

Comparaison théorique des SGBD-R* Oracle, Informix et Ingres

Suzy PASLEAU

L'informatique est un monde où il est extrêmement difficile de se faire une opinion correcte sur la qualité et l'avenir des produits offerts. Il faut tenir compte en effet de l'intoxication publicitaire à laquelle se livrent les différents constructeurs, afin d'accroître leurs marchés. L'effet de mode joue également un rôle et le nouveau logiciel qui vient de sortir et que peu d'utilisateurs ont encore réussi à se procurer, apparaît d'emblée supérieur à tout ce qui se faisait précédemment.

Dans ce contexte, il n'est pas inutile de s'interroger sur le SGBD-R le mieux adapté aux traitements des sources historiques. Plusieurs systèmes se font concurrence. Chacun d'eux possède ses faiblesses et ses points forts : le choix de l'un ou de l'autre dépend avant tout des besoins spécifiques de chaque utilisateur en particulier. Par ailleurs, si nous sommes des défenseurs des SGBD-R, notre expérience nous enseigne cependant la prudence. Le concept commercial de SGBD-R est encore récent, le marché est en pleine mutation et les produits ne sont pas toujours bien adaptés à tous les environnements.

C'est pourquoi nous avons sélectionné les trois SGBD-R les plus performants en 1990, afin de comparer leurs mérites respectifs. Il ne s'agit évidemment pas de décrire ces produits dans leurs moindres détails, mais d'évaluer la capacité de chacun à réaliser les objectifs du concepteur d'applications tout en minimisant les temps de développement et de maintenance. Nous avons donc privilégié le point de vue théorique dans la mesure où il faudrait émettre certaines réserves quant à l'opportunité d'une étude ORACLE/INFORMIX/INGRES qui ne tiendrait compte que du contexte de notre laboratoire et des applications que nous y développons.

* SGBD-R : Système de gestion de bases de données relationnelles.

Les conclusions d'un tel type d'étude ne permettraient pas à un historien de prendre une décision en toute connaissance de cause¹.

Présentation des produits

On peut considérer une application informatique comme un ensemble de données que l'on crée, modifie et interroge à l'aide d'écrans de saisie et de recherche, que l'on exploite par des traitements plus ou moins complexes, et que l'on visualise sous forme d'états imprimés (éventuellement sous forme graphique).

Chaque SGBD-R inclut un gestionnaire de données, des générateurs d'écrans et d'états imprimés, ainsi que les facilités nécessaires pour effectuer les traitements adéquats.

Le tableau 1 reprend, pour chaque produit, les modules qui réalisent les fonctionnalités nécessaires. Nous y avons ajouté des fonctions fort importantes, mais pas toujours indispensables : la possibilité d'intégrer les langages LG3 aux potentialités du SGBD-R, les facilités de connexions entre machines hétérogènes, et enfin les aides au niveau du développement.

Selon nos besoins, nous serons amenés à utiliser le SGBD-R de diverses façons :

SGBD-R interactif. Une première méthode est d'utiliser directement les outils du SGBD-R pour effectuer rapidement des opérations relativement simples et classiques. Ils permettent de créer aisément de petites applications, sans connaissance particulière en informatique. D'autre part, le programmeur pourra réaliser le "prototype" de son application, qui consiste à tester en interactif tous les accès à la base de données avant de passer à la réalisation complète de son programme. Enfin, c'est aussi grâce à ces logiciels que l'administrateur du système gère les données, les applications et le nombre d'utilisateurs. Un administrateur du SGBD est souvent indispensable pour mener à bien l'informatisation d'un site : il joue un rôle fondamental en aidant les développeurs à utiliser correctement les potentialités du système.

SGBD-R intégré. Une deuxième possibilité consiste à continuer la programmation avec un langage classique de la troisième génération (L3G) comme le C ou le COBOL, et à intégrer des écrans, requêtes SQL et états imprimés créés à partir des outils du SGBD-R. C'est souvent une étape intermédiaire qui utilise le "know-how" des programmeurs et qui autorise déjà un gain de temps appréciable. En effet, toute la gestion des fichiers et des écrans est réalisée par le SGBD-R.

¹ Cet article s'adresse avant tout à des praticiens de l'informatique. Cet article nous fournit l'occasion d'exprimer en toute objectivité les résultats de notre expérience en la matière.

Tableau n° 1 : Outils des SGBD-R, ORACLE, INFORMIX et INGRES.

FONCTIONS	ORACLE	INFORMIX	INGRES
Gestion interactive des données	SQL*PLUS	RDSQL	SQL, QUEL
Génér. menus	SQL*MENU	4GL MENUS	ABF
Génér. écrans	SCREEN PAINTER (SQL*FORMS)	4GL PERFORM	VIFRED
Génér. états imp.	SQL*REPORT SQL*REPORT-WRITER	4GL ACE	REPORT WRITER REPORT BY FORMS
Génér. applic.	SQL*FORMS	4GL	ABF, QFB, VIFRED, INGRES 4GL
Tableur	ORACLE FOR 1, 2, 3 SQL*CALC	DATASHEET ADD IN	VQL
Graphiques	SQL*GRAPH		VIGRAPH
Intégration LG3	PRO*C, ... (6 langages)	ESQL/C, ... (3 langages)	ESQL (6 langages)
Connexions distantes	SQL*CONNECT SQL*NET SQL*STAR	INFORMIX-NET	PC-LINK INGRES-NET INGRES-STAR
Aides au développement	outils CASE	4GL-RDS/ID FOUR-GEN	

SGBD-R programmé. Dans ce mode, le développeur écrit ses applications à l'aide d'un générateur intégré au SGBD-R (souvent un L4G). Toutes les finesses de traitements particuliers sont théoriquement possibles. Par rapport à l'optique "SGBD-R intégré", les potentialités du système sont pleinement exploitées et les temps de développement sont sans commune mesure. Par ailleurs, il est généralement possible d'intégrer des routines en L3G pour des traitements très spécifiques.

