

# **TecnoDB: Desarrollo de una metodología de aprovechamiento de Metadatos de los diccionarios de Datos de Bases de Datos Relacionales para lograr un generador de sentencias SQL PROMETEO**

## **Integrantes**

Ing. Iris Gastañaga  
Ing. Calixto Maldonado  
Ing. Cesar Martínez Spessot

**Laboratorio de Investigación de Software  
Departamento de Sistemas  
Facultad Regional de Córdoba  
Universidad Tecnológica Nacional**

## **Dirección Postal**

Universidad Tecnológica Nacional - Facultad Regional Córdoba  
Maestro M. López esq. Cruz Roja Argentina  
Ciudad Universitaria - C.P. (5016)  
Córdoba - República Argentina.

## **Dirección Electrónica**

cispessot@hotmail.com  
calixtomaldonado@hotmail.com  
iris.gastanaga@gmail.com

## **Resumen**

El proyecto de construcción de la herramienta de generación de sentencias SQL llamado provisoriamente 'Prometeo' tiene como objetivo lograr un sistema de software que se ejecute en múltiples plataformas de sistemas operativos, en principio, Windows 2000 y Red Hat Linux 7.1 y que sea capaz de generar sentencias SQL basados en los metadatos de los diccionarios de datos de Oracle y PostgreSQL:

- ❑ Consultas simples con SELECT,
- ❑ Consultas multitas de las tablas relacionadas con Constraints Referenciales
- ❑ DML para insertar, borrar y actualizar columnas
- ❑ DDL para creación, modificación y borrado de objetos como Vistas
- ❑ OQL o lenguaje de consulta de Objetos cuando la base de datos estudiada contenga esa opción (Oracle y PostgreSQL)

Las etapas del proyecto son

- ❑ Estudio de Factibilidad
- ❑ Desarrollo de la metodología y un prototipo de interfaz grafica
- ❑ Desarrollo de las versiones Alfa y Beta del producto

Con la recopilación de información en el estudio de Factibilidad inicial, se concluyó que, si bien existen herramientas al estilo enunciado, estas agregan valor y facilitan la explotación de información almacenada.

Forma parte del presente trabajo el desarrollo de la metodología para lograr las sentencias de SQL en base a los metadatos del motor Oracle 10g.

## **Palabras Clave**

Bases de Datos Relacionales, Motor de Base de Datos Relacionales, SQL, Generador de Consultas, Asistente de SQL, SQL autogenerado, Motor de Base de Datos orientado a Objetos, Generador de consultas con, Generador de DDL, Generador de DML. Sentencias SQL, Sentencias Metadatos.

## Introducción

Durante el proyecto “Desarrollo de un motor de Base de Datos Relacional, Tecnodb” acreditado por la Secretaria de Ciencia y Técnica de la Universidad Tecnológica Nacional, cuya directora es la Ing. Iris Gastañaga, creció la percepción sobre la falta de herramientas que asistan al usuario de nuestro motor para escribir sentencias SQL en forma automática basada en la información residente en el diccionario de Datos de las Base de Datos objetivo.

Al finalizar el estudio de Factibilidad, técnica y operativa de TecnoDB, las conclusiones y el plan de trabajo nos indicaron que los tiempos del proyecto principal, encargado a dos integrantes encargados, entre otras cosas, de la programación, permitirían a otros integrantes encarar las tareas de análisis, diseño y construcción de un software capaz de analizar la información de los metadatos de TecnoDB y como aún no estaba finalizada ese aspecto del Desarrollo, decidimos enfocarnos en bases de datos ya existentes en el mercado para construir una herramienta de características aplicables a múltiples plataformas.

Entendimos que avanzar usando otras Bases de Datos serviría para el proyecto original como originador de otra visión del diccionario de datos a implementar en TecnoDB y que ayudaría así a lograr un modelo de datos completo para los Metadatos. Además el hecho de estudiar los diccionarios de otras implementaciones, hará que este subproyecto nos dé la perspectiva de demandadores de información, otro tipo de usuarios del diccionario de Datos, ayudando a mejorar el producto final de TecnoDB.

Si le agregamos que el software ‘Prometeo’ ayudará a superar la barrera inicial, inherente al dominio del lenguaje SQL al usuario no experto y que también ayudara a facilitar la enseñanza de SQL a los Alumnos de las materias Gestión de Datos (GDA) y Sistemas de Soporte de Decisiones (SSD) de la Carrera de Ing. En Sistemas de Información de la Facultad Regional de Córdoba de la Universidad Tecnológica Nacional, vemos muy justificado el esfuerzo de abrir este subproyecto.

Estos motivos, mejorar el diccionario de datos de TecnoDB, generar una herramienta multiplataforma, ayudar al alumno de dos materias de la Carrera de Ingeniería en Sistemas y aumentar la productividad en desarrollo de sistemas, nos llevan a enunciar el objetivo de desarrollar una herramienta de generación de SQL 'Prometeo'.

