgreSQL nativement, mais peut accéder à ce type de bases de données grâce à ODBC\* en utilisant les ordres SQL. L'accès est également possible grâce au support ODBC via OLEDB\* en utilisant les ordres W-Langage.

### *Microsoft SQL Server<sup>6</sup>*

SQL Server est développé et commercialisé par Microsoft. Ce SGBD fonctionne uniquement sous Windows. il représente 24,9 %<sup>7</sup> des SGBD en production. Le support est assuré par Microsoft et son réseau de partenaires.

### *Historique*

Il était développé à l'origine par Microsoft et Sybase avant la sortie de sa version 6.0. Suite à un différent entre les deux sociétés, chacune a continué le développement de son côté. Microsoft a choisi de le commercialiser sous le nom de SQL Server.

La dernière version de SQL Server qui est sortie est la 2005.

### *Fonctionnement*

SQL Server est un SGBD relationnel et transactionnel. Il s'appuie par défaut sur le système d'utilisateurs de Windows. Comme il existe des cas où l'utilisateur ne peut pas être identifié par son login Windows, une méthode d'identification directe à SQL Server peut être mise en place.

### *Outils associé à Microsoft SQL Server*

*Il existe quelques outils permettant de concevoir, modifier, ou administrer les base de données Microsoft SQL Server :*

- Entreprise manager, jusqu'à la version 2000, cet outil sert à gérer la base de données de façon entièrement graphique. Il permet en outre d'accéder et de configurer tous les outils annexes de SQL Server.
- Query Analyzer (Analyseur de requête), il s'agit d'un outil qui permet d'exécuter des requêtes sur un serveur SQL. Il permet aussi d'analyser le fonctionnement des requêtes pour optimiser la requête elle-même ou la base de données qu'il interroge.
- Profiler (générateur de profil), cet outil permet de capturer l'activité d'une base de données. Il permet aux administrateurs de vérifier les éléments et les requêtes qui prennent du temps sur la base de données.

---

<sup>6</sup> Voir le site français de Microsoft SQL Server : <http://www.microsoft.com/france/sql/>

<sup>7</sup> Référence : Alain CLAPAUD. SGBD : l'open source stimule le marché. 01 Réseaux. 25/01/2006. Disponible sur : <http://www.01net.com/article/302634.html>



## *Licences*

*SQL Server supporte deux systèmes de licences :*

- Licence par utilisateur : SQL Server peut utiliser tous les processeurs du serveur mais est limité au nombre d'utilisateurs spécifié. Chaque personne physique se servant d'une application qui utilise SQL Server est considérée comme un utilisateur de la base de données
- Licence par processeur : SQL Server utilise le nombre de processeurs spécifiés dans la licence et peut accepter un nombre indéfini d'utilisateurs

## *Versions*

*Microsoft SQL Server est décliné en plusieurs versions :*

- Express Edition (anciennement MSDE) : Microsoft a édité une version gratuite composée uniquement du moteur MSDE. Elle doit en principe être distribuée conjointement à une autre application. Microsoft la distribue avec la suite Microsoft Office. Cette version est limitée à 5 utilisateurs simultanés. Au-delà les utilisateurs supplémentaires partagent les processus des utilisateurs déjà connectés et les performances s'écroulent. De ce fait elle est adaptée à l'utilisation sur un poste de travail par un utilisateur. En outre, cette version ne permet pas de gérer des bases de données de plus de deux gigaoctets. L'indexation des champs « texte plein » est impossible dans cette version. Cette version est la seule fournie sans les clients de gestion.

- Workgroup Edition : Cette version propose une couverture fonctionnelle essentiellement axée sur des besoins transactionnels dans le cadre d'un serveur départemental ou de serveurs de bases de données indépendants les uns des autres. Les fonctions de haute disponibilité, de montée en charge, de réplication et de reporting sont limitées dans cette édition.

Ordre de prix : 3900 \$ par processeur ou 740 \$ pour 5 utilisateurs.

- Standard Edition : L'édition standard est fournie avec l'indexation texte plein elle permet en outre de gérer un grand nombre d'utilisateurs et peut utiliser plusieurs processeurs. Cette version ne peut pas s'intégrer dans un cluster et ne gère pas les cubes OLAP. Elle ne peut s'installer que sur un Windows version serveur. Elle est particulièrement adapté pour l'hébergement de bases de données qui ne nécessitent pas de statistiques. Elle peut gérer jusqu'à 8 instances simultanées sur un serveur.

Ordre de prix : 6000 \$ par processeur ou 2800 \$ pour 10 utilisateurs.

- Entreprise Edition : Cette version permet la prise en charge des cubes OLAP qui permettent de faire des statistiques poussées sur les bases de données sans faire chuter les performances du serveur lors des requêtes. Elle peut en outre fonctionner en cluster jusqu'à 8 machines. Elle peut gérer jusqu'à 16 instances simultanée sur un serveur.

Ordre de prix : 25 000 \$ par processeur ou 13 500 \$ pour 25 utilisateurs.

- Developer Edition : Cette version est fonctionnellement identique à la version Enterprise, elle est cependant réservée aux tests et aux développements, et fait l'objet d'un mode de licence spécifique.

Ordre prix : 60 €.

- Mobile Edition : Il s'agit d'une version légère adaptée à l'utilisation sur des PDA munis

de Windows for Pocket PC. Cette version est gratuite.

### *Utilisation avec WinDev*

Windev supporte nativement l'accès à Microsoft SQL Server grâce aux ordres W-Langage. Le prix de la licence d'utilisation de l'accès natif SQL Server est d'environ 750 €<sup>HT</sup> par serveur.

### *Conclusion*

On mettra à l'écart le SGBD PostgreSQL pour la suite de l'étude. Car d'une part, WinDev ne fournit pas d'accès natif pour ce SGBD. Et d'autre part, on a besoin d'une société assurant la formation des développeurs, techniciens et ingénieurs réseau qui interviendront sur la bases de données SQL. De plus, en cas de problème avec la base de données d'un client, on a besoin d'une société qui soit capable d'intervenir sur cette base chez le client.

On ne rencontre pas ces problèmes avec MySQL et SQL Server car ils possèdent un accès natif sous WinDev et un support de la part des sociétés qui les développent. MySQL offre un service de formation et de maintenance via la société MySQL AB. Elle est implantée dans plusieurs pays dont la France. SQL Server offre également un service de formation et de maintenance via la société Microsoft et ses partenaires partout en France.

## *Migration*

Avant de pouvoir effectuer les tests de remontée de données, il faut d'abord effectuer la migration de ces données à partir d'Hyper File vers les Systèmes de Gestion de Bases de Données SQL. Des outils de migration sont fournis avec MySQL et Microsoft SQL Server. Ils importent dans leur SGBD respectif, la structure et les données de bases accessibles grâce à des pilotes ODBC. Ces pilotes sont distribués avec la plupart des SGBD. Au mieux, ces outils de migration fonctionnent mal avec les pilotes ODBC d'Hyper File. Après des essais peu concluants, on étudie le fonctionnement d'Hyper File puis on essaie de mettre en place notre propre système de migration.

## *Hyper File*

### *Présentation*

Le moteur de base de données Hyper File est apparu à la fin des années 80, il est maintenant intégré dans WinDev. Une caractéristique de cette base de données est d'être déployable gratuitement dans toutes ses versions.

*Hyper File existe en plusieurs modes :*

- Hyper File monoposte/réseau, également appelé "Classic" :

Ce mode est principalement orienté fichier et gère automatiquement les accès concurrents aux données en local (applications multi-instances) ou en réseau.

Il permet de joindre les fichiers dans le répertoire de l'application ou un dossier de la machine pour une utilisation monoposte (voire même nomade si les données et le programme sont stockés sur un support amovible), ou d'installer les fichiers sur un serveur.