Bien que couvrant l'ensemble des trois modes, le présent article s'attache plus particulièrement au mode "programmé".

Le logiciel ORACLE

ORACLE diffère très peu selon le matériel envisagé à cela près que les outils CASE sont disponibles seulement sur mainframe et que la version Macintosh présente comme toujours des caractères spécifiques : elle contient des instruments particulièrement bien adaptés à l'environnement du Macintosh. Par ailleurs, les langages

supportés L3G diffèrent d'une machine à l'autre et d'un système d'exploitation à l'autre. Cela dit, le grand avantage d'ORACLE est sa portabilité sur la plupart des matériels (du PC au mainframe MVS). Ainsi, la nouvelle version PC inclut dorénavant les modules SQL*ReportWriter et SQL*Menu, jusqu'ici réservés aux autres versions.

ORACLE n'offre plus de menu général : tous les modules sont distincts et appelés directement à partir de l'OS.

SQL*PLUS

Ce module permet la manipulation directe des données par le langage SQL. La version SQL d'ORACLE est l'une des plus complètes du marché et entièrement compatible avec DB2, SQL/DS et ANSI.

SQL*PLUS possède en outre des commandes de mise en forme à l'écran des résultats des requêtes SQL.

SQL*FORMS

1.- Philosophie

Le module de développement d'applications ne renferme pas de langage de programmation : le programmeur construit son application à l'aide de "fenêtres" de dialogues. Cette démarche originale a pour principal intérêt de permettre une grande rapidité de développement. Nous y reviendrons.

La philosophie de SQL*FORMS peut se résumer comme suit :

- Les tables sont créées grâce à SQL*PLUS.
- SQL*FORMS prévoit des écrans de saisie par défaut de tables.
- SQL*FORMS offre aussi une quantité importante d'options par défaut pour toutes les opérations courantes (interrogations, saisies, ...). Par conséquent, dès qu'une table est créée, elle est opérationnelle au niveau des traitements classiques et peut être intégrée immédiatement dans une application.
- Le programmeur a la possibilité de modifier ces options par défaut et de préciser les exceptions et traitements particuliers. La grande efficacité de SQL*FORMS vient de ce que ces modifications sont introduites de façon interactive à l'aide de menus et d'options, sans écrire la moindre ligne de code. L'interactivité optimale est obtenue par l'emploi intensif de touches fonctions, entièrement programmables, aussi bien pour la conception des applications que pour leur utilisation.
- L'application obtenue est représentée sous la forme d'un fichier ASCII aisément "portable" d'une machine à l'autre. Pour être exécuté, ce fichier est alors compilé par un utilitaire extérieur non intégré.

2.- Caractéristiques

- L'entité de base est la FORME, constituée d'un ou de plusieurs blocs, et peut occuper plusieurs pages écran.
- Un BLOC correspond à une table de la base de données dans laquelle les opérations seront exécutées. Il contient des zones réservées aux informations et mises en relation avec la table de base, des constantes de textes et des caractères semi-graphiques (lignes, cadres, ...). Un bloc multi-enregistrements permet de visualiser morceau par morceau le contenu d'une table et est particulièrement utile lorsque la forme contient deux blocs et que la relation entre ceux-ci est 1 : N (à un enregistrement de l'un correspond N enregistrements de l'autre, comme par exemple un ménage et ses individus).
- Les opérations par défaut sont modifiées à l'aide de TRIGGERS. Un TRIGGER est une séquence d'instructions SQL ou SQL*FORMS, exécutée lorsqu'un événement survient. La séquence contient des opérations de divers ordres : *lookup* de données dans d'autres tables pour intégrité référentielle avec affichage ou non, mise à jour des données, passage dans une autre forme, exécution d'une routine L3G, ...
- Un EVENEMENT peut être associé à un mouvement de curseur, à une touche-fonction, à une forme, un bloc, une zone ou appelé par un autre *trigger*.
- Les *triggers* permettent de simuler très rapidement des alternatives (IF et CASE) et des boucles. Les "USER TRIGGERS" conviennent pour les procédures répétitives.
- Le curseur dans l'écran est déplacé directement par l'utilisateur ou par programme.
- Lors du passage d'une forme à l'autre, le passage de paramètres est possible dans les deux sens via des variables globales. En effet, la forme que l'on quitte est abandonnée.
- C'est ORACLE qui gère les variables-mémoire de zones de l'écran. Le programmeur n'y a pas accès.
- Les touches-fonctions peuvent être redéfinies à différents niveaux dans l'application, grâce encore à des *triggers*.
- Il n'existe pas de facilité particulière pour les aides de caractère textuel : le texte d'aide peut être placé en constante dans une forme qui contiendrait plusieurs blocs correspondant aux différentes pages. C'est une solution qui utilise beaucoup de ressources.

3.- Générateur d'écrans

SCREEN PAINTER, le générateur d'écrans, est intégré à SQL*FORMS. Les écrans sont décrits de façon interactive, grâce aux fenêtres de dialogue. Au départ d'écrans "standard" et d'options par défaut très utiles, le programmeur peut tout modifier de façon rapide et aisée : supprimer certaines zones, en rajouter, modifier leur type, leur longueur, leur localisation ...

