Database-SQL-RDBMS HOW-TO pour Linux (PostgreSQL Object Relational Database System), version française.

Al Dev (Alavoor Vasudevan) *alavoor@yahoo.com*
Adaptation française :
Albert-Paul Bouillot *apb@club-internet.fr*

v13.0,23 Octobre 1999, (version française 28 Octobre 1999)

Ce document est un "guide pratique" pour la mise en place rapide d'un moteur de Base de Données SQL et des outils frontaux sur un système Unix. Il traite également du langage standard International ANSI/ISO SQL et passe en revue les mérites/avantages d'un moteur de Base de Données développé dans le cadre mondial de l'environnement de développement ouvert d'internet. Ce document explique COMMENT mettre en place une Base de Données Relationnelle SQL Objet de la nouvelle génération "PostgreSQL" sur votre système unix qui pourra être utilisée comme Base de Données Serveur d'Application ou Serveur Web. PostgreSQL implémente un sous-ensemble des normes Internationales ISO et ANSI SQL 1998,92,89. Ce document donne aussi des informations sur les programmes d'interface à la base de données tels que interfaces utilisateur graphiques (GUI), outils RAD (Développement Rapide d'Application), interfaçage des langages de programmation ("C", "C++", Java, Perl), pilotes ODBC, JDBC ainsi que sur les outils et programmes interfaçage d'une Base de Données Web . L'information donnée ici est valable pour toutes autres plates-formes unix et autres Bases de Données. Cette information sera très utile aux nouveaux utilisateurs des Bases de Données, de PostgreSQL et du langage SQL. Ce document comporte aussi un didacticiel SQL et les spécifications de la syntaxe SQL qui devrait être très utile aux débutants. Les gens expérimentés trouveront dans ce document un guide de référence très pratique. Pour les étudiants, l'information donnée ici leur permettra d'obtenir le code source du Système de Gestion de Base de Données Relationnel (SGBDR) PostgreSQL à partir duquel ils pourront étudier comment un moteur de SGBDR SQL est écrit.

Table des matières

1 Introduction

Le but de ce document est de fournir une liste de pointeurs/URL détaillée de façon à mettre rapidement en oeuvre PostgreSQL et de mettre en avant l'intérêt des systèmes à Code Source Ouvert tel que PostgreSQL, Linux.

Tout système informatique au monde a besoin d'une base de données pour stocker/retrouver les informations. La première raison pour laquelle on utilise un ordinateur est de stocker, retrouver et traiter l'information et de faire cela très rapidement, et donc, de faire économiser du temps. En même temps, le système doit être simple, robuste, rapide, fiable, économique et d'utilisation aisée. La Base de Données constitue une **PARTIE VITALE DU SYSTÈME** des plus importantes car elle contient les données indispensables de toute compagnie au monde. Les systèmes de gestion de base de données les plus courants sont basés sur les spécifications ISO (International Standard Organisation) SQL lesquelles sont également basées sur les normes américaines ANSI SQL. Les spécifications les plus courantes généralement utilisées sont l'ISO/ANSI SQL 1992. La prochaine norme, encore en cours de développement est la SQL 1998/99 aussi appelée SQL-3. Les systèmes de gestion de bases de données les plus répandus tels que Oracle, Sybase et Informix s'appuient sur ces standards ou essaient de les implémenter.

Il y a plus de 20 variétés, commerciales/internet, de systèmes de gestion de base de données actuellement utilisés dans le monde et beaucoup plus dans un futur proche. En l'absence d'une norme telle que l'ANSI/ISO SQL, il serait très difficile pour l'utilisateur de développer une application une fois pour toutes et de l'utiliser avec tous les systèmes de gestion de bases de données. L'utilisateur veut développer son application une fois pour toutes en utilisant les normes ISO SQL, ODBC, JDBC afin qu'elle puisse être déployée sur une grande variété de systèmes de gestion de base de données a travers le monde.

PostgreSQL est la Base de Donnée LIBRE la plus populaire au monde qui implémente la plupart des normes ISO SQL, ANSI SQL/98, SQL/92 et ANSI SQL/89 RDBMS. PostgreSQL est une base de données relationnelle Objet de la nouvelle génération et les futures normes ANSI SQL tels que le SQL 1998 (SQL-3) et au-delà traiteront de manière croissante de bases de données Objet et de types de données Objet. PostgreSQL est le seul SGBD (RDBMS - Relational Data Base Management System) au monde qui supporte les bases de données Objet et SQL. Ce document vous explique comment installer le système de gestion de base de données, comment mettre en place la base de données Web, la base de données application , les frontaux GUIs et les programmes interfaçage. On ne saurait trop RECOMMANDER d'écrire vos applications bases de données 100 % compatibles avec les normes ISO/ANSI SQL, ODBC, JDBC ceci rendant votre application portable sur de multiples SGBD (Systèmes de gestion de Bases de Données) tels que PostgreSQL, Oracle, Sybase, Informix etc.

La haute qualité, et un large ensemble de possibilités de PostgreSQL viennent du fait que ce système est développé suivant le principe du «Modèle de développement à Code Source Ouvert». Les modèles à Code Source Ouvert sont ceux ou la totalité du code source est fournie et ou le développement se fait à travers l'internet par une très vaste communauté de cerveaux humains en réseau. La tendance future des développement logiciels réside dans ce que l'on appelle les "super autoroutes de l'information" qui s'étendent à travers le monde entier. Dans les années à venir, la croissance d'internet va être explosive et de ce fait favorisera l'adoption de PostgreSQL par la communauté des utilisateurs de bases de données.

Par l'application des principes de la statistique, des mathématiques et de la science à la qualité des logiciels, on obtient la meilleure qualité des logiciels 'à Code Source Ouvert' tel que PostgreSQL en mettant le code source des programmes à la disposition d'un grand nombre de cerveaux humains interconnectés par les super-autoroutes de l'information. Plus le nombre de cerveaux au travail sera grand, meilleure sera la qualité du logiciel produit. Le modèle "à Code Source Ouvert" évite aussi de **RÉINVENTER LA ROUE**, supprime **LA DUPLICATION DU TRAVAIL**tout en étant particulièrement économique, en diminuant les délais de distribution et en suivant les lois économiques modernes d'optimisation des ressources nationales et globales. Une fois qu'un logiciel a été réalisé par d'autres, vous **N'AVEZ PAS** à recommencer le travail. Vous ne perdrez pas un temps précieux sur quelque chose qui a déjà été **CORRECTEMENT** réalisé. Votre temps est extrêmement précieux et il faut l'utiliser de manière efficace, car vous ne disposez que de 8 heures par jour pour travailler!!. Dans un futur proche, à l'aube du 21-ième siècle, la manière de se procurer un logiciel va changer. Les utilisateurs accorderont en premier lieu leur préférence aux logiciels ouverts tel PostgreSQL, Linux etc...

En achetant des binaires vous n'obtenez pas la véritable propriété du logiciel et ce n'est pas juste. Acheter un logiciel *peut* devenir une attitude archaïque. On a seulement besoin d'acheter un bon matériel, il vaut mieux dépenser de l'argent en matériel et trouver le logiciel sur internet. Le point important, c'est que **le gros du travail** est effectué par le matériel. Le matériel est le cheval de bataille et le logiciel est là pour le piloter. Le matériel est tellement plus compliqué que seulement 6 pays sur 180 au monde ont démontré leur capacité à concevoir et à fabriquer les composants d'ordinateurs. La conception et la fabrication de composants d'ordinateurs est une technologie avancée. C'est un processus très complexe, requérant de très larges investissements en usine et en machines de production maîtrisant la technologie des 0.18 micron. Des milliards de transistors/circuits sont implantés sur une petite puce de silicium. Des compagnies telles que Applied Material, AMD, Intel, Cyrix, Hitachi, Ibm et autres ont dépensé d'importantes quantités d'hommes-ans de travail pour maîtriser des hautes technologies telles que la Micro-électronique et la Nano-électronique (Micro signifie millionième de mètre et Nano signifie milliardième de mètre). La technologie actuelle utilise la micro-électronique à 0.35 microns avec des conducteurs en aluminium, celle de 0.25 microns utilise des conducteurs de cuivre. Dans un futur proche, la technologie des 0.10 microns avec conducteurs de cuivre et même la nano-électronique sera utilisées pour les circuits intégrés d'ordinateurs. Les conducteurs en aluminium seront détrônés par ceux en cuivre sur les puces d'ordinateurs, car le cuivre est meilleur conducteur d'électrons. En photo-lithographie, l'extrême ultraviolet, les rayons X ou les techniques de faisceau d'electrons seront utilisées pour graver des circuits de taille inférieure à 0.15 microns. Dans environ 20 ans, les circuits en silicium seront remplacées par des ordinateurs moléculaires et des circuits biologiques qui seront des millions de fois plus rapides que les circuits en silicium. Les molécules sont des groupes d'atomes, et les atomes sont de minuscules particules constituant la matière première première de tout ce qui nous entoure. Les ordinateurs moléculaires utiliseront les molécules en tant qu'interrupteurs électroniques ultra rapides. Lorsque l'interrupteur est fermé (ON), on obtient la valeur 1 et quand il est ouvert (OFF), la valeur 0. Tous les programmes d'ordinateurs au monde utilisent l'algèbre binaire (nombres 0 et 1). Le tableau ci-dessous montre les progrès et les tendances futures des puces d'ordinateurs.

```
Progrès dans les capacités des puces dans le futur
**************************************************
```

Item/Année	1997	1999	2001	2003	2012	2020
Technologie (micron)	0.25	0.18	0.15	0.13	0.05	< 0.00001
Taille de la puce(mm)	200	300	300	300	45	Mol/bio
Tension mini. de fonctionnement	1.8-2.5	1.5-1.8	1.2-1.5	1.2-1.5	0.5-0.6	< 0.001
Puissance dissipée maximum	70	90	110	130	175	600
fréquence de fonct. puce (MHz)	750	1250	1500	2100	10000	> 50 000
Capacité mémoire DRAM	256 MB	1 GB	2 GB	4 GB	256 GB	> 1000GB

Comme on peut le voir c'est le matériel qui est important et de haute technologie, le logiciel est une technologie moins difficile à maîtriser. Donc la fabrication de matériel est vitale pour l'économie nationale!. Des compagnies comme Compaq, Dell, Sun Microsystems, HP, IBM qui fabriquent des ordinateurs contribuent de façon importante à l'économie U.S. aujourd'hui et dans le futur!!

D'un autre côté, chaque pays au monde développe/produit du logiciel. En fait, n'importe qui dans le monde, avec un petit PC économique, peut créer un système serveur de base de données Oracle. Mais cela lui prendra environ 10 ans (Le serveur de base de données Oracle correspond à plus de 10 hommes-ans de travail). Un homme-an correspond au travail d'un homme à temps plein pendant une année. Si 10 personnes travaillent pendant un an cela correspond à 10 hommes-ans.

Les SGBD tels que Oracle, Informix, Sybase, IBM DB2 (Unix) sont écrits en langage "C" et les binaires créés par compilation de ces sources sont fournis aux clients. Les SGBD Oracle, Sybase, Informix sont des programmes "C" à 100 %!!.

Puisqu'une masse importante de travail a été effectuée sur PostgreSQL au cours des 14 dernières années, cela n'aurait aucun sens de recréer ex-nihilo un autre système de gestion de base de données satisfaisant aux normes ANSI/ISO SQL. Il est bien plus intéressant de prendre le code existant de PostgreSQL, de commencer à l'utiliser, de l'améliorer et d'y ajouter les fonctionnalités manquantes.

PostgreSQL n'est pas seulement un SGBD libre mais aussi un "Produit Internet" de bonne qualité.

On peut prédire que la demande de produits "Fabriqué sur Internet" va croître exponentiellement. Ces produits étant de haute qualité, de faible coût et ayant une base d'utilisateurs et de développeurs extrêmement large. Les pays qui n'utilisent pas les produits "Fabriqué sur Internet" seront les oubliés de "la Révolution mondiale Internet" et resteront loin derrière les autres pays. La raison provient de ce "qu'Internet" lui-même est la **PLUS GRANDE** compagnie de logiciels!

2 Autres Formats de ce Document

Ce document est publié en 10 formats différents qui sont - DVI, Postscript, Latex, LyX, GNU-info, HTML, RTF(Rich Text Format), Texte simple, Pages de manuel Unix et SGML.

- Vous pouvez obtenir ce document HOWTO en tant que fichier tar dans les formats HTML, DVI, Postscript ou SGML de - `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO/other-formats/>`
- Le format texte simple se trouve dans: `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>`
- Des traductions dans d'autres langues telles que le Français, l'Allemand, l'Espagnol, le Chinois se trouvent dans `<ftp://sunsite.unc.edu/pub/Linux/docs/HOWTO>` Toute aide que vous pourriez apporter pour le traduire dans d'autres langues sera la bienvenue.

Ce document est rédigé à l'aide d'un outil appelé "SGML tool" que l'on peut obtenir de - `<http://www.xs4all.nl/~cg/sgmltools/>` Pour compiler le texte source, vous utilisez des commandes telles que :

- sgml2html databasehowto.sgml (pour générer un fichier html)
- sgml2rtf databasehowto.sgml (pour générer un fichier RTF)
- sgml2latex databasehowto.sgml (pour générer un fichier latex)

Ce document se trouve à -

- `<http://sunsite.unc.edu/LDP/HOWTO/PostgreSQL-HOWTO.html>`

Vous pouvez également le trouver sur les sites miroirs suivants -

- `<http://www.caldera.com/LDP/HOWTO/PostgreSQL-HOWTO.html>`
- `<http://www.WGS.com/LDP/HOWTO/PostgreSQL-HOWTO.html>`
- `<http://www.cc.gatech.edu/linux/LDP/HOWTO/PostgreSQL-HOWTO.html>`
- `<http://www.redhat.com/linux-info/ldp/HOWTO/PostgreSQL-HOWTO.html>`
- Vous pouvez trouver d'autres sites miroirs proches de chez vous (au sens d'adresse réseau) à `<http://sunsite.unc.edu/LDP/hmirrors.html>` choisissez un site et allez dans le répertoire /LDP/HOWTO/PostgreSQL-HOWTO.html

Pour visualiser le document au format dvi, utilisez le programme xdvi. Le programme xdvi se trouve dans le paquetage tetex-xdvi*.rpm de Redhat Linux que l'on trouve en choisissant les menus : Panneau de contrôle | Applications | Publication | TeX. Pour lire le document, tapez la commande -

```
        xdvi -geometry 80x90 howto.dvi
man xdvi
```

Et redimensionnez l'écran à l'aide du mulot. Consultez la page de manuel concernant xdvi. Pour naviguer dans le document, utilisez les touches Flèches, Page précédente ,Page suivante. Vous pouvez aussi utiliser les lettres 'f', 'd', 'u', 'c', 'l', 'r', 'p', 'n' pour vous déplacer vers le haut, vers le bas, au centre, vers la page précédente, Pour désactiver le menu expert, tapez 'x'.

Vous pouvez lire le fichier postscript en utilisant le programme 'gv' (ghostview) ou 'ghostscript'. Le programme ghostscript se trouve dans le paquetage ghostscript*.rpm et le programme gv dans le paquetage gv*.rpm de Linux Redhat que l'on peut trouver en choisissant les menus : Panneau de contrôle | Applications | Graphique. Le programme gv est beaucoup plus convivial que ghostscript. Les programmes ghostscript et gv sont disponibles sous d'autres plates-formes telles que OS/2, Windows 95 et NT, vous pouvez donc visualiser ce document sur ces plates-formes. Pour lire le document postscript tapez la commande -

```
gv howto.ps
```

ATTENTION: ce document est volumineux, le nombre total de pages (postscript) imprimées est approximativement de 134.

Vous pouvez lire ce document en format HTML en utilisant Netscape Navigator, Microsoft Internet explorer, Redhat Baron Web browser ou n'importe lequel des 10 butineurs web.

Vous pouvez le lire en format latex de sortie de LyX en utilisant LyX, frontal X-Window pour latex.

3 Les lois de la Physique s'appliquent au logiciel !

Dans ce chapitre, on montrera comment la science joue un rôle important dans la création de nombreux objets comme le logiciel, cet univers, la masse, les atomes, les énergies et même vous-même !. Ce chapitre montre également pourquoi il est important d'avoir des connaissances en science AVANT de commencer à en utiliser les produits. Tout en fait partie - par exemple PostgreSQL, le temps, la masse, l'énergie, les planètes, le soleil, la lune, les étoiles, les super-amas, les humains etc... sont des objets créés par la science. Ce chapitre montre également comment les lois de la science et les statistiques témoignent en faveur de systèmes à code source ouvert comme PostgreSQL et Linux. La vitesse d'internet s'accroissant de jour en jour, et internet devenant de PLUS EN PLUS fiable, les systèmes à code source ouvert gagnent rapidement de l'audience. Et, si les règles de la statistique et les lois de la physique sont correctes, les systèmes à code source fermés sont appelés à DISPARAÎTRE de cette planète.

Les paragraphes ci-dessous montreront - "combien la science est vaste, combien elle est importante pour les hommes et quel est son impact sur des projets logiciels comme PostgreSQL, Linux".

Le développement d'un projet tel que PostgreSQL requiert des ressources telles que de l'énergie et du temps, de ce fait, PostgreSQL est un produit d'énergie et de temps. Puisque l'énergie et le temps ne peuvent d'expliquer que par la science, il y a une corrélation directe entre la physique et les projets tels que PostgreSQL, Linux.

Les lois des sciences (Physique) s'appliquent partout, toujours, à tout ce que vous faites et même aux projets logiciels. La physique est en action pendant que vous parlez (ondes sonores), marchez (frottement entre le sol et vos pieds), lisez un livre ou écrivez un logiciel. Différentes branches des sciences, telles que la physique, la chimie, etc... se rejoignent toutes dans un vaste ensemble, les mathématiques (également nommées Reine des Sciences).

Et tout, dans ce monde, a des racines profondes dans les mathématiques, y compris PostgreSQL. PostgreSQL utilise «L'algèbre moderne» qui constitue une petite branche des mathématiques. L'algèbre moderne concerne la «Théorie des ensembles », «l'algèbre relationnelle», la science des Groupes, des Anneaux, les Collections, Ensembles, Unions, Intersections, Exclusions,Domaines, Listes, etc...

Un logiciel tel que PostgreSQL existe aujourd'hui grâce à l'énergie et au temps. C'est l'énergie qui a créé ce monde, les cerveaux humains et beaucoup d'autres choses!. Et, la masse et l'énergie sont UNE et la **MÊME** entité!. Il y a 100 ans, le fait que la masse et l'énergie soient la même chose n'était pas connu des gens !

Les cellules des cerveaux humains consomment de l'énergie en travaillant (à la création d'un logiciel), transformant de l'énergie chimique en chaleur. Même pendant la lecture de ce paragraphe, les cellules de votre cerveau consomment du carburant et utilisent de l'énergie. Donc INTERROMPEZ IMMÉDIATEMENT VOTRE LECTURE!

L'énergie résultant de l'activité des neurones (cellules du cerveau) peut être mesurée en laboratoire. Par exemple, il y a de nombreux instruments comme le "détecteur de mensonge" et autres instruments médicaux qui peuvent mesurer l'activité du cerveau. Tout cela sous-entend que le cerveau humain est une machine thermodynamique. Et comme le cerveau humain est une machine thermodynamique, les lois de la thermodynamique d'appliquent au cerveau et, par conséquent, la thermodynamique a des effets indirects sur les logiciels comme PostgreSQL.

Suivant la science, il est impossible de construire tout système ou théorie (y compris un système logiciel de gestion de base de données) qui soit parfait à 100 % et sans erreur. C'est comme poursuivre un mirage, nous ne pourrons **JAMAIS** atteindre l'objectif d'obtenir la perfection dans un système ou une théorie. Les équations/discussions mathématiques détaillées pour prouver qu'un "système parfait" (de même pour prouver qu'un "système imparfait") est impossible sont des sujets avancés et hors sujet dans ce document. Et une telle étude mathématique implique la prise en compte d'un nombre infini de dimensions (ainsi que les dimensions primaires) que l'on trouve dans la nature. Malheureusement, les humains ne peuvent voir et conceptualiser que 4 dimensions alors que les mathématiques peuvent facilement en explorer beaucoup d'autres. Les autres dimensions sont **"infiniment plus petites"** que les atomes, et les atomes eux-mêmes sont si minuscules que les yeux humains ne peuvent pas les voir ! Pour analyser et expliquer la naissance et la mort de notre univers, les mathématiques constituent un outil très puissant. Notre univers, si vous le regardez à partir d'un autre univers possède pratiquement une **taille nulle** et vice versa. Ce qui signifie que notre univers n'est pas visible (n'existe pas) pour les gens d'un autre univers ! Et vice versa ! Théoriquement, il est possible de sortir de notre univers et de voyager sur des distances immenses (milliard d'années lumière) en un temps nul et de rentrer en un point différent de l'espace-temps !. La distance entre notre univers et les autres univers est en réalité nulle !.

Bien qu'il y ait un nombre infini de dimensions, celles-ci peuvent être déduites/générées à partir d'un nombre restreint de dimensions PRIMAIRES. C'est-à-dire qu'un nombre infini de dimensions peut se combiner et se ramener aux dimensions primaires. Les dimensions primaires absorbent simplement les autres dimensions sans être détruites. Les mathématiciens utilisent ces dimensions primaires pour comprendre la naissance et la mort des univers. L'univers où vous vivez actuellement a commencé à exister lors d'un **BIG BANG**, causé par les interactions de particules atomiques d'autres dimensions, qui s'est produit il y a des milliards d'années (environ 20 milliards d'années). Juste avant le big bang il y avait un petit point où la longueur, la largeur, la hauteur et le temps étaient NULS (c'est-à-dire que notre univers N'EXISTAIT PAS !!) alors que d'autres univers et les dimensions primaires existaient. Le temps lui-même n'existait PAS là et les atomes, les étoiles, les planètes et les galaxies n'existaient PAS! Les atomes qui sont à l'intérieur de notre corps n'existaient PAS !!. Tellement de choses se sont passées AVANT qu le temps n'existe !

Le big bang et donc la naissance de notre univers résulte de quelques atomes de dimensions primaires. **QUELQU'UN**, (quelque chose ?), a causé le fracassement de quelques atomes minuscules d'autres dimensions pour créer notre univers, et de nouvelles dimensions de temps, de longueur, de largeur et de hauteur étaient nées !. Nous voyons l'effet d'une quelconque main dans se processus. Ce processus **n'est pas parfaitement compris** par les hommes. Ceux-ci essaient de générer un autre univers en laboratoire en simulant le big bang (Un accélérateur très puissant est en construction en Europe, un autre, en cours de construction à Dallas, aux USA, a été arrêté par le congrès US pour cause de coupes budgétaires). Il y a des atomes dans d'autres dimensions exactement comme nous avons des atomes dans notre univers. Théoriquement, on peut créer/générer un nombre infini d'univers !. Ce processus est réversible, ce qui veut dire que notre univers peut complètement s'effondrer sur lui-même et disparaître en quelques atomes d'autres dimensions !. C'est une analogie similaire à la façon dont VOUS êtes nés à partir de de deux cellules minuscules qui se sont FRACASSÉES l'une contre l'autre pour créer une cellule unique. Cette toute petite cellule s'est divisée et multipliée en 6 billions de cellules pour devenir cet être humain de 1.80 mètre (que vous êtes !). Il y a une ressemblance entre les êtres humains et les univers qui naissent puis meurent, un peu comme les humains.

Puisqu'un état PARFAIT (comme IMPARFAIT) est impossible, des univers comme le nôtre sont nés et plus tard mourront selon un processus cyclique. MAIS il ne peut pas y avoir de mort PARFAITE, seules des transformations sont permises par la science !. Actuellement notre univers est en expansion à un rythme formidable, il n'est pas statique mais très dynamique. Cet univers continuera son expansion jusqu'à ce que quelque chose interfère pour entraîner son effondrement par l'application de pressions ayant des dimensions externes qui l'annihileront !!. Il y a deux possibilités : suppression de masse de l'univers via les trous noirs (processus lent), ou ajout de masse à l'univers via les trous blancs (processus rapide d'un big crunch). Des millions d'univers peuvent exister mais seuls très peu d'univers peuvent être bâtis à partir des dimensions primaires. Vous n'êtes pas constitués uniquement d'atomes de cet univers mais aussi d'atomes d'autres dimensions !. Dans les trous noirs les atomes et les particules de notre univers sont engloutis et entièrement transformés et convertis en particules d'autres dimensions !. Au centre d'un trou noir, le temps est nul comme le sont la longueur, la largeur et la hauteur !. Et le trou noir constitue la porte d'ENTRÉE/SORTIE pour d'autres univers. Ces portes d'entrée/sortie peuvent s'ouvrir n'importe où, même à l'intérieur de votre corps !

Il peut y avoir un nombre infini de couleurs, de langages informatiques, d'architectures de circuits d'ordinateurs et de théories, mais il NE PEUT PAS y avoir LA PERFECTION pour UN(E) SEUL(E) couleur, langage informatique, architecture ou système ! Vous ne pouvez obtenir qu'une couleur (longueur d'onde, qu'un système, qu'une base de données ou une théorie PRESQUE PARFAITE ! La nature est comme un kaléidoscope.

En combinant les énergies de millions de personnes à travers le monde via internet il est possible d'obtenir un système **PRESQUE PARFAIT** (y compris un système logiciel de gestion de base de données). Prise individuellement, l'énergie de chacun est microscopique, mais en mettant en réseau un très grand nombre de personnes, l'énergie totale sera énorme, et elle peut être canalisée pour créer un système presque parfait. Les êtres humains sont des créatures à l'énergie et aux ressources finies, portés à faire des erreurs allant de la plus triviale à la plus profonde. Puisque notre propension à faire des erreurs varie largement en fonction de qualifications particulières, d'expérience particulière et des humeurs du moment, la découverte et l'élimination des erreurs logicielles dues à des erreur humaines ordinaires peut être largement facilitée en concentrant un grand nombre d'esprits humains à ce problème.

Il est très clair qu'internet peut mettre en réseau un grand nombre de personnes, ce qui implique qu'internet possède beaucoup d'énergie et de temps qui peuvent créer des produits logiciels de bien meilleure qualité, en un temps plus court, que des compagnies commerciales. Même de grandes compagnies comme Microsoft, IBM ne peuvent pas s'affranchir des lois de la physique mais finalement **SE PLIER** aux lois de la science !

Aujourd'hui, il y a beaucoup trop de systèmes de base de données relationnelles SQL RDBMS dans le monde se référant à une spécification ANSI/ISO SQL unique. **L'humanité ne doit pas perdre son temps à créer de trop nombreux paquetages logiciels SIMILAIRES OU IDENTIQUES** alors qu'il y a trop d'autres sujets en sciences qui mériteraient l'attention.

En conclusion, à cause des lois de la science, les systèmes à "code source ouvert" comme PostgreSQL, Linux seront toujours meilleurs que les systèmes à "code source fermé", et il est possible de prouver cette affirmation scientifiquement.

4 Qu'est-ce que PostgreSQL ?

PostgreSQL Version 6.5.2 est un SGBD libre, son code source complet est fourni. De plus, c'est un SGBD Relationnel-Objet pratiquement conforme (de plus en plus conforme) aux normes ANSI SQL1998,92,89. Il fonctionne sur diverses plates-formes matérielles sous différents Systèmes d'Exploitation.

```
                 Quelquefois des corrections urgentes d'erreurs sont diffusées après la diffusion
                 d'une révision de PostgreSQL. Vous pouvez appliquer ces rustines optionnelles
                 selon les besoins de votre application. Effectuez les étapes suivantes pour
                 appliquer ces rustines -
                         cd /usr/src/postgresql6.5.2
            man patch
                         patch -p0 < patchfile
                         make clean
                         make
```

Les fichiers de mise à jour se trouvent dans

- Rustines PostgreSQL : <ftp://ftp.postgresql.org/pub/patches>

L'objectif ultime et le but final de PostgreSQL est de devenir compatible à 100 % à la norme ANSI/ISO SQL ainsi que de devenir le numéro UN mondial des systèmes de gestion de Base de Données ouverts génériques. PostgreSQL guidera, conduira, contrôlera, suivra et dictera le futur de l'ANSI/ISO SQL. C'est-à-dire, l'implémentation et les idées apparaîtront en premier dans PostgreSQL puis, plus tard, seront incorporés dans l'ANSI/ISO SQL. PostgreSQL représente "l'état de l'art" de la technologie des bases de données.

Le serveur Informix Universal (diffusé en 1997) est basé sur une version précédente de PostgreSQL car Informix a acheté Illustra Inc. et l'a intégrée à Informix. La base de données Illustra était complètement basée sur Postgres (version précédente de PostgreSQL).

PostgreSQL est une amélioration du SGBD POSTGRES, prototype de recherche de SGBD de la prochaine génération. PostgreSQL, tout en conservant le puissant modèle de données et la richesse des types de données de POSTGRES, remplace le langage de requête PostQuel par un sous-ensemble étendu de SQL.

Le développement de PostgreSQL est réalisé par une équipe de développeurs Internet qui sont tous inscrits à la liste de diffusion "PostgreSQL développement mailing list". Son coordinateur actuel est Marc G. Fournier

- *scrappy@postgreSQL.org*

Cette équipe est maintenant responsable des développements actuels et futurs de PostgreSQL. Naturellement, l'utilisateur de la base de données est lui-même développeur de PostgreSQL!. La charge de développement est répartie au sein d'un grand nombre d'utilisateurs finaux de bases de données sur internet.

Les auteurs de PostgreSQL 1.01 sont Andrew Yu et Jolly Chen. De nombreux autres ont contribué au portage, aux tests, à la mise au point et a l'amélioration du code. Le code original Postgres, duquel PostgreSQL est issu, est le résultat de l'effort de nombreux étudiants de troisième cycle, de deuxième cycle et d'enseignants sous la direction du Professeur Michael Stonebraker de l'université de Californie, Berkeley.

Le nom original du logiciel à Berkeley était Postgres. Lors de l'ajout des fonctionnalités SQL en 1995, il fut renommé Postgres95. Ce nom fut changé à la fin de 1996 en PostgreSQL.

Des millions d'exemplaires du SGBD PostgreSQL sont installés comme serveurs, serveurs Web et serveurs d'application. Ce SGBD est très avancé, c'est un SGBD Relationnel-Objet (ORDBMS).

PostgreSQL peut stocker plus de types de données que les types traditionnels entier, caractères, etc. - L'utilisateur peut créer des types, des fonctions, de l'héritage de type etc. PostgreSQL fonctionne sur Solaris, SunOS, HPUX, AIX, Linux, Irix, Digital Unix, BSDi, NetBSD, FreeBSD, SCO unix, NEXTSTEP, Unixware et toutes sortes d'unix. Un portage pour Windows 95/NT est en cours de réalisation.

- Titre : PostgreSQL SQL RDBMS Database (Système de Gestion de Base de Données Relationnelle Objet)
- Version actuelle : 6.5.2
- Age : PostgreSQL a 14 ans. En développement depuis 1985
- Auteurs : Développé par des millions d'universités/compagnies sur internet au cours de ces 14 dernières ANNÉES

PostgreSQL et les éléments qui s'y rapportent dans ce document sont couverts par le COPYRIGHT de l'Université de Californie, Berkeley.

5 Où le trouver?

On peut acheter le CDROM Linux Redhat 5.0, le CDROM Linux Debian ou le CDROM Linux Slackware qui contiennent déjà postgresql sous forme de paquetage (a la fois en code source et en binaires) au :

- site Web Linux System Labs : `<http://www.lsl.com/>` (7 U.S. dollars)
- site Web Cheap Bytes Inc : `<http://www.cheapbytes.com/>` (7 U.S. dollars)
- site Web Principal Debian : `<http://www.debian.org/vendors.html>`

L'organisation PostgreSQL vend un 'CDROM PostgreSQL' qui contient le code source complet et les binaires pour de nombreux systèmes d'exploitation Unix, ainsi que la documentation complète.

- CDROM PostgreSQL au site Web principal à : `<http://www.postgresql.org>` 30 (U.S. dollars)

Distributions uniquement en binaire de PostgreSQL :

- On peut exécuter PostgreSQL sans compiler le source. Récupérer les binaires pour Intel-Linux de l'url `<ftp://www.redhat.com/pub/contrib/i386/>` le fichier se nomme postgresql-6.5.2.i386.rpm. Son format est de type paquetage redhat 'rpm'. Il contient à la fois le source et les binaires de PostgreSQL.
- Sites binaires pour Solaris, HPUX, AIX, IRIX, Linux : `<ftp://ftp.postgresql.org/pub/bindist>` Si vous faites une compilation pour n'importe laquelle des plates-formes envoyer le binaire à ce site, cela peut être utile aux autres.
- site ftp : récupérer les binaires pour Intel-Linux de `<ftp://ftp.redhat.com/pub/contrib/i386/>` le fichier se nomme postgresql-6.5.2.i386.rpm. Son format est de type paquetage redhat 'rpm'. Il contient à la fois le source et les binaires de PostgreSQL.