El nombre surge de la mitología Griega, que cuenta que Prometeo y su hermano [Epimeteo](#) recibieron el encargo de crear la humanidad y de proveer a los seres humanos y a los animales de todo lo necesario para vivir. Epimeteo, procedió en consecuencia a conceder a los animales atributos como el valor, la fuerza o la rapidez y los proveyó de todos los elementos necesarios para poder vivir en el mundo, tales como plumas, patas, o piel. Sin embargo, [Epimeteo](#) debía crear un ser superior a todos los demás pero no le quedaban más virtudes para ello y no tenía nada que conceder, así que le pidió ayuda a su hermano Prometeo, nombre que significa "prudencia", para que los seres humanos fueran superiores a los animales, Prometeo decidió darles una forma más noble y permitirles caminar erguidos. Como don les dio el fuego, que había obtenido quemando una antorcha en el sol. El fuego era, sin duda, el don más valioso que Prometeo podía haber dado a la humanidad

Identificamos al gesto de Prometeo de darle el fuego al humano con el de poder obtener información de los metadatos trabajosamente incorporados al Diccionario de las Bases de Datos para lograr las consultas que se pueden derivar de ellos, generar vistas, sentencias de actualización, detectar estructuras Maestro Detalle, relaciones transitivas, etc., el usuario con algún entrenamiento básico, y esta herramienta podrá obtener mayor productividad reduciendo los errores de sintaxis y generar todas las consultas posibles a priori y que el usuario pueda guardarlas en el mismo diccionario, para su posterior uso.

Vimos como características necesarias para 'Prometeo' que debería ser **Multiplataforma**, refiriéndonos a multiplataforma a la posibilidad de ser utilizado independientemente del sistema Operativo del equipo cliente por un lado y diseñar la herramienta de tal forma que escrito en un lenguaje como JAVA, se ensanche la base de utilización del mismo.

Ser Multiplataforma, es la característica considerada como uno de los factores de éxito de Internet, la posibilidad de acceder todo tipo de contenido, texto, gráfico, video, sonido, independiente de la máquina y sistema operativo que el navegante use. Además las aplicaciones desarrolladas en lenguajes como Java hacen de su característica multiplataforma la bandera principal de su conveniencia.

Esta característica de Multiplataforma surgió entonces como el mayor valor, aparte del funcionar correctamente, de Prometeo y lo orientamos a dos aspectos multi-sistema operativo y que sea capaz de leer los metadatos de diferentes bases de datos, orientándonos primero hacia Oracle y PostgreSQL, para que una vez funcionando aplicarlo a TecnoDB, luego de haber diseñado el Diccionario de Datos.

El obtener como resultado sentencias SQL estándares se podría valorar como una extensión de esta característica de Multiplataforma, referido a múltiples sistemas operativos y metadatos de múltiples bases de datos, ya que adoptar el estándar SQL 3 permitirá que el resultado sea aplicable en otras tantas bases de datos, independientemente de la existencia de metadatos en la base de datos de producción donde aplicaremos las consultas y operaciones generadas automáticamente con 'Prometeo'.

### **Antecedentes**

#### **Enseñanza de SQL en la Universidad**

En la experiencia docente de enseñar el lenguaje SQL, la cátedra a la que pertenece uno de los Coautores, elabora anualmente las estadísticas de los resultados de los Alumnos, en ella podemos extraer los resultados del éxito medido en regularización, logro de promoción y fracaso de los estudiantes de 3º año de la Carrera de Ingeniería de Sistemas de Información de la Universidad Tecnológica Nacional, con una base de 500 alumnos por cuatrimestre, alcanzando el 30 % de alumnos libres, la regularización el 70% restante, con un 15% de promoción práctica, es decir habilidad en diseño de un modelo relacional y competencia en escribir sentencias SQL.

La conclusión es que el desarrollo de sentencias SQL es una tarea compleja, con una serie de dificultades puntuales no derivadas de la pericia con el lenguaje SQL en sí del desarrollador, sino de factores como dependencia del conocimiento del modelo de datos objetivo, que muy comúnmente no cuenta con documentación elaborada por los desarrolladores.

Una herramienta que ayude a lograr el objetivo, ayudando al desarrollador con las complejidades del medio es un factor que agrega valor al trabajo del desarrollador.

### **La experiencia de SQL en la empresa**

El argumento inicial de SQL como el lenguaje que le permitiría conocer al gerente ó supervisor sobre los hechos de la empresa, totales, promedios, listas de producto con sólo escribir la sentencia en un lenguaje similar al ingles, fue probado como tan difícil de concretar como otras promesas de marketing de la Industria de la Informática (que no es la única en abusar de los mensajes engañosos).

La complejidad del modelo relacional hace que el desarrollador tenga que escribir unas pocas, de un par a unas decenas de, líneas de código SQL para lograr que las filas buscadas, aparezcan en el monitor o en un archivo en un formato carácter, y así poder ofrecer al ocupado gerente, un reporte de algún indicador de importancia de un sistema de la empresa.

El esfuerzo enunciado así, podría indicar al lector inexperto que para el desarrollador es objetivamente muy sencillo resolver esto y que su conocimiento de SQL es el único requisito, pero la experiencia nos indica que por mas que se domine el lenguaje y sus formas de escribir las consultas, el conocimiento del modelo de Datos es definitivamente el factor que influye en la productividad del desarrollador.