Le serveur de fichier peut être un serveur windows ou compatible Samba (Unix), ce qui permet d'installer les fichiers sur un serveur Linux.

- Hyper File client/serveur :

L'accès aux données est ici réservé à un programme "serveur". Ce programme serveur existe pour Windows et Linux.

Ce mode de fonctionnement permet d'accélérer les performances d'accès aux données en réseau par rapport à Hyper File réseau en limitant le volume des données qui circulent sur le réseau. Ce mode permet un accès à distance sécurisé offrant de bonnes performances.

Les fichiers de données Classic sont compatibles avec les fichiers de données Client/Serveur. La programmation d'accès est également compatible.

- Hyper File Mobile :

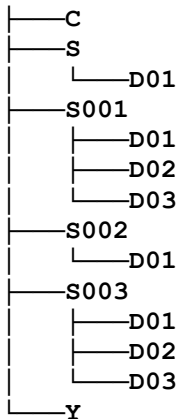
Cette version de Hyper File, compatible avec la version classic et client/serveur, est destinée aux terminaux mobiles, sous Windows CE ou Windows Mobile: pocket pc, smartphone, terminal industriel,...

### *Fonctionnement*

*Avec Hyper File, une table correspond à trois fichier :*

- .fic pour **fichier**, en jargon WinDev, un fichier est une table. Ce fichier contient les données de la table.
- .ndx pour **index**, ce fichier contient les index sur les données de la table.
- .mmo pour **mémo**, ce fichier contient les données de type « texte formaté ». Ces textes sont au format RTF.

Ces tables peuvent être placées dans une arborescence de répertoires, contrairement à ce qui peut se faire avec les systèmes SQL. Le passage à une base de données SQL nécessite donc une mise à plat de cette hiérarchie. Une arborescence typique de G.I.S. est :



S001, S002, S003 correspondent aux différentes sociétés d'une entreprise.

S correspond aux données communes à ces sociétés.

D01, D02, D03 correspondent aux différents exercices de ces sociétés.

*Sachant que ces répertoires contiennent des tables qui ont le même nom et la même structure, on a quatre solutions pour faire cette mise à plat :*

- Créer une base de données par répertoire.
- Préfixer le nom des tables, par exemple, pour une table Hyper File dont le nom est GF et qui est située dans le répertoire S001/D01, on aurait S001\_D01\_GF.
- Ajouter des champs dans les tables : Par exemple, faire une table GF qui contiendrait les données des GF de toutes les sociétés et de tous les exercices en ajoutant un champ Société où on mettrait le numéro de la société. Et un champ Exercice pour le numéro de l'exercice ou l'année de l'exercice. Dans ce cas, il faut également modifier les clefs primaires.
- Mixer ces solutions. Par exemple : Créer une base de données par société puis ajouter des champs dans les tables situées dans les répertoires D01, D02, etc.

WinDev utilise aussi un système d'alias qui permet de répartir les enregistrements d'une table dans plusieurs fichiers. Par exemple, dans G.I.S., les documents de la table DD, sont répartis dans 8 fichiers selon leur type (DDCC, DDCCD, DDCCF, DDCL, DDFC, DDFFD, DDFF, DDFFL). Un champ indiquant le type du document est déjà inclus dans ces tables donc tous ces fichiers peuvent être assemblés en un seul.

## Utilisation d'ODBC

On utilise donc les pilotes ODBC d'Hyper File fournis par PCSoft, l'éditeur de WinDev.

J'utilise le langage interprété PHP en mode console qui possède de nombreuses fonctions natives permettant de manipuler facilement les bases de données.

### *Structure*

Tout d'abord, il s'agit de récupérer le détail de la structure des tables et des colonnes (fonctions `odbc_tables` et `odbc_columns`). On rencontre alors des problèmes de récupération des types de données. On contourne partiellement ce problème en utilisant d'autres fonctions ODBC (fonctions `odbc_field_name`, `odbc_field_len`, `odbc_field_scale`, `odbc_field_type`). Ce contournement ne fonctionne pas si la table (le fichier Hyper File) ne contient aucune donnée. On ne peut par contre pas résoudre le problème de récupération de l'information Nullable qui permet de savoir si un champ doit être obligatoirement renseigné ou non. On récupère sans problème les clés primaires (fonction `odbc_primarykeys`) et index (`odbc_statistics`) mais il n'est pas possible de récupérer les clés étrangères (la fonction `odbc_foreignkeys` ne fonctionne pas avec le pilote Hyper File 7)

On transcrit toutes ces informations en SQL selon le Système de Gestion de Bases de Données Relationnelles (MySQL, PostgreSQL, SQL Server). On observe de légères différences de syntaxe entre ces différents SGBDR.

De plus, il faut traduire les types de données ODBC en types de données adaptés à chaque SGBD. En effet, les types de données et le nom de ces types varient beaucoup entre chaque SGBD. Il a fallu étudier tous ces types pour effectuer les conversions sans risquer des pertes de données.

### *Données*

La migration des données pose moins de problème une fois que la structure a été correctement migrée.

On constate lors du lancement des migrations des données, que la « moulinette » ne fonctionne que pour la première table (fichier Hyper File) ensuite aucune donnée n'est renvoyée pour les tables suivantes. Le problème est contourné en fermant et en réouvrant la connexion ODBC à Hyper File pour chaque table.

Le problème rencontré le plus dérangeant est la migration des données de type `TIMESTAMP` ou « DATE et HEURE ». Le pilote ODBC Hyper File 7 nous retourne systématiquement la valeur `NULL`.

### *Utilisation de l'accès natif WinDev*

Après les tests peu concluants effectués avec les pilotes ODBC, on se tourne vers WinDev pour automatiser cette migration.

### *Structure*

En mode natif, la structure d'une table Hyper File est définie dans la fenêtre Analyse. Cette analyse est automatiquement convertie lorsqu'on demande la création d'une table avec un SGBD SQL. Cependant, si on modifie la structure de la table dans la fenêtre d'Analyse, la structure de la table SQL n'est pas mise à jour. Quand on fait la même manipulation en utilisant Hyper File, la structure des tables Hyper File peut être mise à jour grâce à l'utilitaire `WModfic`. Cet utilitaire ne peut pas être utilisé avec les autres SGBD. Sinon il existe aussi un utilitaire qui permet de mettre à

jour l'Analyse WinDev si la structure d'une table SQL est changée mais pas l'inverse.

Pour tester l'import d'une base Hyper File vers SQL, j'ai pris les tables DD (documents) et DF (lignes de documents) d'une base créée par G.I.S..

G.I.S. peut gérer plusieurs sociétés en même temps. Les données de chaque société sont réparties dans des répertoire S001, S002, ..., Snnn. Il faut assembler toutes ces données dans une seule table, j'ai donc modifié l'analyse et j'ai ajouté la colonne DD\_Soc pour pouvoir indiquer la société à laquelle appartient un document.

## Données

Un utilitaire WinDev permet d'importer une base SQL vers Hyper File mais il n'a pas son équivalent pour faire l'inverse.

L'utilitaire que j'ai réalisé pour tester les imports, parcourt les répertoires S, S001, S002, ..., Snnn pour rechercher les enregistrements Hyper File DD et DF. Chaque répertoire contient 8 fichiers DD et 8 fichiers DF (*cf. Fonctionnement Hyper File*). Il insère tous les enregistrements trouvés dans des tables MySQL ou SQL Server.

Bien que l'intégrité des données soit faite sur l'ensemble des répertoires d'une base, pour éviter les doublons de clés primaires lors de l'import, j'ai agrandi leur taille dans l'analyse (de type chaîne de caractères) pour pouvoir ajouter le numéro de société et le type de document devant l'ancienne clé primaire. Ainsi l'enregistrement numéro 090734 de type CC (du fichier DDCC) du répertoire S001 devient l'enregistrement numéro 001CC090734.