Il est possible d'attacher des attributs à chaque zone d'un écran : zone affichable ou cachée, critère de recherche, zone requise, conversion en majuscules,... mais pas d'attribut d'affichage (couleur, inverse, clignotement ...).

Il est permis de donner à chaque zone des critères de validation (contraintes de domaines, valeur par défaut), ainsi que des contraintes de relation avec d'autres zones, d'autres blocs (intéressant pour les jointures entre blocs).

La ligne et les cadres peuvent être tracés en caractères semi-graphiques.

Dans les zones à choix multiple, on peut associer une zone à une colonne d'une table de la base pour permettre l'affichage successif de toutes les valeurs possibles de cette colonne, parmi lesquelles l'utilisateur effectuera sa sélection.

Une forme de SQL*FORMS est "physiquement" limitée aux dimensions de l'écran, mais rien n'empêche qu'elle s'étende sur plusieurs. Il n'est pas possible de créer des fenêtres.

SQL*MENU

Ce module, non intégré à SQL*FORMS, est utilisé lorsque toutes les formes ont été générées. La création de menus hiérarchiques s'effectue aisément par fenêtres de dialogue. A l'intérieur des formes, l'opération est plus laborieuse et peu satisfaisante. Il faut recourir à certaines astuces de SQL*FORMS pour créer ce que l'on appelle des pseudo-menus (en fait un menu est une forme particulière).

SQL*REPORT

Le module, ancien, est encore disponible sous DOS et sur certaines machines. Il fait appel à un langage hermétique de la troisième génération pour créer laborieusement des états imprimés. On lui préférera le module suivant, nettement plus convivial.

SQL*ReportWriter

Il fonctionne, comme SQL*FORMS, par touches fonctions et sélections d'options dans des fenêtres de dialogue. Le résultat, sans être pleinement "WYSIWYG", est obtenu très rapidement. Les attributs de sortie sont assez pauvres, à nouveau par souci de compatibilité, mais de nombreux traitements en mode calcul sont possibles.

Le logiciel Ingres

INGRES, de RTI, n'est plus réellement disponible que sous VMS, UNIX et DOS. RTI a décidé d'abandonner sa version MVS, bien qu'une application INGRES puisse se connecter sans problème sur un site DB2. La version DOS qui, il y a peu, souffrait de nombreux problèmes de gestion de mémoire, a été depuis lors sensiblement améliorée.

INGRES est un très bon logiciel intégré, où tous les modules sont accessibles à partir d'un menu général. L'utilisateur est constamment guidé par une structure arborescente de menus, un commentaire spécifique à chaque option et une aide contextuelle très utile.

Au départ, INGRES a été développé autour du puissant langage QUEL propre à RTI, jusqu'à l'adoption de la norme SQL, voici quelques années. INGRES est cependant moins bien adapté à SQL qu'à QUEL et la version SQL s'écarte de façon parfois embarrassante des normes en vigueur (ANSI, DB2).

Le module ABF

1.- Philosophie

ABF (Application by Forms) est plus classique que SQL*FORMS. Il implique l'utilisation d'un langage de programmation (INGRES 4GL) et utilise les fonctionnalités des outils VIFRED et QBF pour créer et gérer les écrans.

La philosophie d'ABF peut se résumer comme suit :

- Les tables sont créées par le langage SQL ou dans un module qui permet une gestion interactive des tables (sans écriture de code). Il est appelé à partir du menu général. La structure d'une table ne peut pas être modifiée, lors du développement (il faut la détruire, puis la reconstruire), ce qui constitue manifestement un handicap.
- Un écran de saisie par défaut est associé à chaque table. Cet écran est entièrement modifiable par VIFRED.
- Une application est un ensemble de "frames" et de procédures 4GL.
- Un "frame" peut être de trois types : QFB, Report ou Utilisateur. Un "frame" QFB est un écran comprenant les zones d'une ou plusieurs tables liées (Joindefs), et géré de façon entièrement automatique. D'office un menu "standard" est présent et propose les opérations classiques de création, suppression, modification et recherche. Un "frame" Report est un état imprimé généré à l'aide du module INGRES REPORT WRITER, tandis qu'un "frame" Utilisateur, de loin le plus courant, est un écran quelconque façonné entièrement par le programmeur.

- Le code 4GL permet de déterminer le flux de dialogue de l'application. Une procédure 4GL n'est pas associée à un écran particulier, mais peut être appelée par un "frame" ou une autre procédure.

2.- Caractéristiques

Le dialogue dans l'application se fait via des menus et des touches fonctions.

La gestion de ces menus est extrêmement simple dans AFB. Quelques lignes de code permettent de décrire les options, un commentaire éventuel à afficher et la séquence d'instructions à exécuter.

Seules certaines touches-fonctions sont définissables, et à un seul niveau simultanément.

Le langage 4GL possède des instructions L3G classiques (définition de variables, alternatives et boucles imbriquées, ainsi que des appels à des sous-routines) et des instructions plus élaborées comme les appels à des "frames" ou l'entrée dans l'OS. Les accès à la base de données se font exclusivement par SQL ou QUEL.

Les sous-routines sont compilées séparément et utilisables autant de fois qu'on le désire.

Lors de l'appel d'un "frame" à partir d'un autre "frame", celui-ci est conservé en mémoire : on y revient instantanément, mais l'espace mémoire occupé devient de ce fait trop important.

Le passage de paramètres peut se faire par référence ou par valeur.