Sites WWW Web :

- Site Web primaire : `<http://www.postgresql.org/>`
- Site Web secondaire : `<http://logical.thought.net/postgres95/>`
- `<http://www.itm.tu-clausthal.de/mirrors/postgres95/>`
- `<http://s2k-ftp.cs.berkeley.edu:8000/postgres95/>`
- `<http://xenium.pdi.net/PostgreSQL/>`
- `<http://s2k-ftp.cs.berkeley.edu:8000/postgres95/>`

Les sites ftp sont indiqués ci-dessous :-

- Primaire FTP : `<ftp://ftp.postgresql.org/pub>`
- Secondaire FTP : `<ftp://ftp.chicks.net/pub/postgresql>`
- `<ftp://ftp.emsi.priv.at/pub/postgres/>`
- `<ftp://ftp.itm.tu-clausthal.de/pub/mirrors/postgres95>`

- <ftp://rocker.sch.bme.hu/pub/mirrors/postgreSQL>
- <ftp://ftp.jaist.ac.jp/pub/dbms/postgres95>
- <ftp://ftp.luga.or.at/pub/postgres95>
- <ftp://postgres95.vnet.net:/pub/postgres95>
- <ftp://ftpza.co.za/mirrors/postgres>
- <ftp://sunsite.auc.dk/pub/databases/postgresql>
- <ftp://ftp.task.gda.pl/pub/software/postgresql>
- <ftp://xenium.pdi.net/pub/PostgreSQL>

Le code source de PostgreSQL est aussi disponible sur tous les sites miroirs de sunsite.unc (soit environ 1000 sites autour du globe). Il se trouve dans la distribution Linux Red Hat dans le fichier /pub/contrib/i386/postgresql.rpm.

- Pour obtenir la liste des sites miroirs aller à l'url <ftp://sunsite.unc.edu>

6 PostgreSQL à grands pas

Ce chapitre est destiné à vous aider à installer et à faire tourner la base de données très rapidement, en moins de 10 minutes.

6.1 Installation et tests

Étapes rapides pour installer, tester, vérifier et faire tourner PostgreSQL Connectez-vous en tant qu'utilisateur *root*.

```
# rpm -qpl postgre*.rpm (pour consulter la liste des fichiers. Pour la documentation 'man rpm')
# rpm -qpi postgre*.rpm (pour prendre connaissance des informations concernant le paquetage)
# cat /etc/passwd | grep postgres
```

Note: Si vous voyez un utilisateur 'postgres', vous pouvez avoir à faire une sauvegarde et un nettoyage du répertoire postgres et à supprimer l'utilisateur unix 'postgres', une autre solution consiste à renommer l'utilisateur unix 'postgres' en quelque chose du genre 'postgres2'. L'installation doit être effectuée dans un "répertoire propre".

```
# rpm -i postgre*.rpm (Doit installer tous les paquetages clients, devel, data
        et main pour que pgaccess puisse fonctionner )
# chkconfig --add postgresql  (pour lancer pg durant le démarrage, voir 'man
chkconfig')
# cp /usr/lib/pgsql/python/_pg.so /usr/lib/python1.5/lib-dynload (pour
python)
# /etc/rc.d/init.d/postgresql start  (pour lancer postgres)
# su - postgres
bash$ createdb mydatabase  (ceci va créer une base de données de nom  :
'mydatabase')
                 (pour lire la documentation 'man createdb')
bash$ psql mydatabase   (pour lire la documentation 'man psql')
..... dans psql utilisez les flèches haut/bas ou \s pour visualiser l'historique
bash$ export DISPLAY=<hostname>:0.0; pgaccess mydatabase; (consulter 'man
pgaccess')
```

```
Dès maintenant, vous pouvez commencer à taper des commandes SQL pour pgaccess ou psql !!
```

```
bash$ cd /usr/doc/postgresql*
```

Arrivé là, lisez toutes les FAQ, les manuels du programmeur et de l'administrateur ainsi que les documents de prise en main (tutorials) existants.

Pour obtenir des informations complémentaires concernant PostgreSQL, consultez <http://www.postgresql.org>

6.2 Maximum RPM

Téléchargez le livre 'Maximum RPM' à l' <http://www.RPM.org>. le nom du fichier est maximum-rpm.ps.gz Lisez-le sous Linux en utilisant la commande gv -

```
# gv maximum-rpm.ps.gz
```

6.3 Test de PyGreSQL - Interface Python

```
bash$ cd /usr/lib/pgsql/python
bash$ createdb thilo
bash$ psql thilo
thilo=> create table test (aa char(30), bb char(30) );
bash$ /usr/bin/python
>>> import _pg
>>> db = _pg.connect('thilo', 'localhost')
>>> db.query("INSERT INTO test VALUES ('ping', 'pong')")
>>> db.query("SELECT * FROM test")
eins|zwei
----+----
ping|pong
(1 row)
>>>CTRL+D
bash$
..... Wow! Cela a l'air de fonctionner - maintenant, installons-le proprement
bash$ su - root
# cp /usr/lib/pgsql/python/_pg.so /usr/lib/python1.5/lib-dynload
```

6.4 Test de Perl - Interface Perl

```
bash$ cd /usr/doc/postgresql-6.5.2/examples/perl5
```

Note: La variable globale @INC doit inclure le module Pg.pm dans le répertoire site_perl donc utilisez l'option -I ci-dessous

```
bash$ perl -I/usr/lib/perl5/site_perl/5.004/i386-linux-thread ./example.newstyle
```

.... Re_Wow! Vous avez fait tourner un perl qui accède à la base de données PostgreSQL!!

Prenez connaissance des fichiers d'exemples pour l'utilisation de l'interface perl.

6.5 Test des interfaces libpq, libpq++

```
bash$ cd /usr/doc/postgresql-6.5.2/examples/libpq++
bash$ su root   --> to change ownership of examples
# chown -R postgres /usr/doc/postgresql-6.5.2/examples
# exit
bash$ g++ testlibpq0.cc -I/usr/include/pgsql -I/usr/include/pgsql/libpq++
-lpq++ -lpq -lcrypt
bash$ ./a.out  (Note: Ignore Error messages if you get any - as below)
> create table foo (aa int, bb char(4));
No tuples returned...
status = 1
Error returned: fe_setauthsvc: invalid name: , ignoring...
> insert into foo values ('4535', 'vasu');
No tuples returned...
status = 1
Error returned: fe_setauthsvc: invalid name: , ignoring...
> select * from foo;
aa   |bb   |
-----|-----|
4535 |vasu |
Query returned 1 row.
>
>CTRL+D
bash$
```

.... Hourra!! Vous venez de faire tourner les interfaces directes de C/C++ à la base de données PostgreSQL!!

6.6 Test des interfaces Java

Pour pouvoir effectuer ceci vous DEVEZ installer le paquetage jdk-*glibc*.rpm (paquetage Java RPM)

```
bash$ cd /usr/doc/postgresql-6.5.2/examples/jdbc
bash$ echo $CLASSPATH
 --> Devrait afficher  CLASSPATH=/usr/lib/jdk-1.1.6/lib/classes.zip
bash$ export CLASSPATH=$CLASSPATH:.:/usr/lib/pgsql/postgresql.jar
Editer tout le fichier psql.java et enlevez le symbole de commentaire de la ligne 'package'.
bash$ javac psql.java
bash$ java psql jdbc:postgresql:template1 postgres < password>[1] select * from pg_tables;
tablename       tableowner      hasindexes      hasrules
pg_type postgres        true    false   false
pg_attribute    postgres        true    false   false
[2]
CTRL+C
bash$
```

.... Re_Hourra!! Vous venez de faire tourner les interfaces directes de Java à la base de données PostgreSQL!!

6.7 Test des interfaces ecpg

```
bash$ cd /usr/doc/postgresql-6.5.2/examples/ecpg
bash$ ecpg test1.pgc -I/usr/include/pgsql
bash$ cc test1.c -I/usr/include/pgsql -lecpg -lpq -lcrypt
bash$ createdb mm
bash$ ./a.out
```

.... Wow!! vous venez de faire tourner le "C"-SQL / base de données PostgreSQL" intégré!!

6.8 Test des exemples SQL - Types and fonctions définis par l'utilisateur

```
bash$ cd /usr/doc/postgresql-6.5.2/examples/sql
Under-development..
```

6.9 Test des interfaces Tcl/Tk

Il y a un exemple d'interfaces Tcl/Tk dans le programme pgaccess. Lire le fichier /usr/bin/pgaccess à l'aide d'un éditeur de texte -

```
bash$ view /usr/bin/pgaccess
bash$ export DISPLAY=<hostname of your machine>:0.0
bash$ createdb mydb
bash$ pgaccess mydb
```

6.10 Test des interfaces ODBC

1. Procurez-vous le pilote win32 pgsql odbc à `<http://www.insightdist.com/psqlodbc/>` 2. Voir également /usr/lib/libpsqlodbc.a

6.11 Test des interfaces de la feuille de travail Motif MPSQL

Procurez-vous les RPMs auprès de `<http://www.mutinybaysoftware.com>`

6.12 Vérification

Pour vérifier le parfait fonctionnement de PostgreSQL, faites tourner le paquetage de test de Régression :- Connectez-vous en tant que root -

```
# rpm -i postgresql*.src.rpm
# cd /usr/src/redhat/SPECS
# more postgresql*.spec   (pour savoir quels sont les paquetage RPM que
vous devez installer)
# rpm -bp postgresql*.spec  (.. ce qui va préparer le paquetage)

Le test de régression a besoin des Makefiles et de quelques fichiers en-tête
comme *fmgr*.h
qui peuvent être construits par -
# rpm --short-circuit -bc postgresql*.spec ( .. utilisez "short circuit" pour
forcer les choses!)
```

```
Interrompre la construction en pressant les touches CTRL+C quand vous voyez
 'make -C common  SUBSYS.o'
À ce moment là, la configuratioin est réussie, tous les "makefiles" et tous les fichiers en-tête (headers) ont été créés.
Il n'y a rien de plus à faire.
# cd /usr/src/redhat/BUILD
# chown -R postgres postgresql*
# su - postgres
bash$ cd /usr/src/redhat/BUILD/postgresql-6.5.2/src/test/regress
bash$ more README
bash$ make clean; make all runtest
bash$ more regress.out
```

7 PostgreSQL Supporte des Base de Données extrêmement grandes > 200 Gigas

Les performances des machines à cpu 32-bit décroissent rapidement quand la taille de la base de données dépasse 5 GigaBytes. Vous pouvez faire tourner des bases de données de 30 gigas avec un cpu 32-bit cpu mais cela ne sera pas très performant. Les machines à CPU 32 bits imposent une limitation de 2 GB de mémoire RAM, de 2 GB pour le système de fichier et d'autres limitations dues au système d'exploitation.

Si vous avez besoin d'utiliser des bases de données extrêmement grandes, il est fortement recommandé d'utiliser des machines 64-bit telles que : Digital Alpha cpu, Sun Ultra-sparc 64-bit cpu, Silicon graphics 64-bit cpu, Intel Merced IA-64 cpu, machines HPUX 64bit , machines IBM 64-bit. Si l'on compile PostgreSQL avec un cpu 64-bit il pourra supporter d'énormes bases de Données et de grosses requêtes. Les performances de PostgreSQL pour des interrogations sur de grandes tables et de grandes bases de données sera plus rapide de plusieurs ordres de grandeurs que sur des machines à cpu 32-bit . L'avantage des machines 64-bit est qu'elles disposent d'un grand espace d'adressage mémoire et que le système d'exploitation peut gérer de très grands systèmes de fichiers. Cela permet des meilleures performances avec de grandes bases de données, plus de mémoire centrale (RAM), plus de possibilités etc...

8 PostgreSQL est-il fiable?

Le paquetage de "Test de Régression" permet de s'assurer de son bon fonctionnement. Il est inclus (src/test/regress) dans la distribution et permet à l'ordinateur d'effectuer la vérification des opérations SQL standard ainsi que des capacités de PostgreSQL. Le paquetage de test contient déjà des centaines de programmes SQL de test.

L'avantage de ce test effectué par l'ordinateur vient du fait que celui-ci peut effectuer plusieurs millions de test SQL très rapidement. La vitesse de l'ordinateur est un milliard de fois plus rapide que celle de l'esprit humain! Il est donc pertinent d'utiliser la puissance de calcul de l'ordinateur pour valider un logiciel plutôt celle de l'esprit humain.

Si c'est nécessaire, vous pouvez en ajouter beaucoup d'autres. Dans ce cas penser à envoyer ces tests au site PostgreSQL primaire si vous pensez que cela peut être utile aux autres. Le paquetage de Test de Régression permet d'asseoir la confiance des utilisateurs en PostgreSQL et facilite le déploiement rapide de PostgreSQL sur des systèmes en production sans inquiétude majeure.

Le paquetage "Test de Régression" peut être considéré comme un document technique **"TRÈS SOLIDE"** reconnu mutuellement par les développeurs et par les utilisateurs finaux. Les développeurs de PostgreSQL utilisent de façon intensive ce paquetage durant la période de développement et également avant de mettre le logiciel à la disposition des utilisateurs pour en assurer la parfaite qualité.

9 Outil GUI frontal pour PostgreSQL (Interface Utilisateur Graphique)

Le navigateur Web deviendra, dans le futur, l'interface utilisateur graphique le (GUI) le plus populaire. La plus grande partie du code doit être écrite en utilisant le langage de scripts de serveurs Web PHP/Zend (et en le compilant) 17 () combiné avec un peu de JavaScript du côté du client web. Il est recommandé de migrer vos applications existantes sous Windows 95/NT vers PHP/Zend.

Les meilleurs outils sont :

- PHP et Zend (script et compilateur) 17 ()
- X-Designer supporte C++, Java et MFC `<http://www.ist.co.uk/xd>`
- Qt pour Windows95 et Unix à `<http://www.troll.no>` et `<ftp://ftp.troll.no>`
- Code Crusader se trouve sur le cdrom linux, logiciel libre basé sur Code Warrior de MetroWorks `<http://www.kaze.stetson.edu/cdevel/code_crusader/about.html>`
- Code Warrior de MetroWorks `<http://www.metrowerks.com>`
- GNU Prof C++ IDE de Cygnus `<http://www.cygnus.com>`
- Borland C++ Builder pour Linux `<http://www.inprise.com>`
- Borland Java JBuilder pour Linux `<http://www.inprise.com>`

Les langages, par ordre de préférence sont :

1. Le langage de script serveur Web PHP avec Javascript pour le client.
2. Le langage de script Perl en utilisant Perl-Qt ou Perl-Tk 11 ()
3. Le langage C++ Omniprésent avec QtEZ, QT, Lesstiff ou Motif.
4. Java mais les programmes obtenus sont très lents.

D'autres outils sont disponibles : PostgreSQL possède une bibliothèque d'interface Tcl/Tk dans la distribution nommée 'pgTcl'. Il existe un Environnement de Développement Intégré (EDI ou IDE pour integrated development environment) pour Tcl/Tk nommé SpecTcl.

- L'outil Lesstiff Motif `<ftp://ftp.redhat.com/pub/contrib/i386/lesstiff*.rpm>`
- Vibe Java/C++ se trouve àt `<http://www.LinuxMall.com/products/00487.html>`
- JccWarrior `<ftp://ftp.redhat.com/pub/contrib/i386/jcc*.rpm>`
- Tcl/Tk `<http://www.scriptics.com>`
- Une extension orientée objet de Tcl appelée INCR se trouve à `<http://www.tcltk.com>`
- Le site Visual TCL `<http://www.neuron.com>`
- Le rpm Visual TCL de Redhat à `<ftp://ftp.redhat.com/pub/contrib/i386/visualtcl*.rpm>`
- `<http://sunscript.sun.com/>`
- `<http://sunscript.sun.com/TclTkCore/>`
- `<ftp://ftp.sunlabs.com/pub/tcl/tcl8.0a2.tar.Z>`
- FreeBuilder `<ftp://ftp.redhat.com/pub/contrib/i386/free*.rpm>`
- SpecTCL `<ftp://ftp.redhat.com/pub/contrib/i386/spec*.rpm>`
- Kanchenjunga Java RAD Tool pour PostgreSQL `<http://www.man.ac.uk/~whaley/kj/kanch.html>`
- L'Applixware Tool `<http://www.redhat.com>`
- XWPE X Windows Programming Environment `<http://www.rpi.edu/~payned/xwpe/>` `<ftp://ftp.redhat.com/pub/contrib/i386/xwpe*.rpm>`
- XWB X Windows Work Bench `<ftp://ftp.redhat.com/pub/contrib/i386/xwb*.rpm>`
- NEdit `<ftp://ftp.redhat.com/pub/contrib/i386/nedit*.rpm>`

Vous pouvez également utiliser Borland C++ Builder, Delphi, Borland JBuilder, PowerBuilder sous Windows95 pour connecter PostgreSQL sous unix à travers des pilotes ODBC/JDBC.

10 Pilotes d'interface pour PostgreSQL

10.1 ODBC Pilotes pour PostgreSQL

ODBC signifie 'Open DataBase Connectivity'. C'est une norme répandue pour accéder aux informations des bases de données de différents vendeurs. Le fonctionnement des applications écrites en utilisant les pilotes ODBC est garanti quelle que soient les bases de données utilisées PostgreSQL, Oracle, Sybase, Informix etc..

- *PostODBC* `<http://www.insightdist.com/psqlodbc>` est déjà inclus dans les distributions. Consultez le site web principal `<http://www.postgresql.org>`. Il est inclus dans ce CDROM PostgreSQL.
- `<http://www.openlinksw.com>` Open Link Software Corporation distribue des ODBC pour PostgreSQL et d'autres bases de données. Open Link dispose aussi d'ODBC gratuits (quantité limitée) vérifier auprès d'eux.
- Insight ODBC pour PostgreSQL `<http://www.insightdist.com/psqlodbc>` C'est le site officiel PostODBC.

Il existe aussi un projet appelé FreeODBC Pack Package . Il n'y a pas de version PostgreSQL, peut-être pouvez vous y participer.

- `<http://www.ids.net/~bjepson/freeODBC/>` c'est une version gratuite d'ODBC.

10.2 Pilotes UDBC pour PostgreSQL

UDBC est une version statique de pilote de gestionnaire et de DLL d'ODBC indépendant, pour intégrer le support de la connectivité base de données directement au niveau des applications.

- `<http://www.openlinksw.com>` Open Link Software Corporation vends UDBC pour PostgreSQL et autres SGBD. Open Link dispose aussi d'UDBC gratuits (quantité limitée) vérifier auprès d'eux.

10.3 Pilotes JDBC pour PostgreSQL

JDBC signifie 'Java DataBase Connectivity'. Java est un langage de développement indépendant de la plate-forme d'utilisation développé par Sun Microsystems. Les programmeurs Java sont encouragés à écrire leurs applications en utilisant JDBC pour faciliter la portabilité entre différentes plates-formes telles que PostgreSQL, Oracle, informix, etc. Si vous écrivez des applications Java applications vous pouvez obtenir les pilotes JDBC pour PostgreSQL à partir des sites suivants :

Le pilote JDBC est inclus dans la distribution PostgreSQL.

- `<http://www.demon.co.uk/finder/postgres/index.html>` Sun's Java connectivity to PostgreSQL
- `<ftp://ftp.ai.mit.edu/people/rst/rst-jdbc.tar.gz>`
- `<http://www.openlinksw.com>` Open Link Software Corporation vends JDBC pour PostgreSQL et autres SGBD. Open Link dispose aussi d'JDBC gratuits (quantité limitée) vérifier auprès d'eux.
- Site JDBC UK `<http://www.retep.org.uk/postgres>`
- Site JDBC FAQ `<http://eagle.eku.edu/tools/jdbc/faq.html>`

Le site JDBC, le guide et les FAQ sont situés à -

- JDBC HOME `<http://splash.javasoft.com/jdbc>`
- JDBC guide `<http://www.javasoft.com/products/jdk/1.1/docs/guide/jdbc>`
- JDBC FAQ `<http://javanese.yoyoweb.com/JDBC/FAQ.txt>`

10.4 Java pour PostgreSQL

Ces classes pour PostgreSQL seront très utiles au programmeur JAVA.

- `<ftp://ftp.redhat.com/pub/contrib/i386>` et regardez postgresql-jdbc-*.rpm
- `<http://www.blackdown.org>`

11 Pilote d'interface de SGBD Perl (Database Interface DBI) pour PostgreSQL

11.1 interface Perl 5 pour PostgreSQL

PERL est l'acronyme de 'Practical Extraction and Report Language'. Perl est disponible pour tous les systèmes d'exploitation et toutes les plates-formes matérielles au monde. Vous pouvez utiliser Perl sous Windows95/NT, Apple Macintosh iMac, toutes les variétés d'Unix (Solaris, HPUX, AIX, Linux, Irix, SCO etc..), ordinateur central MVS, ordinateur de bureau OS/2, OS/400, Amdahl UTS et beaucoup d'autres. Perl tourne **MÊME** sur de nombreux matériels et systèmes d'exploitations peu courants/généralement inconnus!! Aussi, ne soyez pas surpris si vous voyez perl tourner sur un système d'exploitation très rarement utilisé.

Cette interface est incluse dans la distribution de PostgreSQL. On la trouve dans le répertoire src/pgsql_perl5.

- Contact pour Pgsql_perl5 par email : *E.Mergl@bawue.de*
- On la trouve aussi dans - `<ftp://ftp.kciLink.com/pub/PostgresPerl-1.3.tar.gz>`
- Page d'accueil Perl : `<http://www.perl.com/perl/index.html>`
- Tutoriel Perl, pour un titre de tutoriel, consultez : `<http://reference.perl.com/>`
- La FAQ Perl est à : *http://www.yahoo.com/Computers_and_Internet/Programming_Languages/Perl/*
- Perl GUI Interfaces Utilisateurs Perl-Qt rpm : `<ftp://ftp.redhat.com/pub/contrib/i386>` et cherchez PerlQt-1.06-1.i386.rpm
- Perl GUI Interfaces Utilisateurs Perl-Qt : `<http://www.accessone.com/~jql/perlqt.html>`
- Perl GUI Interfaces Utilisateurs Perl-XForms : `<ftp://ftp.redhat.com/pub/contrib/i386>` et cherchez Xforms4Perl-0.8.4-1.i386.rpm
- Perl GUI Interfaces Utilisateurs Perl-Tk : `<ftp://ftp.redhat.com/pub/contrib/i386>`
- Perl GUI kits : `<http://reference.perl.com/query.cgi?ui>`
- Interfaces de Base de Données Perl : `<http://reference.perl.com/query.cgi?database>`
- Traducteur de Perl en "C" : `<http://www.perl.com/CPAN-local/modules/by-module/B/>` et cherchez Compiler-a3.tar.gz
- Traducteur de Bourne shell en Perl : `<http://www.perl.com/CPAN/authors/id/MERLYN/sh2perl-0.02.tar.gz>`
- awk en perl "a2p" et sed en perl "s2p" sont inclus dans la distribution PERl.
- Consultez également le forum PERL à comp.lang.perl.*

11.2 DBI Interface de Base de Données Perl

11.2.1 Qu'est-ce que DBI?

L'interface de SGBD Perl (Perl Database Interface - DBI) est une interface logicielle d'accès à un SGBD (Application Programming Interface - API) pour le langage PERL. Les spécifications de l'API DBI perl définissent un ensemble de fonctions, de variables et de conventions d'accès à un SGBD cohérent et indépendant du SGBD utilisé. Les informations concernant cette section DBI ont été prises dans la "DBI FAQ" dont l'auteur est Alligator Descartes et sont reproduites ici avec sa permission.

– Alligator Descartes Hermetica peut être contacté à *descarte@hermetica.com*

11.2.2 Pilote DBI pour PostgreSQL DBD-Pg-0.89

Vous pouvez obtenir DBD-Pg-0.89.tar.gz de l'un des sites indiqués ci-dessous :

– DBD-Pg-0.89 : `<http://www.perl.com/CPAN/modules/by-module/DBD/>`
– Comprehensive Perl Archive Network CPAN `<http://www.perl.com/CPAN>`
– Liste des pilotes DBI et pages sur les modules DBI `<http://www.hermetica.com/technologia/perl/DBI>`
– L'information sur DBI se trouve à `<http://www.fugue.com/dbi/>`
– Site ftp primaire `<ftp://ftp.demon.co.uk/pub/perl/db>`
– Liens divers sur DBI `<http://www-ccs.cs.umass.edu/db.html>`
– Liens divers sur DBI `<http://www.odmg.org/odmg93/updates_dbarry.html>`
– Liens divers sur DBI `<http://www.jcc.com/sql_stnd.html>`
– Base de Données PostgreSQL `<http://www.postgresql.org>`

CONTRAINTES :

– - Construire, tester et installer Perl 5 (au moins 5.002)
– - Construire, tester et installer le module DBI (au moins 0.89)
– - Construire, tester et installer PostgreSQL (au moins 6.2)

11.2.3 Support technique sur DBI

Veuillez envoyer vos commentaires et vos comptes-rendus d'erreurs à

– *E.Mergl@bawue.de*

Pensez à inclure la sortie de perl -v, et perl -V, la version de PostgreSQL, la version de DBD-Pg, et la version de DBI dans votre compte-rendu d'erreur.

11.2.4 Qu'est-ce que c'est que DBI, DBperl, Oraperl and *perl?

Pour citer Tim Bunce, l'architecte et l'auteur de DBI :

"DBI est une interface logicielle d'accès aux bases de données (Application Programming Interface -API) pour le langage Perl. Les spécifications DBI API définissent un ensemble de fonctions, de variables et de conventions cohérents d'interfaçage à une base de données indépendant de la base de données utilisée."

En langage simple, l'interface DBI permet aux utilisateurs d'accéder de manière transparente à de multiples base de données. Ainsi, Si vous vous connectez à une base de données Oracle, Informix, mSQL, Sybase ou n'importe quelle autre, vous n'avez pas besoin de connaître les mécanismes sous-jacents de la couche 3GL. L'API définie par DBI fonctionnera sur tous ces types de bases de données.

On obtient un bénéfice du même ordre en ayant la possibilité de se connecter à deux bases de données de différents fournisseurs à l'aide du même script perl, i.e., je veux lire des données d'une base de données Oracle et les insérer dans une Informix à partir du même programme. La couche logicielle DBI permet de le réaliser simplement et efficacement.

DBperl est le nom ancien des spécifications de l'interface. Il est utilisé maintenant pour désigner les modules perl4 d'interfaçage des bases de données tels que oraperl, isqlperl, ingperl et autres. Ces interfaces n'ont pas d'API standard et ne sont généralement pas supportés.

Voici une liste des modules DBperl, de leur équivalent DBI correspondants et du support d'information. Notez que les auteurs cités ici ne maintiennent généralement pas le module DBI de la base de données. Les adresses E-mail n'ont pas été vérifiées et ne doivent être utilisées que pour les questions concernant les modules perl4 listés ci-dessous. Les questions sur les pilotes DBI doivent être directement adressées aux listes de diffusion des utilisateurs DBI.

```
Nom du module SGBD requis         Auteur          DBI
  ----------- -----------------   ------          ---
  Sybperl     Sybase              Michael Peppler DBD::Sybase
                                  <mpeppler@itf.ch>
  Oraperl     Oracle 6 & 7        Kevin Stock     DBD::Oracle
                                  <dbi-users@fugue.com>
  Ingperl     Ingres              Tim Bunce &     DBD::Ingres
                                  Ted Lemon
                                  <dbi-users@fugue.com>
  Interperl   Interbase           Buzz Moschetti  DBD::Interbase
                                  <buzz@bear.com>
  Uniperl     Unify 5.0           Rick Wargo      None
                                  <rickers@coe.drexel.edu>
  Pgperl      Postgres            Igor Metz       DBD::Pg
                                  <metz@iam.unibe.ch>
  Btreeperl   NDBM                John Conover    SDBM?
                                  <john@johncon.com>
  Ctreeperl   C-Tree              John Conover    None
                                  <john@johncon.com>
  Cisamperl   Informix C-ISAM     Mathias Koerber None
                                  <mathias@unicorn.swi.com.sg>
  Duaperl     X.500 Directory     Eric Douglas    None
              User Agent
```

Cependant, certains modules DBI possèdent des couches logicielles d'émulation. Ainsi DBD::Oracle est livré avec une couche d'émulation Oraperl, ce qui permet d'exécuter d'anciens scripts oraperl sans modification. La couche logicielle d'émulation traduit les appels oraperl API en appels DBI et les exécute.

Voici une table des couches d'émulation :

```
Module         Couche d'émulation     État
------           ---------------     ------
DBD::Oracle      Oraperl             Complète
DBD::Informix    Isqlperl            En cours de  développement
DBD::Sybase      Sybperl             Fonctionnelle? ( Nécessite une
                                     vérification)
DBD::mSQL        Msqlperl            En version expérimentale avec
                                     DBD::mSQL-0.61
```

L'émulation Msqlperl est un cas particulier. Msqlperl est un pilote perl5 pour les bases de données mSQL , mais il ne se conforme pas aux spécifications DBI. On désapprouve son utilisation en faveur de DBD::mSQL. On peut télé-charger Msqlperl à partir du site CPAN via :

- <http://www.perl.com/cgi-bin/cpan_mod?module=Msqlperl>

11.2.5 Spécifications DBI

Il existe quelques sources d'information sur DBI. Spécifications DBI

- <http://www.hermetica.com/technologia/perl/DBI/doc/dbispec>

On trouve deux spécifications disponibles à cette adresse: la nouvelle spécification Draft (édition provisoire) DBI qui est un document en évolution rapide à mesure que l'équipe de développement s'approche d'une version stable de l'interface, et l'ancienne spécification historique DBperl à partir de laquelle l'interface DBI actuelle a évolué.

Il faut considérer ce dernier document comme ne présentant qu'un intérêt historique et ne pas l'utiliser en tant que manuel de programmation ou document de référence. Il demeure cependant une source d'informations très utile.

Documentation POD (Plain Old Documentation) Les PODs sont des morceaux de documentation généralement noyés à l'intérieur des programmes perl qui documentent le code "sur place". Ce sont des ressources très utiles pour les programmeurs et les utilisateurs des modules. Les PODs pour DBI et pour les pilotes deviennent monnaie courante et la documentation pour les modules contenant ces PODs peut être lue avec les commandes suivantes.

La Spécification DBI Les PODs pour la spécification DBI peut être lue avec la commande :

perldoc DBI

Oraperl Les utilisateurs de la couche d'émulation fournie avec DBD::Oracle, peuvent s'informer sur la manière de programmer en utilisant l'interface Oraperl en tapant:

perldoc Oraperl

Ce qui permettra d'obtenir une copie à jour de la page de manuel originale écrite par Kevin Stock pour perl4. L'API oraperl y est entièrement listée et décrite.

DBD::mSQL Les utilisateurs du module DBD::mSQL peuvent lire des informations sur quelques fonctions privées et bizarreries de ce pilote en tapant :

perldoc DBD::mSQL

Foire Aux Questions (FAQ) Ce document, la Foire Aux Questions, est aussi disponible en tant que documentation POD! Vous pouvez le lire sur votre propre système en tapant :

perldoc DBI::FAQ

Ceci peut être plus pratique pour ceux qui ne sont pas connectés à l'Internet ou le sont d'une manière peu pratique.

Les POD en général On peut lire des informations sur la manière d'écrire des PODs, ainsi que sur la philosophie des PODs en général en tapant :

perldoc perlpod

Les utilisateurs ayant le module Tk installé seront peut-être intéressés d'apprendre qu'il existe un lecteur de POD basé sur Tk nommé tkpod. Il formate les POD de manière pratique et lisible.

Discussions, Cancans et Observations

- <http://www.hermetica.com/technologia/perl/DBI/tidbits>

Il y a , de temps en temps, une série de discussions de la part de certaines personnes, dans les listes de diffusion sur DBI.

“DBI – L'interface de SGBD en perl5” C'est un article écrit par Alligator Descartes et Tim Bunce sur la structure de DBI. Il a été publié dans le numéro 5 de “The Perl Journal”. Il est extrêmement bon. Allez acheter ce magazine. En fait, achetez les tous. Le site WWW de "The Perl Journal" est :

- <http://www.tpj.com>

"DBperl" Cet article, publié dans l'édition de novembre 1996 du "Dr. Dobbs Journal" traitait de DBperl.

"The Perl5 Database Interface" Cette référence est celle d'un livre à écrire par Alligator Descartes publié par O'Reilly et Associés.

Listes de diffusion Il y a trois listes de diffusion pour DBI gérées par Ted Lemon. On peut s'inscrire à toutes et résilier cette inscription à travers le World Wide Web à :

- Listes de diffusion <http://www.fugue.com/dbi>

Les listes où les utilisateurs peuvent participer sont:

dbi-announce Cette liste de diffusion est réservée uniquement aux annonces. Si vous n'arrivez pas à utiliser le formulaire sur la page WWW indiquée ci-dessus, inscrivez-vous à cette liste de la manière suivante :

- Email: *dbi-announce-request@fugue.com*

avec le mot 'subscribe' dans le corps du message.

dbi-dev Cette liste de diffusion est à l'usage des développeurs pour discuter des idées et des concepts de l'interface DBI, API et des mécanismes des pilotes. Seulement utiles pour les développeurs et les personnes intéressées. Trafic faible. Si vous n'arrivez pas à utiliser le formulaire sur la page WWW indiquée ci-dessus, inscrivez-vous à cette liste de la manière suivante :

- Email: *dbi-dev-request@fugue.com*

avec 'subscribe' dans le corps du message.

dbi-users Cette liste de diffusion est un lieu de discussion générale utilisée pour les rapports d'erreurs, la discussion sur différents problèmes et des demandes de renseignement d'intérêt général. Si vous n'arrivez pas à utiliser le formulaire sur la page WWW indiquée ci-dessus, inscrivez-vous à cette liste de la manière suivante :

- Email: *dbi-users-request@fugue.com*

avec 'subscribe' dans le corps du message.

