Introducción

El mundo de las Bases de Datos ha estado históricamente dominado por tres grandes compañías (Oracle, Informix e IBM) que ofrecen productos destinados a la gestión de grandes cantidades de información. Estos productos tienen unos precios nomalmente inalcanzables para una empresa media y no hablemos de un usuario doméstico.

Dirigidos a un mercado medio encontramos multitud de productos que pueden cubrir perfectamente las necesidades de cualquier empresa mediana sin hacer que el presupuesto se dispare.

Menos conocidos por la mayoría de usuarios, existen también varios Gestores de Bases de Datos gratuitos desarrollados generalmente por Universidades o por grupos de voluntarios. Algunos de estos productos tienen una gran calidad y en muchos casos superan las cualidades de sus compedirores comerciales. Entre estos productos gratuitos y de gran calidad encontramos a PostgreSQL.

La historia de PostgreSQL comienza como un proyecto de investigación del grupo del Profesor Michael Stonebraker en Berkeley. Su proyecto original, *Ingres*, se comercializa actualmente con el mismo nombre. El original *University Ingres* está aún disponible de forma gratuita y usa su propio lenguaje de consulta *'QUEL'*.

Su siguiente proyecto fue 'Postgres' (*post−ingres*), que usaba un lenguaje de consulta más avanzado: *POSTQUEL*. En 1987, Postgres introdujo reglas, procedimientos, tipos definibles y conceptos de orientación a objetos. Postgres fue comercializado con el nombre de *Illustra* y forma parte actualmente del *Universal Server* de Informix. La versión original de Postgres 4.2 (basada aún en *POSTQUEL*) todavía se puede encontrar por Internet.

Dos alumnos del laboratorio de Stonebraker, Andrew Yu y Jolly Chen, decidieron reescribir gran parte del código de Postgres y sustituir *POSTQUEL* por un subconjunto ampliado de SQL. Esta nueva versión fue lanzada en 1995 como Postgres95 y en ese momento el desarrollo pasó a manos de la comunidad de Internet. Postgres95 evolucionó hasta lo que es ahora PostgreSQL.

PostgreSQL es un gestor de Bases de Datos basado en el modelo relacional, aunque incorpora algunos conceptos del modelo Orientado a Objetos, tales como la herencia. Usa un subconjunto ampliado de SQL como lenguaje de consulta y está implementado siguiendo la arquitectura cliente−servidor. PostgreSQL ofrece gran variedad de herramientas y librerías para acceder a las Bases de Datos. En su versión 6.3.2 (la última a la hora de escribir este documento), existen interfaces para los siguientes lenguajes:

- C
- C++
- C Embebido
- Java (a través de JDBC)
- Perl
- ODBC (existen drivers ODBC para PostgreSQL para Windows, Max, UNIX y VMS)
- Python
- TCL

Objetivos

A pesar de ofrecer diversos interfaces de programación, PostgreSQL carece de un lenguaje de cuarta generación (4GL) que facilite la programación de aplicaciones de bases de datos. Con este proyecto se pretende cubrir este hueco, desarrollando un compilador para un lenguaje 4GL.

Tomamos como referencia el lenguaje 4GL de Informix, siendo el objetivo conseguir que programas diseñados para Informix puedan ser recompilados y usados con PostgreSQL con el mínimo de modificaciones.

Informix−4GL ofrece herramientas para crear menús, formularios de entrada de datos y generadores de listados. Será necesario definir estas mismas herramientas manteniendo, a ser posible, la sintaxis original.

En principio se generarán aplicaciones que funcionen en modo texto, dejando para una futura ampliación la generación de aplicaciones en entornos gráficos.

Objetivos que se han cubierto

La sintaxis del lenguaje es casi idéntica a la de Informix−4GL. Se han implementado los tipos de datos básicos, los de fecha y hora, intervalos y *'datetime'*.

Las sentencias de asignación, presentación en pantalla, estructuras repetitivas (bucles *while* y *for*) y condicionales (*if..else..*), generación de menús y listados se han implementado completamente.

El manejo de los formularios de entrada de datos se ha implementado siguiendo la sintaxis original de Informix−4GL, sin embargo se ha modificado la sintaxis de la definición de los formularios para dar cabida a nuevos tipos de objetos, como botones o casillas de verificación.

Se ha introducido una nueva sentencia (*browse*), que permite visualizar todas las tuplas devueltas por una consulta y *'navegar'* por ellas mediante movimientos del cursor.

La sintaxis de algunas sentencias se ha ampliado, permitiendo nuevas funcionalidades no encontradas en Informix−4GL, como asignaciones múltiples. Estas modificaciones están explicadas con detalle en el manual de usuario.

Posibilidades de ampliación

Se ha generado únicamente una interfaz en modo texto. Sin embargo, la implementación de las funciones de manejo de formularios de entrada se ha realizado de manera independiente de la implentación a bajo nivel (*curses*). Gracias a esto es posible desarrollar librerías de funciones que gestionen los formularios de entrada de manera gráfica, generando interfaces en X−Window.

La generación de código está actualmente muy enfocada a la interfaz de PostgreSQL. Una posible ampliación sería la generalización de este código, para permitir el desarrollo de aplicaciones que accedan a datos almacenados en otro tipo de servidores, o incluso usando ODBC (el API de acceso a bases de datos definido por Microsoft y que se ha convertido en estándar).