C'est INGRES qui gère les variables-mémoire des zones des "frames". Le programmeur n'y a pas accès.

La position du curseur est programmable.

Une aide contextuelle est rapidement constituée grâce à une facilité de INGRES, elle réclame toutefois la création d'un fichier ASCII.

3.- Générateur d'écrans VIFRED

- Le générateur VIFRED est utilisé par QBF pour modifier les écrans par défaut des tables de base, et par ABF pour créer les "frames" Utilisateur. Comme tout le logiciel INGRES, VIFRED est essentiellement basé sur l'utilisation de menus, et se révèle parfois plus long d'utilisation que le SCREEN PAINTER d'ORACLE.
- Un "frame" peut être plus grand que l'écran et est accessible par lecture horizontale ou verticale, il ne comporte pas de fenêtres.
- Les attributs des zones comprennent toutes les possibilités d'affichage (couleur ...). Par contre, il n'existe guère de possibilité dans le domaine des caractères semi-graphiques. Des constantes-textes peuvent être insérées n'importe où.

- Les critères de validation et contraintes d'intégrité sont introduits très facilement.
- La notion de "zone à choix multiples" pour la sélection par l'utilisateur d'une valeur parmi plusieurs est obtenue par le code 4GL lors de l'entrée dans la zone.
- Pour afficher plusieurs enregistrements en même temps (par exemple les individus d'un ménage), on définit une zone comme étant un tableau de longueur précise et le défilement est bien sûr possible.

Le module QBF

Comme nous l'avons déjà signalé, QBF (Query By Forms) est un outil puissant permettant la gestion complète d'une table ou d'un ensemble de tables liées par des attributs communs (*Joindef*). L'écran par défaut de chaque table est personnalisable par VIFRED. QBF peut être appelé à partir d'une application ("frame" QBF).

Le module REPORT WRITER

Très puissant, comme l'outil équivalent d'ORACLE, il permet la constitution rapide d'états imprimés de façon interactive.

Le logiciel Informix

En fait, il existe deux grandes lignes de produits dans INFORMIX : INFORMIX-SQL et INFORMIX-4GL.

INFORMIX-SQL contient les modules RDSQL, PERFORM, ACE et MENUS. RDSQL permet la gestion interactive des données par SQL, PERFORM crée les écrans et ACE les états imprimés. Grâce à MENUS, ces modules sont intégrés pour constituer une application. INFORMIX-SQL reste cependant limité à des applications simples, et s'adresse plutôt à l'utilisateur final. L'instrument est très simple à utiliser : des menus, commentaires et aides diverses guident l'utilisateur pour créer et manipuler ses données. SQL n'est pas nécessaire, et toutes les opérations peuvent être réalisées sans écrire une ligne de code.

Pour des applications professionnelles, les produits INFORMIX-4GL sont nettement plus adaptés. INFORMIX-4GL utilise le module RDSQL et existe sous deux formes : le logiciel "standard" et le nouveau produit RDS/ID comprenant un *debugger* interactif. Dans la suite, nous nous attacherons seulement à cette ligne de produits, bien que l'acquisition de INFORMIX-SQL soit fortement conseillée lors de l'achat de INFORMIX-4GL.

Logiciel le plus utilisé sous UNIX, INFORMIX est encore disponible sous VMS et DOS et le concepteur n'est pas tenté par d'autres environnements (MVS, ...).

Le produit INFORMIX-4GL

1.- Philosophie

Dans les grandes lignes, nous retrouvons la philosophie de INGRES (code 4GL et outils pour les écrans et les états imprimés). Toutefois, INFORMIX a été créé pour l'environnement UNIX et le code 4GL en est fortement influencé. En fait, il n'est rien d'autre qu'une large couche de logiciel posée sur C, le langage privilégié de UNIX. Comme le C est aussi largement utilisé sous DOS et VMS, on imagine bien qu'il va jouer un rôle fondamental dans INFORMIX...

- Le code 4GL compilé est un langage C agrémenté de requêtes SQL. On passe par une seconde compilation qui transforme le SQL en C, puis le C est à son tour compilé en code-objet.
- Le code 4GL est structuré en fonctions, comme le C, et des fonctions C extérieures sont appelables à partir du 4GL.
- Au niveau de l'éditeur de liens, rien n'empêche le programmeur d'insérer des bibliothèques C venant de tous horizons.
- Le code 4GL est d'un abord très aisé et les instructions (curseurs et tableaux notamment) sont particulièrement puissantes.
- Les écrans et les états imprimés sont générés très simplement à l'aide d'outils assez performants.
- Les applications bénéficient de tous les attributs d'affichage (couleur ...) et les fenêtres sont permises, même sous terminaux ASCII.

On comprend dès lors qu'INFORMIX-4GL apparaisse comme un outil de développement sans pareil dont les seules limites sont celles du C.

2.- Caractéristiques

- La démarche pour créer des menus est aussi simple que dans INGRES. On peut associer un commentaire, une aide et un "prompt" (ou confirmation) à chaque option du menu. Une option permet d'appeler une fonction quelconque qui est soit une fonction 4GL comprenant des requêtes SQL, un menu ou une autre forme, soit une fonction C créée indépendamment. A noter qu'une fonction C peut contenir des requêtes SQL si l'on respecte la syntaxe de ESQ/C.
- Le programme réagit aux options de menus, aux touches fonctions programmables, et à d'autres événements comme l'entrée et la sortie d'une zone.
- Le code 4GL comprend alternatives et boucles, ainsi que des fonctions de haut niveau gérant l'affichage, les formes et les états imprimés.
- Contrairement à ORACLE et INGRES, INFORMIX-4GL ne travaille pas directement sur le buffer de la base de données et gère les zones des formes par des

variables-mémoire accessibles au programmeur. Celui-ci bénéficie donc d'une grande souplesse.