Archives des Listes de Diffusion

- Archive des listes de diffusion US <http://outside.organic.com/mail-archives/dbi-users/>
- Archive des listes de diffusion Européennes <http://www.rosat.mpe-garching.mpg.de/mailing-lists/PerlDB-Interest>

11.2.6 Problèmes de compilation ou "Il échoue aux tests"

Si vous avez un vidage mémoire, essayez le module Devel::CoreStack pour générer une trace de la pile du vidage mémoire. On peut trouver Devel::CoreStack à :

- <http://www.perl.com/cgi-bin/cpan_mod?module=Devel::CoreStack>

Envoyez un courrier électronique sur la Liste dbi-users contenant la trace de la pile, les versions des modules, la version de perl, les situations de test, la version du système d'exploitation et toutes autres informations pertinentes. Plus vous fournirez d'informations plus vite les développeurs pourront résoudre les problèmes. Si vous ne nous envoyez rien, n'attendez rien en retour.

11.2.7 DBI est-il supporté sur les plates-formes Windows 95 / NT?

Les portages de DBI et de DBD::Oracle pour Win32 ports font maintenant partie intégrante de DBI, donc, la récupération d'une version de DBI supérieure à 0.81 doit donner satisfaction. Vous pouvez accéder aux bases de données Microsoft Access et SQL-Server à partir de DBI via ODBC. Une "couche d'émulation" DBI expérimentale est fournie avec DBI-0.79 (et suivants) pour le module Win32::ODBC. Son nom est DBI::W32ODBC. Vous aurez besoin du module Win32::ODBC.

- Win32 DBI `<http://www.hermetica.com/technologia/perl/DBI/win32>`
- Win32 ODBC `<http://www.roth.net>`

11.2.8 Qu'est-ce que DBM? Et pourquoi dois-je utiliser DBI à sa place?

A l'origine UNIX était bienheureux avec sa "Base de Données" rustique reposant sur des fichiers, nommée système dbm. Avec dbm vous enregistrez les données dans des fichiers et les retrouvez rapidement. Cependant, il souffre de sérieux inconvénients.

Verrouillage des fichiers

Les systèmes dbm ne permettent par un verrouillage particulièrement robuste des fichiers, de même qu'il n'y a pas de possibilité de corriger les problèmes survenants lors d'écritures [dans la base de données] simultanées.

Structures de Données Arbitraires

Les systèmes dbm permettent seulement une simple structure de données fixe: paires clé-valeur. Cette valeur peut être un objet complexe, tel qu'une structure [C], mais la clé doit être unique. Ce fut une grande limitation dans l'utilité des systèmes dbm.

Cependant, les systèmes dbm continuent d'offrir des fonctions utiles pour les utilisateurs ayant des ensembles de données simples et des ressources limitées, puisqu'ils sont rapides, robustes et extrêmement bien testés. Les modules Perl pour accéder aux systèmes dbm font maintenant partie intégrante de la distribution Perl via le module AnyDBM_File."

Pour résumer, DBM est une solution parfaitement satisfaisante pour les bases de données essentiellement en lecture seule, ou pour des ensembles de données simples et réduits. Toutefois, pour des ensembles de données plus importants, sans mentionner un verrouillage des transactions robuste, on recommandera aux utilisateurs de préférer DBI.

11.2.9 Est-ce que < insérez une fonctionnalité ici > est supporté par DBI?

Si l'on suppose que la fonctionnalité en question n'est pas, en standard, spécifique d'un SGBD, alors la réponse sera non.

DBI représente un API qui doit fonctionner avec la plupart des SGBD, et n'a pas de fonctionnalité spécifique à un SGDB particulier.

Cependant, les auteurs d'un pilote peuvent, s'ils le désirent, ajouter une fonctionnalité spécifique à un SGBD à travers les méthodes func définies dans l'API DBI. Les développeurs de Scripts doivent noter que l'utilisation de cette fonctionnalité au travers de ces méthodes func a de bonnes chances d'en sacrifier la portabilité entre les différents SGBD.

11.2.10 Est-ce que DBI est d'une quelconque utilité pour la programmation CGI?

En un mot, oui! DBI est extrêmement utile pour la programmation CGI! En fait, la programmation CGI est une des deux principales utilisation de DBI.

DBI confère aux programmeurs CGI la possibilité d'offrir des base de données WWW à leurs utilisateurs, ce qui donne à ces utilisateurs la possibilité d'utiliser de grandes quantités de données bien organisées. DBI donne aussi la possibilité , si un site reçoit un trafic trop important pour les performances du serveur, d'améliorer ce serveur de base de données de façon transparente, sans modifier les scripts CGI.

11.2.11 Comment puis-je obtenir un temps de connexion plus rapide avec DBD::Oracle et CGI CGI?

Le serveur httpd Apache maintient un ensemble de processus fils httpd pour servir les requêtes clients.

En utilisant le module mod_perl Apache de Doug MacEachern, l'interpréteur perl est inclus dans le processus fils httpd. Les modules CGI, DBI, et vos autres modules favoris peuvent être chargés au lancement de chaque processus fils. Ces modules ne seront pas rechargés à moins d'être modifiés sur disque.

Pour de plus amples informations sur Apache, consultez le site WWW du Projet Apache à :

- Site WWW du Projet Apache `<http://www.apache.org>`
- Module Mod_perl `<http://www.perl.com/cgi-bin/cpan_mod?module=mod_perl>`

11.2.12 Comment puis-je obtenir une connexion persistante avec DBI et CGI?

En utilisant le module Apache::DBI de Edmund Mergl, les connexions à la base de données sont enregistrées dans une table avec chacun des processus httpd fils. Si votre application utilise une base de données simple utilisateur, cette connexion peut être lancée avec chaque processus fils. Actuellement, les connexions à la base de données ne peuvent pas être partagées entre processus httpd fils. Apache::DBI peut être télé-chargé de CPAN via :

- `<http://www.perl.com/cgi-bin/cpan_mod?module=Apache::DBI>`

11.2.13 "Quand je lance un script perl de la ligne de commande, ça marche, mais, quand je le lance sous httpd, ça échoue!" Pourquoi?

Fondamentalement, il y a une bonne chance que cela provienne du fait que l'utilisateur à partir duquel vous avez lancé la ligne de commande a un ensemble de variables d'environnement correctement configuré, ce sont, dans le cas de DBD::Oracle, des variables telles que $ORACLE_HOME, $ORACLE_SID ou TWO_TASK. Le processus httpd s'exécute habituellement sous un utilisateur id ne correspondant pas à un utilisateur, ce qui implique qu'il n'y a pas d'environnement configuré. Tous scripts essayant de s'exécuter dans ces circonstances échoueront. Pour résoudre ce problème, initialisez l'environnement de votre base de données dans un bloc BEGIN ( ) en tête de votre script. Ceci devrait résoudre votre problème. De même, vous devriez regarder votre fichier registre d'erreurs pour y trouver des indices, ainsi que les guides "Idiot's Guide To Solving Perl / CGI Problems" et "Perl CGIProgramming FAQ" pour avoir des informations complémentaires. Il est peu probable que ce problème concerne DBI. Lisez ces DEUX documents très soigneusement !

- Idiot's Guide to Solving Perl / CGI problems `<http://www.perl.com/perl/faq/index.html>`

11.2.14 Puis-je faire de l'exécution en parallèle avec DBI?

A la date de ce document, non. perl ne permet pas l'exécution en parallèle. Cependant, l'exécution en parallèle doit faire partie de la distribution perl de base à compter de la version 5.005, ce qui sous-entend que le support de l'exécution en parallèle pour DBI devrait suivre rapidement. Pour quelques exemples de code OCI pour Oracle ayant des instructions SELECT avec exécution en parallèle, voir :

- <http://www.hermetica.com/technologia/oracle/oci/orathreads.tar.gz>

11.2.15 Comment puis-je invoquer des procédures enregistrées avec DBI?

En supposant que vous avez créé une procédure enregistrée à l'intérieur de la base de données cible, e.g., une base de données Oracle, vous pouvez utiliser $ dbh-> do pour exécuter immédiatement cette procédure. Par exemple,

$ dbh-> do("BEGIN someProcedure END");

11.2.16 Comment puis-je récupérer les valeurs de retour de procédures enregistrées avec DBI?

N'oubliez pas d'effectuer un test d'erreur, strict!

```
$sth = $dbh->prepare( "BEGIN foo(:1, :2, :3); END;" );
$sth->bind_param(1, $a);
$sth->bind_param_inout(2, \$path, 2000);
$sth->bind_param_inout(3, \$success, 2000);
$sth->execute;
```

11.2.17 Comment puis-je créer ou supprimer une base de données avec DBI?

La création et la suppression de bases de données sont des concepts qui sont beaucoup trop abstraits pour être supportés par DBI. Par exemple, Oracle ne supporte pas le concept de détruire une base de données du tout ! Ainsi, dans Oracle, le serveur de base de données est essentiellement la base de données elle-même alors que dans mSQL, le processus serveur s'exécute tranquillement sans aucune base de données créée. C'est un problème trop hétérogène pour s'y attaquer. Quelques pilotes, cependant, supportent la création et la suppression de bases de données à travers des méthodes func privées. Il vous faut regarder dans la documentation des pilotes que vous utilisez pour vérifier s'ils supportent de tels mécanismes.

11.2.18 Comment les valeurs NULL sont-elles prises en compte par DBI?

Les valeurs NULL dans DBI sont traitées comme la valeur undef. Des NULLs peuvent être insérés dans les bases de données en tant que NULL, par exemple :

```
$rv =
    $dbh->do( "INSERT INTO table VALUES( NULL )" );
```

mais lors d'une interrogation, les NULLs devront être testés comme des undef. C'est une norme pour tous les pilotes.

11.2.19 Qu'est-ce que c'est que ces histoires de méthodes func?

Une méthode func est définie à l'intérieur de DBI comme étant un point d'entrée pour une fonctionnalité d'une base de données spécifique, eg, la possibilité de créer ou supprimer des bases de données. L'invocation de ces méthodes spécifiques aux pilotes est simple. Par exemple, pour invoquer une méthode createDatabase qui n'a qu'un seul argument, on écrira :

```
$rv = $dbh->func( 'argument', 'createDatabase' );
```

Les développeurs de logiciels doivent cependant noter que ces méthodes func ne sont pas portables entre SGBD.

11.2.20 Assistance et formation payante

L'interface aux SGBD Perl5 est un logiciel LIBRE. IL EST DISTRIBUE SANS GARANTIE D'AUCUNE SORTE. Cependant, quelques organisations fournissent soit une assistance technique soit des programmes de formation pour DBI.

PERL CLINIC : La société "Perl Clinic" peut offrir des contrats d'assistance payants pour Perl, DBI, DBD::Oracle et Oraperl. Cette assistance est fournie par la compagnie où travaille Tim Bunce, auteur de DBI. Pour de plus amples informations concernant leurs services, consultez :

- `<http://www.perl.co.uk/tpc>`

12 Outils de Gestion PostgreSQL

12.1 PGACCESS - Un GUI pour la gestion de PostgreSQL

PgAccess est une interface Tcl/Tk à PostgreSQL. Il est déjà inclus dans la distribution de PostgreSQL. Pour une copie plus récente, vous pouvez accéder au site web :

- `<http://www.flex.ro/pgaccess>`
- Si vous avez des commentaires, des suggestions pour des améliorations, n'hésitez pas à envoyer un courrier électronique à : *teo@flex.ro*

Fonctionnalités de PgAccess

Fenêtres de PgAccess : Fenêtre principale, créateur de tables, vue de table (interrogation), créateur visuel d'interrogation .

Tables

- - Ouverture de tables pour la visualisation, 200 enregistrements maximum (modifiable dans le menu des préférences)
- - re-dimensionnement d'une colonne en tirant la ligne verticale de la grille (plus pratique à réaliser à l'intérieur même de la table que dans l'en-tête)
- - texte encadré dans les cellules - disposition enregistrée pour chacune des tables)
- - importation/exportation vers des fichiers externes (SDF,CSV)
- - possibilité de filtres (filtre de saisie, tel que (prix>3.14)
- - possibilités de tri (saisie manuelle de(s) champ(s) à trier)
- - édition en ligne
- - assistant générateur de table amélioré
- - édition de champ améliorée

Requêtes

- - définition, édition et enregistrement de "requêtes définies par l'utilisateur"

- - enregistrement de requêtes en tant que vues
- - exécution des requêtes
- - visualisation des résultats de requêtes de type select
- - suppression et changement de nom de requête
- - créateur visuel de requêtes avec possibilités de "glisser/déposer". Pour tous ceux d'entre vous qui ont l'extension pour Netscape Navigator de Tcl/Tk installée, vous pouvez le voir au travail en cliquant ici

Séquences

- - définition de séquences, ainsi que leurs suppression et inspection Fonctions
- - définition, inspection et suppression de fonctions en langage SQL

Les futures implémentations disposeront

- - de la conception des tables (ajout de nouveaux champs, changement de nom, etc.)
- - de la définition de fonction
- - d'un générateur de rapport
- - d'un langage de script de base

Si vous avez des commentaires, ou des suggestions d'amélioration, envoyez un courrier électronique à :

- *teo@flex.ro*

Informations concernant libgtcl

Vous aurez également besoin de la bibliothèque d'interface de PostgreSQL à Tcl, disponible comme module Tcl/Tk chargeable. Son nom est libpgtcl et le source est situé dans le répertoire PostgreSQL /src/interfaces/libpgtcl. Précisément, vous aurez besoin d'une bibliothèque libpgtcl qui soit "chargeable" à partir de Tcl/Tk. Ce qui est techniquement différent d'un fichier objet chargeable PostgreSQL ordinaire, car libpgtcl est constituée d'un ensemble de fichiers objets. Sous Linux, on l'appelle libpgtcl.so. Vous pouvez télé-charger à partir d'ici une version déjà compilée pour les systèmes Linux i386. La seule chose à faire est de copier libpgtcl.so dans le répertoire bibliothèque système (/usr/lib) et c'est tout. Une solution possible est de supprimer dans le source la ligne contenant load libpgtcl.so et de charger pgaccess.tcl non pas avec wish, mais avec pgwish (ou wishpg) le wish qui a été lié avec la bibliothèque libpgtcl.

12.2 Outil Windows interactif de génération de requête pour PostgreSQL

("Windows Interactive Query Tool" WISQL ou MPSQL) MPSQL offre une interface SQL graphique à PostgresSQL. MPSQL est identique à la feuille de programmation SQL d'Oracle et au serveur de requête SQL de Microsoft WISQL! Il dispose d'un GUI agréable et d'un historique des commandes. Vous pouvez utiliser le coupé/collé ainsi que d'autres fonctionnalités sympathiques pour accroître votre productivité.

- `<http://www.troubador.com/~keidav/index.html>`
- Email : *keidav@whidbey.com*
- `<http://www.ucolick.org/~de/>` dans le fichier tcl_syb/wisql.html
- `<http://www.troubador.com/~keidav/index.html>`
- Email : *de@ucolick.org*

12.3 Outil interactif de génération de requête pour PostgreSQL - ISQL

ISQL est destiné aux terminaux fonctionnants en mode ligne de commande. C'est inclus dans la distribution et nommé PSQL. Il est très similaire à Sybase ISQL, et à Oracle SQLplus. A l'invite unix tapez la commande 'psql' qui en retour vous affichera le caractère d'attente de commande: psql> .

```
Tapez /h pour obtenir de l'aide sur les commandes utilisables.
```

C'est très convivial et d'utilisation facile. C'est aussi très utile pour écrire des scripts pour les interpréteurs de commandes.

12.4 MPMGR - Un outil de gestion de Base de Données pour PostgresSQL

MPMGR fournira un outil graphique de gestion de Base de Données pour PostgresSQL. Vous pouvez le trouver à

- `<http://www.mutinybaysoftware.com/>`
- Email: *keidav@mutinybaysoftware.com*
- `<http://www.troubador.com/~keidav/index.html>`
- Email: *keidav@whidbey.com*
- `<http://www.troubador.com/~keidav/index.html>`
- `<http://www.ucolick.org/~de>` dans le fichier tcl_syb/wisql.html
- WISQL for PostgreSQL `<http://www.ucolick.org/~de/Tcl/pictures>`
- Email: *de@ucolick.org*

13 Mise en place de plusieurs machines PostgreSQL avec un seul moniteur

Vous pouvez empiler plusieurs cpu et les connecter à un unique moniteur en utilisant un commutateur KVM (Keyboard, Video, Monitor) pour choisir la connexion avec une unité centrale particulière. Ce qui économise de l'espace et évite l'utilisation de plusieurs écrans, claviers et mulots (économisant environ de 100 à 500 US dollars par ensemble) et évite aussi pas mal de pagaille.

En utilisant ce commutateur, vous pouvez empiler plusieurs serveurs PostgreSQL (développement, test, production), serveurs Web, serveurs ftp, serveurs Intranet, serveur de Courrier, serveurs de Nouvelles, sur une étagère. Le boîtier commutateur peut être également utilisé pour contrôler des machines Windows 95/NT ou OS/2.

Veuillez consulter ces sites :

- DataComm Warehouse Inc.au 1-800-328-2261. Ils fournissent toutes sortes de matériels informatique `<http://www.warehouse.com>` Un commutateur manuel 4-ports KVM (PS/2) vaut environ $89.99. Référence de l'article : DDS1354
- Network Technologies Inc `<http://www.networktechinc.com/servswt.html>` (120 dollars/PC 8 ports) qui propose des commutateurs de serveurs 'Server Switches' et des commutateurs uniquement vidéo 'Video only switches'
- Scene Double Inc, England `<http://www.scene.demon.co.uk/qswitch.htm>`
- Cybex corporation `<http://www.cybex.com>`
- Raritan Inc `<http://www.raritan.com>`
- RealStar Solutions Inc `<http://www.real-star.com/kvm.htm>`
- Belkin Inc `<http://www.belkin.com>`
- DataComm Warehouse Inc Appels 24 heures sur 24, 7 jours sur 7 au 1-800-328-2261 Ils fournissent toutes sortes de matériels d'ordinateur `<http://www.warehouse.com>`
- Better Box Communications Ltd. `<http://www.betterbox.com/info.html>`
- Allez chez votre marchand de matériel le plus proche et demandez un commutateur de serveur "Server Switch" également connu sous le nom "KVM Auto Commutateurs".

Utilisez le moteur de recherche yahoo pour trouver d'autres compagnies en utilisant la clé de recherche 'Server Switches' ou "KVM Switches".

Pour avoir de meilleures performances, on recommande instamment d'avoir une machine unix pour chaque processus serveur PostgreSQL. Aucun autre programme/traitement ne doit tourner sur cette machine. Consultez la page Affaires de votre journal local pour trouver des vendeurs de machines intel nues, avec un moniteur monochrome de 13" (moniteur particulièrement économique). Les vendeurs locaux vendent uniquement du matériel **SANS** un quelconque Microsoft Windows/DOS (ce qui vous fait économiser environ $ 150). Vous n'avez pas besoin d'un moniteur couleur pour un serveur de base de données puisque vous pouvez en effectuer l'administration à distance à partir d'une station de travail PC couleur. Procurez-vous le cdrom Linux à l'adresse ci-dessous -

- Site Web Linux System Labs : `<http://www.lsl.com/>` 7 (U.S. dollars)
- Site Web Cheap Bytes Inc : `<http://www.cheapbytes.com/>` 7 (U.S. dollars)

Assurez-vous que le matériel que vous achetez est supporté par le Linux Redhat. Vérifiez sur le site ftp de Redhat le matériel recommandé tels que les coupleurs SCSI, les cartes video avant d'acheter. Pour seulement $ 600 vous obtiendrez une machine intel puissante avec un Linux Redhat faisant tourner PostgreSQL. Utilisez odbc/jdbc/perl/tcl pour vous connecter à PostgreSQL à partir de Windows95, OS/2, Unix Motif ou de butineurs web (e.g. Redbaron, Opera, Netscape, 20 autres). (Les butineurs web sont en train de rapidement devenir le client standard).

En suivant ce conseil, vous pouvez contrôler de nombreux systèmes à l'aide de seulement un clavier et un moniteur!

On trouvera ci-dessous un **extrait du catalogue de networktechnic Inc**

Commandez maintenant, appelez le 800-742-8324 (appel gratuit à partir des USA)

Pour recevoir notre catalogue, veuillez envoyer votre adresse par courrier électronique à : sales@networktechinc.com

PILOTEZ PLUSIEURS PC's avec un seul clavier, écran et mulot Ces commutateurs électroniques permettent de piloter jusqu'à 64 PC's avec un unique clavier, écran et mulot. Son microprocesseur interne simule la présence constante du clavier, de l'écran et du mulot pour chacun des PC's connectés.

Fonctionnalités et Applications

- Clavier, face avant ou commande à distance
- utilisation de connecteurs de mulots 9 broches D Série ou 6 broches miniDIN
- utilisation de connecteurs de clavier 5 broches DIN ou 6 broches miniDIN
- résolution vidéo 1600x1200 sans dégradation
- Utilisation de câbles standards
- Boîtier pour bureau ou pour montage en rack

Spécifications - Clavier

- Tous les connecteurs dont femelles
- 5 broches DIN ou 6 broches miniDIN
- Permettent un Re-démarrage (boot) matériel ou logiciel de tous les PC's à chaque instant

Contrôles - Boutons en face avant

- Le toucher d'un bouton permet la connexion au PC correspondant
- L'appui sur un bouton pendant plus de 0.5 secondes entraîne le fonctionnement dans les modes SCAN, BROADCAST ou COMMAND

- Les LEDs en face avant indiquent le mode d'opération du clavier
- L'appui sur CTRL+* entraîne le fonctionnement en mode COMMAND
- Les modes SCAN, BROADCAST ou COMMAND sont disponibles

Mulot

* 9 broches D série

- o Le commutateur NTI émule le mulot Microsoft série pour tous les PC's
- o 9 broches D mâle pour mulot
- o 9 broches D femelle pour PC's

* 6 broches miniDIN

- o Le commutateur NTI émule le mulot IBM PS/2 pour tous les PC's
- o Tous les connecteurs sont femelles

Par câble, à distance

- Optionnel–doit être acquis séparément
- Fonctionnement identique à celui "Boutons en face avant"

Moniteur (écran) - VGA

- Bande passante 150 MHz
- Résolution 1600X1200 sans dégradation
- Tous les connecteurs sont des connecteurs femelles

14 Outils et Applications pour PostgreSQL

14.1 PostgreSQL 4GL pour les applications de bases de données web - Le système de développement AppGEN

AppGEN peut être télé-chargé de

- `<http://www.man.ac.uk/~whaley/ag/appgen.html>`
- `<ftp://ftp.mcc.ac.uk/pub/linux/ALPHA/AppGEN>`.

AppGEN est un langage de haut niveau de la quatrième génération ainsi qu'un générateur d'application pour produire des applications destinées au World Wide Web (WWW). Ces applications sont typiquement utilisées à travers l'internet ou dans le cadre d'un réseau intranet d'entreprise. Les applications AppGEN sont implantées en scripts C respectants la norme Common Gateway Interface (CGI) utilisé par la plupart des serveurs Web.

Pour utiliser AppGEN il vous faudra disposer de :-

PostgresSQL, système de gestion de base de données relationnel

Un serveur web compatible CGI tel que HTTPD du NCSA

D'un compilateur ANSI C tel que GCC

AppGEN est constitué des exécutables Unix (Linux) suivants :-

- defgen, qui produit une application cadre, de base, à partir de la structure logique des données. Ces applications sont capables d'ajouter, de mettre à jour, de supprimer et de rechercher les enregistrements d'une base de données tout en conservant automatiquement l'intégrité référentielle de cette base.

- appgen, le compilateur AppGEN qui compile le code source appgen en code source C CGI exécutable et en documents formatés HTML prêts pour leur déploiement sur un serveur web.
- dbf2sql, un utilitaire de conversion de fichiers .dbf compatibles dBase III en scripts SQL exécutables. Ce qui permet de migrer les données stockées dans la plupart des bases de données DOS/Windows vers un serveur SQL tel que PostgresSQL.
- De plus, AppGEN comprend un ensemble de documents HTML , de fichiers GIF et d'applets Java qui sont utilisés, à l'exécution, par le système. Et, naturellement, comme pour tout bon logiciel, la totalité du code source est fournie.

L'auteur, Andrew Whaley, peut être contacté à

- *andrew@arthur.smuht.nwest.nhs.uk*

14.2 WWW Web interface pour PostgresSQL - DBENGINE

dbengine est une interface Web "plug 'n play" pour Postgres95 créée par Ingo Ciechowski. Il est à :

- `<http://www.cis-computer.com/dbengine/>`

A propos de DBENGINE : dbengine est une interface entre le WWW et Postgres95 qui permet un accès simple à n'importe quelle base de données existante en seulement quelques minutes.

PHP 3 offre une sorte de langage Perl dans vos documents, mais pas le vrai Perl alors que AppGen et wdb-p95 nécessite la création d'un fichier de configuration pour chacune de vos bases de données – c'est un peu comme si vous deviez apprendre un nouveau méta-langage avant de pouvoir commencer à travailler.

A la différence des autres outils, vous n'avez pas à apprendre un langage de programmation particulier ou un langage de script pour commencer à utiliser dbengine. De plus, il n'y a pas à créer de fichier de configuration pour chaque base de données, et vous n'avez donc pas besoin de vous familiariser avec sa structure. Cependant - au cas où vous voudriez profiter de toutes les possibilités de dbengine, ce serait une bonne idée de connaître le langage Perl.

Le système tout entier peut être configuré à l'aide de simples manipulations d'une base de données complémentaire qui contient les informations de détail sur la manière de visualiser vos accès à la base de données. Vous pouvez même spécifier des Champs Virtuels qui sont calculés en temps réel avant d'être affichés à l'écran.

Licence :

dbengine est un logiciel libre selon les mêmes conditions que Perl. Lisez sa licence si vous n'êtes pas sûr de ce que vous pouvez ou ne pouvez pas faire. La dernière ligne indique que c'est une version plus gentille et plus modérée que celle de la licence GNU – une de celle qui n'affecte pas votre travail si vous extrayez des parties de dbengine ou du paquetage pour l'inclure dans un produit commercial!

15 Module Apache Webserver pour PostgreSQL - NeoSoft NeoWebScript

Apache est un serveur Web bien connu. On trouvera un module d'interface de PostgreSQL pour le serveur Web Apache à -

- `<http://www.neosoft.com/neowebscript/>`

NeoWebScript est un langage de programmation qui permet d'inclure des programmes simples ou compliqués dans des fichiers HTML.

Quand une page HTML contenant un script NeoWebScript est appelée, le serveur Web ayant le NeoWebScript activé, exécute le(s) script(s) inclus, produisant une page Web dont le contenu, personnalisé, a été créé par le programme.

NeoWebScript est un moyen rapide, sûre, facile à apprendre de créer des programmes interactifs puissants directement en code HTML dans des pages Web. Avec NeoWebScript, les compteurs, les formulaires de courrier électronique, murs à graffitis, livres d'or et suivi des visiteurs sont aisés, même pour un programmeur débutant. Regardez comment NeoWebScript se défend bien par rapport à PERL et JavaScript.

Au cas où vous souhaiteriez installer NeoWebScript sur votre serveur web, votre webmestre doit lire notre "FAQ Sysop" pour se lancer. Le document "Théorie de fonctionnement" explique comment NeoWebScript fonctionne, celui d'Installation est un guide pas à pas des opérations à effectuer. Le guide de Gestion traite de la manière de configurer et de faire fonctionner le serveur, celui de Tests permet de vérifier le fonctionnement correct de NeoWebScript, enfin, celui de Maintenance traite des problèmes de serveur.

Il n'y a aucuns frais pour l'utilisation de NeoWebScript-2.2 que ce soit pour votre ISP, votre intranet, ou votre extranet. Vous pourrez voir un licence complète quand vous vous enregistrerez pour télé-charger le logiciel, mais il en coûte $ 99 si vous désirez l'inclure dans votre propre produit ou l'utiliser dans un serveur commercial (eg. SSL).

NeoWebScript est un module pour le serveur web Apache qui vous permet d'inclure , en tant que langage de script, pour vos pages web, le langage de programmation Tcl/Tk . Il a été inventé par Karl Lehenbauer, Directeur Technique chez NeoSoft, et documenté, renforcé et étendu par les programmeurs de NeoSoft et par des rédacteurs techniques.

Le serveur Apache est le serveur web le plus populaire au monde, totalisant 68 % des sites interrogés.

Tcl/Tk est un langage de script puissant, libre et multi-plateformes développé par le Dr. John Ousterhout. Selon ses propres termes :

"Tcl/Tk permet aux développeurs de logiciels de réaliser un travail dix fois plus rapidement qu'avec des outils basés sur C ou C++. C'est aussi un grand langage de synthèse permettant de faire travailler ensemble des applications existantes en les rendant plus graphiques et orientées Internet."

Karl Lehenbauer, Fondateur et Directeur Technique de NeoSoft, a participé au développement de Tcl/Tk dès le tout début. Avec Mark Diehkans, ils sont les auteurs de Tcl étendu (Extended Tcl), connu également sous le nom de TclX ou NeoSoft Tcl, qui constitue un ensemble puissant d'extensions au langage. Beaucoup des commandes courantes essentielles de Tcl proviennent de Tcl étendu, et furent introduites dans le langage par le Dr. Ousterhout.

NeoSoft Inc., 1770 St. James Place, Suite 500, Houston, TX 77056 USA

15.1 HEITML, extension, côté serveur de HTML et langage 4GL pour PostgreSQL

Heitml est un autre outil d'interfaçage de postgres avec le monde du world wide web. Pour plus de détails contacter

```
Helmut Emmelmann H.E.I. Informationssyteme GmbH
Wimpfenerstrasse 23 Tel. 49-621-795141
68259 Mannheim Germany Fax. 49-621-795161
```

- E-mail Mr.Helmut Emmelmann at *emmel@h-e-i.de*
- Heitml site web principal `<http://www.heitml.com>`
- Heitml site web secondaire `<http://www.h-e-i.deom>`

Heitml est à la fois une extension, côté serveur, de HTML et un langage de la quatrième génération (4GL). Avec lui, on peut écrire des applications web dans un style HTML et en utilisant de nouvelles balises de style HTML.

heitml (prononcé "H-ail"-TML) est une extension de HTML et un langage de quatrième génération complet permettant aux Applications utilisant le web d'interagir avec des données stockées dans des bases de données SQL, sans mériter l'écriture de scripts CGI complexes.

heitml étend HTML côté serveur, convertissant de manière dynamique les fichiers ".hei" au format HTML et ainsi, les rendant compatibles avec n'importe quel butineur web. Il englobe la syntaxe simple et familière de HTML et apporte un large assortiment de Balises et de Bibliothèques pré-développées pour prendre en charge les tâches qui auparavant nécessitaient CGI. De même que XML, heitml permet l'utilisation de balises définies par l'utilisateur. Avec heitml les marqueurs définis par l'utilisateur peuvent être traduits en HTML et envoyés à un butineur.

heitml est destiné à la fois aux concepteurs HTML et aux programmeurs professionnels. Les concepteurs HTML peuvent utiliser les Balises heitml pour fabriquer des pages web dynamiques, des accès aux bases de données SQL ou créer des applications web complètes. On peut créer des Compteurs, des bases de données d'inscriptions, des formulaires de recherche, des formulaires de courrier électronique ou des menus hiérarchisés en utilisant simplement des Balises de style HTML pré-développées que l'on trouve dans les nombreuses Bibliothèques de Composants.

Pour les programmeurs, heitml inclut un langage de quatrième génération complet en HTML

```
(e.g. <if>, <while>, et <let> Balises),
```

plus un évaluateur d'expression puissant pour les types de données entiers, réels, booléens chaîne de caractères et tuple. Les tuples ont une référence sémantique comme dans les langages orientés objets modernes et sont stockés sur un tas. Les variables heitml y compris toutes les structures de données complexes stockées sur le tas conservent leur valeur de page en page par l'utilisation du Mode Session. Il vous est possible de définir vos propres balises ou vos balises d'environnement et même de redéfinir les balises HTML.

Avec heitml il devient possible de

- - - développer des Sites Web de manière structurée et modulaire, tout en réduisant de façon drastique la surcharge due à la maintenance.

- - - développer des Sites Web intelligents et interactifs, dont le contenu s'adapte dynamiquement aux besoins de l'utilisateur.

- - - de visualiser le contenu de bases de données SQL sans autre programmation que l'utilisation de notre bibliothèque de Balises pré-définies "dba".

- - - de développer des applications de bases de données complexes et de Catalogue d'Achat en utilisant les Variables Session

heitml tourne sous Linux avec n'importe quel Serveur Web utilisant l'interface CGI, et il est particulièrement rapide (mis à part la surcharge due à CGI) avec le Serveur Web APACHE (version 1.1.1, 1.1.3, ou 1.2.4) en utilisant l'API apache. Actuellement MSQL (Version 1 et 2), PostgreSQL (Version 6), mysql, et les bases de données yard sont supportés. heitml tourne sous Linux, BSDi, Solaris et SunOS, de même que sous Windows NT avec CGI, ISAPI et ODBC et Windows 95.

heitml (sous linux) est libre pour la recherche, et pour une utilisation privée et non commerciale. Les Sites Web commerciaux doivent payer une licence. La version totalement opérationnelle de heitml est disponible pour une période d'essai et peut être télé-chargé librement. (Notez, cependant, que chaque page Web ".hei" que vous développerez affichera un message indiquant qu'il s'agit d'une version à usage non commerciale. Après enregistrement, vous recevrez une clé pour effacer ce message sans avoir à réinstaller le programme.)