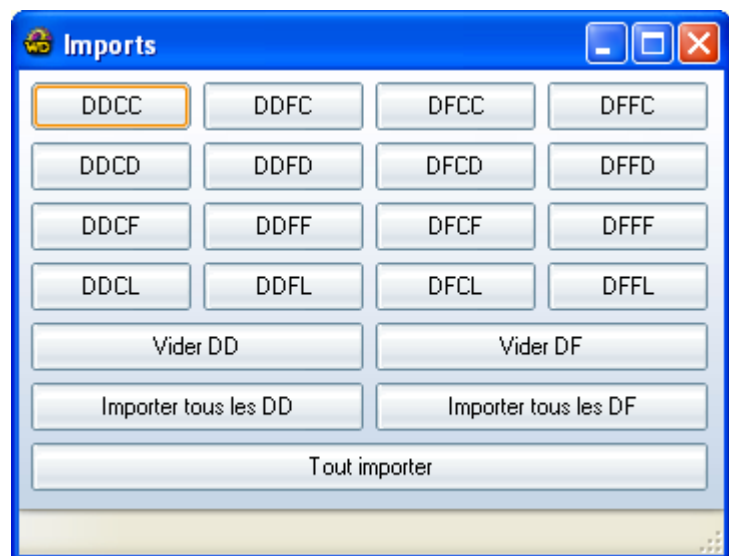


Illustration 2: Fenêtre d'import des tables DD et DF

## *Remontée des données*

### *Types d'accès aux bases de données<sup>8</sup>*

#### *Accès en natif*

Pour chaque type d'accès natif, il est nécessaire de posséder un module complémentaire à WinDev. L'accès natif de MySQL est gratuit et téléchargeable sur le site de PCSoft<sup>9</sup>. L'accès natif de SQL Server est payant, il est également téléchargeable sur le site de PCSoft mais nécessite un numéro de licence pour être installé.

Pour les accès de type MySQL ou SQL Server, il est nécessaire d'installer la couche client (libmysql.dll ou ntwdblib.dll) sur le poste de l'utilisateur.

Les fonctions SQL et Hyper File peuvent être utilisées avec ce type d'accès. Il est plus rapide que les accès par ODBC ou par un provider OLE DB.

#### *Accès par pilote ODBC direct*

Les caractéristiques de la connexion à la base de données doivent être définies dans l'administrateur ODBC de Windows. Seules les fonctions de type SQL sont utilisables pour ce type d'accès. Les fonctions Hyper File ne sont pas utilisables.

#### *Accès par pilote ODBC via provider OLE DB*

Ce type d'accès utilise un provider OLE DB spécifique. Ce type d'accès est déconseillé car plus lent qu'un accès par un driver ODBC. En effet, les performances sont moins bonnes que par un driver ODBC direct car l'accès se fait à la fois par le driver ODBC et par le provider OLE DB. Les fonctions Hyper File et SQL peuvent être utilisées avec ce type d'accès.

Il est nécessaire de définir les caractéristiques de la connexion à la base de données dans l'administrateur ODBC de Windows. Le provider ainsi que le MDAC 2.6 (ou supérieur) doivent être installés sur le poste.

#### *Accès par provider OLE DB*

Ce type d'accès utilise un provider OLE DB. Le provider ainsi que le MDAC 2.6 (ou supérieur) doivent être installés sur le poste. Les fonctions SQL et Hyper File peuvent être utilisées avec ce type d'accès.

### *Conclusion*

On utilisera l'accès natif qui offre les meilleurs performances et qui permet de réutiliser les fonctions de manipulation des bases de données Hyper File avec les bases de données MySQL et SQL Server.

---

<sup>8</sup> Voir le tableau récapitulatif en annexe.

<sup>9</sup> Site de PCSoft : <http://www.pcsoft.fr>

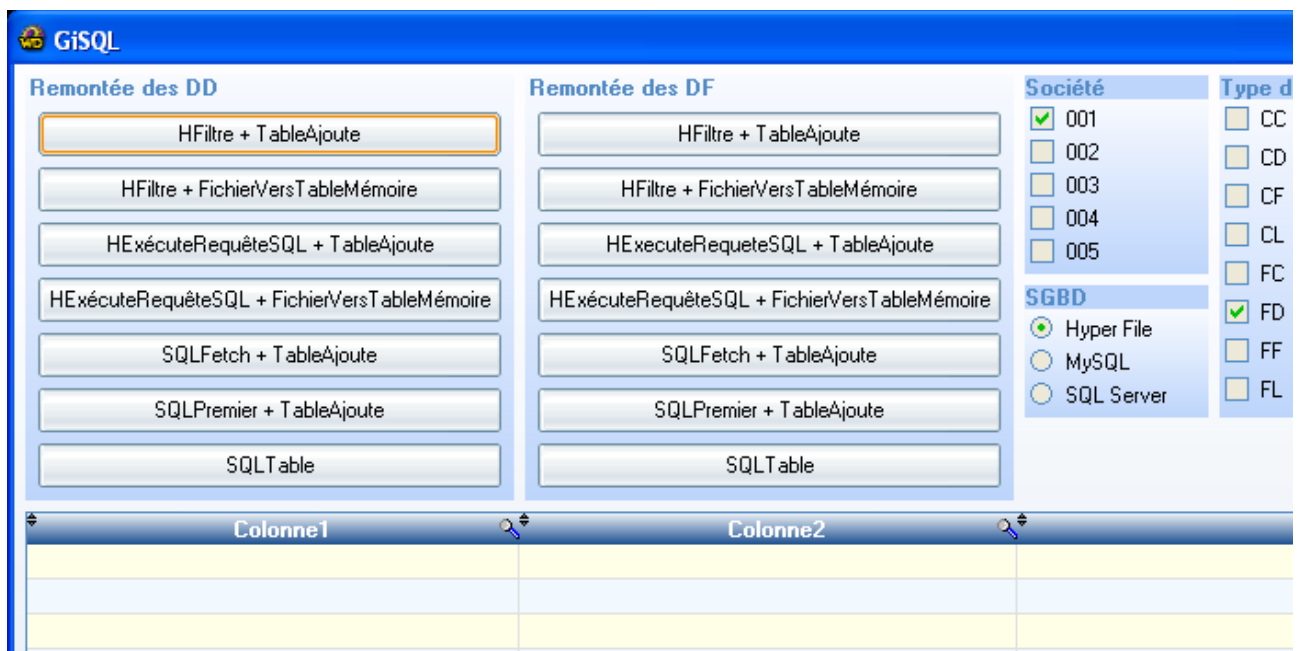


Illustration 3: Fenêtre de test

## Méthode HFilter + TableAjoute

### Code

```
HFilter(DD, DD_Code, Caract(0), Caract(255), "DD_Soc='001'")
TableSupprimeTout (TAM)
HLitPremier (DD)
TANTQUE PAS HEnDehors (DD)
    TableAjoute (TAM,...
        + DD.DD_Code + TAB...
        + DD.CE_Nom + TAB...
        + DD.CA_Ville + TAB)
    HLitSuivant (DD)
FIN
```

TAM est le tableau dans lequel les enregistrements sont affichés.  
 DD est le nom de la table à partir de laquelle sont extraits les enregistrements (236 929 au total).  
 DD\_Code, CE\_Nom, CA\_Ville sont les colonnes qu'on souhaite afficher.

### Résultats avec MySQL 5.0

On constate que les enregistrements à peu près remontés 1000 par 1000 avec un temps de latence très long entre chaque remontée (10 à 30 secondes).

### Résultat avec Microsoft SQL Server 2000

Comparé à MySQL, les enregistrements sont remontés plus régulièrement, sans temps de latence, mais on observe un plantage entre le 90 000<sup>ème</sup> et le 111 000<sup>ème</sup> enregistrement remonté. Le message d'erreur indique un manque de mémoire vive ou de mémoire virtuelle.