- Le passage d'une forme à une autre forme quelconque s'effectue simplement par superposition des fenêtres d'encodage. Le transfert de paramètres est possible dans les deux sens. Par exemple, lors de l'encodage d'un individu, si le ménage n'existe pas encore, on peut ouvrir la fenêtre d'encodage d'un nouveau ménage. Lorsque celui-ci a été introduit, on retrouve la fenêtre d'encodage de l'individu où les informations ménages sont déjà complétées.
- La notion de "zone à choix multiples" est réalisée de façon intelligente. Lors de l'entrée dans la zone, on peut ouvrir une fenêtre où apparaissent les valeurs. Celles-ci proviennent éventuellement d'une colonne de tables grâce à une requête SQL. L'utilisateur fait défiler les valeurs, et celle qui est sélectionnée est affichée dans la zone. Ce "lookup" dans une autre table est basé sur la notion de "curseur" bien connue en SQL, et notamment le "scroll cursor".
- Les zones multi-enregistrements trouvent aussi une solution immédiate sous la forme de tableaux comme dans INGRES.
- L'aide contextuelle est particulièrement simple à implanter. Un seul fichier ASCII convient pour toutes les aides.

3.- *Le générateur d'écrans*

Le générateur est différent des modules correspondants des deux autres produits. Une forme est ici entièrement définie par un fichier ASCII qui contient le masque, la liste des zones à afficher, les attributs et les mises en forme nécessaires. Cette forme est ensuite compilée puis associée à un code 4GL qui précisera le menu courant, les touches-fonctions disponibles, les contraintes de validation et d'intégrité, ainsi que les traitements spécifiques.

Une forme peut être une fenêtre qui se superpose alors à l'écran précédent. Tous les attributs d'affichage sont permis et la mise en forme est agrémentée de lignes et cadres en caractères semi-graphiques. Toutefois, il convient de noter que certaines fonctions de mises en forme sont disponibles uniquement sur la version DOS et suscitent, le cas échéant, des problèmes de compatibilité.

Une facilité intéressante d'INFORMIX réside dans l'utilitaire UPSCOL. Le programmeur peut introduire sur les colonnes des tables de contraintes de validation valables pour toutes les formes où ces colonnes interviennent. Par exemple, si le code ménage est un entier compris entre 1 et 4, on introduira cette contrainte au niveau de UPSCOL, avec un message à afficher en cas d'erreur d'encodage. Tout se passe alors comme si la contrainte avait été introduite dans toutes les formes où le code ménage intervient.

Une forme a toujours la dimension de l'écran.

Le produit INFORMIX-4GL RDS/ID

RDS = Rapid Development System

ID = Interactive Debugger

Ce produit récent présente trois avantages majeurs que nous ne retrouvons pas dans ORACLE et INGRES :

- La compilation transforme le 4GL et les fonctions C en p-code. Celui-ci sera alors interprété ligne par ligne pour un "debugging" fouillé. Il est possible notamment de visualiser toutes les variables et des "breakpoints" peuvent être placés à tout endroit du programme.
- La compilation en p-code présente aussi l'avantage d'être particulièrement rapide, ce qui augmente d'autant la productivité du programmeur.
- Ce p-code est totalement transparent pour tous les matériels (contrairement aux compilateurs C). Le p-code est donc particulièrement bien adapté pour le portage des applications sur différentes machines. Il suffit d'utiliser le produit INFORMIX-4GL RDS/ID sur PC pour transporter aisément les applications sur toute autre machine.

Le générateur d'états imprimés

Ce module fonctionne de la même façon que le générateur d'écrans : il constitue un fichier ASCII qui décrit le document désiré avec sa mise en forme et les attributs nécessaires. Ce fichier est alors compilé avec les traitements appropriés (calculs).

Les possibilités de ce générateur sont relativement restreintes bien que suffisantes pour la plupart des applications.

Le produit FOUR-GEN

La constitution des fichiers ASCII qui décrivent les écrans et les états imprimés devient vite fastidieuse dans le cas d'applications complexes. Une société américaine a réalisé un *package* qui génère du code 4GL sur base d'une description d'écrans et d'états imprimés. Le code résultat, bien documenté, reprend tous les traitements classiques possibles et très rapidement, les écrans et les traitements par défaut sont encodés. Le programmeur peut alors se concentrer sur l'application proprement dite en modifiant le code résultat pour tenir compte des traitements particuliers.

Comparaison des produits

Relationnalité

Ce critère de comparaison est très théorique. Un SGBD qui ne respecte pas toutes les caractéristiques du modèle relationnel n'est pas nécessairement inefficace. C'est ainsi qu'il existe de très bons SGBD de type réseau et donc non conformes au modèle relationnel. Cependant, étant donné les avantages considérables du modèle relationnel, il est intéressant de voir comment se comportent les trois produits étudiés.

Nous nous basons sur le modèle relationnel de Codd, et notamment sur les treize règles qu'il a mises au point.