Les nouvelles fonctionnalités de la version 1.2 sont

- - - Bibliothèque de Composants pour des Formulaires de Recherche dans un Base de Données, menus hiérarchisés d'ouverture et de fermeture, formulaires de courrier électronique

- - - Le Mode Session a été re-conçu et amélioré pour conserver toutes les variables (y compris le tas tout entier) au travers des différentes pages. Cela signifie que les données, de n'importe quelle taille sont conservées à l'intérieur d'une session. Ceci offre l'opportunité d'un champ plus large pour créer de nouvelles applications, e.g. stockage complet des résultats d'une interrogation dans la mémoire de la session.

- - - Les instructions d'installation, la documentation et les bibliothèques d'exemples ont été augmentées de manière significative, un cours d'auto-formation a été ajouté

- - - Des balises pour exécuter des commandes shell et pour envoyer des courriers électronique ont été ajoutées

- - - Le support du formatage d'impression et de la mise au point. heitml affiche votre code source en couleurs dans votre butineur et signale les erreurs de façon intuitive. En cas d'erreur à l'exécution, toutes les structures de données internes sont affichées dans le butineur en indiquant et préservant leurs positions dans le code source original.

- - - Diverses nouvelles variables serveur et fonctions intégrées ont été ajoutées

- - - En mode production, heitml collecte maintenant des informations complètes sur les erreurs survenues lors de l'accès, par des utilisateurs, à votre site.

Voici ce que dit la page d'accueil de heitml -

heitml (prononcer "H-ail"-TML) étend et augmente de façon significative les fonctionnalités de HTML grâce aux balises définissables et aux possibilités complètes de programmation. Ce qui permet de créer simplement des applications au contenu dynamique et orientées bases de données dans le monde HTML, sans CGI et sans scripts externes ou langages de programmation. Cela signifie que vous, en tant qu'auteur HTML, vous pouvez inclure des applications dans vos pages web, simplement, en utilisant quelques nouvelles balises sans CGI et sans programmation. D'un autre côté, comme utilisateur avancé ou comme programmeur vous pouvez créer et programmer de puissantes bibliothèques de balises. Cette approche rend heitml souhaitable à la fois pour les utilisateurs nouveaux de HTML et pour les programmeurs professionnels. heitml tourne sur le serveur web et génère dynamiquement du HTML, aussi heitml est compatible avec les normes internet et avec n'importe quel butineur web. Il permet un accès total aux bases de données tout en évitant à l'utilisateur toute la complexité inutile de CGI. heitml a été développé selon les plus récents critères en matière de construction de compilateurs et de systèmes transactionnels.

Les pages heitml sont développées exactement de la même façon que les pages HTML, à l'aide d'un éditeur de texte ou d'un éditeur HTML, et placées comme d'habitude sur le serveur web. Cependant, maintenant, les pages peuvent contenir des balises heitml dynamiques et des accès à des bibliothèques de balises. Vous pouvez utiliser ces balises pour accéder à une base de données, pour créer un contenu dynamique, pour envoyer des courriers électronique, et même pour créer de puissantes applications telles que les bases de données d'inscriptions ou de systèmes d'achats.

Les nouveaux venus à HTML et les programmeurs professionnels seront stupéfaits de la vitesse et de la facilités avec lesquelles il peuvent concevoir des applications passionnantes telle que notre Livre d'Or Interactif sans nécessiter la complexité et l'apprentissage difficiles des scripts CGI, simplement en utilisant les outils fournis dans notre bibliothèque dba.

heitml est accompagné d'un large éventail de bibliothèques de balises, pour créer des livres d'or, des applications de maintenance de bases de données, des formulaires puissants de courrier électronique ou de structuration de votre site web à l'aide de menus hiérarchiques. Ces outils sont prêts à être utilisés, il suffit simplement d'ajouter les balises correspondantes à votre site web.

En tant que programmeur expérimenté, vous pouvez pleinement utiliser l'architecture dynamique persistante de tuple d'heitml : heitml n'est pas simplement un langage de script à typage dynamique, avec évaluateur d'expression, procédures récursives et capacités de passage de paramètres étendues, mais il apporte aussi des possibilités de tuples persistants dynamiques pour conserver automatiquement des données de session de n'importe quelle taille.

15.2 Serveur Web America On-line AOL pour PostgreSQL

Le serveur web commercial gratuit, AOLserver version 2.3 supporte des connexions à la base de données PostgreSQL version 6.2.1 et ultérieure. Pour des informations complémentaires consultez

- AOL Web Server `<http://www.aolserver.com>`

15.3 Outil Système de suivi de Problème/Projet pour PostgreSQL

Se trouve à l'url :

- `<http://www.homeport.org/~shevett/pts/>`

15.4 Conversion de fichiers dbase dbf en fichiers PostgreSQL

Le programme dbf2msql fonctionne parfaitement avec mSQL et PostgreSQL. Vous pouvez le trouver à

- `<ftp://ftp.nerosworld.com/pub/SQL/dbf2sql/>`
- `<ftp://ftp.postgresql.org/pub/incoming/dbf2pg-3.0.tar.gz>`

Ce programme a été écrit par Maarten Boekhold, Faculty of Electrical Engineering TU Delft, NL Computer Architecture and Digital Technique section

- *M.Boekhold@et.tudelft.nl*

Vous pouvez également utiliser une méthode python pour lire des fichiers dbf et les charger dans une base de données postgres.

- Voir `<http://www.python.org>`

16 Outil de Conception/Implémentation pour SGBD Web pour PostgreSQL - EARPW

- `<http://www.oswego.edu/Earp>`
- `<ftp://ftp.oswego.edu>` in the directory 'pub/unix/earp'.

16.1 EARP, c'est quoi ?

Le "Programme de Réponse Facilement Ajustable" ("Easily Adjustable Response Program" - EARP) créé par David Dougherty. EARP est un outil de Conception/Implémentation pour Base De Données Web, réalisé au-dessus du système de gestion de base de données PostgresSQL. Ses fonctionnalités comprennent:

- Un système de conception visuel.
- Une interface d'envoi de courrier électronique. (pouvant prendre en charge le courrier entrant et sortant)
- Un Mécanisme de Sécurité amélioré.
- Un pilote cgi.

16.2 Implantation

L'implantation principale de EARP consiste en un binaire CGI qui s'exécute sous le démon http pour fournir l'accès au serveur de base de données. Tous les outils de conception sont intégrés dans le pilote, aucune conception ne se fait sur le web. Les outils eux-mêmes requièrent un navigateur graphique, la compatibilité des objets conçus avec ces outils est indépendante de l'implantation, elle ne dépend uniquement que des préférences de conception individuelles.

16.3 Ce dont vous avez besoin pour faire fonctionner EARP

EARP est sensé fonctionner sur une grande variété de plates-formes avec peu de modifications sinon pas du tout. Les plates-formes pour lesquelles le fonctionnement est connu sont les suivantes :

- Solaris 2.5
- Linux 1.2.13+
- GNU C++
- PostgreSQL (Version 1.01 / 1.02)
- netsite server
- NCSA httpd

- GNU C++
- PostgreSQL (Version 1.01 / 1.02)
- NCSA httpd
- Apache httpd

La version actuelle (1.3) de Earp a été conçue au-dessus de la version de libpq livrée avec PostgreSQL v1.01/1.02. Si vous utilisez une version plus récente de Postgres, vous devez vous attendre à ce que le programme nécessite quelques modifications pour fonctionner correctement. Dans la version de développement (Earp 2.0), la prise en charge de libpq est en cours de prise en compte en tant que module.

16.4 Comment fonctionne-t-il?

Une des fonctionnalités principales de EARP est d'utiliser une approche orientée objet pour produire des pages html qui s'interfacent à la base de données. La plupart des pages sont constituées de plusieurs objets. Chaque objet est produit par une sorte d'outil et reçoit un nom, les objets sont alors liés ensembles et appelés en séquence par l'outil de gestion de pages. Les objets sont également réutilisables à l'intérieur de plusieurs pages. Il existe des outils de base pour HTML, pour les Requêtes, pour la saisie dans des formulaires, le formatage variable des requêtes et des objets en entrée, ainsi que pour lier des objets pour en former de nouveaux. On trouve également des outils plus avancés tels que l'outil de courrier électronique et l'outil de création de requêtes en parallèle.

La gestion perfectionnée de la sécurité est une des autres fonctionnalités de EARP. L'accès aux différentes zones du système EARP peut être limité par une grande variété de moyens. Pour faciliter cette sécurité perfectionnée, EARP effectue des contrôles à chaque connexion au système, et détermine à quels "ids" et "groups" appartient l'agent qui se connecte. Les accès aux zones sont définis séparément, et la combinaison des deux permet de décider si l'accès à une certaine zone de Earp est autorisé. De plus , tout ce qui est requis pour réaliser les fonctionnalités de sécurité se trouve dans un serveur http qui effectue une authentification de l'utilisateur minimale (ou meilleure).

16.5 Où puis-je le trouver?

EARP est disponible via un ftp anonyme à :

- <ftp://ftp.oswego.edu> in the directory 'pub/unix/earp'.

17 PHP Pré-processeur Hypertexte - Langage de script côté serveur, avec HTML intégré, pour PostgreSQL

L'outil d'interfaçage WWW se trouve à :

- <http://www.php.net>
- <http://www.vex.net/php>

PHP possède également un compilateur appelé Zend qui améliore grandement les performances. En premier lieu, pendant le développement, les tests et la mise au point, vous écrivez votre application en utilisant le langage de script PHP. Une fois que le projet est prêt à être déployé, vous utilisez le compilateur Zend pour compiler le script PHP et créer un exécutable qui tournera très rapidement.

PHP 3 est l'ancien nom des Pages d'accueil professionelles et le nouveau nom est Pré-Processeur Hypertexte PHP

- Des sites miroirs existent dans de nombreux pays à l'adresse www.COUNTRYCODE.php.org
- <http://www.fe.de.php.net>
- <http://www.sk.php.net>
- <http://php.iquest.net/>

Pour les questions envoyer un courrier électronique à :

- *rasmus@lerdorf.on.ca*

PHP 3 est un langage de script côté serveur, avec langage HTML intégré. Il vous permet d'écrire des scripts simples directement dans vos fichiers .HTML à la manière JavaScript, sauf que, à la différence de JavaScript, PHP 3 n'est pas dépendant du butineur utilisé. JavaScript est un langage, côté client, avec HTML intégré alors que PHP 3 est un langage côté serveur. PHP 3 est similaire, dans son concept, au produit LiveWire Pro pour Netscape. Si vous en avez les moyens, vous utilisez le serveur du commerce Netscape et l'un des systèmes d'exploitation supportés, et vous jetterez probablement un oeil sur LiveWire Pro. Si vous préférez un logiciel libre, évoluant rapidement, qui est disponible avec la totalité de son code source, vous apprécierez probablement PHP 3.

17.1 Fonctionnalités principales

Support de CGI Standard, FastCGI et du module Apache. Comme programme CGI standard, PHP 3 peut être installé sur n'importe quelle machine Unix sur laquelle tourne n'importe quel serveur web Unix. Avec le support de la nouvelle norme FastCGI, PHP 3 peut trouver avantage des gains de vitesse apportés par ce mécanisme. Comme module Apache, PHP 3 devient une alternative extrêmement puissante et brillante à la programmation CGI.

- Enregistrement des accès Avec les possibilités d'enregistrement des accès de PHP 3, les utilisateurs peuvent entretenir leur propre compteur d'accès et l'enregistrer. Il n'utilise en aucune façon les fichiers de d'enregistrement des connexions du système central, et il permet un suivi des accès en temps réel. Le Script du Visualisateur de Connexions fournit un résumé rapide des accès à un ensemble de pages possédées par un utilisateur individuel. De plus, le paquetage peut être configuré pour générer un pied de page, sur chaque page, qui montre l'information sur les accès. Regardez au bas de cette page pour en avoir un exemple.

- Contrôle d'accès Un écran de configuration intégré, basé sur le web, permet la configuration du contrôle des accès. Il est possible de créer des règles d'accès pour toutes ou quelques unes des pages web possédées par une certaine personne qui met diverses restrictions sur qui peut voir ces pages et comment elles seront vues. Les pages peuvent être protégées par un mot de passe, complètement interdites, à connexion désactivée et l'accès basé sur le domaine du client, le butineur, l'adresse de courrier électronique ou même le document auquel on se réfère.
- Support de Postgres Postgres est un SGBDR (RDBMS) avancé libre. PHP 3 supporte les requêtes Postgres95 et PostgreSQL SQL incluses directement dans les fichiers .html.
- RFC-1867 Support du télé-chargement de fichier Le télé-chargement de fichier est une nouvelle fonctionnalité de Netscape 2.0. Il permet aux utilisateurs de télé-charger des fichiers vers le serveur web. PHP 3 réalise le décodage Mime réel pour faire ce travail et fournit aussi le cadre additionnel pour faire quelque chose d'utile avec le fichier télé-chargé une fois que celui-ci a été reçu.
- Contrôle d'authentification basé sur HTTP PHP 3 peut être utilisé pour créer des mécanismes d'authentification personnalisés basés sur HTTP pour le serveur web Apache.
- Variables, Tableaux, Tableaux associatifs PHP 3 supporte des variables typées, des tableaux et même des tableaux associatifs à la Perl. Ils peuvent être passés d'une page web à l'autre en utilisant les méthodes GET ou POST.
- Conditions, Boucles Tant que (While) PHP 3 possède un langage de script aux fonctionnalités complètes de style C. Vous disposez des instructions de tests conditionnels if/then/elseif/else/endif ainsi que des boucles while et des instructions switch/case pour contrôler l'ordre logique et la manière dont les pages html doivent être affichées.
- Expressions Régulières étendues Les expressions régulières sont très utilisées pour le filtrage, le remplacement de séquences et les manipulations générales de chaînes de caractères. PHP 3 supporte toutes les opérations communes sur les expressions régulières.
- Contrôle de l'En-tête HTTP brute La possibilité d'envoyer à partir de pages web des en-têtes HTTP brutes personnalisées en fonction d'une condition est essentielle pour créer un site web de haut niveau. Un usage fréquent est l'envoi d'un emplacement: en-tête URL pour Rediriger le client appelant vers une autre URL. Il peut aussi être utilisé pour stopper le stockage ou manipuler la dernière mise à jour de l'en-tête de pages.
- Création d'images KIF à la volée PHP 3 prend en compte la bibliothèque d'image GD de Thomas Boutell ce qui offre la possibilité de générer des images GIF à la volée.
- Prise en charge du mode sécurité ("Safe Mode") ISP PHP 3 prend en charge un "Mode de Sécurité" exceptionnel qui permet d'avoir de multiples utilisateurs exécutant des scripts PHP en toute sécurité sur le même serveur.
- C'est Libre! Pour finir, et c'est un point essentiel. Le paquetage est entièrement libre. Il est sous les conditions de la licence GPL qui vous permet d'utiliser ce logiciel pour n'importe quel objectif, commercial ou autre. Reportez-vous au document de la Licence Publique GNU pour des renseignements détaillés.

17.2 Crédits

* De grandes parties de ce code ont été développées à et pour l'université de Toronto. De grands Mercis à Lee Oattes du Département de Développement des Réseaux à l'université pour ses critiques constructives permanentes.

* Le code de prise en charge de Postgres95 a été écrit par Adam Sussman

  – *asussman@vidya.com*

* d'autres, innombrables, ont participé aux tests et à la mise au point du paquetage.

17.3 PHP 3 - Bref Historique

PHP a commencé sa vie comme simple petite enveloppe cgi écrite en Perl. Son utilisation en dehors d'un usage privé n'avait jamais été envisagée. Le nom de ce premier paquetage était "Personal Home Page Tools" (Outils pour Page d'Acceuil Personnelle"), qui devint plus tard "Kit de Construction d'une Page d'Acceuil".

Un outil pour inclure facilement des requêtes SQL dans des pages web a été écrit. C'était fondamentalement un autre petit programme enveloppe CGI qui analysait les requêtes SQL et facilitait la création de formulaires et de tables basés sur ces requêtes. Cet outil fut appelé FI "Interpréteur de Formulaire" (Form Interpreter).

La version 2.0 PHP/FI est une réécriture complète de ces deux paquetages combinés pour former un simple programme. il a maintenant évolué au point de devenir un simple langage de programmation intégré dans les fichiers HTML. L'acronyme original, PHP, lui est resté. Il n'est plus réellement approprié. PHP/FI est aujourd'hui plus utilisé pour mettre en place des sites web entiers que pour de petites pages d'accueil personnelles. Quel que soit son nom, il élimine le besoin de recourir à de nombreux petits programmes cgi Perl en permettant d'insérer de simples scripts directement dans vos fichiers HTML. Ceci améliore les performances globales de vos pages web puisque la surcharge due au lancement de processus Perl successifs a été éliminée. La gestion de larges sites web a également été facilitée en plaçant tous les composants d'une page web dans un fichier html unique. En incluant le support de différentes bases de données, il devient trivial de développer des pages web mettant en oeuvre des bases de données. Nombreux sont ceux qui trouvent que cette intégration est plus facile à gérer que d'essayer de créer des fichiers HTML et CGI séparés.

Tout au long de cette documentation, toute référence à PHP, FI ou PHP/FI traite de la même chose. La différence entre PHP et FI est seulement conceptuelle. Les deux sont construits à partir de la même distribution du code source. Maintenant, PHP/FI a été renommé PHP 3.

17.4 Bon, que puis-je faire avec PHP/FI?

La première chose que vous allez noter, si vous faites tourner une page par l'intermédiaire de PHP/FI, c'est qu'il ajoute un pied de page vous informant du nombre d'accès à votre page (si vous incluez le support des connexions d'accès dans votre binaire). C'est seulement un tout petit exemple de ce que PHP/FI peut faire pour vous. Il joue également le rôle très important d'interpréteur de formulaire cgi, d'où la partie FI de son nom. Par exemple si vous créez un formulaire dans l'une de vos pages web, vous avez besoin de quelque chose pour traiter l'information contenue dans ce formulaire. Même si vous désirez simplement passer l'information à une autre page web, vous aurez besoin d'un programme cgi pour le faire. PHP/FI facilite grandement le processus de prise en charge des données du formulaire pour en faire quelque chose.

17.5 Un simple exemple

Supposons que vous ayez le formulaire :

```
<FORM ACTION="/cgi-bin/php.cgi/~userid/display.html" METHOD=POST>
<INPUT TYPE="text" name="name">
<INPUT TYPE="text" name="age">
<INPUT TYPE="submit">
<FORM>
```

Votre fichier display.html doit contenir quelque chose du genre :

```
< ?echo "Hi $ name, vous avez $ age ans!<p>" >
```

C'est aussi simple que cela! PHP/FI crée automatiquement une variable pour chaque champ de saisie de votre formulaire. Vous pouvez alors utiliser ces variables dans le fichier ACTION URL.

L'étape suivante, après avoir vu comment utiliser ces variables, est de commencer à jouer avec quelques balises de gestion du déroulement logique de vos pages. Par exemple, si vous voulez afficher différents messages dépendants de des informations entrées par l'utilisateur, vous pouvez utiliser la logique si/alors (if/else). Dans notre exemple ci-dessus, on peut afficher différentes choses dépendant de l'âge entré par l'utilisateur en modifiant notre fichier display.html en :

```
<?
    if($age>50);
        echo "Hi $name, vous êtes un fossile!<p>";
    elseif($age>30);
        echo "Hi $name, vous êtes très vieux!<p>";
    else;
        echo "Hi $name.";
    endif;
>
```

PHP/FI fournit un langage de script très puissant qui peut faire beaucoup plus que ce qui est exposé dans l'exemple ci-dessus. Regardez la section sur le langage de Script PHP/FI pour des informations complémentaires.

Vous pouvez également utiliser PHP/FI pour configurer qui a le droit d'accéder à vos pages. Ceci est effectué en utilisant l'écran de configuration inclus. Par son intermédiaire , vous pourrez, par exemple, spécifier que seulement les gens appartenant à un certain domaine seront autorisés à voir vos pages, vous pouvez également créer une règle protégeant certaines pages par un mot de passe. Voir la section Contrôle d'Accès pour plus de détails.

PHP/FI est également capable de recevoir des fichiers télé-chargés par n'importe quel butineur conforme à la norme RFC-1867. Cette fonctionnalité permet aux gens de télé-charger également des fichiers de texte ou binaire. Avec le contrôle de PHP/FI et les fonctions logiques, vous disposez d'un contrôle total sur qui est autorisé à télé-charger un fichier et sur ce que l'on peut faire de ce fichier une fois télé-chargé. Voir la section Télé-chargement de fichier pour de plus amples détails.

PHP/FI supporte le paquetage Postgres95. Il supporte les requêtes SQL incluses dans vos fichiers .HTML. Consultez la section Support Postgres95 pour plus d'informations.

PHP/FI supporte également le paquetage du SGBD mysql. Il supporte les requêtes SQL incluses dans vos fichiers .HTML. Consultez la section Support mysql pour plus d'informations.

17.6 Redirection CGI

17.6.1 Notes Apache 1.0.x

Une bonne manière de faire tourner PHP/FI est d'utiliser un module de redirection cgi avec le serveur Apache. Veuillez noter que vous n'avez pas à vous soucier des modules de redirection si vous utilisez la version module Apache de PHP/FI. Deux de ces modules de redirection sont disponibles. L'un est développé par Dave Andersen

– *angio@aros.net*

et est disponible à

- *ftp://ftp.aros.net/pub/util/apache/mod_cgi_redirect.c*

et l'autre est inclus avec Apache et est appelé mod_actions.c. Ces deux modules sont extrêmement similaires. Ils diffèrent très légèrement dans leur utilisation. Les deux ont été testés et fonctionnent avec PHP/FI.

Regardez dans la documentation Apache la manière d'ajouter un module. En général vous ajoutez le nom de module dans un fichier appelé Configuration. La ligne à ajouter si vous voulez utiliser le module mod_actions est :

```
Module action_module mod_actions.o
```

Si vous envisagez d'utiliser le module mod_cgi_redirect.c ajoutez cette ligne :

```
Module cgi_redirect_module mod_cgi_redirect.o
```

Puis compilez votre httpd et installez-le. Pour configurer la redirection cgi il vous faudra soit créer un nouveau type mime dans votre fichier mime.types soit utiliser la commande AddType dans votre fichier srm.conf pour ajouter le type mime. Le type mime à ajouter doit être quelque chose comme :

```
application/x-httpd-php phtml
```

Si vous vous apprêtez à utiliser le module mod_actions.c il vous faudra ajouter la ligne suivante dans votre fichier srm.conf:

```
Action application/x-httpd-php /cgi-bin/php.cgi
```

Si vous vous apprêtez à utiliser mod_cgi_redirect.c vous devrez ajouter cette ligne à srm.conf :

```
CgiRedirect application/x-httpd-php /cgi-bin/php.cgi
```

N'essayez pas d'utiliser en même temps mod_actions.c et mod_cgi_redirect.c .

Une fois que vous avez l'un de ces modules de redirection cgi installé et configuré correctement, vous pouvez spécifier que vous voulez qu'un fichier soit filtré par php/fi en mettant simplement l'extension .phtml au fichier. De plus, si vous ajoutez index.phtml à votre ligne de configuration DirectoryIndex dans votre fichier srm.conf alors la page de plus haut niveau d'un répertoire sera automatiquement filtrée par php si vous appelez votre fichier index index.phtml.

17.6.2 HTTPD Netscape

Vous pouvez rediriger automatiquement les requêtes pour des fichiers ayant une extension donnée de façon à ce qu'ils soient pris en compte par PHP/FI en utilisant le module de Redirection du Serveur CGI Netscape. Ce module est disponible dans le Fichier Archives de la Page d'Accueil de PHP/FI. Le fichier LISEZ_MOI (README) dans le paquetage explique clairement comment le configurer pour l'utiliser avec PHP/FI.

17.6.3 HTTPD NCSA

Actuellement NCSA ne supporte pas les modules, donc, pour effectuer une redirection cgi avec ce serveur, il vous faudra modifier le code source de votre serveur. Une rustine pour faire cela avec NCSA 1.5 est disponible dans le fichier archives de PHP/FI.

17.7 Lancer PHP/FI à partir de la ligne de commande

Si vous fabriquez la version CGI de PHP/FI, vous pouvez l'utiliser simplement à partir de la ligne de commande, en tapant: php.cgi nom_de_fichier où nom_de_fichier est le fichier que vous voulez filtrer.

Vous pouvez également créer des scripts PHP/FI autonomes en faisant ressembler la première ligne de votre script à quelque chose comme :

```
#!/usr/local/bin/php.cgi -q
```

L'option "-q" supprime l'impression des en-têtes HTTP. Vous pouvez vous passer de cette option si vous le désirez.

18 Interface Python pour PostgreSQL

Python est un langage de script, interprété et orienté objet. Il est simple à utiliser (syntaxe légère, instructions simples et sans détours). Il possède de nombreuses extensions pour bâtir des interfaces utilisateurs (GUI), pour s'interfacer au WWW, etc. Un butineur de web intelligent (à la HotJava), est actuellement en cours de développement (novembre 1995), et cela devrait ouvrir de nombreuses portes aux programmeurs. Le copyrightde Python est détenu par Stichting S Mathematisch Centrum, Amsterdam, Pays-Bas, et est librement distribuable Il supporte le chargement dynamique d'objets, de classes, de modules et d'exceptions. L'ajout d'interfaces à de nouvelles bibliothèques système, à l'aide de code C est aisé, rendant Python facile à utiliser en cas de personnalisation. Python est un langage de script de très haut niveau possédant une interface X. Le paquetage Python est distribué dans les cdroms Linux, et il comprend la plupart des modules Python standards, ainsi que des modules d'interfaçage à l'ensemble de widgets Tix pour Tk.

PyGresSQL est un module Python qui s'interface à une base de données PostgreSQL. Il intègre la bibliothèque de requêtes de PostgreSQL pour permettre l'utilisation aisée des possibilités de PostgreSQL à partir d'un script Python. PyGreSQL a été écrit par D'Arcy J.M. Cain et Pascal Andre.

- Nouveau site de PyGreSQL `<http://www.druid.net/pygresql/>`
- Maintenu par D'Arcy à `<http://www.druid.net/~darcy/>`
- L'ancien site est à `<ftp://ftp.via.ecp.fr/pub/python/contrib/Database/PyGres95.README>`
- D'Arcy J.M. Cain *darcy@druid.net*
- Pascal Andre *andre@chimay.via.ecp.fr*
- Pascal Andre *andre@via.ecp.fr*

18.1 Où le trouver PyGres?

Les sites d'accueil des différents paquetages sont :

- Python `<ftp://ftp.python.org:/pub/www.python.org/1.5/python1.5b2.tar.gz>`
- PyGreSQL `<ftp://ftp.druid.net/pub/distrib/PyGreSQL-2.1.tgz>`
- Old site `<ftp://ftp.via.ecp.fr/pub/python/contrib/Database/PyGres95-1.0b.tar.gz>`

Vous devriez cependant essayer de trouver un site miroir proche de votre propre site. Consultez les sources d'information pour trouver ces sites. PyGres95 devrait se trouver dans les répertoires contrib des sites Python et PostgresSQL.

18.2 Information et support

Si vous avez besoin d'informations concernant ces paquetages veuillez consulter leurs sites web:

- Python : `<http://www.python.org/>`

- PostgreSQL : `<http://epoch.cs.berkeley.edu:8000/postgres95/index.html>`
- PyGreSQL `<ftp://ftp.druid.net/pub/distrib/PyGreSQL-2.1.tgz>`
- Ancien site PyGreSQL : `<http://www.via.ecp.fr/via/products/pygres.html>`

Pour un support :

- Liste de diffusion de PyGreSQL. Vous pouvez vous abonner en envoyant un courrier électronique à *majordomo@vex.net* en mettant la ligne "subscribe pygresql name@domain" dans le corps du message et en remplaçant "name@domain" par votre propre adresse électronique.
- Forum de discussion pour Python : newsgroup comp.lang.python
- PyGreSQL : contacter Andre à *andre@via.ecp.fr* pour les comptes-rendus d'erreurs, idées, remarques

19 Passerelle entre PostgreSQL et WWW - WDB-P95

WDB-P95 - Une interface Web interface aux bases de données PostgresSQL a été créée par J. Douglas Dunlop. C'est à :

- La nouvelle WDB de J Rowe se trouve à `<http://www.lava.net/beowulf/programming/wdb>`
- La nouvelle version de WWW-WDB est à `<http://www.eol.ists.ca/~dunlop/wdb-p95/>`
- Pour les questions ou pour s'abonner à une liste de diffusion contacter : *dunlop@eol.ists.ca*

C'est une version modifiée de wdb-1.3a2 qui fournit une passerelle WWW à PostgresSQL. Cette version nécessite également un Butineur capable de prendre en compte les Tables HTML pour avoir une sortie tabulée. Ceci n'est pas requis pour le wdb original et on peut facilement y revenir.

Vous pouvez essayer ma Bande CASI et ma Requête d'Image. Vous pouvez jeter un coup d'oeil au Fichier de Définition de Formulaire (FDF) qui est utilisé pour créer la Bande CASI et également la Requête d'Image, qui comprends une jointure (JOIN) de 2 tables.

Cette version contient tous les fichiers nécessaires pour installer et faire tourner WDB-P95 en tant qu'interface à vos bases de données PostgresSQL. Le portage de ce système pour une autre base de données devrait être relativement facile - à condition qu'elle supporte le SQL standard et possède une interface Perl.

19.1 Le serveur PostgresSQL, pgperl, et httpd doivent-ils résider sur le même hôte?

Non - Le serveur PostgresSQL n'a pas à être sur le même hôte. Comme WDB-P95 est appelé par le démon http, ils doivent résider sur le même hôte. - Et comme WDB-P95 a été écrit pour utiliser Pg.pm - pgperl doit être sur le même hôte également. Pgperl a été écrit en utilisant la bibliothèque libpq, donc, il sera capable d'accéder à n'importe quel serveur Postgres95 n'importe où sur le réseau, juste comme n'importe quel autre client PostgresSQL. Comme illustré ci-dessous

{WWW Client (Netscape)} => {HTTP Server (NCSA's http) + WDB-P95 + pgperl + libpq}=> {PostgresSQL server}

Les parenthèses () représentent les machines.

Chaque machine peut être d'un type différent : NT, SUN, HP, ... mais il faut que vous ayez la bibliothèque d'interface libpq pour le type de machine sur lequel vous envisagez d'utiliser WDB-P95, puisqu'il vous faut compiler pgperl. (Le système a été conçu pour utiliser les tables HTML donc un client WWW récent est meilleur)

20 Interfaces au langage "C", "C++", ESQL/C et Opérateurs sur les bits pour PostgreSQL

20.1 Interface "C"

Elle est incluse dans la distribution et s'appelle 'libpq'. Elle est similaire aux bibliothèques OCI Oracle, DB-lib Sybase, ou CLI Informix.

20.2 "C++" interface

Elle est incluse dans la distribution et est nommée 'libpq++'.

20.3 ESQL/C

C'est un pré-compilateur C intégré pour PostgreSQL ESQL/C comme Pro*C d'Oracle, et ESQL/C d'Informix. ESQL/C pour PostgreSQL est une interface de programmation d'application SQL (API) qui permet au programmeur C de créer des applications personnalisées avec des possibilités de gestion de base de données. ESQL/C pour PostgreSQL vous permet d'utiliser un langage de troisième génération avec lequel vous êtes familiarisé tout en profitant des avantages d'un Langage de Requêtes Structuré (SQL).

ESQL/C est composé des éléments logiciels suivants:

- Les bibliothèques ESQL/C de fonctions C fournissent les accès au serveur de base de données.
- Les fichiers d'en-tête ESQL/C apportent les définitions des structures de données, les constantes et les macros utiles dans un programme ESQL/C.
- Le pré-processeur ESQL/C, qui est un pré-processeur de code source, transforme un fichier C contenant des instructions SQL en fichier exécutable.

Il se trouve à :

- ESQL/C pour PostgreSQL est déjà inclus dans la distribution.
- Site principal `<ftp://ftp.lysator.liu.se/pub/linus>`
- Email : *linus@epact.se*

20.4 Opérateurs sur les bits pour PostgreSQL

Les opérateurs sur les bits ont été écrits par Nicolas Moldavsky

- *nico@overnet.com.ar*

Ce sont des fonctions "C" qui implantent les opérateurs bit-par-bit (AND, OR, XOR, bit complement) dans pgsql. Si quelqu'un désire les utiliser, il peut les récupérer par ftp anonyme de

- `<ftp://ftp.overnet.com.ar/pub/utils/linux/bitpgsql.tgz>`

De plus, il y a un "Makefile" pour Linux.