Les tests ont donc été effectués sur deux machines : l'une équipée de 512Mo de RAM, l'autre



de 1024 Mo avec la même quantité de mémoire virtuelle (de 1024Mo min à 2048Mo max).

*Résultats :*

- « Plantage » systématique au 90 000<sup>ième</sup> enregistrement remonté pour la machine à 1024 Mo de mémoire vive.
- « Plantage » systématique au 110 364<sup>ième</sup> enregistrement remonté pour la machine à 512 Mo de mémoire vive.

*Remarque*

On obtient exactement le même résultat en enlevant la ligne `TableAjoute (TAM, [...])`.

*Comparatif*

SQL Server se montre jusqu'à 30 fois plus rapide que MySQL avec cette méthode.

*Méthode HFilter + FichierVersTableMémoire*

*Code*

```
HFilter(DD, DD_Code, Caract(0), Caract(255), "DD_Soc='001'")
TableSupprimeTout (TAM)
FichierVersTableMémoire (TAM,DD)
```

*Résultat avec Microsoft SQL Server 2000*

Cette méthode ne permet pas d'observer la vitesse de la remontée d'enregistrements pendant son execution. Aucun message d'erreur ne survient, cependant, on constate après la remontée des enregistrements que le tableau dans lequel les enregistrements sont affichés ne contient pas tous les enregistrements attendus (103 576 dans le tableau sur 236 929 attendus).

*Résultat avec MySQL 5.0*

Après 45 min, la remontée n'était pas finie. Cette méthode n'est donc pas une bonne solution avec MySQL.

*Comparatif*

Micorsoft SQL Server se montre jusqu'à 60 fois plus rapide que MySQL avec cette méthode.

## Méthode HExécuteRequêteSQL + TableAjoute

### Code

```
HExécuteRequêteSQL(l_Requete, "SELECT DD_Code, CE_Nom, CA_Ville FROM DD WHERE
DD_Soc='001'")
TableSupprimeTout (TAM)
HLitPremier (l_Requete, hSansRafraîchir)
TANTQUE PAS HEnDehors (l_Requete)
    TableAjoute (TAM, ...
        + l_Requete.DD_Code + TAB...
        + l_Requete.CE_Nom + TAB...
        + l_Requete.CA_Ville + TAB)
    HLitSuivant (l_Requete)
FIN
```

TAM est le tableau dans lequel les enregistrements sont affichés. DD est le nom de la table à partir de laquelle sont extraits les enregistrements. DD\_Code, CE\_Nom, CA\_Ville sont les colonnes qu'on souhaite afficher.

### Résultats avec MySQL 5.0

Les données sont toutes remontées à une vitesse régulière.

Par contre si on remplace la requête SQL :

"SELECT DD\_Code, CE\_Nom, CA\_Ville FROM DD WHERE DD\_Soc='001'" par

"SELECT \* FROM DD WHERE DD\_Soc='001'",

On obtient le même résultat qu'avec HFilter, à savoir, des remontées d'enregistrements d'environ 1000 par 1000 et des temps de latences élevées entre chacune de ces remontées.

### Résultat avec Microsoft SQL Server 2000

Les données sont toutes remontées à une vitesse régulière.

Par contre si on remplace la requête SQL :

"SELECT DD\_Code, CE\_Nom, CA\_Ville FROM DD WHERE DD\_Soc='001'" par

"SELECT \* FROM DD WHERE DD\_Soc='001'",

on obtient le même résultat qu'avec HFilter, à savoir, un plantage à la remontée des enregistrements entre le 90 000<sup>ième</sup> et le 111 000<sup>ième</sup>.

### Remarque

En sélectionnant toutes les colonnes sauf celles de type TEXT avec SQL Server ou LONGBLOB avec MySQL, les plantages disparaissent. Ces colonnes correspondent à des Mémos Hyper File qui peuvent contenir des textes formatés en RTF. La remontée d'un grand nombre de ces colonnes sature la mémoire de la machine cliente.

### Comparatif

Microsoft SQL Server est jusqu'à qu'à 10 fois plus rapide que MySQL avec cette méthode.

## Méthode HExécuteRequêteSQL + FichierVersTableMémoire

### Code

```
TableSupprimeTout (TAM)
HExécuteRequêteSQL (l_Requete, "SELECT DD_Code, CE_Nom, CA_Ville FROM DD WHERE
DD_Soc='001'")
HLitPremier (l_Requete, hSansRafrâchir)
FichierVersTableMémoire (TAM, l_Requete)
```

### Résultats avec MySQL 5.0

Avec cette méthode, on obtient des temps du même ordre qu'avec HExécuteRequête + TableAjoute, elle est 4 fois plus rapide que les méthodes HFilter.

### Résultat avec Microsoft SQL Server 2000

C'est avec cette méthode qu'on obtient les meilleurs temps avec SQL Server. Les temps sont du même ordre qu'avec un HFilter + FichierVersTableMémoire avec un léger gain.

### Comparatif

Microsoft SQL Server est 25 fois plus rapide que MySQL avec cette méthode.

## Méthode SQLFetch + TableAjoute

Aucun enregistrement n'est chargé en mémoire. La base de données gère le résultat de la requête. SQLFetch est également appelé SQLAvance.

### Code

```
TableSupprimeTout (TAM)
SQLExec ("SELECT DD_Code, CE_Nom, CA_Ville FROM DD WHERE DD_Soc='001'", "RequeteSQL")
TANTQUE PAS SQLFetch ("RequeteSQL")
    TableAjoute (TAM, ...
        + SQLLitCol ("RequeteSQL", 1) + TAB...
        + SQLLitCol ("RequeteSQL", 2) + TAB...
        + SQLLitCol ("RequeteSQL", 3) + TAB)
FIN
SQLFerme ("RequeteSQL")
```

### Résultats avec MySQL 5.0

Avec cette méthode, on obtient de meilleurs résultats qu'avec les fonctions Hyper File.

### Résultat avec Microsoft SQL Server 2000

Avec cette méthode, on obtient de moins bons résultats qu'avec les fonctions Hyper File.

### Comparatif

MySQL est de 1,5 à 2 fois plus rapide que Microsoft SQL Server avec cette méthode.

## Méthode SQLPremier + TableAjoute

Tous les enregistrements résultat de la requête sont chargés en mémoire lors de l'exécution de la fonction SQLPremier. Le premier appel à la fonction SQLPremier peut donc être relativement long.

### Code

```
TableSupprimeTout (TAM)
SQLExec ("SELECT DD_Code, CE_Nom, CA_Ville FROM DD WHERE DD_Soc='001'", "RequeteSQL")
SQLPremier ("RequeteSQL")
TANTQUE PAS SQL.EnDehors
    TableAjoute (TAM, ...
        + SQLCol ("RequeteSQL", 1) + TAB...
        + SQLCol ("RequeteSQL", 2) + TAB...
        + SQLCol ("RequeteSQL", 3) + TAB)
    SQLSuivant ("RequeteSQL")
FIN
SQLFerme ("RequeteSQL")
```

### Résultats avec MySQL 5.0

On obtient de meilleurs résultats qu'avec les fonctions Hyper File. Cependant, les temps sont moins bons qu'avec la fonction SQLAvance.

### Résultat avec Microsoft SQL Server 2000

On obtient de moins bons temps qu'avec les fonctions Hyper File ou la fonction SQLAvance. On note un plantage de la remontée de données autour du 100 000<sup>ième</sup> enregistrement.

### Comparatif

MySQL est de 1,5 à 2 fois plus rapide que Microsoft SQL Server avec cette méthode.

## Méthode SQLTable

Tous les enregistrements résultat de la requête sont chargés en mémoire et affichés.