- Contrairement à INGRES, les deux autres logiciels permettent de définir et de travailler sur des valeurs "NULL" (règle 3).
- Le langage SQL est supporté par les trois produits (avec certains écarts à la norme, très gênants pour INGRES). Il faut savoir que la norme SQL elle-même ne respecte pas toujours le modèle relationnel : les vues ne sont pas réellement manipulables comme les autres tables et les contraintes d'intégrité ne font généralement pas partie du dictionnaire (règles 5, 6 et 10).
- Certaines contraintes d'intégrité peuvent être définies dans INGRES, mais de façon limitée, et malheureusement non exploitables dans les applications (règle 10).
- La règle 12 de non-subversibilité n'est pas respectée par les trois produits, en raison de dictionnaires trop incomplets.
- Enfin, les bases de données réellement distribuées restent encore à réaliser (règle 11).

Il s'agit en fin de compte de trois bons produits relationnels, si ce n'est des dictionnaires perfectibles et une version SQL décevante dans INGRES.

Portabilité

En fait, il faut distinguer différentes notions dans ce concept de portabilité :

- Portabilité proprement dite.
Le produit est-il disponible sur un grand nombre de machines?
- Distribution des traitements.
Peut-on, à partir d'une machine X, se connecter sur une machine Y où se trouvent les données, et exécuter sur X une application qui utilise les données de Y?
- Distribution des données.
Peut-on, à partir d'une application située en X, accéder simultanément à des données réparties sur divers sites Y, Z, ... et ce, de façon transparente?

– *Gateways* (passerelles).

Peut-on manipuler des données du produit X avec le produit Y et inversement?

Portabilité proprement dite

ORACLE est disponible sur plus de trente systèmes d'exploitation, du PC au *mainframe*. Il est le produit-clé si l'on désire adapter un SGBD-R sur un parc de machines particulièrement hétérogènes. Il convient toutefois de vérifier si la machine utilisée est bien compatible. Ainsi, ne suffit-il pas de dire que UNIX est "supportée", car les versions UNIX sont presque aussi nombreuses que les marques de machines.

Dans ORACLE, le "portage" sur l'AS400 est prévu, en raison même d'une forte demande. ORACLE étant écrit en C, c'est la disponibilité d'un compilateur C non buggé sur l'AS400 qui décidera en fait de la date de mise en service du produit².

INGRES et INFORMIX sont limités aux environnements DOS, UNIX et VMS. Cette limitation est cependant toute relative, dans la mesure où certains systèmes UNIX peuvent supporter jusqu'à 200 terminaux.

Les versions DOS de INGRES et INFORMIX se satisfont de 640 Kb de mémoire vive, mais détectent et utilisent toute mémoire supplémentaire.

Distribution des traitements

De ce point de vue, les logiciels se valent. Ils offrent la possibilité d'exécuter en local une application qui puise ses données dans un serveur distant. Les protocoles de réseaux supportés sont en principe les divers standards du marché³.

Distribution des données

Dans ce domaine, on en est toujours aux premiers balbutiements. ORACLE et INGRES semblent les plus avancés, bien que INFORMIX annonce une version 4.0 distribuée pour la fin de l'année.

Gateways

Les trois sociétés éditrices des logiciels, prévoient des utilitaires de liaison avec DB2 de IBM (qui tourne sur des *mainframes*).

L'intérêt de ces passerelles est double :

² Il ne faudrait pas espérer le produit avant 6 mois; le retard provient de ce que les langages privilégiés de l'as sont le RPG et le COBOL, et le C ne sera que très peu utilisé sur ce type de machines (Nous avons reçu le texte en juin 91. NdE).

³ Il est préférable de vérifier que tout ce qui est annoncé fonctionne effectivement. Le mieux est toujours de demander une démonstration probante.

- Elles permettent à une application INGRES, ORACLE ou INFORMIX de se connecter sur un site distant DB2, de façon transparente : les données DB2 seraient visualisées et manipulées via les outils locaux.
- Elles rendent possible la saisie des données sur un site DB2 et leur traitement local sur ORACLE, INFORMIX ou INGRES. Inversement, on peut transférer des données locales sur un site DB2 distant.

ORACLE propose aussi une passerelle vers SQL/DS, INGRES vers IMS et RDB.

Fonctionnalité du générateur d'écrans

Il s'agit d'un point essentiel, car l'encodage des écrans constitue une part importante de chaque application.

Les générateurs d'ORACLE et d'INGRES offrent une excellente convivialité : la constitution des écrans est interactive, ne nécessitant l'écriture d'aucun code. On donnera ici la meilleure mention au SCREEN PAINTER d'ORACLE, au contraire du générateur d'INFORMIX. Certes, la description des écrans s'y fait très aisément, mais sous forme de programme. C'est bien sûr pour cette raison que le logiciel FOUR-GEN est maintenant proposé en complément du 4GL d'INFORMIX.

Par contre, les écrans obtenus par INFORMIX sont nettement plus ergonomiques, étant donné les possibilités de créer des fenêtres. Les applications en deviennent plus dynamiques, sans que l'utilisateur ne perde le fil de sa procédure, puisque les écrans successifs se superposent. De ce point de vue, ORACLE est loin de présenter les mêmes avantages, il ne gère pas en effet les mêmes attributs d'affichage.

Fonctionnalités du générateur d'applications

L'inconvénient majeur des langages de la quatrième génération est leur manque de souplesse. Généralement, dans un 4GL, on augmente l'efficacité d'un programme en réalisant automatiquement certaines fonctions et en fournissant un ensemble d'options par défaut qui ne sont pas toujours modifiables. En fait, on fige une partie du travail du programmeur avec l'idée que la plupart des applications de gestion opèrent selon le même mode. Ce raisonnement pertinent pour des applications simples, ne se vérifie pas pour des développements plus complexes gérant un grand nombre de tables et donc d'écrans. Si l'on désire réaliser un logiciel ergonomique qui répond efficacement aux besoins des utilisateurs, on est vite confronté à des flux de dialogues spécifiques qui s'écartent des traitements par défaut. On peut voir là une des raisons de la percée relativement lente des 4GL vis-à-vis des langages plus traditionnels.