21 Les Codes Kanji Japonais pour PostgreSQL

Sont très utiles pour les Japonais. On les trouvera au site suivant :

- `<ftp://ftp.sra.co.jp/pub/cmd/postgres/>`

22 Portage de PostgreSQL pour Windows 95/Windows NT

Un portage pour Windows 95/Windows NT est en cours de réalisation. Ce portage de fait en utilisant gcc, gmake pour Win NT/95. Le programme gnu-win32 est utilisé pour compiler le code source sous win32. GNU gcc est disponible pour win32. Consultez ce site -

- `<http://www.cygnus.com/misc/gnu-win32>`

Récupérez-y le fichier cdk.exe (fichier auto-extractible pour gnu-win32)

Le portage peut également être réalisé en utilisant l'outil "Emulateur Unix sous NT" (Unix-Emulator on NT) suivant de

- `<http://www.softway.com>`

23 Mailing Lists

23.1 Obtenir un compte e-mail Gratuit

Regardez les offres gratuites - obtenez des comptes e-mail gratuit chez

- Yahoo `<http://www.yahoo.com>` cliquer sur e-mail
- Lycos `<http://www.lycos.com>` cliquer sur nouveaux comptes e-mail
- hotmail `<http://www.hotmail.com>` cliquer sur nouveaux comptes e-mail

Inscrivez vous à la liste de diffusion PostgreSQL et Yahoo dispose de la fonctionnalité complémentaire de création d'un classeur séparé pour les courriers électroniques PostgreSQL e-mails, ainsi votre e-mail habituel ne sera pas pollué. Sélectionnez le menu Email- > Options- > Filters et prenez un classeur séparé pour e-mail. Avec ce compte e-mail vous pouvez accéder au courrier de n'importe où dans le monde à partir du moment où vous avez accès à une page web.

Si vous disposez d'un autre e-mail, vous pouvez utiliser les "Filtres de Courrier" pour recevoir automatiquement les courriers PostgreSQL dans un dossier séparé. De cette façon, vous éviterez de voir votre courrier pollué.

23.2 Liste de Diffusion Anglaise

Regardez les Titres des Listes de Diffusion sur la page web principale à :

- `<http://www.postgresql.org/>`
- Envoyez vos questions par courrier électronique à: *pgsql-questions@postgresql.org*
- Développeurs *pgsql-hackers@postgresql.org*
- Questions spécifiques à un portage *pgsql-ports@postgresql.org*
- Questions concernant la documentation *pgsql-docs@postgresql.org*

Vous recevrez une réponse, par courrier électronique, dans la journée qui suit!!

Vous pouvez également vous inscrire aux listes de diffusion. Pour vous inscrire ou vous radier d'une liste, envoyez un courrier à

- *pgsql-questions-request@postgresql.org*
- *pgsql-hackers-request@postgresql.org*
- *pgsql-ports-request@postgresql.org*
- *pgsql-docs-request@postgresql.org*

Le corps du message doit uniquement contenir la simple ligne

subscribe

(ou)

unsubscribe

23.3 Archive des listes de diffusion

Il y a aussi des listes de diffusion qui sont archivées au format html à l'endroit suivant -

- Liste accessible par dates via MHonarc sur le WWW à `<http://www.postgresql.org/mhonarc/pgsql-questions>`
- `<ftp://ftp.postgresql.org>` dans le répertoire /pub/majordomo

23.4 Liste de Diffusion Espagnole

Il y a maintenant une liste "non officielle" pour postgreSQL en Espagnol. Consultez également la section 'Compte gratuit pour Organiser vos e-mails PostgreSQL e-mails' ci-dessus. Pour souscrire, l'utilisateur doit envoyer un message à :

- *majordomo@tlali.iztacala.unam.mx*

Le corps du message doit contenir la simple ligne :

inscripcion pgsql-ayuda

24 Livres et Documentations

24.1 Guides utilisateur et Manuels

On trouve dans la distribution PostgreSQL aux formats postscript, HTML et pages de manuel unix les documents suivants. Si vous avez accès à l'internet, vous pouvez trouver les documents indiqués ci-dessous à `<http://www.postgresql.org/docs>`

- Le 'Guide Utilisateur' pour PostgreSQL,
- Le 'Guide de Réalisation' détaillant la constitution interne de PostgreSQL.
- Les manuels "en ligne".
- Le manuels en ligne au format HTML.
- Également les manuels au format Postscript pour faire des éditions papier.

24.2 Documentation en ligne

- Liste et description des types de données et des opérateurs par défaut

  ```
  Fait partie des commandes PSQL de la version 6.5.2.
  ```
- Liste des mots-clés SQL supportés

  ```
  Il y a un script dans le répertoire /tools qui fait cela
  ```
- Liste des instructions supportées -

  ```
  Utilisez la commande psql \h
  ```

- Concepts de base des bases de données relationnelles sous PostgreSQL (en implémentation) et des tonnes d'exemples en ligne (requêtes) -

```
Consultez les tests de régression dans src/test. Là, vous y trouver les
répertoires regress/sql et suite/*.sql.
```

- Didacticiel pour PostgreSQL.

```
Les scripts du didacticiel SQL sont dans le répertoire src/tutorial
```

Voir également "Didacticiel SQL pour les débutants" dans l'appendice B de ce document B ()

24.3 Documents de Référence: Ouvrages de référence utiles:

- "Understanding the New SQL: A Complete Guide" (Comprendre le Nouveau SQL: Un Guide Complet)- par Jim Melton et Alan R.Simon

```
Morgan Kaufman Publisher. C'est un des meilleurs livres sur SQL. Il
concerne SQL92.
```

- "A Guide to THE SQL STANDARD" (Un Guide du STANDARD SQL) - par C.J.Date

```
Addison-Wesley Publishing company. C'est également un bon livre. Très
populaire en ce qui concerne SQL.
```

- SQL - The Standard Handbook, (Le Manuel Standard) November 1992

```
Stephen Cannan and Gerard Otten
McGraw-Hill Book Company Europe , Berkshire, SL6 2QL, England
```

- SQL Instant Reference, (Référence Immédiate SQL) 1993

```
Martin Gruber, Technical Editor: Joe Celko
SYBEX Inc.  2021 Challenger Drive Alameda, CA 94501
```

- C.J.Date, "An introduction to Database Systems" (Une introduction aux Systèmes de Bases de Données) (6th Edition), Addison-Wesley, 1995, ISBN 0-201-82458-2

```
Ce livre est la Bible des Systèmes de Gestion de Bases de Données.
Le livre détaille la normalisation, SQL, la récupération, la concurrence,
la sécurité, l'intégrité, les extensions au modèle relationnel original,
des sujets actuels tels que  les systèmes client/serveur et le(s) modèle(s)
Orienté(s) Objet. De nombreuses références sont données pour des lectures
complémentaires. Recommandé pour la plupart des utilisateurs.
```

- Stefan Stanczyk, "Theory and Practice of Relational Databases", (Théorie et pratique des bases de Données Relationnelles) UCL Press Ltd, 1990, ISBN 1-857-28232-9

```
Ce livre détaille la théorie des bases de données relationnelles, l'algèbre
relationnelle, le calcul et la normalisation. Mais il ne couvre pas les sujets
du mode réel et les exemples sont un peu simplistes. Recommandé pour la plupart
des utilisateurs.
```

- "The Practical SQL Handbook" (Le Manuel Pratique de SQL) Third Edition, Addison Wesley Developers Press ISBN 0-201-44787-8

```
Recommandé pour la plupart des utilisateurs.
```

- Michael Stonebraker, "Readings in Database Systems" (lectures sur les Systèmes de Bases de Données), Morgan Kaufmann, 1988, ISBN 0-934613-65-6

```
Ce livre est un recueil de papiers, sur les bases de données, qui ont été
publiés sur plusieurs années. Il n'est pas pour l'utilisateur occasionnel
mais, c'est réellement une référence pour les étudiants en classes supérieures
(troisième cycle) ou pour les développeurs de systèmes de bases de données.
```

- C.J.Date, "Relational Database - Selected Readings" (Bases de Données Relationnelles - Morceaux choisis), Addison-Wesley, 1986, ISBN 0-201-14196-5

```
Ce livre est un recueil de papiers, sur les bases de données, qui ont été
publiés sur plusieurs années. Il n'est pas pour l'utilisateur occasionnel
mais c'est réellement une référence pour les étudiants en classes
supérieures (troisième cycle) ou pour les développeurs de systèmes de
bases de données.
```

- Nick Ryan and Dan Smith, "Database Systems Engineering", (Ingénierie de Systèmes de Bases de Données), International Thomson Computer Press, 1995, ISBN 1-85032-115-9

```
Ce livre traite des détails des méthodes d'accès, et des technique de stockage.
```

- Bipin C. Desai, "An introduction to Database Systems", (Une introduction aux systèmes de bases de données) West Publishing Co., 1990, ISBN 0-314-66771-7

```
Il n'est pas pour l'utilisateur occasionnel mais c'est réellement une
référence pour les étudiants en classes supérieures (troisième cycle)
ou pour les développeurs de systèmes de bases de données.
```

- Joe Celko "INSTANT SQL Programming" (Programmation SQL IMMEDIATE)

```
Wrox Press Ltd.
Unit 16, 20 James Road, Tyseley
Birmingham, B11 2BA, England
1995
```

- Michael Gorman "Database Management Systems: Understanding and Applying Database" (Systèmes de Gestion de Bases de Données : comprendre et appliquer les bases de données)

```
Technology
QED and John Wiley
1991
```

- Michael Gorman "Enterprise Database for a Client/Server Environment" (Base de Données d'Entreprise pour un Environnement Client/Serveur) QED and John Wiley

```
Présente les exigences de construction d'applications de base de données
client/server par l'intermédiaire de méta-modèles d'entrepôt et l'utilisation
de la norme SQL ANSI 1993
```

Des centaines d'autres titres concernant SQL sont disponibles! Vérifiez-le dans une librairie.

24.4 Documents de Spécifications ANSI/ISO SQL - SQL 1992, SQL 1998

Les documents de Spécifications ANSI/ISO SQL sont situés à l'endroit indiqué ci-dessous.

- `<http://www.naiua.org/std-orgs.html>`
- `<http://www.ansi.org/docs>` et cliquez sur le fichier cat_c.html puis rechercher "Database SQL"
- Standard SQL92 `<http://www.jcc.com>` et cliquez sur le fichier sql_stnd.html
- Spécifications ANSI/ISO SQL `<http://www.contrib.andrew.cmu.edu/~shadow/sql.html>` Vous trouverez la Référence SQL ici.

24.5 Syntaxe de l'ANSI/ISO SQL 1992

Voir l'Appendice A de ce document A ()

24.6 Syntaxe de l'ANSI/ISO SQL 1998

Les spécifications de SQL 1998 (SQL 3) sont encore en cours de développement. Voir le paragraphe 'Electronic Access to the SQL3 Working Draft' de ce document document à 24.9 ()

24.7 Didacticiel SQL pour les débutants

Voir l'Appendice B de ce document B ()

24.8 Extension Temporelle à SQL92

- Document pour l'Extension Temporelle à SQL-92 `<ftp://FTP.cs.arizona.edu/tsql/tsql2/>`
- Spécification SQL-3 Temporelle `<ftp://FTP.cs.arizona.edu/tsql/tsql2/sql3/>`

Ce répertoire contient les spécifications pour une extension temporelle au langage SQL-92 standard. Ce nouveau langage est appelé TSQL2.

Les spécifications du langage présentées ici correspondent à la version finale du langage.

La correspondance doit être adressée au bureau du Comité de Conception du Langage TSQL2, Richard T.Snodgrass, Department of Computer Science, University of Arizona, Tucson, AZ 85721,

- *rts@cs.arizona.edu*

Les affiliations et les adresses électroniques des membres du Comité de Conception du langage TSQL2 peuvent être trouvées, dans un chapitre séparé, à la fin des spécifications du langage.

Le contenu du répertoire est le suivant :

spec.dvi,.ps Spécifications du langage TSQL2, publié en Septembre, 1994

bookspec.ps Spécifications du langage TSQL2, telles qu'elles apparaissent dans le livre sur TSQL2, publié en Septembre, 1995 (voir ci-dessous).

sql3 proposition de modification soumise au comités ANSI et ISO SQL3.

Un ensemble de commentaires, associés aux spécifications du langage, parle des décisions de conception, fournit des exemples, et traite de la façon dont le langage peut être implanté. Ces commentaires sont, à l'origine, des propositions faites au Comité de Conception du Langage TSQL2. Ils poursuivent actuellement un objectif différent: fournir des exemples de syntaxe, expliquer les nombreuses décisions prises durant la conception du langage, et comparer TSQL2 aux nombreuses autres propositions de langage faites au cours des quinze dernières années. Il faut insister sur le fait que ces commentaires ne font pas partie intégrante des spécifications du langage TSQL2 en lui-même, mais plutôt qu'elles le complètent et constituent un apport. Le mot de la fin est donné par les spécifications de TSQL2 proprement dit.

Les commentaires, ainsi que les spécifications du langage, plusieurs index, et d'autres éléments de support ont été publiés dans un livre :

Snodgrass, R.T., éditeur, The TSQL2 Temporal Query Language, (Le Langage de Requêtes Temporel TSQL2) Kluwer Academic Publishers, 1995, 674+xxiv pages.

Les commentaires d'évaluation sont donnés sous forme abrégée dans le livre; La totalité des commentaires est fournie dans le fichier eval.ps situé dans ce répertoire

Le fichier tl2tsql2.pl est un programme prolog qui traduit la logique temporelle autorisée en TSQL2. Ce programme a été écrit par Michael Boehlen

- *boehlen@iesd.auc.dk*

On peut le contacter pour obtenir un papier qui décrit cette traduction. C'est une version tout à fait ancienne du programme. Les nouvelles versions sont disponibles à

- `<http://www.cs.auc.dk/general/DBS/tdb/TimeCenter/Software>`

(the TimeDB and Tiger systems).

24.9 Partie 0 - Acquisition des Documents ISO/ANSI SQL

Ce document vous explique comment acquérir (légalement) une copie de la norme SQL-92 et comment acquérir une copie du Brouillon de Travail du SQL3 "actuel".

La norme ANSI est protégée par un copyright de l'ANSI, et la norme ISO par un copyright de l'ISO.

Il y a deux (2) normes SQL courantes, une publication ANSI et une publication ISO. Les deux normes sont identiques mot-pour-mot excepté pour quelques sujets triviaux tel que le titre du document, les en-tête de page, la phrase "International Standard" au lieu de "American Standard", et ainsi de suite.

Achat du Standard SQL-92

La norme ISO, ISO/IEC 9075:1992, Information Technology - Database Languages - SQL, est actuellement (Mars, 1993) disponible et en stock à l'ANSI :

```
American National Standards Institute
1430 Broadway
New York, NY 10018 (USA)
Phone (sales): +1.212.642.4900
```

au prix de US$ 230.00. La version ANSI, ANSI X3.135-1992, American National Standard for Information Systems - Database Language SQL, n'était pas disponible en stock au moment de la rédaction, mais il devrait l'être entre fin Mars et début Mai, 1993). Il devrait coûter US$ 225.00.

Si vous achetez l'un ou l'autre document à l'ANSI, il y aura un supplément de 7% pour le traitement (soit environ US$ 9.10). Les frais d'envoi à l'étranger accroîtront sans doute le coût. ANSI exige des sociétés un document de commande écrit pour chaque commande; d'un autre côté, vous pouvez envoyer un chèque tiré sur une banque US en dollars US, qu'ils encaisseront avant d'expédier votre commande. (Il y a une exception : si votre organisation est membre de l'ANSI, alors l'ANSI enverra les documents et une facture à votre compagnie.)

La norme ISO est aussi disponible en dehors des États-Unis à partir de représentations nationales locales (entités de standardisation du pays) qui sont membres soit de l'ISO (International Organization for Standardization) soit de l'IEC (International Electrotechnical Commission). Des copies de la liste des représentations nationales, avec leurs adresses, sont disponibles de l'ANSI ou d'autres entités nationales. Ils sont aussi disponibles à l'ISO :

```
International Organization for Standardization
Central Secretariat
1, rue de Varembi
CH-1211 Genève 20
Switzerland
```

Si vous préférez commander la norme d'une façon plus facile et plus rapide, il vous faudra payer pour ce privilège. Vous pouvez commander le ISO/IEC 9075:1992, Information Technology - Database Languages - SQL, à :

```
Global Engineering Documents
2805 McGaw Ave
Irvine, CA 92714 (USA)
USA
Téléphone (accessible de partout): +1.714.261.1455
Téléphone (uniquement aux USA): (800)854-7179
```

pour un prix de US$ 308.00. Je ne sais pas si le port est compris ou non, mais je parierais que l'envoi international (au moins) sera en supplément. Ils peuvent vous envoyer un document très rapidement et acceptent même les "plus grandes cartes de crédit". Global n'a pas encore la version ANSI pas plus qu'un prix ou qu'une date prévisible de disponibilité (bien que je l'attende dans les quelques semaines suivant la publication par l' ANSI et à un prix aux alentours de US$ 300.00).

Achat d'une copie du Document de Travail SQL3

Vous pouvez acheter une copie du document de travail SQL3 au secrétariat de l'ANSI X3, CBEMA (Computer and Business Equipment Manufacturers Association). Leur objectif est d'avoir la "plus récente" version du document de travail de SQL3 disponible et de la vendre pour environ US$ 60.00 à US$ 65.00. Vous pouvez contacter CBEMA à:

```
CBEMA, X3 Secretariat
Attn: Lynn Barra
1250 Eye St.
Suite 200
Washington, DC 20005 (USA)
```

Lynn Barra peut aussi être contactée par téléphone au +1.202.626.5738 pour une demande de copie, bien qu'un courrier soit probablement plus courtois.

Accès Électronique au Document de Travail SQL3

La plus récente version (à la date de rédaction) du document de travail SQL3 (à la fois ANSI et ISO, et de toutes ses parties) est disponible par "ftp anonyme" ou par "ftpmail" sur:

```
gatekeeper.dec.com
```

`dans`

```
/pub/standards/sql/
```

Dans ce répertoire , il y a de nombreux fichiers. Ce sont des fichiers PostScript et "texte ordinaire" (pas très bien formatés, mais lisibles sur un écran sans logiciel spécifique).

En général, vous pouvez trouver des fichiers avec des noms tels que :

```
sql-bindings-mar94.ps
sql-bindings-mar94.txt
sql-cli-mar94.ps
sql-cli-mar94.txt
sql-foundation-mar94.ps
sql-foundation-mar94.txt
sql-framework-mar94.ps
sql-framework-mar94.txt
sql-psm-mar94.ps
sql-psm-mar94.txt
```

A mesure que de nouvelles versions des documents sont produites le "mar94" changera pour indiquer la nouvelle date de publication (e.g., "aug94" est la date attendue de la prochaine publication suivant "mar94").

De plus, pour les lecteurs ne pouvant pas obtenir une liste de répertoire par FTP, nous avons ajouté un fichier ayant pour nom :

```
ls
```

dans le même répertoire. Ce fichier (surprise!) contient une liste du répertoire.

Récupérer les Fichiers Directement par Ftp

Voici un exemple sur la façon d'utiliser FTP. Spécifiquement, il montre comment se connecter à gatekeeper.dec.com, aller au répertoire où le document de base est gardé et à transférer ce document sur votre machine. Notez que votre machine doit pouvoir accéder à l'internet pour faire cela. Le nom de connexion est 'ftp' et le mot de passe est votre adresse e-mail (on appelle quelquefois cela 'ftp anonyme'). La commande 'type binaire' est utilisée pour s'assurer qu'aucuns bits ne seront supprimés du (des) fichier(s) reçu(s). La commande 'get' récupère un fichier à la fois. Les commentaires dans le script ci-dessous sont placés entre les signes $<$ ... $>$, $<$ comme ceci $>$.

```
% ftp gatekeeper.dec.com
Connected to gatekeeper.dec.com.
220- *** /etc/motd.ftp ***
     Gatekeeper.DEC.COM is an unsupported service of DEC Corporate Research.
     <...il en va ainsi pendant un certain temps...>
220 gatekeeper.dec.com FTP server (Version 5.83 Sat ... 1992) ready.
Name (gatekeeper.dec.com:<votre nom de connexion ici>): ftp  <anonyme fonctionne également>
331 Guest login ok, send ident as password.
Password: <entrez votre adresse e-mail ici>
230 Guest login ok, access restrictions apply.
Remote system type is UNIX.  <ou autre>
Using binary mode to transfer files.
ftp> cd pub/standards/sql
250 CWD command successful.
ftp> dir
200 PORT command successful.
150 Opening ASCII mode data connection for /bin/ls.
total 9529
-r--r--r--  1 root     system     357782 Feb 25 10:18 x3h2-93-081.ps
-r--r--r--  1 root     system     158782 Feb 25 10:19 x3h2-93-081.txt
-r--r--r--  1 root     system     195202 Feb 25 10:20 x3h2-93-082.ps
-r--r--r--  1 root     system      90900 Feb 25 10:20 x3h2-93-082.txt
-r--r--r--  1 root     system    5856284 Feb 25 09:55 x3h2-93-091.ps
-r--r--r--  1 root     system    3043687 Feb 25 09:57 x3h2-93-091.txt
226 Transfer complete.
ftp> type binary
200 Type set to I.
ftp> get x3h2-93-082.txt
200 PORT command successful.
150 Opening BINARY mode data connection for x3h2-93-082.txt (90900 bytes).
226 Transfer complete.
90900 bytes received in 0.53 seconds (166.11 Kbytes/s)
ftp> quit
% <le fichier est maintenant dans votre répertoire sous le nom x3h2-93-082.txt>
```

Récupération de Fichiers Sans Faire un ftp Direct

Digital Equipment Corporation, comme plusieurs autres compagnies, fournit un service de ftp par courrier électronique. La réponse peut prendre plusieurs jours, mais cela fournit un service équivalent à ftp pour ceux qui n'ont pas un accès direct au ftp par Internet. L'adresse du serveur est:

ftpmail@decwrl.dec.com

Le script suivant récupérera la dernière version PostScript du document SQL3 :

```
reply joe.programmer@imaginary-corp.com
connect gatekeeper.dec.com anonymous
binary
compress
```

Le script suivant récupérera la dernière version PostScript du document SQL3 :

```
reply joe.programmer@imaginary-corp.com
```

```
connect gatekeeper.dec.com anonymous
binary
compress
uuencode
chdir /pub/standards/sql
get x3h2-93-091.ps
quit
```

La première ligne du script commande au serveur de vous retourner les fichiers demandés; vous devrez remplacer "joe.programmer@imaginary-corp.com" par votre adresse Internet. Le fichier x3h2-93-091.ps, dans cet exemple, est retourné sous forme de 34 messages e-mail séparés, "compressés" et "uuencodés". Si votre environnement ne dispose pas des outils pour reconstruire de tels fichiers, alors, il vous faudra récupérer le fichier sous forme de texte pur en utilisant le script suivant:

```
reply joe.programmer@imaginary-corp.com
connect gatekeeper.dec.com anonymous
chdir /pub/standards/sql
get x3h2-93-091.ps
quit
```

Mais, attention, le fichier .ps vous sera probablement envoyé en plus de 70 morceaux!

Pour récupérer un fichier particulier, autre que x3h2-93-091.ps, remplacez simplement "x3h2-93-091.ps" par le nom du fichier désiré. Pour obtenir la liste de tous les fichiers disponibles dans un répertoire, remplacez "get x3h2-93-091.ps" par "dir".

24.10 Partie 1 - Etat Actuel de l'ISO/ANSI SQL

Ce chapitre constitue une source d'information à propos du processus de normalisation de SQL et de son état actuel.

État Actuel

Le développement pour améliorer SQL afin d'en faire un langage complet pour la définition et la gestion d'objets persistants et complexes. Ceci comprend : les hiérarchies de généralisation et de spécialisation, l'héritage multiple, les types de données définies par l'utilisateur, les déclencheurs et les assertions, le support des systèmes à base de connaissances, les requêtes récursives, et des outils complémentaires d'administration de données. Cela inclue également la spécification des types de données abstraits (ADTs), des identificateurs d'objets, des méthodes, de l'héritage, du polymorphisme, de l'encapsulation et de tous les autres moyens normalement associés à la gestion d'objets.

A l'automne 1996, plusieurs parties de SQL3 ont été soumises à un vote de l'ISO CD. Ce sont SQL/Framework, SQL/Foundation et SQL/Bindings. Ce vote échoua (comme prévu) avec environ 900 commentaires. Fin Janvier, un comité de rédaction traita un grand nombre de solutions aux problèmes qui furent soit incluses avec les commentaires du vote ou soumises en tant que documents séparés. Depuis le comité de rédaction DBL n'a pas pu traiter tous les commentaires et sa durée a été augmentée. La fin du comité de rédaction est prévue pour la fin Juillet 1997, à Londres.

A la suite du comité de rédaction de Juillet, on attend qu'un vote final du CD soit demandé pour ces parties de SQL. Le processus du CD Final prendra environ 6 mois et un comité de rédaction du DBL, après lesquels il y aura un vote IS et un très rapide vote IS.

Les procédures ISO ont changé depuis SQL/92, et les comités SQL travaillent encore selon les règles du processus.

Si tout va bien, ces parties du SQL3 deviendront une norme officielle ISO/IEC en fin 1998, mais le programme est très serré.

En 1993, les comités de développement ANSI et ISO ont décidé de d'éclater le développement futur de SQL en une norme multi-parties. Ces parties sont :

- Partie 1: Structure Description non technique de la manière dont le document doit être structuré.
- Partie 2: Base Spécifications de base, incluant toutes les nouvelles fonctionnalités ADT.
- Partie 3: SQL/CLI Interface du niveau appel (Call Level Interface).
- Partie 4: SQL/PSM Spécifications des procédures enregistrées
- Partie 5: SQL/Bindings SQL Dynamique et SQL Embedded SQL liens venant de SQL-92.
- Partie 6: SQL/XA Une spécialisation de l'interface populaire XA développée par X/Open
- Partie 7:SQL/Temporal Ajoute les capacités liées au temps à la norme SQL.

Aux USA, la totalité de SQL3 est en cours de traitement à la fois comme projet ("D") national et comme projet ISO. La fin de SQL3 est attendue pour courant 1999.

Le SQL/CLI et le SQL/PSM sont en cours de traitement aussi rapidement que possible comme addenda à SQL-92. Aux USA, ils sont traités uniquement comme projet International. SQL/CLI a été terminé en 1995. SQL/PSM devrait être terminé fin 1996.

En plus du travail sur SQL3, quelques projets additionnels sont poursuivis :

- SQL/MM Un effort en cours pour définir une norme multi-media utilisant les possibilités de SQL3 ADT.
- Accès aux données à distance (RDA)

Comités et Processus de Standardisation

Il y a effectivement de nombreux comités de normalisation de par le monde. Il y a un groupe international de normalisation SQL faisant partie de l'ISO. Un certain nombre de pays on des comités qui se consacrent à SQL. Ces pays (habituellement) envoient des représentants aux réunions ISO/IEC JTC1/SC 21/WG3 DBL. Les pays qui participent activement au processus de normalisation ISO SQL sont :

- Australie
- Brésil
- Canada
- France
- Allemagne
- Japon
- Corée
- Pays-Bas
- Angleterre
- États-Unis

Validation NIST

Les implémentations SQL sont validées (aux États-Unis) par le National Institute of Standards and Testing (NIST). Le NIST possède une suite de tests de validation pour le niveau de base SQL-92. Les détails exacts des exigences de la validation NIST sont définis comme un Federal Information Processing Standard (FIPS). Les exigences actuelles pour SQL sont définies dans le document FIPS 127-2. Les versions Postscript et Texte de ce document peuvent être obtenues au NIST. La Liste des Produits Validés à jour peut également être obtenue au NIST.

Publications et Articles sur le Standard SQL

Il y a deux versions de la norme SQL. Les deux sont disponibles à l'ANSI:

- ISO/IEC 9075:1992, "Information Technology — Database Languages — SQL"
- ANSI X3.135-1992, "Database Language SQL"

Les deux versions de la norme SQL sont identiques sauf la page de garde et les références aux autres normes. Les deux versions sont disponibles au :

```
American National Standards Institute
1430 Broadway
New York, NY 10018
USA
Téléphone (ventes): +1.212.642.4900
```

En supplément à la norme SQL-92, il y a maintenant un Technical Corrigendum (correction d'erreurs) :

```
* Technical Corrigendum 1:1994 to ISO/IEC 9075:1992
```

Le TC 1 doit aussi être disponible à l'ANSI. Il n'y a qu'une version du TC 1 – qui s'applique à la fois aux versions ISO et ANSI de SQL-92.

En supplément à ces normes, plusieurs livres ont été écrits sur la norme SQL 1992. Ces livrent donnent une description bien plus lisible de la norme que celle donnée par le norme elle-même.

Standards connexes

Un certain nombre d'autres normes présentent de l'intérêt pour la communauté SQL. Cette section contient des pointeurs donnant des informations sur ces efforts. Ces pointeurs seront plus nombreux à mesure que des informations complémentaires seront disponibles sur le web.

- SQL Environnements (FIPS 193) - (Environnements SQL)
- Next Generation Repository Systems (X3H4) - (Systèmes Dépositaires de la Prochaine Génération) Une version des Nouvelles appelant à une participation au "Développement des Standards pour les Systèmes Dépositaires de la Prochaine Génération."

24.11 Partie 2 - Fondements ISO/ANSI SQL

Une partie importante de l'effort pour SQL3 réside dans le document de base SQL :

- Capacités de Base SQL/PSM (venant de SQL/PSM-92)
- Nouveaux types de données
- Déclencheurs
- Sous-tables
- Types de Données Abstraits (ADT)
- Capacités Orientées Objets

Il y a plusieurs pré-requis pour les capacités orientées objets :

- Capacité de pouvoir définir des opérations complexes
- Stockage des opérations complexes dans la base de données
- Appels de Procédures Externes Quelques opérations peuvent ne pas exister dans SQL ou peuvent nécessiter des interactions externes

Ces capacités sont définies comme faisant partie de SQL/PSM

Une grande quantité de travail est en cours de réalisation pour affiner le modèle objet SQL-3 et l'aligner sur le modèle objet proposé par l'ODMG. Cet effort est décrit dans les papiers X3H2 et ISO DBL : Accommoder SQL3 et ODMG. Une mise à jour récente sur le Fusionneur SQL3/OQL est également disponible.

Planification de SQL3

Le travail sur SQL3 se poursuit normalement, mais la publication de la norme finale se situe encore à un horizon de plusieurs années.

- Le vote International pour passer du Brouillon de Travail au Brouillon du Comité (CD) aura lieu à l'automne 1996.
- On s'attend à ce que le vote génère de nombreux commentaires
- Un second vote sera probablement nécessaire
- Le vote pour le Brouillon du Standard International devrait avoir lieu à la mi-1998
- Le Standard International doit être terminé à la mi-1999.

Le programme de la version ANSI sera similaire.

24.12 Partie 3 - Interface d'Appel SQL ISO/ANSI

SQL/CLI est une interface de programmation d'appel pour les bases de données SQL. Il est conçu pour accéder aux bases de données à partir d'applications fermées. CLI a, à l'origine, été créé par un sous-comité du groupe SQL Access (SAG). Les spécifications de SAG/CLI ont été publiées, en 1992, en tant que spécifications Microsoft Open DataBase Connectivity (ODBC) en 1992. En 1993, le SAG a soumis le CLI aux comités ANSI et ISO SQL. (Le groupe "SQL Access Group" a maintenant été fusionné avec le consortium X/Open.)

SQL/CLI fournit une norme internationale pour :

- Un CLI dépendant de l'implémentation pour accéder aux bases de données SQL
- Des outils Client-serveur pour accéder facilement à une base de données au moyen de Bibliothèques de Lien Dynamique
- Supporter et encourager un riche ensemble d'outils Client-serveur

Planification de SQL/CLI

Le processus de normalisation de SQL/CLI se poursuit à une vitesse aveuglante.

- SQL/CLI est un addendum à la norme SQL de 1992 (SQL-92)
- Achevé comme norme ISO en 1995
- ISO/IEC 9075-3:1995 Information technology – Database languages – SQL – Partie 3: Call-Level Interface (SQL/CLI)
- L'effort actuel porte sur l'ajout des fonctionnalités SQL3 à SQL/CLI

24.13 Partie 4 - Modules Mémorisés Persistants ISO/ANSI SQL

SQL/PSM étend SQL en ajoutant :

- Des extensions de langage procédural
- Des procédures mémorisées multi-instructions

- Des appels de procédures et de fonctions externes

SQL/PSM apporte, en plus du fait d'être un bon outil de développement d'application, une base pour des fonctionnalités orientées objet dans SQL3.

Procédures mémorisées et multi-instructions

Multi-instructions et procédures mémorisées offrent une grande variété d'avantages dans un environnement client/serveur :

- Performance - Puisqu'une procédure mémorisée peut effectuer de multiples instructions SQL, les interactions sur le réseau, avec le client, sont réduites.
- Sécurité - On peut donner à un utilisateur le droit d'appeler une procédure mémorisée qui met à jour une table ou un ensemble de tables mais lui refuser le droit de mettre à jour les tables directement
- Code partagé - Le code d'une procédure mémorisée n'a pas à être réécrit ni re-testé pour chaque outil d'un client qui accède à la base de données.
- Contrôle - Fournit un point unique de définition et de contrôle dans la logique de l'application.

Extensions de Langage Procédural

Le langage procédural ajoute la puissance d'un langage de programmation traditionnel à SQL grâce à des instructions de contrôle de flot et à d'autres constructions utilisées en programmation.