### Code

```
TableSupprimeTout (TAM)
SQLExec ("SELECT DD_Code, CE_Nom, CA_Ville FROM DD WHERE DD_Soc='001'", "RequeteSQL")
SQLTable ("RequeteSQL", TAM)
SQLFerme ("RequeteSQL")
```

### Résultats avec MySQL 5.0

C'est avec cette méthode qu'on obtient les meilleurs temps avec MySQL. Lors des tests, l'ordre de temps de remontée de données était de la minute pour 200 000 enregistrements.

## *Résultat avec Microsoft SQL Server 2000*

Cette méthode ne fonctionne pas avec Microsoft SQL Server 2000 au delà des 100 000 enregistrements.

## *Comparatif*

MySQL est 2,5 fois plus rapide que Microsoft SQL Server avec cette méthode.

# *Résultats*

## *Microsoft SQL Server*

Les méthodes HFilter offrent des temps de remontée d'enregistrements satisfaisants, de l'ordre de la minute pour 100 000 avec HFilter et de la minute pour 160 000 avec HExécuteRequête. Cependant, elles sont peu fiables au delà des 100 000 enregistrements, en effet, on observe soit des plantages soit des remontées incomplètes après dépassement de cette limite.

Les méthodes HExécuteRequêteSQL offrent des temps de remontée légèrement meilleurs qu'avec les HFilter mais gardent le problème de fiabilité au delà de la limite des 100 000 enregistrements. Pour fiabiliser les méthodes HExécuteRequêteSQL, on recommandera de ne pas remonter tous les champs des enregistrements, et d'éviter particulièrement, lors de remontée de nombreux enregistrements, de remonter les champs de type Mémo qui sont responsables de la saturation de la mémoire de la machine cliente et donc responsable des plantages et remontées incomplètes. Ces problèmes disparaissent lorsqu'on ne remonte plus les mémos, même pour des remontée de plus de 230 000 enregistrements. Il faut donc tout simplement ne remonter que les champs dont on a besoin. HFilter n'offre malheureusement pas la possibilité de sélectionner les champs qu'on souhaite remonter et remonte donc tous les champs.

## *MySQL*

Les méthodes HFilter ou HExécuteRequête offrent des temps de remontée d'enregistrements insatisfaisants, de l'ordre de l'heure pour 100 000.

Les méthodes SQLFetch, SQLPremier, ou SQLTable offrent de bonnes performances. De l'ordre de la minute pour 100 000 enregistrements pour SQLFetch, SQLPremier et de l'ordre de la minute pour 200 000 enregistrements avec SQLTable.

Comme pour SQL Server, on rencontre le problème de fiabilité au delà de la limite des 100 000 enregistrements avec les fonctions Hyper File. Pour fiabiliser les méthodes HExécuteRequêteSQL, on recommandera donc également de ne pas remonter tous les champs des enregistrements.

## *Conclusion*

Microsoft SQL Server offre une rapidité régulière quelque soit le type d'accès. La fiabilité n'est pas au rendez-vous pour les HFilter à partir de 100 000 enregistrements à remonter. Pour obtenir une rapidité optimale avec SQL Server, HExécuteRequêteSQL semble être la meilleure solution.

MySQL quant à lui offre de bonnes performances avec les fonctions SQL (SQLFetch, SQLPremier, etc.). Par contre, l'accès natif MySQL ne permet pas d'obtenir de bonnes performances avec les fonctions Hyper File (HFilter, HExécuteRequêteSQL, etc.) que ce soit au niveau performances ou au niveau fiabilité.

*Pour MySQL et Microsoft SQL Server, la fiabilité et les performances s'améliorent quand :*

- on limite le nombre de colonnes à remonter
- on évite de remonter les colonnes de type mémo

PCSoft préconise l'utilisation des fonctions SQL avec les bases de données autres que Hyper File. Que ce soit avec MySQL ou SQL Server, la fonction SQLTable est celle qui offre les meilleures performances parmi les fonctions SQL.

Un problème important subsiste en ce qui concerne la structure des bases SQL, le gestionnaire d'analyse de WinDev est seulement capable de créer une base de données SQL correspondant à une analyse, mais après modification d'une analyse, il est incapable de répercuter les changements sur la structure d'une base de données SQL.

# Réalisations diverses

Mon sujet de stage est centré sur G.I.S. et la migration de son système de gestion de bases de données. Cependant, j'ai été amené à étendre mes réalisations autour de ce progiciel.

## Comparatif des Progiciels de Gestion Intégrés

### But

Ce comparatif a plusieurs objectifs. D'abord évaluer l'offre du marché des PGI (*Progiciels de Gestion Intégrés*) et la richesse fonctionnelle de ces PGI. A partir de cette évaluation, on peut situer G.I.S. (*la solution de gestion de SATTI*) dans ce marché et répondre aux questions « Doit-on continuer à développer G.I.S. ? Doit-on abandonner G.I.S. et distribuer un PGI en partenariat avec un éditeur ? Doit-on continuer le développement de G.I.S. et créer un partenariat avec un éditeur de PGI de sorte qu'on puisse proposer une solution alternative aux clients de SATTI ? »

### Principe

Il a tout d'abord fallu déterminer comment on allait procéder pour faire ce comparatif :

- Il faut commencer par déterminer les caractéristiques des PGI auxquels on va s'intéresser.
- Il est ensuite nécessaire de rechercher ces PGI et d'en établir la liste.
- Et enfin, pour obtenir des informations sur les PGI :
  - Télécharger des documentations et des logiciels de démonstration.
  - Prendre contact directement auprès de l'éditeur pour se faire parvenir des plaquettes d'information et CD de démonstration.
  - Puis, pour les PGI intéressants, demander des démonstrations sur site ou participer à des conférences de démonstration.

### Réalisation

Un dossier avait déjà été préparé par le service commercial sur les PGI qui, comme G.I.S., sont spécialisés dans la GPAO\* (*Gestion de Production Assistée par Ordinateur*). Ludovic GAUDIN, mon maître de stage, responsable du service développement, s'est chargé d'établir la liste des PGI intéressants dans ce dossier.

J'ai ensuite contacté les éditeurs pour qu'ils me fassent parvenir la documentation de leur PGI. Et enfin, j'ai invité ces éditeurs à venir nous faire une démonstration de leur logiciel pour qu'on puisse juger de la qualité de leur produit.

---

\* Voir le glossaire en annexe

## *Terminaux de saisie de temps en confection*

J'ai eu l'occasion, dans le cadre de la découverte des activités du service développement, de participer à l'audit des établissements de confection Marty. Ce client souhaite renouveler son système de suivi de production vieillissant. Il possède pour cela un réseau d'une quarantaine de terminaux de suivi de temps qui permet de suivre en temps réel le déroulement des opérations.

Chaque ouvrière possède un terminal. Lorsqu'elles reçoivent un vêtement, elles tapent le numéro de l'opération qu'elles vont effectuer sur leur terminal. Un chronomètre en compte à rebours s'affiche alors indiquant le temps qu'il reste pour effectuer l'opération saisie. Lorsqu'elles ont fini, elles le signalent grâce au terminal. La responsable de production peut savoir en temps réel comment se déroule les opérations. Si on observe des retards réguliers pour une opération chez une ouvrière, une personne va alors analyser les gestes de l'ouvrière et chronométrer les temps qu'elle passe à faire les manipulations de l'opération. Cette personne va ensuite trouver avec l'ouvrière tous les gestes superflus effectués et les améliorations de manipulations envisageables pour éliminer les retards sur l'opération.

### *But*

Les terminaux de saisie de temps de cet établissement doivent être remplacés. En effet, le constructeur de ces terminaux n'existe plus, donc, chaque fois qu'une entreprise de confection qui utilise le même type de terminaux fait faillite, une personne des établissements Marty participe à la vente aux enchères du matériel de cette entreprise pour récupérer des terminaux de saisie de temps pour pouvoir remplacer les terminaux de suivi de temps en panne. Donc, ce que demande le client, c'est de remplacer ces terminaux par des terminaux neufs et d'interfacer ce réseau avec G.I.S. Gestion de Production.