Qu'en est-il de nos trois produits ?

Le logiciel qui permet la plus grande rapidité de développement est incontestablement ORACLE : pas une ligne de code à écrire ; tout se joue avec les *triggers*. Malheureusement, de nombreuses limitations freinent les procédures dès que l'on sort des traitements classiques.

Citons, entre autres :

- Pas de solution simple pour les menus associés aux formes.
- Seules certaines fonctions très limitées sont permises dans les *triggers* navigationnels (c'est-à-dire attachés aux mouvements du curseur dans l'écran).
- Lors du passage d'une forme à l'autre, la première et toutes les valeurs déjà introduites sont perdues, si l'on ne les sauve pas (pas de passage de paramètres par référence ou par valeur).
- Toutes les requêtes SQL ne sont pas permises dans SQL*FORMS (notamment l'importante option GROUP BY) , ce qui rend difficile certaines opérations comme par exemple l'affichage de statistiques.

Avec INGRES, les développements seront plus longs, mais le langage 4GL assure davantage de souplesse. Les types d'instructions prévus dans ce langage sont cependant encore limités et surtout la démarche à suivre pour réaliser un traitement précis est loin d'être toujours évidente. INGRES est desservi par une documentation particulièrement mauvaise où certaines potentialités du langage ne sont qu'effleurées, quand elles ne sont pas simplement ignorées. Les programmeurs qui abordent INGRES passent par une longue période d'adaptation pendant laquelle ils essayent diverses options du langage.

Les quelques contacts que nous avons eus avec INFORMIX-4GL nous ont réellement enthousiasmés. L'accès aux fonctionnalités du 4GL est aisé, la documentation est excellente et la philosophie du langage, claire et astucieuse. La responsabilité en incombe au langage C sous-jacent, dont le 4GL a la structure simple et modulaire. Nous voyons mal quelle limitation présenterait le produit puisque du code C, dont la souplesse n'est plus à démontrer, peut être inséré à tout endroit, même après compilation de 4GL. L'acquisition de la version RDS/ID constitue un atout supplémentaire, étant donné les nombreux avantages qui en découlent et que nous avons déjà décrits dans la partie précédente.

INFORMIX-4GL est incontestablement ici notre outil préféré.

Intégration vis-à-vis de l'environnement

Les trois logiciels permettent l'accès au système d'exploitation et à toutes ses commandes.

Il est tout à fait possible d'insérer du code C dans une application créée avec chacun d'eux, mais nous donnerons la meilleure cote à INFORMIX qui est particulièrement souple à ce niveau.

Inversement, rien n'empêche d'insérer des requêtes SQL dans des applications en C ou en COBOL (et d'autres langages éventuellement). Introduire une forme ou un bloc d'application 4GL est encore possible dans un programme C ou COBOL, avec certaines restrictions sur le transfert des paramètres notamment.

Contraintes d'intégrité

Les trois logiciels ne sont guère satisfaisants sur ce point. Les contraintes sur les données ne font en effet pas partie du dictionnaire. Une information est seulement connue du SGBD-R par son type et sa longueur.

En conséquence, une contrainte associée à une donnée doit être spécifiée pour chaque écran de saisie où celle-ci intervient. Rappelons cependant que dans le cas d'INFORMIX, l'utilitaire UPSCOL évite cet encodage répétitif pour les contraintes associées à une seule donnée, ce qui constitue le cas le plus fréquent...

Pour les contraintes correspondant à des clés étrangères, c'est-à-dire les contraintes entre tables, nos produits autorisent évidemment la prise en compte des liaisons 1 : 1 (maître/maître) et 1 : N (maître/esclave).

Sécurité et cohérence des données

Les trois produits sont entièrement satisfaisants.

La confidentialité de l'application est assurée par un mot de passe connu au niveau du dictionnaire de données.

Chaque table et, éventuellement, certaines colonnes ou lignes, peuvent être cachées ou rendues inaccessibles pour certains utilisateurs. En effet, ces SGBD-R bénéficient des facilités offertes par SQL :

- Des privilèges d'accès sont distribués par l'administrateur du système à chaque utilisateur, pour chaque table ou morceau de table distinctement.
- La notion de vue permet de spécifier très précisément la partie de la base de données que "voit" l'utilisateur.

Les accès concurrents à une même donnée sont gérés en partie par le SGBD-R et en partie par le programmeur grâce à des fonctions de "locking" de table ou de *tuple*. Les cas de "deadlock" sont détectés et résolus.

Enfin, il est heureux que les trois produits travaillent conformément à la notion de transaction qui assure la cohérence des données en cas de problème. Il est donc toujours possible de revenir à l'état cohérent précédent en cas de panne ou d'erreur. La gestion des transactions est basée sur les concepts de "before image file" et d'"after image file". Aucune donnée n'est perdue grâce à un "journaling" optionnel.

Générateur d'états imprimés

Les trois logiciels présentent des potentialités similaires, mais, ici aussi, les générateurs de ORACLE et INGRES offrent une meilleure convivialité. Le logiciel FOUR-GEN est un bon complément d'INFORMIX-4GL, sans doute moins indispensable que dans le cas de la génération des écrans.