Instructions de contrôle de flot

- If-then-else (si-alors-sinon)
- Instructions de boucle
- Prise en compte de erreurs
- Instruction de choix multiple (Case)
- Blocks (Begin-End blocks)

Les extensions de langage procédural comprennent d'autres constructions des langages de programmation :

- Déclaration des variables
- Instructions d'affectation de valeur
- Lecture de diagnostiques sur les processus et l'état

De plus, toutes les instructions SQL traditionnelles peuvent être inclues dans de procédures multi-instructions.

Appels de Fonctions et de Procédures externes

Une fonctionnalité souvent mentionnée dans la liste des souhaits pour de nombreuses base de données et produits et qui est implémentée dans quelques-uns, est la capacité de pouvoir augmenter les fonctionnalités du logiciel de base de données d'origine par des appels à des procédures externes écrites par les utilisateurs.

- Cela permet à un site ou a une application en particulier d'ajouter leurs propres fonctions base de données
- Peuvent être utilisées tout au long des applications bases de données

Le bénéfice de cette capacité est de donner accès, pour la base de données (et, par là, aux applications base de données), à un riche ensemble de procédures et de fonctions trop nombreuses pour être définies par un comité de normalisation.

Planification de SQL/PSM

SQL/PSM évolue rapidement :

- SQL/PSM est un addendum à SQL-92
- Un vote international pour faire passer SQL/PSM de Brouillon de Norme Internationale à Norme Internationale s'est achevé en fin janvier 1996.
- Une réunion de travail en mai 1996 n'a pas apporté de solution à tous les commentaires
- La poursuite de la réunion de travail du PSM est programmée du 30 septembre au 4 octobre 1996
- Le programme est serré, mais il y a des chances que le PSM le publie en 1996.
- Sa désignation officielle sera : ISO/IEC DIS 9075-4:199? Information technology – Database languages – SQL – Partie 4: SQL Persistent Stored Modules (SQL/PSM)
- L'ajout du support de SQL/PSM dans les fonctionnalités de SQL3 est en bonne voie.

24.14 Partie 5 - ISO/ANSI SQL/Liens

Pour en faciliter l'accès, la description des liens avec le langage de programmation a été mise dans un document séparé. La version actuelle consiste simplement en un extrait concernant les liens dynamiques et intégrés de SQL-92.

Un certain nombre de sujets concernant les liens avec le langage de programmation restent en suspens.

Pour les langage de programmation traditionnels, il existe des transformations pour les types de données de SQL-92. Cependant, des transformations doivent être définis entre les objets SQL et les variables des langages de programmation.

Pour les langages orientés objets, la transformation doit être définie pour les types de données courants SQL et entre le modèle objet SQL et le modèle objet du langage orienté objet.

Le modèle objet doit se stabiliser avant de résoudre ces problèmes.

Les liens avec le langage feront partie intégrante de SQL3.

24.15 Partie 6 - Interface de Spécialisation (SQL/XA) ISO/ANSI SQL XA)

Cette spécification devrait normaliser l'interface de programmation d'application (API) entre un Gestionnaire de Transaction global et le Gestionnaire de Ressources SQL. Il devrait normaliser les appels de fonction, en s'appuyant sur la sémantique de l'ISO/IEC 10026, "Distributed Transaction Processing", qu'un Gestionnaire de Ressources SQL devrait supporter pour la validation en deux phases. Le document de base est dérivé d'une publication de l'X/Open, avec sa permission, qui spécifie explicitement les paramètres d'entrée, de sortie et la sémantique, en termes de types de données SQL, pour les fonctions suivantes : xa_close, xa_commit, xa_complete, xa_end, xa_forget, xa_open, xa_prepare, xa_recover, xa_rollback, et xa_start.

ISO essaye actuellement de coller aux spécifications XA de l'X/Open. Ce processus de suivi, au plus près, consiste à adopter les spécifications de l'industrie sans changements. Le vote pour l'XA à l'ISO SC21, JTC 1 a débuté le 27 avril 1995 et se termine le 27 octobre 1995. Si les spécifications XA sont approuvées par 75% des votants, et par 2/3 des membres participants de JTC 1, cela deviendra une Norme Internationale. Si le vote est approuvé, SQL/XA pourrait devenir une norme en 1996.

24.16 Partie 7 - ISO/ANSI SQL Temporel

Le SQL Temporel traite des données en relation avec le temps. Conceptuellement, cela consiste à dire qu'il est utile d'aller chercher des données pour découvrir l'allure qu'elles avaient à un certain moment. L'origine du SQL Temporel réside dans un papier de décembre 1994 de Rick Snodgrass qui en décrivait les concepts.

X3 Annonce l'Approbation d'un Nouveau Projet, ISO/IEC 9075 Partie 7 : SQL/Temporel est un communiqué de presse concernant le SQL/Temporel.

```
-----------------------------------------------------------------------------
                                SQL Temporel
                                ************
Rick Snodgrass (Président du comité TSQL2)
31-Dec-1994
```

Plusieurs personnes se sont posé la question de la nécessité de la prise en charge de la gestion du temps dans SQL3 (comme cela a été proposé par DBL RIO-75, demandant qu'une nouvelle partie de SQL prenne en charge les bases de données temporelles). Le sentiment est que les types de données abstraits (ADT) sont suffisants pour gérer les questions de temps. Dans ce papier, je démontre, par des exemples concrets, que l'utilisation de colonnes typées, avec des types de données abstraits, est inadaptée pour des requêtes temporelles. En particulier, beaucoup de requêtes temporelles courantes sont soit difficiles à simuler en SQL, soit nécessitent d'intégrer SQL dans un langage procédural. Des solutions alternatives sont exposées dans TSQL2, une extension temporelle à SQL-92.

24.16.1 INTRODUCTION

La prise en charge d'un instant de validité va au-delà de celle de types de données temporels abstraits (ADT temporel). Avec ce dernier, on spécifie qu'une colonne sera celle consacrée au domaine temporel, tel que la DATE ou un INTERVALLE (je donnerai des exemples bientôt). Avec un instant de validité, les lignes d'une table varient en fonction du temps, reflétant les modifications de la réalité. Le timbre dateur associé à une ligne d'une table à instant de validité est interprété par le langage d'interrogation comme l'instant où la combinaison des valeurs dans les colonnes d'une ligne était valide. Ce timbre dateur implicite permet d'exprimer des requêtes de manière simple et intuitive.

24.16.2 ÉTUDE DE CAS - ENREGISTREMENT DE L'INFORMATION COURANTE

Le bureau du personnel (BP) salarié de l'université de l'Arizona possède un certain nombre d'informations, dans une base de données, dont le nom de chaque employé, son salaire actuel ainsi que son titre. On peut représenter cela dans une simple table.

```
Employee(Name, Salary, Title)
```

Étant donné cette table, retrouver le salaire d'un employé est une opération aisée.

```
SELECT Salary
FROM Employee
WHERE Name = 'Bob'
```

Maintenant, le BP souhaite enregistrer les dates de naissance. Pour ce faire, on ajoute une colonne à la table, ce qui donne le schéma suivant.

```
Employee(Name, Salary, Title, DateofBirth DATE)
```

Retrouver la date de naissance d'un employé est analogue à retrouver son salaire.

```
SELECT DateofBirth
FROM Employee
WHERE Name = 'Bob'
```

24.16.3 ÉTUDE DE CAS - ENREGISTREMENT DE L'INFORMATION HISTORIQUE

Le BP souhaite informatiser l'historique de l'emploi. Pour ce faire, ils ajoutent deux colonnes, l'une pour indiquer quand l'information contenue dans le ligne est devenue valide, l'autre indiquant la fin de validité de cette information.

Employee (Name, Salary, Title, DateofBirth, Start DATE, Stop DATE)

Pour le modèle de données, ces nouvelles colonnes sont identiques à la date de naissance. Cependant, leur présence à des conséquences à grande échelle.

24.16.4 ÉTUDE DE CAS - PROJECTION

Pour trouver le salaire actuel d'un salarié, les choses sont plus difficiles.

```
SELECT Salary
FROM Employee
WHERE Name = 'Bob' AND Start <= CURRENT_DATE AND CURRENT_DATE <= Stop
```

Cette requête est plus compliquée que la précédente. Le coupable se situe, de toute évidence, dans ces deux nouvelles colonnes. Le BP veut fournir à chaque employé l'historique de leur salaire. Pour chaque personne, spécifiquement, il est nécessaire de déterminer les intervalles correspondants à chacun des salaires. Malheureusement, cela n'est pas possible en SQL. Un employé peut avoir un nombre arbitraire de changements de titre entre deux changements de salaire.

```
Name    Salary  Title             DateofBirth   Start         Stop
----    ------  -----             -----------   -----         ----
Bob     60000   Assistant Provost 1945-04-09    1993-01-01    1993-05-30
Bob     70000   Assistant Provost 1945-04-09    1993-06-01    1993-09-30
Bob     70000   Provost           1945-04-09    1993-10-01    1994-01-31
Bob     70000   Professor         1945-04-09    1994-02-01    1994-12-31
```

Figure 1

Notez qu'il y a trois lignes pour lesquelles le salaire de Bob reste constant à $ 70,000. De là, le résultat sera de deux lignes pour Bob.

```
Name    Salary  Start         Stop
----    ------  -----         ----
Bob     60000   1993-01-01    1993-05-30
Bob     70000   1993-06-01    1994-12-31
```

Une alternative est de donner à l'utilisateur une édition des informations Salaire et Titre, et de laisser l'utilisateur déterminer quand son salaire a changé. Cette alternative n'est ni très attirante ni très réaliste. Une seconde alternative est d'utiliser SQL autant que possible.

```
CREATE TABLE Temp(Salary, Start, Stop)
AS      SELECT Salary, Start, Stop
        FROM Employee;
```

répéter

```
UPDATE Temp T1
SET (T1.Stop) = (SELECT MAX(T2.Stop)
                 FROM Temp AS T2
                 WHERE T1.Salary = T2.Salary AND T1.Start < T2.Start
                       AND T1.Stop >= T2.Start AND T1.Stop < T2.Stop)
WHERE EXISTS (SELECT *
```

```
                    FROM Temp AS T2
                    WHERE T1.Salary = T2.Salary AND T1.Start < T2.Start
                            AND T1.Stop >= T2.Start AND T1.Stop < T2.Stop)
        jusqu'à ce qu'aucune ligne ne soit mise à jour;

DELETE FROM Temp T1

WHERE EXISTS (SELECT *
              FROM Temp AS T2
              WHERE T1.Salary = T2.Salary
                        AND ((T1.Start > T2.Start AND T1.Stop <= T2.Stop)
                        OR (T1.Start >= T2.Start AND T1.Stop < T2.Stop))
```

La boucle trouve des intervalles qui se chevauchent ou sont adjacents et, doivent donc être fusionnés. La boucle est exécutée log N fois dans le pire des cas, où N est le nombre de lignes dans une chaîne de lignes de valeur équivalente qui se chevauchent ou sont adjacentes. Le lecteur peut simuler cette requête sur la table de l'exemple pour se convaincre de sa vérité.

Une troisième alternative consiste à utiliser SQL uniquement pour ouvrir un curseur sur la table. Une liste chaînée de périodes est entretenue, chacune avec un salaire. La liste chaînée doit être initialisée vide.

```
DECLARE emp_cursor CURSOR FOR
        SELECT Salary, Title, Start, Stop
        FROM Employee;
OPEN emp_cursor;
boucle:
        FETCH emp_cursor INTO :salary, :start, :stop;
        si pas-de-données est retourné alors aller à fin;
        trouver la position dans la liste chaînée pour insérer cette information;
        aller à boucle;
fin:
CLOSE emp_cursor;
```

Répéter tout au long de la liste chaînée, édition des dates et des salaires

La liste chaînée peut ne pas être nécessaire au cas où le curseur est ORDER BY Start.

Dans tous les cas, la requête, tout ce qu'il y a de naturelle, est tout à fait difficile à exprimer en utilisant les possibilités offertes par SQL-92. La requête est triviale en TSQL2.

```
SELECT Salary
FROM Employee
```

24.16.5 ÉTUDE DE CAS - JOINTURE

Une approche plus drastique consiste à éviter le problème d'extraction de l'historique du salaire en réorganisant le schéma pour séparer les informations de salaire, de titre, et de date de naissance (dans ce qui suit, nous ignorerons la date de naissance pour raison de simplicité).

```
Employee1 (Name, Salary, Start DATE, Stop DATE)
Employee2 (Name, Title, Start DATE, Stop DATE)
```

La table Employee1 est la suivante.

```
Name    Salary  Start           Stop
----    ------  -----           ----
Bob     60000   1993-01-01      1993-05-30
Bob     70000   1993-06-01      1993-12-31
```

Voici un exemple de la table Employee2.

```
Name    Title                   Start           Stop
----    ------                  -----           ----
Bob     Assistant Provost       1993-01-01      1993-09-30
Bob     Provost                 1993-10-01      1994-01-31
Bob     Professor               1994-02-01      1994-12-31
```

Avec cette modification, aller chercher l'information de salaire d'un employé devient maintenant facile.

```
SELECT Salary, Start, Stop
FROM Employee1
WHERE Name = 'Bob'
```

Mais qu'en est-il si le BP veut une table des salaires, des intervalles de titre (c'est-à-dire, supposons que le BP veuille qu'une table soit calculée dans la forme de la Figure 1)? Une alternative est d'éditer les deux tables, et de laisser l'utilisateur en calculer les combinaisons. Une seconde alternative est d'utiliser entièrement SQL. Malheureusement, cette requête doit faire une étude de cas sur la manière dont chaque ligne de Employee1 se chevauche avec chaque ligne de Employee2; il y a quatre cas possibles.

```
SELECT Employee1.Name, Salary, Dept, Employee1.Start, Employee1.Stop
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
     AND Employee2.Start <= Employee1.Start AND Employee1.Stop < Employee2.Stop
UNION
SELECT Employee1.Name, Salary, Dept, Employee1.Start, Employee2.Stop
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
     AND Employee1.Start >= Employee2.Start AND Employee2.Stop < Employee1.Stop
        AND Employee1.Start < Employee2.Stop
UNION
SELECT Employee1.Name, Salary, Dept, Employee2.Start, Employee1.Stop
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
     AND Employee2.Start > Employee1.Start AND Employee1.Stop < Employee2.Stop
        AND Employee2.Start < Employee1.Stop
UNION
SELECT Employee1.Name, Salary, Dept, Employee2.Start, Employee2.Stop
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
     AND Employee2.Start > Employee1.Start AND Employee2.Stop < Employee1.Stop
```

Prendre en compte, d'une façon correcte, tous les cas constitue une tâche qui relève du challenge. En TSQL2, réaliser une jointure temporelle correspond simplement à ce que l'on peut espérer.

```
SELECT Employee1.Name, Salary, Dept
FROM Employee1, Employee2
WHERE Employee1.Name = Employee2.Name
```

24.16.6 ÉTUDE DE CAS - AGRÉGATS

Maintenant, on demande au BP quel est le salaire maximum? Avant d'ajouter les données de temps, c'était facile.

```
SELECT MAX(Salary)
FROM Employee
```

Maintenant que l'historique du salaire est enregistré, nous obtiendrons quelque chose comme l'historique du salaire maximum à travers le temps. Le problème est que, naturellement, SQL ne fournit pas d'agrégats temporels. Le moyen simple de réaliser cela consiste à éditer l'information, et à chercher, à la main, le maximum. Une alternative consiste à être plus astucieux et à convertir l'instantané de la requête agrégée en une requête non agrégée, puis à convertir le résultat en requête temporelle. La requête non agrégée retrouve les salaires pour lesquels un salaire plus élevé n'existe pas.

```
SELECT Salary
FROM Employee AS E1
WHERE NOT EXISTS (SELECT *
                  FROM Employee AS E2
                  WHERE E2.Salary > E1.Salary)
```

Convertir cette requête en une requête temporelle est loin d'être évident. Ce qui suit constitue une approche possible.

```
CREATE TABLE Temp (Salary, Start, Stop)
AS      SELECT Salary, Start, Stop
        FROM Employee;
INSERT INTO Temp
        SELECT T.Salary, T.Start, E.Start
        FROM Temp AS T, Employee AS E
        WHERE E.Start >= T.Start AND E.Start < T.Stop AND E.Salary > T.Salary;

INSERT INTO Temp
        SELECT T.Salary, T.Stop, E.Stop
        FROM Temp AS T, Employee AS E
        WHERE E.Stop > T.Start AND E.Stop <= T.Stop AND E.Salary > T.Salary;
DELETE FROM Temp T
WHERE EXISTS (SELECT *
              FROM Employee AS E
              WHERE ((T.Start => E.Start AND T.Start < E.Stop)
                        OR (E.Start >= T.Start AND E.Start < T.Stop))
                    AND E.Salary > T.Salary;
```

Cette approche crée une table auxiliaire. Nous ajoutons à cette table la période inférieure soustraite d'une période et la partie supérieure soustraite d'une période. Nous supprimons alors toutes les périodes qui se chevauchent avec une ligne définie dans la sous-requête, de ce fait réalisant le NOT EXISTS(n'existe pas). Finalement nous générons à partir de la table auxiliaire les périodes maximales, de la même façon que l'information de salaire avait été calculée ci-dessus. Comme on peut l'imaginer, un tel code SQLest extrêmement inefficace à l'exécution, compte tenu des requêtes complexes imbriquées contenant des prédicats d'inégalité.

Une troisième alternative consiste à utiliser le moins possible SQL, et plutôt de calculer le maximum désiré dans un langage hôte en utilisant des curseurs.

La requête en TSQL2 est de nouveau simple et intuitive.

```
SELECT MAX(Salary)
FROM Employee
```

24.16.7 RÉSUMÉ

Dans la plupart des applications de base de données on manipule des données qui changent en fonction du temps. La prise en compte de l'instant de validité est absente de SQL. Beaucoup de requêtes temporelles ordinaires sont difficiles à simuler en SQL, ou requièrent du SQL intégré dans un langage procédural, ceci étant dû au manque de support des tables avec instant de validité dans son modèle de données et dans la structure des requêtes.

Ailleurs, nous avons montré que l'ajout du support de l'instant de validité ne requiert que peu de modifications dans l'implémentation des SGDB, peut radicalement simplifier un certain nombre de requêtes et en permettre d'autres, et peut, plus tard, permettre des optimisations dans les structures de mémoires, les méthodes d'indexation et les stratégies d'optimisation qui peuvent entraîner des améliorations notables dans les performances.

Avec une nouvelle partie de SQL3 supportant les informations qui varient avec le temps, permettant une meilleure gestion des données temporelles par SQL3, nous pouvons commencer à apporter une réponse à de telles application.

```
------------------------------------------------------------------------
          Comité de Normalisation Accrédité* X3, Technologie de L'Information
COMMUNIQUE

Doc. No.:       PR/96-0002

Reply to:       Barbara Bennett at bbennett@itic.nw.dc.us

            X3 Annonce l'approbation d'un Nouveau Projet, ISO/IEC

                        9075 Partie 7:  SQL/Temporal

Washington D.C., Janvier 1996
------------------------------------------------------------------------
```

– Le Comité de Normalisation Accrédité X3, Technologie de l'Information, annonce l'approbation d'un nouveau projet concernant le support de SQL/Temporel , ISO/IEC 9075 Partie 7, le travail devant être réalisé au sein du Comité Technique X3H2, Bases de Données. Cette proposition de norme concerne un nouvelle partie de la norme naissante SQL3, e.g., Partie 7, SQL Temporel, et spécifie les extensions au langage SQL pour gérer le stockage, l'extraction et la manipulation des données temporelles dans un environnement de base de données SQL.

La prochaine réunion du comité X3H2 est planifié pour la période du 11 au 14 Mars 1996 dans le Kansas.

Les questions concernant ce projet doivent être envoyées au président de X3H2 :

```
Chairman of X3H2,
Dr. Donald R. Deutsch,
Sybase, Inc., Suite 800,
6550 Rock Spring
Drive, Bethesda, MD  20817.
Email: deutsch@sybase.com.
```

Un appel initial pour de possibles brevets ou autres sujets pertinents (copyrights, trademarks) est sur le point d'être émis. Soumettez, s'il vous plaît, les informations à ce propos au secrétariat de X3 à :

```
X3 Secretariat
1250 Eye Street
NW, Suite 200,
Washington DC  20005.
Email: x3sec@itic.nw.dc.us
FAX:  (202)638-4922.
```

24.17 Partie 8 - SQL MULTIMEDIA ISO/ANSI (SQL/MM)

Un nouveau projet international de normalisation ISO/IEC pour le développement d'une bibliothèque de classe SQL pour les applications multi-media a été approuvé début 1993. Cette nouvelle activité de normalisation, appelée SQL Multi-media (SQL/MM), spécifiera les paquetages de définition de types de données abstraites SQL (ADT) en utilisant les fonctions fournies dans les spécifications SQL3 naissantes pour ADT. SQL/MM a pour objectif de normaliser des bibliothèques de classes pour la science et l'ingénierie, le traitement des textes et des documents, ainsi que des méthodes pour la gestion d'objets multi-media tels que les images, les sons, les animations, la musique et la vidéo. Il fournira vraisemblablement un lien entre le langage SQL et les objets multi-media définis par les autres entités de normalisation du JTC1 (e.g. SC18 pour les documents, SC24 pour les images, et SC29 pour les photographies et les images animées).

Le Plan du Projet pour SQL/MM prévoit que ce sera une norme multi-parties dont le nombre de parties pourra évoluer. La partie 1 consistera en un cadre spécifiant comment les autres parties doivent être architecturé. Chacune de ces autres parties sera consacrée à un paquetage d'application SQL spécifique. En août 1994, la structure existante était la suivante :

- Partie 1: Cadre - Description non technique de la manière dont le document est structuré.
- Partie 2: Méthodes pour le Texte et ADT (types de données abstraits) pour le traitement des textes. Environ 45 pages.
- Partie 3: Méthodes Spatiales et ADT pour la gestion des données spatiales Environ 200 pages avec l'active contribution d'experts en Données Spatiales de 3 entités nationales.
- Partie 4: Méthode à Usage Général et ADT pour les nombres complexes, et des fonctionnalités incluant la trigonométrie et les fonctions exponentielles les vecteurs, les ensembles etc. Actuellement environ 90 pages.

On assiste à des efforts de normalisation dans le domaine des informations Spatiales et Géographiques :

- ANSI X3L1 - Systèmes d'Information Géographique. Mark Ashworth de Unisys est chargé de la liaison entre X3L1 et ANSI X3H2. Il est également le rédacteur des parties 1, 3, and 4 du brouillon de SQL/MM.
- ISO TC 211 - Information géographique/Géomatique.

25 Support Technique pour PostgreSQL

Si vous avez des questions techniques ou si vous rencontrez des problèmes vous pouvez envoyer un courrier électronique à :

- *pgsql-questions@postgresql.org*

et vous recevrez par courrier électronique une réponse dans la journée qui suit. Comme le nombre de produits internet est très large, et que les utilisateurs aident les autres utilisateurs, internet est capable d'apporter facilement un support technique à des millions d'utilisateurs. Le support par Email est de loin beaucoup plus adapté que celui effectué par téléphone car on peut récupérer les messages d'erreur, les sorties de programme etc... et les transmettre facilement dans les listes de diffusion ou les forums.

Dans un futur proche, l'organisation PostgreSQL vendra un support technique aux petites et grandes compagnies, les revenus de cette activité serviront à entretenir plusieurs sites miroirs (web et ftp) partout dans le monde. Ces revenus pourront aussi servir à créer une documentation imprimée, des guides, des livres pour aider les clients. Les profits seront également utilisés à construire des immeubles/salles de classe et à offrir des cours techniques sur l'administration et la gestion des bases de données et à organiser des conférences annuelles pour les utilisateurs. Ceci devrait aider l'organisation PostgreSQL à se débrouiller toute seule.

Vous pouvez aussi vous faire aider par des sociétés de conseil professionnelles telles que Anderson, WGS (Work Group Solutions). Pour de l'aide, puisqu'ils ont une très bonne expertise en "C", "C++" (PostgreSQL est écrit en "C"), contactez les à -

- Work Group Solutions `<http://www.wgs.com>`
- Anderson Consulting `<http://www.ac.com>`

26 Aspects Économiques et Commerciaux

Les bases de données commerciales paient de nombreuses taxes telles que des taxes fédérales, d'état, sur les ventes, sur les salariés, la sécurité sociale, les taxes pour les soins médicaux, des indemnités pour les employés, des coûts de marketing et de publicité. Tous ces coûts ne sont pas destinés directement au développement de la base de donnée. Quand vous achetez une base de données commerciale une partie du montant est destiné aux taxes, aux dépenses de recherche et développement (R&D). Donc la valeur réelle pour la base de données est beaucoup plus faible.

De plus les bases de données commerciales doivent payer pour leurs immeubles/biens et pour l'achat de machines Unix, leur installation et leur maintenance. Tous ces coûts sont répercutés sur les clients.

PostgreSQL possède l'avantage, sur les bases de données commerciales, de ne pas supporter de taxes puisque développée sur l'internet. Un très grand nombre de personnes contribuent à son développement. Par exemple, cas hypothétique, s'il y a un million de compagnies aux U.S.A et que chacune contribue pour environ $ 10 (en valeur de logiciel pour PostgreSQL) alors, chaque compagnie recevra dix millions de dollars!! C'est cela la **GRANDE MAGIE** du développement sur internet.

Actuellement, le code source de PostgreSQL est constitué d'environ 200 000 lignes de code "C" et "C++". Si le coût de chaque ligne de code "C" est évalué à $ 10, alors le coût total de PostgreSQL, tel qu'il est aujourd'hui est de $ 2 000 000 (deux millions de dollars!!).

De nombreuses compagnies ont déjà développé de grandes quantités de code "C", "C++" maison. Donc, en prenant le code source de PostgreSQL et en collaborant avec les autres compagnies sur internet, cela leur bénéficierait beaucoup en leur faisant économiser du temps et des efforts.

27 Liste des autre Bases de Données

On trouvera ci-dessous la liste des autres bases de données SQL pour Unix, Linux.

- Cliquez et allez à Applications->databases. `<http://www.caldera.com/tech-ref/linuxapps/linapps.html>`
- Cliquez et allez à Applications->databases. `<http://www.xnet.com/~blatura/linapps.shtml>`
- Ressources pour Bases de Données `<http://linas.org/linux/db.html>`
- Liste des Bases de Données libres `<http://cuiwww.unige.ch:80/~scg/FreeDB/FreeDB.list.html>`
- Liste des SGBDR de Browne `<http://www.hex.net/~cbbrowne/rdbms.html>` écrit par Christopher B. Browne *cbbrowne@hex.net*
- Liste des SGBDR de SAL `<http://SAL.KachinaTech.COM/H/1/>`
- Liste des SGBD Orientés Objets de SAL `<http://SAL.KachinaTech.COM/H/2/>`
- Liste des Utilitaires et d'Autres SGDB de SAL `<http://SAL.KachinaTech.COM/H/3/>`
- Index des Logiciels de Base de Données Publics de l'ACM SIGMOD `<http://bunny.cs.uiuc.edu/sigmod/databaseSoftware/>`

28 Indications pour la recherche sur le World Wide Web Internet

Internet est immense, recèle un très grand nombre de logiciels et un océan d'informations. Il s'accroît à la vitesse de 500% par an. On estime le nombre de sites Web à environ 90 million dans le monde entier!

Utilisez des moteurs de recherche tels que "Yahoo", "Netscape", "Lycos" etc. pour trouver une information. Allez sur Yahoo, et cliquez sur recherche. Utilisez les options de filtrage pour affiner vos critères de recherche. La méthode de recherche par défaut consiste en une "recherche intelligente" qui est plus générale et donne toutes les possibilités. Cliquez sur "Options" et sélectionnez la recherche par "EXACT phrase" , par "AND" , par "OR" , etc... De cette manière vous trouverez l'information que vous cherchez beaucoup plus rapidement. Dans le menu de recherche, il y a également des boutons pour une recherche dans Usenet, dans les sites Web et les sites Yahoo.

29 Conclusion

Après avoir cherché toutes les bases de données disponibles qui soient libres et dont le code source soit disponible, j'ai trouvé que SEUL PostgreSQL, le SGBDR (Relationnel Objet)SQL libre, est le PLUS mature et le plus largement utilisé au monde. PostgreSQL est très attractif en ce que beaucoup de travail a déjà été effectué et qu'il possède des pilotes ODBC et JDBC permettant d'écrire des applications indépendantes des bases de données. Les applications écrites en PostgreSQL qui utilisent les pilotes ODBC, JDBC sont facilement portables sur d'autres systèmes tels que Oracle, Sybase et Informix et vice versa.

Vous pouvez vous demander "Mais pourquoi PostgreSQL?" La réponse à cela est que puisqu'il faut pas mal de temps pour développer un système de gestion de base de données ex nihilo, il est plus raisonnable d'en utiliser un qui satisfasse aux conditions suivantes -

Un système de gestion de base de données

- Dont le code source soit disponible, ce doit être un système à "code source ouvert"
- Qui ne nécessite pas de licence, qui n'a pas de doits de propriété
- Qui puisse être distribué sur internet
- Qui a nécessité plusieurs années de développement.
- Qui satisfasse des standards tels que ISO/ANSI SQL 92 (and SQL 89)
- Qui puisse répondre aux besoins futurs telles que SQL 3 (SQL 98)
- Qui possède des fonctionnalités avancées

Et, justement, il appert que c'est 'PostgreSQL' qui répond à tous ces critères et que c'est un logiciel approprié dans ce cas.