### *Principe*

Mon rôle est d'entrer en contact avec un fabricant de terminaux de saisie de temps, pour qu'il nous fasse une offre de prix. Après acceptation du devis par le client, il faut emprunter le matériel nécessaire auprès du fabricant pour qu'on puisse interfacier leur système de réseau de terminaux avec G.I.S..

### *Réalisation*

Le nom du fabricant nous a été donné par les établissements Marty. C'est CaptorGroup, situé en Belgique. J'ai d'abord consulté leur site pour voir leur panel de solutions matérielles en ce qui concerne le suivi de temps en confection.

Ensuite j'ai pris contact avec eux par e-mail et par téléphone pour leur présenter les activités de SATTI et nos besoins. Puis je leur ai demandé de la documentation sur leur matériel et sur le fonctionnement de leur système de réseau de terminaux.

*Leur système de réseau de terminaux est basé sur un multiplexeur :*

- Il est chargé de maintenir la communication avec l'ordinateur qui gère le suivi de production via une ligne série RS232-C ou via une interface Ethernet. A partir de l'ordinateur, le réseau est vu comme un simple terminal ASCII.



- Il alimente les terminaux en 24V.
- Il communique avec les terminaux (255 maximum).



Illustration 4: Terminal Microcap

J'ai ensuite été mis en contact avec un technicien de CaptorGroup, avec qui j'ai choisi le modèle de terminaux qui correspond le mieux aux besoins des établissements Marty. Suite à ce choix, j'ai fait une demande de prix sur le matériel.

Après réception de l'offre de prix, le devis a pu être établi et transmis à notre client. Après le feu vert du client, il a donc fallu s'intéresser de plus près à l'interfaçage de G.I.S. avec le réseau de terminaux Captor. On a donc réalisé un emprunt de matériel pour pouvoir faire des tests de dialogue avec le réseau Captor. Un multiplexeur et un terminal nous a alors été livré, et, à partir de la documentation technique du multiplexeur Capdrive et du terminal Microcap, j'ai développé une classe sous WinDev permettant de dialoguer avec les terminaux Microcap. La classe est décrite dans le tableau ci-dessous. Les noms des

méthodes ont été choisis pour correspondre aux noms des fonctions natives de WinDev.

### Structure de la classe :

<b>Méthode</b>	<b>Commentaire</b>
OuvreConnexion ()	Ouvre et paramètre le port série RS232
FermeConnexion ()	Ferme le port série.
Ecrit ()	Formate et envoie des données au multiplexeur
Lit ()	Lit les données en provenance du multiplexeur
MAffiche ()	Affiche un message sur l'écran d'un Microcap
MEfface ()	Efface une ligne sur l'écran d'un Microcap
MEffaceTout ()	Efface complètement l'écran d'un Microcap
MBip ()	Déclanche le bip sonore d'un Microcap
TDemarre ()	Démarre un timer d'un Microcap
TArrete ()	Arrête un timer d'un Microcap
TAffiche ()	Affiche le timer sur l'écran d'un Microcap
TCache ()	Cache le timer sur l'écran d'un Microcap
TCroissant ()	Met un timer en mode incrémentation
TDecroissant ()	Met un timer en mode compte à rebour
TMinSecDix ()	Met un timer en mode affichage <Minutes>:<Secondes>:<Dixièmes>
TSecDix ()	Met un timer en mode affichage <Secondes><Dixièmes>
TLit ()	Demande la valeur d'un timer à un Microcap

<i>Méthode</i>	<i>Commentaire</i>
TReinitialise ()	Réinitialise un timer à la valeur chargée dans son registre
TChargeValeur ()	Charge une valeur dans le registre d'un timer
TActiveStopMode ()	Stoppe le timer quand il arrive à la valeur chargée dans son registre
TDesactiveStopMode ()	Remet le timer à zéro quand il arrive à la valeur dans son registre
TActiveHeure ()	Ajoute l'heure dans les messages envoyés par un terminal
TDesactiveHeure ()	Enlève l'heure dans les messages envoyés par un terminal
CActiveF8 ()	Active l'incréméntation d'un compteur de terminal lors de l'appui sur F8
CActiveLigneExterne ()	Active l'incréméntation d'un compteur de terminal lors d'un signal sur la ligne externe
CArrete ()	Arrête l'incréméntation d'un compteur
CAffiche ()	Affiche le compteur sur l'écran d'un terminal
CCache ()	Cache le compteur sur l'écran d'un terminal
CCroissant ()	Met le compteur en mode incréméntation
CDecroissant ()	Met le compteur en mode décréméntation
CLit ()	Demande la valeur d'un compteur à un terminal
CRéinitialise ()	Réinitialise un compteur à la valeur chargée dans son registre
CChargeValeur ()	Charge une valeur dans le registre d'un compteur
CActiveModeStop ()	Stoppe un compteur lorsqu'il arrive à la valeur chargée dans son registre
CDesactiveModeStop ()	Remet le compteur à zéro lorsqu'il arrive à la valeur dans son registre
MActiveClavier ()	Active le clavier d'un terminal
MDesactiveClavier ()	Désactive le clavier d'un terminal
MActiveEcho ()	Active l'écho du clavier sur l'écran du terminal
MDesactiveEcho ()	Désactive l'écho du clavier sur l'écran du terminal
MIdentifie ()	Demande à un terminal de s'identifier
MAllumeLED ()	Allume la LED d'un terminal
MEteintLED ()	Éteint la LED d'un terminal

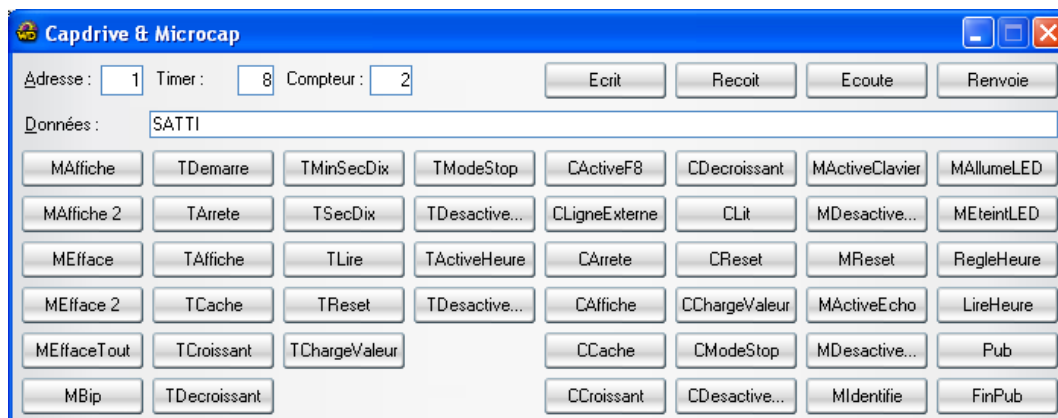


Illustration 5: Fenêtre de test de la classe



## Microsoft SharePoint

SharePoint est un produit de Microsoft qui permet à tous les collaborateurs d'une entreprise de publier et consulter des documents (Word, Excel, PDF, Images, etc.) sur l'intranet de l'entreprise. Ce produit permet également de créer des formulaires grâce à Microsoft Office InfoPath et de les mettre en ligne sur l'intranet. Les formulaires papier de l'entreprise peuvent ainsi être placés et remplis directement sur l'intranet. Chaque fois qu'un formulaire est rempli sur l'intranet, les réponses sont placées dans un fichier, ou insérées dans la base de données de l'intranet ou envoyées par mail.

### But

SATTI est un partenaire de Microsoft et va bientôt distribuer ce produit, c'est pourquoi elle cherche à informer les entreprises sur ce nouveau produit et à réaliser des démonstrations.