Ergonomie des produits

Facilité d'apprentissage

ORACLE et INFORMIX sont deux excellents produits, très bien documentés. On regrettera dans ORACLE le manque d'intégration des divers modules et utilitaires, ce qui implique de fréquents chargements et déchargements de logiciels. Il manque aussi dans ORACLE un utilitaire permettant une gestion des tables, des vues et des index à la portée d'un non-informaticien. INGRES est un bon produit, relativement bien intégré, mais nous attribuerons une mauvaise cote au module ABF. La documentation d'INGRES est un exemple convaincant de ce qu'il ne faudrait pas proposer au développeur, et encore moins à l'utilisateur final.

INFORMIX-SQL et INGRES sont de bons logiciels pour les non-informaticiens, et pour le "prototypage".

Aide à la mise au point

ORACLE, INGRES et INFORMIX dans sa version de base, sont relativement pauvres en matière d'aide. Les erreurs de compilation ne sont pas toujours signalées (ORACLE) ou sont parfois identifiées de façon peu précise (INGRES et INFORMIX). Les applications ORACLE et INGRES peuvent fonctionner dans un mode de mise au point. Par exemple, dans ORACLE, on dispose d'une touche spéciale qui affiche le résultat de la dernière opération sur la base de données ou l'instruction qui a provoqué l'erreur. C'est à peu près tout ce qui permet d'accélérer la mise au point. L'expérience nous a montré avec ORACLE que celle-ci est parfois fastidieuse à cause de la structure et des différents niveaux de *triggers* qui interviennent en même temps. Par ailleurs, la documentation d'INGRES ne contient même pas la liste des erreurs avec leur signification.

Dans ce contexte, la version RDS/ID de INFORMIX-4GL apparaît comme un atout majeur. Toutes les possibilités courantes d'un *debugger* sont disponibles pour une mise au point efficace de l'application.

Vitesse de développement

En fin de compte, SQL*FORMS reste sans doute l'outil le plus rapide, du moins pour des applications pas trop complexes nécessitant peu de traitements spécifiques. INFORMIX-4GL se détache cependant nettement par rapport au module ABF de INGRES.

Le gain de vitesse par rapport aux langages traditionnels (COBOL, DBASE, C + outils) est réellement spectaculaire, même pour INGRES. Il n'est pas excessif d'avancer des facteurs de 10 à 20 d'augmentation de performance pour un programmeur expérimenté.

Conclusions

Nous avons, au cours de cette étude, analysé les trois logiciels dans le contexte d'applications classiques de gestion. Dans cette optique, il s'agit incontestablement de bons produits, nettement plus performants que ceux fournis par les constructeurs comme SQL/DS et DB2 (IBM) ou RDB (DEC). Ils sont néanmoins perfectibles. Par exemple, sur le plan de la relationnalité, la prise en compte, dans le dictionnaire, des contraintes d'intégrité de tous types, devrait accroître encore sensiblement l'efficacité de la programmation. Cette amélioration peut être attendue dans les prochaines versions des logiciels.

Ces trois produits se tiennent de fort près et sont, d'une façon générale, relativement bien cotés par les utilisateurs. Cela ne signifie pas cependant qu'ils sont équivalents, loin s'en faut.

Si l'on désire s'orienter vers un *mainframe* ou une machine particulière, ORACLE présente un avantage incontestable : il est le seul à supporter le système VMS. De même, le "portage" vers l'AS400 est prévu à moyen terme, alors que cette extension n'est prévue ni pour INFORMIX, ni pour INGRES.

Par contre, pour la grande majorité des installations, où un système UNIX ou VMS s'avère suffisant, le choix sera sans doute plus dépendant des qualités intrinsèques de chaque produit. Dans ce créneau, les trois SGBD-R proposent des solutions réseaux assez semblables : des machines UNIX, VMS et DOS peuvent se côtoyer et partager les données d'un même serveur. Sous réserve que ces solutions soient réellement opérationnelles, on ne trouve pas ici matière à les départager.

Sur le plan du développement d'applications, la version INFORMIX-4GL RDS/ID nous paraît un choix incontournable. Nous sommes particulièrement enthousiasmés par ce 4GL qui ne décevra pas un programmeur exigeant, comme cela pourrait être le cas pour ORACLE et INGRES. ORACLE présente en effet des restrictions non négligeables, conséquence d'un compromis entre la rapidité de programmation et la souplesse finale de l'application. Quant à INGRES, nous déplorons une version SQL peu satisfaisante et une documentation tout simplement indigne de tels logiciels. La période d'adaptation du développeur à INGRES risque d'être fort longue.

INFORMIX présente deux atouts principaux. D'abord une souplesse exceptionnelle de programmation. On possède ici un 4GL a priori sans limites puisqu'il peut être complété par le langage C dès que le besoin s'en fait sentir. Ensuite, la version RDS/ID constitue un outil redoutable pour la mise au point. Grâce au logiciel FOUR-GEN, il est fortement conseillé.

Ajoutons encore qu'INFORMIX et INGRES favorisent un "prototypage" efficace.

Ces conclusions doivent, bien sûr, être tempérées par les aspects commerciaux et surtout par le contexte du travail à effectuer (type d'applications, type de matériel informatique, formation et niveau des informaticiens).

Dans notre cas, par exemple, seul ORACLE répond parfaitement à nos besoins, dans la mesure où c'est l'unique logiciel micro qui nous permette d'envisager d'emblée une saisie directe dans une architecture conforme à la banque de données qui elle, vu la taille de nos fichiers, sera stockée sur mainframe.