30 FAQ - Questions sur PostgreSQL

Veuillez vous référer à la dernière version de la FAQ pour les questions générales, celles concernant Linux et Irix à :

- <http://www.postgresql.org/docs/faq-english.shtml>

31 Notice de Copyright

A Appendice A - Syntaxe de l' ANSI/ISO SQL 1992

```
Ce fichier contient la grammaire en BNF du langage, arborescence
explorée d'abord en profondeur, réalisée le 27-AOÛT-1992 11:03:41.64.
La version spécifique de BNF donnée ici est :
ANSI et SQL2-seulement.


<SQL terminal character> ::=
      <SQL language character>
    | <SQL embedded language character>

<SQL language character> ::=
      <simple Latin letter>
    | <digit>
    | <SQL special character>

<simple Latin letter> ::=
      <simple Latin upper case letter>
    | <simple Latin lower case letter>

<simple Latin upper case letter> ::=
          A | B | C | D | E | F | G | H | I | J | K | L | M | N | O
    | P | Q | R | S | T | U | V | W | X | Y | Z

<simple Latin lower case letter> ::=
          a | b | c | d | e | f | g | h | i | j | k | l | m | n | o
    | p | q | r | s | t | u | v | w | x | y | z

<digit> ::=
    0 | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9

<SQL special character> ::=
      <space>
    | <double quote>
    | <percent>
    | <ampersand>
```

```
    | <quote>
    | <left paren>
    | <right paren>
    | <asterisk>
    | <plus sign>
    | <comma>
    | <minus sign>
    | <period>
    | <solidus>
    | <colon>
    | <semicolon>
    | <less than operator>
    | <equals operator>
    | <greater than operator>
    | <question mark>
    | <underscore>
    | <vertical bar>

<space> ::= !! <EMPHASIS>(space character in character set in use)

<double quote> ::= "

<percent> ::= %

<ampersand> ::= &

<quote> ::= '

<left paren> ::= (

<right paren> ::= )

<asterisk> ::= *

<plus sign> ::= +

<comma> ::= ,

<minus sign> ::= -

<period> ::= .

<solidus> ::= /

<colon> ::= :

<semicolon> ::= ;

<less than operator> ::= <

<equals operator> ::= =

<greater than operator> ::= >

<question mark> ::= ?
```

```
<underscore> ::= _

<vertical bar> ::= |

<SQL embedded language character> ::=
      <left bracket>
    | <right bracket>

<left bracket> ::= [

<right bracket> ::= ]

<token> ::=
      <nondelimiter token>
    | <delimiter token>

<nondelimiter token> ::=
      <regular identifier>
    | <key word>
    | <unsigned numeric literal>
    | <national character string literal>
    | <bit string literal>
    | <hex string literal>

<regular identifier> ::= <identifier body>

<identifier body> ::=
    <identifier start> [ ( <underscore> | <identifier part> )... ]

<identifier start> ::= <EMPHASIS>(!! See the Syntax Rules)

<identifier part> ::=
      <identifier start>
    | <digit>

<key word> ::=
      <reserved word>
    | <non-reserved word>

<reserved word> ::=
      ABSOLUTE | ACTION | ADD | ALL
    | ALLOCATE | ALTER | AND
    | ANY | ARE
    | AS | ASC
    | ASSERTION | AT
    | AUTHORIZATION | AVG
    | BEGIN | BETWEEN | BIT | BIT_LENGTH
    | BOTH | BY
    | CASCADE | CASCADED | CASE | CAST
    | CATALOG
    | CHAR | CHARACTER | CHAR_LENGTH
    | CHARACTER_LENGTH | CHECK | CLOSE | COALESCE
    | COLLATE | COLLATION
    | COLUMN | COMMIT
```

```
| CONNECT
| CONNECTION | CONSTRAINT
| CONSTRAINTS | CONTINUE
| CONVERT | CORRESPONDING | COUNT | CREATE | CROSS
| CURRENT
| CURRENT_DATE | CURRENT_TIME
| CURRENT_TIMESTAMP | CURRENT_USER | CURSOR
| DATE | DAY | DEALLOCATE | DEC
| DECIMAL | DECLARE | DEFAULT | DEFERRABLE
| DEFERRED | DELETE | DESC | DESCRIBE | DESCRIPTOR
| DIAGNOSTICS
| DISCONNECT | DISTINCT | DOMAIN | DOUBLE | DROP
| ELSE | END | END-EXEC | ESCAPE
| EXCEPT | EXCEPTION
| EXEC | EXECUTE | EXISTS
| EXTERNAL | EXTRACT
| FALSE | FETCH | FIRST | FLOAT | FOR
| FOREIGN | FOUND | FROM | FULL
| GET | GLOBAL | GO | GOTO
| GRANT | GROUP
| HAVING | HOUR
| IDENTITY | IMMEDIATE | IN | INDICATOR
| INITIALLY | INNER | INPUT
| INSENSITIVE | INSERT | INT | INTEGER | INTERSECT
| INTERVAL | INTO | IS
| ISOLATION
| JOIN
| KEY
| LANGUAGE | LAST | LEADING | LEFT
| LEVEL | LIKE | LOCAL | LOWER
| MATCH | MAX | MIN | MINUTE | MODULE
| MONTH
| NAMES | NATIONAL | NATURAL | NCHAR | NEXT | NO
| NOT | NULL
| NULLIF | NUMERIC
| OCTET_LENGTH | OF
| ON | ONLY | OPEN | OPTION | OR
| ORDER | OUTER
| OUTPUT | OVERLAPS
| PAD | PARTIAL | POSITION | PRECISION | PREPARE
| PRESERVE | PRIMARY
| PRIOR | PRIVILEGES | PROCEDURE | PUBLIC
| READ | REAL | REFERENCES | RELATIVE | RESTRICT
| REVOKE | RIGHT
| ROLLBACK | ROWS
| SCHEMA | SCROLL | SECOND | SECTION
| SELECT
| SESSION | SESSION_USER | SET
| SIZE | SMALLINT | SOME | SPACE | SQL | SQLCODE
| SQLERROR | SQLSTATE
| SUBSTRING | SUM | SYSTEM_USER
| TABLE | TEMPORARY
| THEN | TIME | TIMESTAMP
| TIMEZONE_HOUR | TIMEZONE_MINUTE
| TO | TRAILING | TRANSACTION
```

```
    | TRANSLATE | TRANSLATION | TRIM | TRUE
    | UNION | UNIQUE | UNKNOWN | UPDATE | UPPER | USAGE
    | USER | USING
    | VALUE | VALUES | VARCHAR | VARYING | VIEW
    | WHEN | WHENEVER | WHERE | WITH | WORK | WRITE
    | YEAR
    | ZONE

<non-reserved word> ::=

      ADA
    | C | CATALOG_NAME
    | CHARACTER_SET_CATALOG | CHARACTER_SET_NAME
    | CHARACTER_SET_SCHEMA | CLASS_ORIGIN | COBOL | COLLATION_CATALOG
    | COLLATION_NAME | COLLATION_SCHEMA | COLUMN_NAME | COMMAND_FUNCTION
    | COMMITTED
    | CONDITION_NUMBER | CONNECTION_NAME | CONSTRAINT_CATALOG | CONSTRAINT_NAME
    | CONSTRAINT_SCHEMA | CURSOR_NAME
    | DATA | DATETIME_INTERVAL_CODE
    | DATETIME_INTERVAL_PRECISION | DYNAMIC_FUNCTION
    | FORTRAN
    | LENGTH
    | MESSAGE_LENGTH | MESSAGE_OCTET_LENGTH | MESSAGE_TEXT | MORE | MUMPS
    | NAME | NULLABLE | NUMBER
    | PASCAL | PLI
    | REPEATABLE | RETURNED_LENGTH | RETURNED_OCTET_LENGTH | RETURNED_SQLSTATE
    | ROW_COUNT
    | SCALE | SCHEMA_NAME | SERIALIZABLE | SERVER_NAME | SUBCLASS_ORIGIN
    | TABLE_NAME | TYPE
    | UNCOMMITTED | UNNAMED

<unsigned numeric literal> ::=
      <exact numeric literal>
    | <approximate numeric literal>

<exact numeric literal> ::=
      <unsigned integer> [ <period> [ <unsigned integer> ] ]
    | <period> <unsigned integer>

<unsigned integer> ::= <digit>...

<approximate numeric literal> ::= <mantissa> E <exponent>

<mantissa> ::= <exact numeric literal>

<exponent> ::= <signed integer>

<signed integer> ::= [ <sign> ] <unsigned integer>

<sign> ::= <plus sign> | <minus sign>

<national character string literal> ::=
    N <quote> [ <character representation>... ] <quote>
      [ ( <separator>... <quote> [ <character representation>... ] <quote> )... ]
```

```
<character representation> ::=
      <nonquote character>
    | <quote symbol>

<nonquote character> ::= !! <EMPHASIS>(See the Syntax Rules.)

<quote symbol> ::= <quote><quote>

<separator> ::= ( <comment> | <space> | <newline> )...

<comment> ::=
    <comment introducer> [ <comment character>... ] <newline>

<comment introducer> ::= <minus sign><minus sign>[<minus sign>...]

<comment character> ::=
      <nonquote character>
    | <quote>

<newline> ::= !! <EMPHASIS>(implementation-defined end-of-line indicator)

<bit string literal> ::=
    B <quote> [ <bit>... ] <quote>
      [ ( <separator>... <quote> [ <bit>... ] <quote> )... ]

<bit> ::= 0 | 1

<hex string literal> ::=
    X <quote> [ <hexit>... ] <quote>
      [ ( <separator>... <quote> [ <hexit>... ] <quote> )... ]

<hexit> ::= <digit> | A | B | C | D | E | F | a | b | c | d | e | f

<delimiter token> ::=
      <character string literal>
    | <date string>
    | <time string>
    | <timestamp string>
    | <interval string>
    | <delimited identifier>
    | <SQL special character>
    | <not equals operator>
    | <greater than or equals operator>
    | <less than or equals operator>
    | <concatenation operator>
    | <double period>
    | <left bracket>
    | <right bracket>

<character string literal> ::=
    [ <introducer><character set specification> ]
    <quote> [ <character representation>... ] <quote>
      [ ( <separator>... <quote> [ <character representation>... ] <quote> )... ]

<introducer> ::= <underscore>
```

```
<character set specification> ::=
      <standard character repertoire name>
    | <implementation-defined character repertoire name>
    | <user-defined character repertoire name>
    | <standard universal character form-of-use name>
    | <implementation-defined universal character form-of-use name>

<standard character repertoire name> ::= <character set name>

<character set name> ::= [ <schema name> <period> ]
      <SQL language identifier>

<schema name> ::=
    [ <catalog name> <period> ] <unqualified schema name>

<catalog name> ::= <identifier>

<identifier> ::=
    [ <introducer><character set specification> ] <actual identifier>

<actual identifier> ::=
      <regular identifier>
    | <delimited identifier>

<delimited identifier> ::=
    <double quote> <delimited identifier body> <double quote>

<delimited identifier body> ::= <delimited identifier part>...

<delimited identifier part> ::=
      <nondoublequote character>
    | <doublequote symbol>

<nondoublequote character> ::= <EMPHASIS>(!! See the Syntax Rules)

<doublequote symbol> ::= <double quote><double quote>

<unqualified schema name> ::= <identifier>

<SQL language identifier> ::=
    <SQL language identifier start>
       [ ( <underscore> | <SQL language identifier part> )... ]

<SQL language identifier start> ::= <simple Latin letter>

<SQL language identifier part> ::=
      <simple Latin letter>
    | <digit>

<implementation-defined character repertoire name> ::=
    <character set name>

<user-defined character repertoire name> ::= <character set name>
```

```
<standard universal character form-of-use name> ::=
    <character set name>

<implementation-defined universal character form-of-use name> ::=
    <character set name>

<date string> ::=
    <quote> <date value> <quote>

<date value> ::=
    <years value> <minus sign> <months value>
        <minus sign> <days value>

<years value> ::= <datetime value>

<datetime value> ::= <unsigned integer>

<months value> ::= <datetime value>

<days value> ::= <datetime value>

<time string> ::=
    <quote> <time value> [ <time zone interval> ] <quote>

<time value> ::=
    <hours value> <colon> <minutes value> <colon> <seconds value>

<hours value> ::= <datetime value>

<minutes value> ::= <datetime value>

<seconds value> ::=
      <seconds integer value> [ <period> [ <seconds fraction> ] ]

<seconds integer value> ::= <unsigned integer>

<seconds fraction> ::= <unsigned integer>

<time zone interval> ::=
    <sign> <hours value> <colon> <minutes value>

<timestamp string> ::=
    <quote> <date value> <space> <time value>
        [ <time zone interval> ] <quote>

<interval string> ::=
    <quote> ( <year-month literal> | <day-time literal> ) <quote>

<year-month literal> ::=
      <years value>
    | [ <years value> <minus sign> ] <months value>

<day-time literal> ::=
      <day-time interval>
    | <time interval>
```

```
<day-time interval> ::=
    <days value>
      [ <space> <hours value> [ <colon> <minutes value>
        [ <colon> <seconds value> ] ] ]

<time interval> ::=
      <hours value> [ <colon> <minutes value> [ <colon> <seconds value> ] ]
    | <minutes value> [ <colon> <seconds value> ]
    | <seconds value>

<not equals operator> ::= <>

<greater than or equals operator> ::= >=

<less than or equals operator> ::= <=

<concatenation operator> ::= ||

<double period> ::= ..

<module> ::=
    <module name clause>
    <language clause>
    <module authorization clause>
    [ <temporary table declaration>... ]
    <module contents>...

<module name clause> ::=
    MODULE [ <module name> ]
      [ <module character set specification> ]

<module name> ::= <identifier>

<module character set specification> ::=
    NAMES ARE <character set specification>

<language clause> ::=
    LANGUAGE <language name>

<language name> ::=
    ADA | C | COBOL | FORTRAN | MUMPS | PASCAL | PLI

<module authorization clause> ::=
      SCHEMA <schema name>
    | AUTHORIZATION <module authorization identifier>
    | SCHEMA <schema name>
          AUTHORIZATION <module authorization identifier>

<module authorization identifier> ::=
    <authorization identifier>

<authorization identifier> ::= <identifier>

<temporary table declaration> ::=
```

```
    DECLARE LOCAL TEMPORARY TABLE
        <qualified local table name>
      <table element list>
      [ ON COMMIT ( PRESERVE | DELETE ) ROWS ]

<qualified local table name> ::=
    MODULE <period> <local table name>

<local table name> ::= <qualified identifier>

<qualified identifier> ::= <identifier>

<table element list> ::=
      <left paren> <table element> [ ( <comma> <table element> )... ] <right paren>

<table element> ::=
      <column definition>
    | <table constraint definition>

<column definition> ::=
    <column name> ( <data type> | <domain name> )
    [ <default clause> ]
    [ <column constraint definition>... ]
    [ <collate clause> ]

<column name> ::= <identifier>

<data type> ::=
      <character string type>
           [ CHARACTER SET <character set specification> ]
    | <national character string type>
    | <bit string type>
    | <numeric type>
    | <datetime type>
    | <interval type>

<character string type> ::=
      CHARACTER [ <left paren> <length> <right paren> ]
    | CHAR [ <left paren> <length> <right paren> ]
    | CHARACTER VARYING <left paren> <length> <right paren>
    | CHAR VARYING <left paren> <length> <right paren>
    | VARCHAR <left paren> <length> <right paren>

<length> ::= <unsigned integer>

<national character string type> ::=
      NATIONAL CHARACTER [ <left paren> <length> <right paren> ]
    | NATIONAL CHAR [ <left paren> <length> <right paren> ]
    | NCHAR [ <left paren> <length> <right paren> ]
    | NATIONAL CHARACTER VARYING <left paren> <length> <right paren>
    | NATIONAL CHAR VARYING <left paren> <length> <right paren>
    | NCHAR VARYING <left paren> <length> <right paren>

<bit string type> ::=
      BIT [ <left paren> <length> <right paren> ]
```

```
    | BIT VARYING <left paren> <length> <right paren>

<numeric type> ::=
      <exact numeric type>
    | <approximate numeric type>

<exact numeric type> ::=
      NUMERIC [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | DECIMAL [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | DEC [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | INTEGER
    | INT
    | SMALLINT

<precision> ::= <unsigned integer>

<scale> ::= <unsigned integer>

<approximate numeric type> ::=
      FLOAT [ <left paren> <precision> <right paren> ]
    | REAL
    | DOUBLE PRECISION

<datetime type> ::=
      DATE
    | TIME [ <left paren> <time precision> <right paren> ]
          [ WITH TIME ZONE ]
    | TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]
          [ WITH TIME ZONE ]

<time precision> ::= <time fractional seconds precision>

<time fractional seconds precision> ::= <unsigned integer>

<timestamp precision> ::= <time fractional seconds precision>

<interval type> ::= INTERVAL <interval qualifier>

<interval qualifier> ::=
      <start field> TO <end field>
    | <single datetime field>

<start field> ::=
    <non-second datetime field>
        [ <left paren> <interval leading field precision> <right paren> ]

<non-second datetime field> ::= YEAR | MONTH | DAY | HOUR
    | MINUTE

<interval leading field precision> ::= <unsigned integer>

<end field> ::=
      <non-second datetime field>
    | SECOND [ <left paren> <interval fractional seconds precision> <right paren> ]
```

```
<interval fractional seconds precision> ::= <unsigned integer>

<single datetime field> ::=
      <non-second datetime field>
          [ <left paren> <interval leading field precision> <right paren> ]
    | SECOND [ <left paren> <interval leading field precision>
          [ <comma> <interval fractional seconds precision> ] <right paren> ]

<domain name> ::= <qualified name>

<qualified name> ::=
    [ <schema name> <period> ] <qualified identifier>

<default clause> ::=
      DEFAULT <default option>

<default option> ::=
      <literal>
    | <datetime value function>
    | USER
    | CURRENT_USER
    | SESSION_USER
    | SYSTEM_USER
    | NULL

<literal> ::=
      <signed numeric literal>
    | <general literal>

<signed numeric literal> ::=
    [ <sign> ] <unsigned numeric literal>

<general literal> ::=
      <character string literal>
    | <national character string literal>
    | <bit string literal>
    | <hex string literal>
    | <datetime literal>
    | <interval literal>

<datetime literal> ::=
      <date literal>
    | <time literal>
    | <timestamp literal>

<date literal> ::=
    DATE <date string>

<time literal> ::=
    TIME <time string>

<timestamp literal> ::=
    TIMESTAMP <timestamp string>

<interval literal> ::=
```

```
    INTERVAL [ <sign> ] <interval string> <interval qualifier>

<datetime value function> ::=
      <current date value function>
    | <current time value function>
    | <current timestamp value function>

<current date value function> ::= CURRENT_DATE

<current time value function> ::=
      CURRENT_TIME [ <left paren> <time precision> <right paren> ]

<current timestamp value function> ::=
      CURRENT_TIMESTAMP [ <left paren> <timestamp precision> <right paren> ]

<column constraint definition> ::=
    [ <constraint name definition> ]
    <column constraint>
      [ <constraint attributes> ]

<constraint name definition> ::= CONSTRAINT <constraint name>

<constraint name> ::= <qualified name>

<column constraint> ::=
      NOT NULL
    | <unique specification>
    | <references specification>
    | <check constraint definition>

<unique specification> ::=
    UNIQUE | PRIMARY KEY

<references specification> ::=
    REFERENCES <referenced table and columns>
      [ MATCH <match type> ]
      [ <referential triggered action> ]

<referenced table and columns> ::=
     <table name> [ <left paren> <reference column list> <right paren> ]

<table name> ::=
      <qualified name>
    | <qualified local table name>

<reference column list> ::= <column name list>

<column name list> ::=
    <column name> [ ( <comma> <column name> )... ]

<match type> ::=
      FULL
    | PARTIAL

<referential triggered action> ::=
```

```
      <update rule> [ <delete rule> ]
    | <delete rule> [ <update rule> ]

<update rule> ::= ON UPDATE <referential action>

<referential action> ::=
      CASCADE
    | SET NULL
    | SET DEFAULT
    | NO ACTION

<delete rule> ::= ON DELETE <referential action>

<check constraint definition> ::=
    CHECK
        <left paren> <search condition> <right paren>

<search condition> ::=
      <boolean term>
    | <search condition> OR <boolean term>

<boolean term> ::=
      <boolean factor>
    | <boolean term> AND <boolean factor>

<boolean factor> ::=
    [ NOT ] <boolean test>

<boolean test> ::=
    <boolean primary> [ IS [ NOT ]
          <truth value> ]

<boolean primary> ::=
      <predicate>
    | <left paren> <search condition> <right paren>

<predicate> ::=
      <comparison predicate>
    | <between predicate>
    | <in predicate>
    | <like predicate>
    | <null predicate>
    | <quantified comparison predicate>
    | <exists predicate>
    | <unique predicate>
    | <match predicate>
    | <overlaps predicate>

<comparison predicate> ::=
    <row value constructor> <comp op>
        <row value constructor>

<row value constructor> ::=
       <row value constructor element>
    | <left paren> <row value constructor list> <right paren>
```

```
    | <row subquery>

<row value constructor element> ::=
      <value expression>
    | <null specification>
    | <default specification>

<value expression> ::=
      <numeric value expression>
    | <string value expression>
    | <datetime value expression>
    | <interval value expression>

<numeric value expression> ::=
      <term>
    | <numeric value expression> <plus sign> <term>
    | <numeric value expression> <minus sign> <term>

<term> ::=
      <factor>
    | <term> <asterisk> <factor>
    | <term> <solidus> <factor>

<factor> ::=
    [ <sign> ] <numeric primary>

<numeric primary> ::=
      <value expression primary>
    | <numeric value function>

<value expression primary> ::=
      <unsigned value specification>
    | <column reference>
    | <set function specification>
    | <scalar subquery>
    | <case expression>
    | <left paren> <value expression> <right paren>
    | <cast specification>

<unsigned value specification> ::=
      <unsigned literal>
    | <general value specification>

<unsigned literal> ::=
      <unsigned numeric literal>
    | <general literal>

<general value specification> ::=
      <parameter specification>
    | <dynamic parameter specification>
    | <variable specification>
    | USER
    | CURRENT_USER
    | SESSION_USER
    | SYSTEM_USER
```

```
    | VALUE

<parameter specification> ::=
    <parameter name> [ <indicator parameter> ]

<parameter name> ::= <colon> <identifier>

<indicator parameter> ::=
    [ INDICATOR ] <parameter name>

<dynamic parameter specification> ::= <question mark>

<variable specification> ::=
    <embedded variable name> [ <indicator variable> ]

<embedded variable name> ::=
    <colon><host identifier>

<host identifier> ::=
      <Ada host identifier>
    | <C host identifier>
    | <COBOL host identifier>
    | <Fortran host identifier>
    | <MUMPS host identifier>
    | <Pascal host identifier>
    | <PL/I host identifier>

<Ada host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<C host identifier> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<COBOL host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<Fortran host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<MUMPS host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<Pascal host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<PL/I host identifier> ::= !! <EMPHASIS>(See the Syntax Rules.)

<indicator variable> ::=
    [ INDICATOR ] <embedded variable name>

<column reference> ::= [ <qualifier> <period> ] <column name>

<qualifier> ::=
      <table name>
    | <correlation name>

<correlation name> ::= <identifier>

<set function specification> ::=
      COUNT <left paren> <asterisk> <right paren>
```

```
    | <general set function>

<general set function> ::=
      <set function type>
          <left paren> [ <set quantifier> ] <value expression> <right paren>

<set function type> ::=
    AVG | MAX | MIN | SUM | COUNT

<set quantifier> ::= DISTINCT | ALL

<scalar subquery> ::= <subquery>

<subquery> ::= <left paren> <query expression> <right paren>

<query expression> ::=
      <non-join query expression>
    | <joined table>

<non-join query expression> ::=
      <non-join query term>
    | <query expression> UNION  [ ALL ]
          [ <corresponding spec> ] <query term>
    | <query expression> EXCEPT [ ALL ]
          [ <corresponding spec> ] <query term>

<non-join query term> ::=
      <non-join query primary>
    | <query term> INTERSECT [ ALL ]
          [ <corresponding spec> ] <query primary>

<non-join query primary> ::=
      <simple table>
    | <left paren> <non-join query expression> <right paren>

<simple table> ::=
      <query specification>
    | <table value constructor>
    | <explicit table>

<query specification> ::=
    SELECT [ <set quantifier> ] <select list> <table expression>

<select list> ::=
      <asterisk>
    | <select sublist> [ ( <comma> <select sublist> )... ]

<select sublist> ::=
      <derived column>
    | <qualifier> <period> <asterisk>

<derived column> ::= <value expression> [ <as clause> ]

<as clause> ::= [ AS ] <column name>
```

```
<table expression> ::=
    <from clause>
    [ <where clause> ]
    [ <group by clause> ]
    [ <having clause> ]

<from clause> ::= FROM <table reference>
    [ ( <comma> <table reference> )... ]

<table reference> ::=
      <table name> [ [ AS ] <correlation name>
          [ <left paren> <derived column list> <right paren> ] ]
    | <derived table> [ AS ] <correlation name>
          [ <left paren> <derived column list> <right paren> ]
    | <joined table>

<derived column list> ::= <column name list>

<derived table> ::= <table subquery>

<table subquery> ::= <subquery>

<joined table> ::=
      <cross join>
    | <qualified join>
    | <left paren> <joined table> <right paren>

<cross join> ::=
    <table reference> CROSS JOIN <table reference>

<qualified join> ::=
    <table reference> [ NATURAL ] [ <join type> ] JOIN
      <table reference> [ <join specification> ]

<join type> ::=
      INNER
    | <outer join type> [ OUTER ]
    | UNION

<outer join type> ::=
      LEFT
    | RIGHT
    | FULL

<join specification> ::=
      <join condition>
    | <named columns join>

<join condition> ::= ON <search condition>

<named columns join> ::=
    USING <left paren> <join column list> <right paren>

<join column list> ::= <column name list>
```

```
<where clause> ::= WHERE <search condition>

<group by clause> ::=
    GROUP BY <grouping column reference list>

<grouping column reference list> ::=
    <grouping column reference>
        [ ( <comma> <grouping column reference> )... ]

<grouping column reference> ::=
    <column reference> [ <collate clause> ]

<collate clause> ::= COLLATE <collation name>

<collation name> ::= <qualified name>

<having clause> ::= HAVING <search condition>

<table value constructor> ::=
    VALUES <table value constructor list>

<table value constructor list> ::=
    <row value constructor> [ ( <comma> <row value constructor> )... ]

<explicit table> ::= TABLE <table name>

<query term> ::=
      <non-join query term>
    | <joined table>

<corresponding spec> ::=
    CORRESPONDING [ BY <left paren> <corresponding column list> <right paren> ]

<corresponding column list> ::= <column name list>

<query primary> ::=
      <non-join query primary>
    | <joined table>

<case expression> ::=
      <case abbreviation>
    | <case specification>

<case abbreviation> ::=
      NULLIF <left paren> <value expression> <comma>
            <value expression> <right paren>
    | COALESCE <left paren> <value expression>
            ( <comma> <value expression> )... <right paren>

<case specification> ::=
      <simple case>
    | <searched case>

<simple case> ::=
    CASE <case operand>
```

```
      <simple when clause>...
      [ <else clause> ]
    END

<case operand> ::= <value expression>

<simple when clause> ::= WHEN <when operand> THEN <result>

<when operand> ::= <value expression>

<result> ::= <result expression> | NULL

<result expression> ::= <value expression>

<else clause> ::= ELSE <result>

<searched case> ::=
    CASE
      <searched when clause>...
      [ <else clause> ]
    END

<searched when clause> ::= WHEN <search condition> THEN <result>

<cast specification> ::=
    CAST <left paren> <cast operand> AS
        <cast target> <right paren>

<cast operand> ::=
      <value expression>
    | NULL

<cast target> ::=
      <domain name>
    | <data type>

<numeric value function> ::=
      <position expression>
    | <extract expression>
    | <length expression>

<position expression> ::=
    POSITION <left paren> <character value expression>
        IN <character value expression> <right paren>

<character value expression> ::=
      <concatenation>
    | <character factor>

<concatenation> ::=
    <character value expression> <concatenation operator>
        <character factor>

<character factor> ::=
    <character primary> [ <collate clause> ]
```

```
<character primary> ::=
      <value expression primary>
    | <string value function>

<string value function> ::=
      <character value function>
    | <bit value function>

<character value function> ::=
      <character substring function>
    | <fold>
    | <form-of-use conversion>
    | <character translation>
    | <trim function>

<character substring function> ::=
    SUBSTRING <left paren> <character value expression> FROM <start position>
                [ FOR <string length> ] <right paren>

<start position> ::= <numeric value expression>

<string length> ::= <numeric value expression>

<fold> ::= ( UPPER | LOWER )
     <left paren> <character value expression> <right paren>

<form-of-use conversion> ::=
    CONVERT <left paren> <character value expression>
        USING <form-of-use conversion name> <right paren>

<form-of-use conversion name> ::= <qualified name>

<character translation> ::=
    TRANSLATE <left paren> <character value expression>
        USING <translation name> <right paren>

<translation name> ::= <qualified name>

<trim function> ::=
    TRIM <left paren> <trim operands> <right paren>

<trim operands> ::=
    [ [ <trim specification> ] [ <trim character> ] FROM ] <trim source>

<trim specification> ::=
      LEADING
    | TRAILING
    | BOTH

<trim character> ::= <character value expression>

<trim source> ::= <character value expression>

<bit value function> ::=
```

```
    <bit substring function>

<bit substring function> ::=
    SUBSTRING <left paren> <bit value expression> FROM <start position>
        [ FOR <string length> ] <right paren>

<bit value expression> ::=
      <bit concatenation>
    | <bit factor>

<bit concatenation> ::=
    <bit value expression> <concatenation operator> <bit factor>

<bit factor> ::= <bit primary>

<bit primary> ::=
      <value expression primary>
    | <string value function>

<extract expression> ::=
    EXTRACT <left paren> <extract field>
        FROM <extract source> <right paren>

<extract field> ::=
      <datetime field>
    | <time zone field>

<datetime field> ::=
      <non-second datetime field>
    | SECOND

<time zone field> ::=
      TIMEZONE_HOUR
    | TIMEZONE_MINUTE

<extract source> ::=
      <datetime value expression>
    | <interval value expression>

<datetime value expression> ::=
      <datetime term>
    | <interval value expression> <plus sign> <datetime term>
    | <datetime value expression> <plus sign> <interval term>
    | <datetime value expression> <minus sign> <interval term>

<interval term> ::=
      <interval factor>
    | <interval term 2> <asterisk> <factor>
    | <interval term 2> <solidus> <factor>
    | <term> <asterisk> <interval factor>

<interval factor> ::=
    [ <sign> ] <interval primary>

<interval primary> ::=
```

```
      <value expression primary> [ <interval qualifier> ]

<interval term 2> ::= <interval term>

<interval value expression> ::=
      <interval term>
    | <interval value expression 1> <plus sign> <interval term 1>
    | <interval value expression 1> <minus sign> <interval term 1>
    | <left paren> <datetime value expression> <minus sign>
          <datetime term> <right paren> <interval qualifier>

<interval value expression 1> ::= <interval value expression>

<interval term 1> ::= <interval term>

<datetime term> ::=
      <datetime factor>

<datetime factor> ::=
      <datetime primary> [ <time zone> ]

<datetime primary> ::=
      <value expression primary>
    | <datetime value function>

<time zone> ::=
    AT <time zone specifier>

<time zone specifier> ::=
      LOCAL
    | TIME ZONE <interval value expression>

<length expression> ::=
      <char length expression>
    | <octet length expression>
    | <bit length expression>

<char length expression> ::=
    ( CHAR_LENGTH | CHARACTER_LENGTH )
        <left paren> <string value expression> <right paren>

<string value expression> ::=
      <character value expression>
    | <bit value expression>

<octet length expression> ::=
    OCTET_LENGTH <left paren> <string value expression> <right paren>

<bit length expression> ::=
    BIT_LENGTH <left paren> <string value expression> <right paren>

<null specification> ::=
    NULL

<default specification> ::=
```

```
    DEFAULT

<row value constructor list> ::=
    <row value constructor element>
        [ ( <comma> <row value constructor element> )... ]

<row subquery> ::= <subquery>

<comp op> ::=
      <equals operator>
    | <not equals operator>
    | <less than operator>
    | <greater than operator>
    | <less than or equals operator>
    | <greater than or equals operator>

<between predicate> ::=
    <row value constructor> [ NOT ] BETWEEN
      <row value constructor> AND <row value constructor>

<in predicate> ::=
    <row value constructor>
      [ NOT ] IN <in predicate value>

<in predicate value> ::=
      <table subquery>
    | <left paren> <in value list> <right paren>

<in value list> ::=
    <value expression> ( <comma> <value expression> )...

<like predicate> ::=
    <match value> [ NOT ] LIKE <pattern>
      [ ESCAPE <escape character> ]

<match value> ::= <character value expression>

<pattern> ::= <character value expression>

<escape character> ::= <character value expression>

<null predicate> ::= <row value constructor>
    IS [ NOT ] NULL

<quantified comparison predicate> ::=
    <row value constructor> <comp op> <quantifier> <table subquery>

<quantifier> ::= <all> | <some>

<all> ::= ALL

<some> ::= SOME | ANY

<exists predicate> ::= EXISTS <table subquery>
```

```
<unique predicate> ::= UNIQUE <table subquery>

<match predicate> ::=
    <row value constructor> MATCH [ UNIQUE ]
        [ PARTIAL | FULL ] <table subquery>

<overlaps predicate> ::=
    <row value constructor 1> OVERLAPS <row value constructor 2>

<row value constructor 1> ::= <row value constructor>

<row value constructor 2> ::= <row value constructor>

<truth value> ::=
      TRUE
    | FALSE
    | UNKNOWN

<constraint attributes> ::=
      <constraint check time> [ [ NOT ] DEFERRABLE ]
    | [ NOT ] DEFERRABLE [ <constraint check time> ]

<constraint check time> ::=
      INITIALLY DEFERRED
    | INITIALLY IMMEDIATE

<table constraint definition> ::=
    [ <constraint name definition> ]
    <table constraint> [ <constraint attributes> ]

<table constraint> ::=
      <unique constraint definition>
    | <referential constraint definition>
    | <check constraint definition>

<unique constraint definition> ::=
            <unique specification> even in SQL3)
    <unique specification>
      <left paren> <unique column list> <right paren>

<unique column list> ::= <column name list>

<referential constraint definition> ::=
    FOREIGN KEY
        <left paren> <referencing columns> <right paren>
      <references specification>

<referencing columns> ::=
    <reference column list>

<module contents> ::=
      <declare cursor>
    | <dynamic declare cursor>
    | <procedure>
```

```
<declare cursor> ::=
    DECLARE <cursor name> [ INSENSITIVE ] [ SCROLL ] CURSOR
      FOR <cursor specification>

<cursor name> ::= <identifier>

<cursor specification> ::=
    <query expression> [ <order by clause> ]
      [ <updatability clause> ]

<order by clause> ::=
    ORDER BY <sort specification list>

<sort specification list> ::=
    <sort specification> [ ( <comma> <sort specification> )... ]

<sort specification> ::=
    <sort key> [ <collate clause> ] [ <ordering specification> ]

<sort key> ::=
      <column name>
    | <unsigned integer>

<ordering specification> ::= ASC | DESC

<updatability clause> ::=
    FOR
        ( READ ONLY |
          UPDATE [ OF <column name list> ] )

<dynamic declare cursor> ::=
    DECLARE <cursor name> [ INSENSITIVE ] [ SCROLL ] CURSOR
        FOR <statement name>

<statement name> ::= <identifier>

<procedure> ::=
    PROCEDURE <procedure name>
        <parameter declaration list> <semicolon>
      <SQL procedure statement> <semicolon>

<procedure name> ::= <identifier>

<parameter declaration list> ::=
      <left paren> <parameter declaration>
          [ ( <comma> <parameter declaration> )... ] <right paren>
    | <parameter declaration>...

<parameter declaration> ::=
      <parameter name> <data type>
    | <status parameter>

<status parameter> ::=
    SQLCODE | SQLSTATE
```

```
<SQL procedure statement> ::=
      <SQL schema statement>
    | <SQL data statement>
    | <SQL transaction statement>
    | <SQL connection statement>
    | <SQL session statement>
    | <SQL dynamic statement>
    | <SQL diagnostics statement>

<SQL schema statement> ::=
      <SQL schema definition statement>
    | <SQL schema manipulation statement>

<SQL schema definition statement> ::=
      <schema definition>
    | <table definition>
    | <view definition>
    | <grant statement>
    | <domain definition>
    | <character set definition>
    | <collation definition>
    | <translation definition>
    | <assertion definition>

<schema definition> ::=
    CREATE SCHEMA <schema name clause>
      [ <schema character set specification> ]
      [ <schema element>... ]

<schema name clause> ::=
      <schema name>
    | AUTHORIZATION <schema authorization identifier>
    | <schema name> AUTHORIZATION
          <schema authorization identifier>

<schema authorization identifier> ::=
    <authorization identifier>

<schema character set specification> ::=
    DEFAULT CHARACTER
        SET <character set specification>

<schema element> ::=
      <domain definition>
    | <table definition>
    | <view definition>
    | <grant statement>
    | <assertion definition>
    | <character set definition>
    | <collation definition>
    | <translation definition>

<domain definition> ::=
    CREATE DOMAIN <domain name>
        [ AS ] <data type>
```

```
      [ <default clause> ]
      [ <domain constraint>... ]
      [ <collate clause> ]

<domain constraint> ::=
    [ <constraint name definition> ]
    <check constraint definition> [ <constraint attributes> ]

<table definition> ::=
    CREATE [ ( GLOBAL | LOCAL ) TEMPORARY ] TABLE
        <table name>
      <table element list>
      [ ON COMMIT ( DELETE | PRESERVE ) ROWS ]

<view definition> ::=
    CREATE VIEW <table name> [ <left paren> <view column list>
                              <right paren> ]
      AS <query expression>
      [ WITH [ <levels clause> ] CHECK OPTION ]

<view column list> ::= <column name list>

<levels clause> ::=
    CASCADED | LOCAL

<grant statement> ::=
   GRANT <privileges> ON <object name>
     TO <grantee> [ ( <comma> <grantee> )... ]
       [ WITH GRANT OPTION ]

<privileges> ::=
      ALL PRIVILEGES
    | <action list>

<action list> ::= <action> [ ( <comma> <action> )... ]

<action> ::=
      SELECT
    | DELETE
    | INSERT [ <left paren> <privilege column list> <right paren> ]
    | UPDATE [ <left paren> <privilege column list> <right paren> ]
    | REFERENCES [ <left paren> <privilege column list> <right paren> ]
    | USAGE

<privilege column list> ::= <column name list>

<object name> ::=
      [ TABLE ] <table name>
    | DOMAIN <domain name>
    | COLLATION <collation name>
    | CHARACTER SET <character set name>
    | TRANSLATION <translation name>

<grantee> ::=
      PUBLIC
```

```
    | <authorization identifier>

<assertion definition> ::=
    CREATE ASSERTION <constraint name> <assertion check>
      [ <constraint attributes> ]

<assertion check> ::=
    CHECK
        <left paren> <search condition> <right paren>

<character set definition> ::=
    CREATE CHARACTER SET <character set name>
        [ AS ]
      <character set source>
      [ <collate clause> | <limited collation definition> ]

<character set source> ::=
      GET <existing character set name>

<existing character set name> ::=
      <standard character repertoire name>
    | <implementation-defined character repertoire name>
    | <schema character set name>

<schema character set name> ::= <character set name>

<limited collation definition> ::=
    COLLATION FROM <collation source>

<collation source> ::=
      <collating sequence definition>
    | <translation collation>

<collating sequence definition> ::=
      <external collation>
    | <schema collation name>
    | DESC <left paren> <collation name> <right paren>
    | DEFAULT

<external collation> ::=
    EXTERNAL <left paren> <quote> <external collation name> <quote> <right paren>

<external collation name> ::=
      <standard collation name>
    | <implementation-defined collation name>

<standard collation name> ::= <collation name>

<implementation-defined collation name> ::= <collation name>

<schema collation name> ::= <collation name>

<translation collation> ::=
    TRANSLATION <translation name>
        [ THEN COLLATION <collation name> ]
```

```
<collation definition> ::=
    CREATE COLLATION <collation name> FOR
        <character set specification>
      FROM <collation source>
        [ <pad attribute> ]

<pad attribute> ::=
      NO PAD
    | PAD SPACE

<translation definition> ::=
    CREATE TRANSLATION <translation name>
      FOR <source character set specification>
        TO <target character set specification>
      FROM <translation source>

<source character set specification> ::= <character set specification>

<target character set specification> ::= <character set specification>

<translation source> ::=
      <translation specification>

<translation specification> ::=
      <external translation>
    | IDENTITY
    | <schema translation name>

<external translation> ::=
    EXTERNAL <left paren> <quote> <external translation name> <quote> <right paren>

<external translation name> ::=
      <standard translation name>
    | <implementation-defined translation name>

<standard translation name> ::= <translation name>

<implementation-defined translation name> ::= <translation name>

<schema translation name> ::= <translation name>

<SQL schema manipulation statement> ::=
      <drop schema statement>
    | <alter table statement>
    | <drop table statement>
    | <drop view statement>
    | <revoke statement>
    | <alter domain statement>
    | <drop domain statement>
    | <drop character set statement>
    | <drop collation statement>
    | <drop translation statement>
    | <drop assertion statement>
```

```
<drop schema statement> ::=
    DROP SCHEMA <schema name> <drop behavior>

<drop behavior> ::= CASCADE | RESTRICT

<alter table statement> ::=
    ALTER TABLE <table name> <alter table action>

<alter table action> ::=
      <add column definition>
    | <alter column definition>
    | <drop column definition>
    | <add table constraint definition>
    | <drop table constraint definition>

<add column definition> ::=
    ADD [ COLUMN ] <column definition>

<alter column definition> ::=
    ALTER [ COLUMN ] <column name> <alter column action>

<alter column action> ::=
      <set column default clause>
    | <drop column default clause>

<set column default clause> ::=
    SET <default clause>

<drop column default clause> ::=
    DROP DEFAULT

<drop column definition> ::=
    DROP [ COLUMN ] <column name> <drop behavior>

<add table constraint definition> ::=
    ADD <table constraint definition>

<drop table constraint definition> ::=
    DROP CONSTRAINT <constraint name> <drop behavior>

<drop table statement> ::=
    DROP TABLE <table name> <drop behavior>

<drop view statement> ::=
    DROP VIEW <table name> <drop behavior>

<revoke statement> ::=
    REVOKE [ GRANT OPTION FOR ]
        <privileges>
        ON <object name>
      FROM <grantee> [ ( <comma> <grantee> )... ] <drop behavior>

<alter domain statement> ::=
    ALTER DOMAIN <domain name> <alter domain action>
```

```
<alter domain action> ::=
      <set domain default clause>
    | <drop domain default clause>
    | <add domain constraint definition>
    | <drop domain constraint definition>

<set domain default clause> ::= SET <default clause>

<drop domain default clause> ::= DROP DEFAULT

<add domain constraint definition> ::=
    ADD <domain constraint>

<drop domain constraint definition> ::=
    DROP CONSTRAINT <constraint name>

<drop domain statement> ::=
    DROP DOMAIN <domain name> <drop behavior>

<drop character set statement> ::=
    DROP CHARACTER SET <character set name>

<drop collation statement> ::=
    DROP COLLATION <collation name>

<drop translation statement> ::=
    DROP TRANSLATION <translation name>

<drop assertion statement> ::=
    DROP ASSERTION <constraint name>

<SQL data statement> ::=
      <open statement>
    | <fetch statement>
    | <close statement>
    | <select statement: single row>
    | <SQL data change statement>

<open statement> ::=
    OPEN <cursor name>

<fetch statement> ::=
    FETCH [ [ <fetch orientation> ] FROM ]
      <cursor name> INTO <fetch target list>

<fetch orientation> ::=
      NEXT
    | PRIOR
    | FIRST
    | LAST
    | ( ABSOLUTE | RELATIVE ) <simple value specification>

<simple value specification> ::=
      <parameter name>
    | <embedded variable name>
```

```
    | <literal>

<fetch target list> ::=
    <target specification> [ ( <comma> <target specification> )... ]

<target specification> ::=
      <parameter specification>
    | <variable specification>

<close statement> ::=
    CLOSE <cursor name>

<select statement: single row> ::=
    SELECT [ <set quantifier> ] <select list>
      INTO <select target list>
        <table expression>

<select target list> ::=
    <target specification> [ ( <comma> <target specification> )... ]

<SQL data change statement> ::=
      <delete statement: positioned>
    | <delete statement: searched>
    | <insert statement>
    | <update statement: positioned>
    | <update statement: searched>

<delete statement: positioned> ::=
    DELETE FROM <table name>
      WHERE CURRENT OF <cursor name>

<delete statement: searched> ::=
    DELETE FROM <table name>
      [ WHERE <search condition> ]

<insert statement> ::=
    INSERT INTO <table name>
      <insert columns and source>

<insert columns and source> ::=
      [ <left paren> <insert column list> <right paren> ]
            <query expression>
    | DEFAULT VALUES

<insert column list> ::= <column name list>

<update statement: positioned> ::=
    UPDATE <table name>
      SET <set clause list>
        WHERE CURRENT OF <cursor name>

<set clause list> ::=
    <set clause> [ ( <comma> <set clause> )... ]

<set clause> ::=
```

```
    <object column> <equals operator> <update source>

<object column> ::= <column name>

<update source> ::=
      <value expression>
    | <null specification>
    | DEFAULT

<update statement: searched> ::=
    UPDATE <table name>
      SET <set clause list>
      [ WHERE <search condition> ]

<SQL transaction statement> ::=
      <set transaction statement>
    | <set constraints mode statement>
    | <commit statement>
    | <rollback statement>

<set transaction statement> ::=
    SET TRANSACTION <transaction mode>
        [ ( <comma> <transaction mode> )... ]

<transaction mode> ::=
      <isolation level>
    | <transaction access mode>
    | <diagnostics size>

<isolation level> ::=
    ISOLATION LEVEL <level of isolation>

<level of isolation> ::=
      READ UNCOMMITTED
    | READ COMMITTED
    | REPEATABLE READ
    | SERIALIZABLE

<transaction access mode> ::=
      READ ONLY
    | READ WRITE

<diagnostics size> ::=
    DIAGNOSTICS SIZE <number of conditions>

<number of conditions> ::= <simple value specification>

<set constraints mode statement> ::=
    SET CONSTRAINTS <constraint name list>
        ( DEFERRED | IMMEDIATE )

<constraint name list> ::=
      ALL
    | <constraint name> [ ( <comma> <constraint name> )... ]
```

```
<commit statement> ::=
    COMMIT [ WORK ]

<rollback statement> ::=
    ROLLBACK [ WORK ]

<SQL connection statement> ::=
      <connect statement>
    | <set connection statement>
    | <disconnect statement>

<connect statement> ::=
    CONNECT TO <connection target>

<connection target> ::=
      <SQL-server name>
        [ AS <connection name> ]
          correspondence with Tony Gordon)
        [ USER <user name> ]
    | DEFAULT

<SQL-server name> ::= <simple value specification>

<connection name> ::= <simple value specification>

<user name> ::= <simple value specification>

<set connection statement> ::=
    SET CONNECTION <connection object>

<connection object> ::=
      DEFAULT
    | <connection name>

<disconnect statement> ::=
    DISCONNECT <disconnect object>

<disconnect object> ::=
      <connection object>
    | ALL
    | CURRENT

<SQL session statement> ::=
      <set catalog statement>
    | <set schema statement>
    | <set names statement>
    | <set session authorization identifier statement>
    | <set local time zone statement>

<set catalog statement> ::=
    SET CATALOG <value specification>

<value specification> ::=
      <literal>
    | <general value specification>
```

```
<set schema statement> ::=
    SET SCHEMA <value specification>

<set names statement> ::=
    SET NAMES <value specification>

<set session authorization identifier statement> ::=
    SET SESSION AUTHORIZATION
        <value specification>

<set local time zone statement> ::=
    SET TIME ZONE
        <set time zone value>

<set time zone value> ::=
      <interval value expression>
    | LOCAL

<SQL dynamic statement> ::=
      <system descriptor statement>
    | <prepare statement>
    | <deallocate prepared statement>
    | <describe statement>
    | <execute statement>
    | <execute immediate statement>
    | <SQL dynamic data statement>

<system descriptor statement> ::=
      <allocate descriptor statement>
    | <deallocate descriptor statement>
    | <set descriptor statement>
    | <get descriptor statement>

<allocate descriptor statement> ::=
    ALLOCATE DESCRIPTOR <descriptor name>
       [ WITH MAX <occurrences> ]

<descriptor name> ::=
    [ <scope option> ] <simple value specification>

<scope option> ::=
      GLOBAL
    | LOCAL

<occurrences> ::= <simple value specification>

<deallocate descriptor statement> ::=
    DEALLOCATE DESCRIPTOR <descriptor name>

<set descriptor statement> ::=
    SET DESCRIPTOR <descriptor name>
        <set descriptor information>

<set descriptor information> ::=
```

```
      <set count>
    | VALUE <item number>
        <set item information> [ ( <comma> <set item information> )... ]

<set count> ::=
    COUNT <equals operator> <simple value specification 1>

<simple value specification 1> ::= <simple value specification>

<item number> ::= <simple value specification>

<set item information> ::=
    <descriptor item name> <equals operator> <simple value specification 2>

<descriptor item name> ::=
      TYPE
    | LENGTH
    | OCTET_LENGTH
    | RETURNED_LENGTH
    | RETURNED_OCTET_LENGTH
    | PRECISION
    | SCALE
    | DATETIME_INTERVAL_CODE
    | DATETIME_INTERVAL_PRECISION
    | NULLABLE
    | INDICATOR
    | DATA
    | NAME
    | UNNAMED
    | COLLATION_CATALOG
    | COLLATION_SCHEMA
    | COLLATION_NAME
    | CHARACTER_SET_CATALOG
    | CHARACTER_SET_SCHEMA
    | CHARACTER_SET_NAME

<simple value specification 2> ::= <simple value specification>

<item number> ::= <simple value specification>

<get descriptor statement> ::=
    GET DESCRIPTOR <descriptor name> <get descriptor information>

<get descriptor information> ::=
      <get count>
    | VALUE <item number>
        <get item information> [ ( <comma> <get item information> )... ]

<get count> ::=
    <simple target specification 1> <equals operator>
         COUNT

<simple target specification 1> ::= <simple target specification>

<simple target specification> ::=
```

```
      <parameter name>
    | <embedded variable name>

<get item information> ::=
    <simple target specification 2> <equals operator> <descriptor item name>>

<simple target specification 2> ::= <simple target specification>

<prepare statement> ::=
    PREPARE <SQL statement name> FROM <SQL statement variable>

<SQL statement name> ::=
      <statement name>
    | <extended statement name>

<extended statement name> ::=
    [ <scope option> ] <simple value specification>

<SQL statement variable> ::= <simple value specification>

<deallocate prepared statement> ::=
    DEALLOCATE PREPARE <SQL statement name>

<describe statement> ::=
      <describe input statement>
    | <describe output statement>

<describe input statement> ::=
    DESCRIBE INPUT <SQL statement name> <using descriptor>

<using descriptor> ::=
    ( USING | INTO ) SQL DESCRIPTOR <descriptor name>

<describe output statement> ::=
    DESCRIBE [ OUTPUT ] <SQL statement name> <using descriptor>

<execute statement> ::=
    EXECUTE <SQL statement name>
      [ <result using clause> ]
      [ <parameter using clause> ]

<result using clause> ::= <using clause>

<using clause> ::=
      <using arguments>
    | <using descriptor>

<using arguments> ::=
    ( USING | INTO ) <argument> [ ( <comma> <argument> )... ]

<argument> ::= <target specification>

<parameter using clause> ::= <using clause>

<execute immediate statement> ::=
```

```
    EXECUTE IMMEDIATE <SQL statement variable>

<SQL dynamic data statement> ::=
      <allocate cursor statement>
    | <dynamic open statement>
    | <dynamic fetch statement>
    | <dynamic close statement>
    | <dynamic delete statement: positioned>
    | <dynamic update statement: positioned>

<allocate cursor statement> ::=
    ALLOCATE <extended cursor name> [ INSENSITIVE ]
        [ SCROLL ] CURSOR
      FOR <extended statement name>

<extended cursor name> ::=
    [ <scope option> ] <simple value specification>

<dynamic open statement> ::=
    OPEN <dynamic cursor name> [ <using clause> ]

<dynamic cursor name> ::=
      <cursor name>
    | <extended cursor name>

<dynamic fetch statement> ::=
    FETCH [ [ <fetch orientation> ] FROM ] <dynamic cursor name>
        <using clause>

<dynamic close statement> ::=
    CLOSE <dynamic cursor name>

<dynamic delete statement: positioned> ::=
    DELETE FROM <table name>
      WHERE CURRENT OF
          <dynamic cursor name>

<dynamic update statement: positioned> ::=
    UPDATE <table name>
      SET <set clause>
          [ ( <comma> <set clause> )... ]
        WHERE CURRENT OF
            <dynamic cursor name>

<SQL diagnostics statement> ::=
    <get diagnostics statement>

<get diagnostics statement> ::=
    GET DIAGNOSTICS <sql diagnostics information>

<sql diagnostics information> ::=
      <statement information>
    | <condition information>

<statement information> ::=
```

```
    <statement information item> [ ( <comma> <statement information item> )... ]

<statement information item> ::=
    <simple target specification> <equals operator> <statement information item name>

<statement information item name> ::=
      NUMBER
    | MORE
    | COMMAND_FUNCTION
    | DYNAMIC_FUNCTION
    | ROW_COUNT

<condition information> ::=
    EXCEPTION <condition number>
      <condition information item> [ ( <comma> <condition information item> )... ]

<condition number> ::= <simple value specification>

<condition information item> ::=
    <simple target specification> <equals operator> <condition information item name>

<condition information item name> ::=
      CONDITION_NUMBER
    | RETURNED_SQLSTATE
    | CLASS_ORIGIN
    | SUBCLASS_ORIGIN
    | SERVER_NAME
    | CONNECTION_NAME
    | CONSTRAINT_CATALOG
    | CONSTRAINT_SCHEMA
    | CONSTRAINT_NAME
    | CATALOG_NAME
    | SCHEMA_NAME
    | TABLE_NAME
    | COLUMN_NAME
    | CURSOR_NAME
    | MESSAGE_TEXT
    | MESSAGE_LENGTH
    | MESSAGE_OCTET_LENGTH

<embedded SQL host program> ::=
      <embedded SQL Ada program>
    | <embedded SQL C program>
    | <embedded SQL COBOL program>
    | <embedded SQL Fortran program>
    | <embedded SQL MUMPS program>
    | <embedded SQL Pascal program>
    | <embedded SQL PL/I program>

<embedded SQL Ada program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL C program> ::=
      !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL COBOL program> ::= !! <EMPHASIS>(See the Syntax Rules.)
```

```
<embedded SQL Fortran program> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL MUMPS program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL Pascal program> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL PL/I program> ::= !! <EMPHASIS>(See the Syntax Rules.)

<embedded SQL declare section> ::=
      <embedded SQL begin declare>
        [ <embedded character set declaration> ]
        [ <host variable definition>... ]
      <embedded SQL end declare>
    | <embedded SQL MUMPS declare>

<embedded SQL begin declare> ::=
    <SQL prefix> BEGIN DECLARE SECTION
        [ <SQL terminator> ]

<SQL prefix> ::=
      EXEC SQL
    | <ampersand>SQL<left paren>

<SQL terminator> ::=
      END-EXEC
    | <semicolon>
    | <right paren>

<embedded character set declaration> ::=
    SQL NAMES ARE <character set specification>

<host variable definition> ::=
      <Ada variable definition>
    | <C variable definition>
    | <COBOL variable definition>
    | <Fortran variable definition>
    | <MUMPS variable definition>
    | <Pascal variable definition>
    | <PL/I variable definition>

<Ada variable definition> ::=
    <Ada host identifier> [ ( <comma> <Ada host identifier> )... ] :
    <Ada type specification> [ <Ada initial value> ]

<Ada type specification> ::=
      <Ada qualified type specification>
    | <Ada unqualified type specification>

<Ada qualified type specification> ::=
      SQL_STANDARD.CHAR [ CHARACTER SET
         [ IS ] <character set specification> ]
          <left paren> 1 <double period> <length> <right paren>
```

```
    | SQL_STANDARD.BIT
          <left paren> 1 <double period> <length> <right paren>
    | SQL_STANDARD.SMALLINT
    | SQL_STANDARD.INT
    | SQL_STANDARD.REAL
    | SQL_STANDARD.DOUBLE_PRECISION
    | SQL_STANDARD.SQLCODE_TYPE
    | SQL_STANDARD.SQLSTATE_TYPE
    | SQL_STANDARD.INDICATOR_TYPE

<Ada unqualified type specification> ::=
      CHAR
          <left paren> 1 <double period> <length> <right paren>
    | BIT
          <left paren> 1 <double period> <length> <right paren>
    | SMALLINT
    | INT
    | REAL
    | DOUBLE_PRECISION
    | SQLCODE_TYPE
    | SQLSTATE_TYPE
    | INDICATOR_TYPE

<Ada initial value> ::=
    <Ada assignment operator> <character representation>...

<Ada assignment operator> ::= <colon><equals operator>

<C variable definition> ::=
      [ <C storage class> ]
      [ <C class modifier> ]
      <C variable specification>
    <semicolon>

<C storage class> ::=
      auto
    | extern
    | static

<C class modifier> ::= const | volatile

<C variable specification> ::=
      <C numeric variable>
    | <C character variable>
    | <C derived variable>

<C numeric variable> ::=
    ( long | short | float | double )
      <C host identifier> [ <C initial value> ]
            [ ( <comma> <C host identifier> [ <C initial value> ] )... ]

<C initial value> ::=
    <equals operator> <character representation>...

<C character variable> ::=
```

```
    char [ CHARACTER SET
             [ IS ] <character set specification> ]
      <C host identifier>
        <C array specification> [ <C initial value> ]
        [ ( <comma> <C host identifier>
          <C array specification>
                 [ <C initial value> ] )... ]

<C array specification> ::=
    <left bracket> <length> <right bracket>

<C derived variable> ::=
      <C VARCHAR variable>
    | <C bit variable>

<C VARCHAR variable> ::=
    VARCHAR [ CHARACTER SET [ IS ]
        <character set specification> ]
        <C host identifier>
            <C array specification> [ <C initial value> ]
          [ ( <comma> <C host identifier>
              <C array specification>
                     [ <C initial value> ] )... ]

<C bit variable> ::=
    BIT <C host identifier>
        <C array specification> [ <C initial value> ]
      [ ( <comma> <C host identifier>
        <C array specification>
                   [ <C initial value> ] )... ]

<COBOL variable definition> ::=
    (01|77) <COBOL host identifier> <COBOL type specification>
      [ <character representation>... ] <period>

<COBOL type specification> ::=
      <COBOL character type>
    | <COBOL bit type>
    | <COBOL numeric type>
    | <COBOL integer type>

<COBOL character type> ::=
    [ CHARACTER SET [ IS ]
          <character set specification> ]
    ( PIC | PICTURE ) [ IS ] ( X [ <left paren> <length> <right paren> ] )...

<COBOL bit type> ::=
    ( PIC | PICTURE ) [ IS ]
        ( B [ <left paren> <length> <right paren> ] )...

<COBOL numeric type> ::=
    ( PIC | PICTURE ) [ IS ]
      S <COBOL nines specification>
    [ USAGE [ IS ] ] DISPLAY SIGN LEADING SEPARATE
```

```
<COBOL nines specification> ::=
      <COBOL nines> [ V [ <COBOL nines> ] ]
    | V <COBOL nines>

<COBOL nines> ::= ( 9 [ <left paren> <length> <right paren> ] )...

<COBOL integer type> ::=
      <COBOL computational integer>
    | <COBOL binary integer>

<COBOL computational integer> ::=
    ( PIC | PICTURE ) [ IS ] S<COBOL nines>
      [ USAGE [ IS ] ] ( COMP | COMPUTATIONAL )

<COBOL binary integer> ::=
    ( PIC | PICTURE ) [ IS ] S<COBOL nines>
      [ USAGE [ IS ] ] BINARY

<Fortran variable definition> ::=
    <Fortran type specification>
    <Fortran host identifier>
        [ ( <comma> <Fortran host identifier> )... ]

<Fortran type specification> ::=
      CHARACTER [ <asterisk> <length> ]
          [ CHARACTER SET [ IS ]
                <character set specification> ]
    | BIT [ <asterisk> <length> ]
    | INTEGER
    | REAL
    | DOUBLE PRECISION

<MUMPS variable definition> ::=
    ( <MUMPS numeric variable> | <MUMPS character variable> )
        <semicolon>

<MUMPS numeric variable> ::=
    <MUMPS type specification>
      <MUMPS host identifier> [ ( <comma> <MUMPS host identifier> )... ]

<MUMPS type specification> ::=
      INT
    | DEC
          [ <left paren> <precision> [ <comma> <scale> ] <right paren> ]
    | REAL

<MUMPS character variable> ::=
    VARCHAR <MUMPS host identifier> <MUMPS length specification>
      [ ( <comma> <MUMPS host identifier> <MUMPS length specification> )... ]

<MUMPS length specification> ::=
    <left paren> <length> <right paren>

<Pascal variable definition> ::=
    <Pascal host identifier> [ ( <comma> <Pascal host identifier> )... ] <colon>
```

```
      <Pascal type specification> <semicolon>

<Pascal type specification> ::=
      PACKED ARRAY
          <left bracket> 1 <double period> <length> <right bracket>
        OF CHAR
          [ CHARACTER SET [ IS ]
                <character set specification> ]
    | PACKED ARRAY
          <left bracket> 1 <double period> <length> <right bracket>
        OF BIT
    | INTEGER
    | REAL
    | CHAR [ CHARACTER SET
                           [ IS ] <character set specification> ]
    | BIT

<PL/I variable definition> ::=
    (DCL | DECLARE)
        (   <PL/I host identifier>
          | <left paren> <PL/I host identifier>
                [ ( <comma> <PL/I host identifier> )... ] <right paren> )
    <PL/I type specification>
    [ <character representation>... ] <semicolon>

<PL/I type specification> ::=
      ( CHAR | CHARACTER ) [ VARYING ]
          <left paren><length><right paren>
          [ CHARACTER SET
                [ IS ] <character set specification> ]
    | BIT [ VARYING ] <left paren><length><right paren>
    | <PL/I type fixed decimal> <left paren> <precision>
          [ <comma> <scale> ] <right paren>
    | <PL/I type fixed binary> [ <left paren> <precision> <right paren> ]
    | <PL/I type float binary> <left paren> <precision> <right paren>

<PL/I type fixed decimal> ::=
      ( DEC | DECIMAL ) FIXED
    | FIXED ( DEC | DECIMAL )

<PL/I type fixed binary> ::=
      ( BIN | BINARY ) FIXED
    | FIXED ( BIN | BINARY )

<PL/I type float binary> ::=
      ( BIN | BINARY ) FLOAT
    | FLOAT ( BIN | BINARY )

<embedded SQL end declare> ::=
    <SQL prefix> END DECLARE SECTION
        [ <SQL terminator> ]

<embedded SQL MUMPS declare> ::=
    <SQL prefix>
      BEGIN DECLARE SECTION
```

```
        [ <embedded character set declaration> ]
        [ <host variable definition>... ]
      END DECLARE SECTION
    <SQL terminator>

<embedded SQL statement> ::=
    <SQL prefix>
      <statement or declaration>
    [ <SQL terminator> ]

<statement or declaration> ::=
      <declare cursor>
    | <dynamic declare cursor>
    | <temporary table declaration>
    | <embedded exception declaration>
    | <SQL procedure statement>

<embedded exception declaration> ::=
    WHENEVER <condition> <condition action>

<condition> ::=
    SQLERROR | NOT FOUND

<condition action> ::=
    CONTINUE | <go to>

<go to> ::=
    ( GOTO | GO TO ) <goto target>

<goto target> ::=
      <host label identifier>
    | <unsigned integer>
    | <host PL/I label variable>

<host label identifier> ::= !!<EMPHASIS>(See the Syntax Rules.)

<host PL/I label variable> ::= !!<EMPHASIS>(See the Syntax Rules.)

<preparable statement> ::=
      <preparable SQL data statement>
    | <preparable SQL schema statement>
    | <preparable SQL transaction statement>
    | <preparable SQL session statement>
    | <preparable implementation-defined statement>

<preparable SQL data statement> ::=
      <delete statement: searched>
    | <dynamic single row select statement>
    | <insert statement>
    | <dynamic select statement>
    | <update statement: searched>
    | <preparable dynamic delete statement: positioned>
    | <preparable dynamic update statement: positioned>

<dynamic single row select statement> ::= <query specification>
```

```
<dynamic select statement> ::= <cursor specification>

<preparable dynamic delete statement: positioned> ::=
   DELETE [ FROM <table name> ]
      WHERE CURRENT OF <cursor name>

<preparable dynamic update statement: positioned> ::=
   UPDATE [ <table name> ]
      SET <set clause list>
      WHERE CURRENT OF <cursor name>

<preparable SQL schema statement> ::=
      <SQL schema statement>

<preparable SQL transaction statement> ::=
      <SQL transaction statement>

<preparable SQL session statement> ::=
      <SQL session statement>

<preparable implementation-defined statement> ::=
    !! <EMPHASIS>(See the Syntax Rules.)

<direct SQL statement> ::=
    <directly executable statement> <semicolon>

<directly executable statement> ::=
      <direct SQL data statement>
    | <SQL schema statement>
    | <SQL transaction statement>
    | <SQL connection statement>
    | <SQL session statement>
    | <direct implementation-defined statement>

<direct SQL data statement> ::=
      <delete statement: searched>
    | <direct select statement: multiple rows>
    | <insert statement>
    | <update statement: searched>
    | <temporary table declaration>

<direct select statement: multiple rows> ::=
    <query expression> [ <order by clause> ]

<direct implementation-defined statement> ::=
    !!<EMPHASIS>(See the Syntax Rules)

<SQL object identifier> ::=
    <SQL provenance> <SQL variant>

<SQL provenance> ::= <arc1> <arc2> <arc3>

<arc1> ::= iso | 1 | iso <left paren> 1 <right paren>
```

```
<arc2> ::= standard | 0 | standard <left paren> 0 <right paren>

<arc3> ::= 9075

<SQL variant> ::= <SQL edition> <SQL conformance>

<SQL edition> ::= <1987> | <1989> | <1992>

<1987> ::= 0 | edition1987 <left paren> 0 <right paren>

<1989> ::= <1989 base> <1989 package>

<1989 base> ::= 1 | edition1989 <left paren> 1 <right paren>

<1989 package> ::= <integrity no> | <integrity yes>

<integrity no> ::= 0 | IntegrityNo <left paren> 0 <right paren>

<integrity yes> ::= 1 | IntegrityYes <left paren> 1 <right paren>

<1992> ::= 2 | edition1992 <left paren> 2 <right paren>

<SQL conformance> ::= <low> | <intermediate> | <high>

<low> ::= 0 | Low <left paren> 0 <right paren>

<intermediate> ::= 1 | Intermediate <left paren> 1 <right paren>

<high> ::= 2 | High <left paren> 2 <right paren>
```