### Principe

Pour préparer une démonstration, deux de mes collègues ont installé les services SharePoint et SQL Server sur un serveur qu'ils ont reliés à l'intranet SATTI. Ils ont ensuite configuré ces deux services. Ils ont commencé à créer un formulaire de demande congés pour illustrer les fonctionnalités de SharePoint. Leurs emplois du temps étant particulièrement chargés, je leur ai proposé de programmer le formulaire.

### Réalisation

Quand le formulaire m'a été confié, il avait son aspect visuel final. Il me restait donc à programmer les différents contrôles du formulaire et les interactions entre le formulaire et les services SharePoint, SQL Server et Exchange de l'intranet SATTI.

J'ai également contribué au préalable à la mise en place de SQL Server sur le serveur de test.

#### Demande de congé

Date de la demande :	01/06/2006	Heure de la demande :	10:46
<b>Employé</b>		<b>Responsable</b>	
Nom :		Nom :	
Numéro d'identification :		Numéro de téléphone :	
Service :		Adresse de messagerie :	
Numéro de téléphone :			
Adresse de messagerie :			

#### Détails du congé

Date de début	Date de fin	Type	Heures
		Congés payés	0,00
<b>Total heures demandées</b>			<b>0,00</b>

#### Congés disponibles

Type de congé	Solde précédent	Heures demandées	Nouveau solde
Congés payés	0,00	0,00	<b>0,00</b>
Congés maladie	0,00	0,00	<b>0,00</b>
Congés mobiles	0,00	0,00	<b>0,00</b>
Congés sans solde	0,00	0,00	<b>0,00</b>
Congés de décès	0,00	0,00	<b>0,00</b>
Autres	0,00	0,00	<b>0,00</b>

Illustration 6: Formulaire InfoPath

# Conclusion

J'ai réalisé ce stage en plusieurs étapes, de l'étude du fonctionnement de différents systèmes de gestion de bases de données jusqu'aux tests de remontées de données via l'utilisation de différentes méthodes, en passant par la migration de bases de données Hyper File vers des bases de données SQL.

J'ai également contribué à une étude comparative des Progiciels de Gestion Intégrés, à la programmation d'une classe dialoguant avec des terminaux de saisie de temps utilisés en confection, à la programmation d'une classe de création de planning de tâches, et à la mise en place d'une démonstration du système SharePoint.

*Ce stage a été enrichissant pour plusieurs raisons :*

- Tout d'abord parce qu'il m'a permis de découvrir les systèmes de gestion bases de données et leurs problématiques au travers de bases de données en exploitation dans des entreprises.
- Ensuite parce qu'en me permettant de mettre un pied dans le monde professionnel, il m'a conduit à intégrer une entreprise et d'en découvrir les différents services, leur fonctionnement et leurs interactions.
- Enfin parce qu'il m'a conduit à nouer des contacts professionnels avec des personnes extérieures à SATTI et à en rencontrer.

# Annexes

## Glossaire

### GPAO

GPAO signifie Gestion de la Production Assistée par Ordinateur. Un logiciel de GPAO est un programme modulaire de gestion de production permettant de gérer les nomenclatures et les gammes d'opérations et d'élaborer des plans de charge.

### ODBC

La technologie ODBC permet d'interfacer de façon standard une application à n'importe quel serveur de bases de données, pour peu que celui-ci possède un driver ODBC (la quasi-totalité des SGBD possèdent un tel pilote, dont tous les principaux SGBD du marché).

En 1992 un sous-comité du SQL Access Group a défini une interface permettant à des applications clientes de communiquer avec des bases de données utilisant le langage SQL. Cette interface, dénommée SQL/CLI pour SQL Call Level Interface a été normalisée aussi bien par ISO que par ANSI en 1993 et a été par ailleurs annexée à la norme SQL-92. Cette spécification a été publiée en 1992 sous la dénomination Microsoft Open DataBase Connectivity (ODBC), mais Microsoft n'en est pas le seul auteur.

### OLEDB

Ensemble d'interfaces fournissant aux applications un accès uniforme à des données stockées dans des sources variées. Les applications se servant d'OLEDB utiliseront donc la même interface pour accéder à des données disponibles dans des sources hétérogènes.

### PGI

Un Progiciel de Gestion Intégré (abrégié PGI, en anglais Enterprise Resource Planning ou ERP) est un logiciel qui permet de gérer l'ensemble des processus d'une entreprise, en intégrant l'ensemble des fonctions de cette dernière comme la gestion des ressources humaines, la gestion comptable et financière, l'aide à la décision, mais aussi la vente, la distribution, l'approvisionnement, le commerce électronique.

### SGBD

La gestion et l'accès à une base de données sont assurés par un ensemble de programmes qui constituent le Système de Gestion de Base de Données (SGBD).

Un système de gestion de bases de données héberge généralement plusieurs bases de données,

---

qui sont destinées à des logiciels ou des thématiques différents.

## *WinDev*

WinDev est un environnement de développement intégré conçu pour permettre de développer rapidement des applications orientées données.

Son éditeur d'interface graphique permet de construire facilement une application sans trop de compétence graphique puisqu'il suffit de choisir un modèle de charte graphique parmi un ensemble proposé, voire même d'en créer un soi même.

## Types d'accès aux SGBD fournis par WinDev

	Hyper File		ORACLE, SQL Server, Access, ...		
	Accès intégré		Accès natif		Driver ODBC (Direct)
	Fonctions SQL	Fonctions Hyper File (HLitxxx, ...)	Fonctions SQL	Fonctions Hyper File (HLitxxx, ...)	Fonctions SQL
Connexion à la base de données	<b>SQLConnecte</b> Paramètre <Type de la base> : "HyperFile"	Si l'analyse est liée au projet, connexion automatique, sinon, utilisez une des fonctions suivantes : <b>HOuvreAnalyse</b> <b>HDéclareExterne</b> <b>HDéclare</b>	<b>SQLConnecte</b> Paramètre <Type de la base> : "ORACLE" ou "SQL SERVER" ou "AS400"	<b>HOuvreConnexion</b> avec le nom de la connexion à ouvrir. La connexion doit être définie dans l'analyse ou dynamiquement (voir ci dessous)	<b>SQLConnecte</b> Paramètre <Type de la base> : "ODBC"
Description dynamique de la connexion	Inutile	Inutile			
Exécution de la requête	<b>SQLExec</b>	<b>HExécuteRequête</b> , <b>HExécuteRequêteSQL</b>	<b>SQLExec</b>	<b>HExécuteRequête</b> , <b>HExécuteRequêteSQL</b>	<b>SQLExec</b>
Lecture du résultat	<b>parcours de type SQLPremier / SQLCol</b> <b>parcours de type SQLAvance / SQLLitCol, SQLLitMémo SQLTable</b>	Toutes les fonctions de lecture classiques : <b>HLitPremier</b> , <b>HLitSuivant</b> , <b>HLitRecherche</b> , ... Les filtres Les vues	<b>parcours de type SQLPremier / SQLCol</b> <b>parcours de type SQLAvance / SQLLitCol, SQLLitMémo SQLTable</b>	Toutes les fonctions de lecture classiques : <b>HLitPremier</b> , <b>HLitSuivant</b> , <b>HLitRecherche</b> , ... Les filtres	<b>parcours de type SQLPremier / SQLCol</b> <b>parcours de type SQLAvance / SQLLitCol, SQLLitMémo SQLTable</b>
Blocage des enregistrements	Impossible	<b>HBloqueFichier</b> <b>HBloqueNumEnr</b> Fonctions de lecture bloquante ( <b>HLitPremier</b> , <b>HLitRecherche</b> , ...)	<b>SQLBloque</b>	Chaque base a ses particularités. Pour l'accès natif SQL Server ou Oracle, il est possible d'utiliser la fonction <b>HBloqueFichier</b> .	Le driver ODBC ne supporte pas les blocages.
Changement de la connexion (passe à une autre connexion ouverte)	<b>SQLChangeConnexion</b>	Inutile	<b>SQLChangeConnexion</b>	La connexion utilisée est précisée dans la fonction <b>HExécuteRequêteSQL</b> ou dans la fonction <b>HExécuteRequête</b> .	<b>SQLChangeConnexion</b>
Fermeture de la connexion	<b>SQLDéconnecte</b>	Si l'analyse est liée au projet, fermeture automatique, sinon utilisez une des fonctions suivantes : <b>HFermeAnalyse</b> , <b>HAnnuleDéclaration</b>	<b>SQLDéconnecte</b>	<b>HFermeConnexion</b>	<b>SQLDéconnecte</b>
Vitesse	**	***	***	**	**