B Appendice B - Tutoriel SQL à l'usage des débutants

B.1 Didacticiel pour PostgreSQL

Le didacticiel SQL est inclus dans la distribution de PostgreSQL. Les scripts du didacticiel SQL se trouvent dans le répertoire src/tutorial

B.2 Pointeurs URL Internet

On peut trouver le didacticiel SQL pour débutants à

- `<http://w3.one.net/~jhoffman/sqltut.htm>`

Pour les commentaires ou suggestions, envoyer un courrier électronique à

- *jhoffman@one.net*

John Hoffman suggère de visiter les sites suivants :

- Référence SQL <http://www.contrib.andrew.cmu.edu/~shadow/sql.html>
- Demandez SQL Pro <http://www.inquiry.com/techtips/thesqlpro/>
- Sites utiles pour les BD relationnelles SQL Pro <http://www.inquiry.com/techtips/thesqlpro/usefulsites.html>
- Le site du programmeur <http://infoweb.magi.com/~steve/develop.html>

- Sites <http://info.itu.ch/special/wwwfiles> Go here and see file comp_db.html
- Ingrédients DB <http://www.compapp.dcu.ie/databases/f017.html>
- Web Authoring <http://www.stars.com/Tutorial/CGI/>
- Dictionnaire informatique <http://wfn-shop.princeton.edu/cgi-bin/foldoc>
- DBMS Lab/Links <http://www-ccs.cs.umass.edu/db.html>
- FAQ SQL <http://epoch.CS.Berkeley.EDU:8000/sequoia/dba/montage/FAQ> Et là, consultez le fichier SQL_TOC.html
- Bases de données SQL <http://chaos.mur.csu.edu.au/itc125/cgi/sqldb.html>
- Page de conception de la base de donnée RIT <http://www.it.rit.edu/~wjs/IT/199602/icsa720/icsa720postings.html>
- Site de renvoi sur les Bases de Données <http://www.pcslink.com/~ej/dbweb.html>
- Didacticiels de programmation sur le Web <http://www.eng.uc.edu/~jtilley/tutorial.html>
- Ressources de développement <http://www.ndev.com/ndc2/support/resources.htp>
- Liste de requête <http://ashok.pair.com/sql.htm>
- Diverses IMAGES SQL <http://jazz.external.hp.com/training/sqltables/main.html>
- Liste de ressources Internet <http://www.eit.com/web/netservices.html>

C Appendix C - Instructions pour une installation rapide de Linux

Si vous envisagez d'utiliser PostgreSQL sous Linux, et avez besoin d'aide dans l'installation de Linux, consultez les pointeurs donnés dans cette annexe. Ils couvrent les sujets suivants :

- Fonctionnalités saillantes de Linux - Pourquoi Linux constitue un meilleur serveur de données que Windows 95/NT
- Instructions pour une installation rapide de Linux en 10 minutes
- Liste des analogies Microsoft-Linux
- Étapes rapides pour recompiler le noyau Linux

- Site principal à <http://members.spree.com/technology/aldev/>
- Site miroir <http://aldev.8m.com>
- Site miroir <http://aldev.webjump.com>
- Site miroir <http://homepages.infoseek.com/~aldev1/index.html>
- Site miroir <http://www3.bcity.com/aldev/>