ORACLE, SQL Server, Access, ...					
ODBC via le provider OLE DB		Provider OLE DB			
Fonctions SQL	Fonctions Hyper File (HLitxxx, ...)	Fonctions SQL	Fonctions Hyper File (HLitxxx, ...)		
Connexion à la base de données	<b>SQLConnecte</b> Paramètre <Type de la base> : "ODBC" Paramètre <provider OLEDB> : "MSSASQL" ou la constante hODBC.	<b>HOuvreConnexion</b> avec le nom de la connexion à ouvrir. La connexion doit être définie dans l'analyse ou dynamiquement (voir ci dessous)	<b>SQLConnecte</b> Paramètre <Type de la base> : "OLEDB" Paramètre <provider OLEDB> : correspondant à celui voulu (chaîne de caractères ou constante)	<b>HOuvreConnexion</b> avec le nom de la connexion à ouvrir. La connexion doit être définie dans l'analyse ou dynamiquement (voir ci dessous).	
Description dynamique de la connexion		<b>HDé critConnexion</b>		<b>HDé critConnexion</b>	
Exécution de la requête	<b>SQLExec</b>	<b>HExécuteRequête, HExécuteRequêteS QL</b>	<b>SQLExec</b>	<b>HExécuteRequête, HExécuteRequêteS QL</b>	
Lecture du résultat	<b>parcours de type SQLPremier / SQLCol</b> <b>parcours de type SQLAvance / SQLLitCol, SQLLitMémo SQLTable</b>	Toutes les fonctions de lecture classiques : <b>HLitPremier, HLitSuivant, HLitRecherche, ...</b> Les filtres	<b>parcours de type SQLPremier / SQLCol</b> <b>parcours de type SQLAvance / SQLLitCol, SQLLitMémo SQLTable</b>	Toutes les fonctions de lecture classiques : <b>HLitPremier, HLitSuivant, HLitRecherche, ...</b> Les filtres	
Blocage des enregistrements	Le provider ne supporte pas les blocages.	Le provider ne supporte pas les blocages.	Le provider ne supporte pas les blocages.	Le provider ne supporte pas les blocages.	
Changement de la connexion (passe à une autre connexion ouverte)	<b>SQLChangeConnexion</b>	La connexion utilisée est précisée dans la fonction <b>HExécuteRequêteS QL</b> ou dans la fonction <b>HExécuteRequête.</b>	<b>SQLChangeConnexion</b>	La connexion utilisée est précisée dans la fonction <b>HExécuteRequêteS QL</b> ou dans la fonction <b>HExécuteRequête.</b>	
Fermeture de la connexion	<b>SQLDéconnecte</b>	<b>HFermeConnexion</b>	<b>SQLDéconnecte</b>	<b>HFermeConnexion</b>	
Vitesse	*	*	**	*	

## Résultats des remontées

SQL Server 2000	0	4	13	51	90	754	910	4 974	10 363	50 552	102 132	160 638	236 929
HFiltre + TableAjoute	0,19	0,28	0,22	0,21	0,31	0,86	0,99	18,83	22,22	59,11	134,57		
HFiltre + FichierVersTableMémoire	0,64	0,15	0,29	0,16	0,21	0,40	0,44	12,43	18,65	30,37	69,49	58,24	77,91
HExécuteRequêteSQL + TableAjoute	0,71	0,37	0,36	0,26	0,33	4,05	0,79	32,56	19,88	52,93	89,74	123,31	174,44
HExécuteRequêteSQL + FichierVersTableMémoire	1,07	0,25	0,25	0,27	0,28	0,36	0,36	13,80	18,25	24,94	34,39	39,16	65,80
SQLFetch + TableAjoute	0,59	0,08	0,25	0,15	0,24	0,79	0,86	18,74	22,04	56,23	100,42	150,31	196,99
SQLPremier + TableAjoute	0,83	0,11	0,26	0,15	0,27	3,26	0,90	20,35	22,48	62,62			
SQLTable	0,55	0,13	0,23	0,15	0,14	0,41	0,41	15,92	17,44	33,07	77,33	76,17	103,44
Analyseur de requête SQL	1,00	0,00	0,00	0,00	0,00	1,00	0,00	15,00	18,00	37,00	59,00	85,00	100,00

MySQL 5.0	0	4	13	51	90	754	910	4 974	10 363	50 552	102 132	160 638	236 929
HFiltre + TableAjoute	2,20	1,26	1,26	1,17	1,15	1,96	1,81	7,44	13,32	669,62	4064,76	6239,64	
HFiltre + FichierVersTableMémoire	1,39	1,37	1,26	1,22	1,23	1,57	1,41	5,63	9,50	354,71	4117,15		
HExécuteRequêteSQL + TableAjoute	0,29	0,17	0,22	0,19	0,14	0,99	0,75	5,62	11,62	261,31	979,49	1760,87	
HExécuteRequêteSQL + FichierVersTableMémoire	1,26	1,22	1,13	1,22	1,22	1,31	1,53	2,89	8,21	314,69	878,86		
SQLFetch + TableAjoute	0,29	0,05	0,12	0,10	0,10	0,69	0,51	2,98	6,08	31,59	59,69	90,13	132,41
SQLPremier + TableAjoute	0,14	0,03	0,02	0,06	0,08	0,54	0,67	3,94	7,56	37,28	75,60	114,79	164,36
SQLTable	0,16	0,01	0,02	0,01	0,04	0,20	0,25	1,73	3,00	14,96	31,20	45,37	66,02
MySQL Query Browser	0,16	0,03	0,04	0,03	0,04	0,15	0,18	1,16	3,21	10,76	22,46	34,85	54,17

Hyper File 7	0	4	13	51	90	754	910	4 974	10 363	50 552	102 132	160 638	236 929
HFiltre + TableAjoute	3,46	6,47	2,56	1,20	0,39	0,68	0,75	4,07	50,06	206,54	164,25	321,97	190,67
HFiltre + FichierVersTableMémoire	1,73	1,50	1,17	1,18	0,30	0,34	0,35	1,84	10,32	26,74	57,75	56,86	97,76
HExécuteRequêteSQL + TableAjoute	1,91	1,59	1,20	1,21	0,59	0,78	0,81	4,07	20,95	48,89	281,02	222,70	430,35
HExécuteRequêteSQL + FichierVersTableMémoire	1,43	1,65	2,42	1,33	0,55	0,44	0,45	2,32	103,95	262,47	683,36	592,92	110,19
SQLFetch + TableAjoute	4,33	1,92	2,96	1,58	0,48	0,92	0,69	8,66	151,01	595,20			
SQLPremier + TableAjoute	1,72	1,81	1,55	0,23	0,46	0,78	0,97	6,52	20,24	517,00			
SQLTable	4,39	6,68	2,96	1,57	0,65	0,97	0,91	8,41	121,75	484,17			
WDSql	3,82	6,76	2,73	0,28	0,62	0,45	0,34	3,88	151,84	753,98			