Del uso profesional conocemos en Oracle, a Query Builder de la suite Developer que ha sido una referencia para este trabajo, pero que nuestro objetivo es un generador mas activo, que provea más opciones en base a la información guardada en el diccionario.

Otro ejemplo que sirve como referencia es Discoverer, y lo que mas nos motivo de esta herramienta es la productividad alcanzada por usuario final, inexperto total en SQL.

En Microsoft existen herramientas al estilo pero que no estudiaremos por falta de tiempo y carencia de la suficiente experiencia previa. En SqlServer, y Fox esta incorporado un asistente de consultas, pero por el mismo motivo anterior no es considerado en el trabajo.

Otros proveedores ofrecen herramientas similares, y usando buscadores de Internet buceamos en este conjunto

Ingresando en Yahoo y Google, como primer intento, nos dan unos 3900 sitios en el primer caso y 8400 en el segundo, luego de un filtrado buscando. Algunos sitios son de productos afines son enunciado en Bibliografía y Referencias.

### **Objetivos Iniciales**

Se definió como objetivo inicial estudiar el mercado para detectar herramientas que tengan el mismo propósito para conocer características positivas y aspectos negativos o limitaciones.

Se definió que se debería lograr un diagnóstico, a través del estudio de Factibilidad Técnica y Operativa, básicamente para contestar a las preguntas:

¿Estamos en condiciones de cumplir el objetivo?

Sobre el Diccionario de Oracle, se elaboro una heurística para definir la programación a realizar. Esta forma parte del estudio y una vez concretada si hicieron pruebas testigo de su factibilidad.

Como producto del la investigación se genero un informe de Factibilidad que en resumidas cuentas, define al trabajo a realizar “programable y factible de contruir” y que la tarea deberia comprometer a un grupo de desarrolladores de dos o tres integrantes y con algunas estimaciones de esfuerzos necesarios.

¿Es utilizable y agregará valor a los motores desarrollados?

Esta pregunta fue respondida satisfactoriamente por los motivos enunciados en la Introducción.

Se definió el algoritmo a emplear y estamos enunciando el plan de trabajo que abarque las tareas de construcción de 'Prometeo'.

## Detalle de las características Iniciales de Prometeo

'Prometeo' deberá contactarse con el diccionario de datos de la base de Datos Objetivo y una vez definido sobre que conjunto de tablas se va a trabajar, creara y guardara en el diccionario todas las consultas posibles generadas en modo lotes, es decir que permitirá definirles preferencias o indicaciones de que hacer con el resultado, que podría ser crear consultas, si la preferencia definida lo indica, crear vistas en base a las consultas, asignarle, vía un patrón, a todas un nombre, o explícitamente cuando termine con el proceso de una consulta que el usuario con el nombre que el usuario decida. También tenemos previsto poder generar los bloques de programación, en PL/Sql (propiedad de Oracle) y Java. Estos bloques y clases permitirán insertar filas y actualizar, en forma de métodos constructores, set y get

'Prometeo' debería permitir:

- Consultar tablas, vistas y objetos y sus columnas del diccionario de Datos, en forma general o a través de criterios de a uno, o por esquemas o por nombres de tablas de diferentes esquemas.
- Crear consultas para guardarlas como tales en las tablas específicas de la Herramienta.
- Consultar las tablas de Prometeo para lista las consultas y vistas creadas anteriormente, y modificarlas creando otra versión sin pisar la original y con la opción de reemplazar la anterior
- Crear en base a las consultas anteriores Vistas con el nombre por defecto o darle la posibilidad de darle un nombre distinto y guardar registro de documentación sobre esa vista en las tablas especiales de la Herramienta.
- Las construcciones generadas se guardaran en Proyectos para agruparlos en una entidad de orden superior que permita organizarlos.



- Recopilar la información documentándole contenido de los Comentarios sobre tablas y columnas disponibles en las vistas del diccionario ALL\_TAB\_COMMENTS y la ALL\_COL\_COMMENTS

Para estos objetivos deberá acceder al modelo de datos que almacene los Metadatos es decir la información sobre las tablas, columnas, constraints o relaciones, índices existentes en la Base de datos definida, en una primera instancia se trabajara sobre Oracle 10g.

Esto ayudaría al desarrollador, no experto en el modelo de datos, a obtener el resultado sin escribir por sí mismo código SQL.

Para una consulta mas compleja, una vez elegidas las tablas y columnas participantes, se debe continuar obteniendo mas información sobre los filtros de la consulta y que permitan al usuario de Prometeo otras subconsultas, los agrupamientos, las restricciones del agrupamiento y finalmente el criterio de Ordenamiento.

Un esquema similar debería seguirse con las sentencias DML Insert, update, delete y DDL create, alter, drop, truncate, para generar código procedimental para PL/SQL en Oracle, PSQL en PostgreSQL y los métodos constructores, set y get de Java.

Un ejemplo de lo expuesto es si una Tabla , a la que se quiere analizar tiene una o varias claves foraneas, 'Prometeo' debería construir lo siguiente:

1 Una consulta con todas las columnas descriptoras, llamamos así a las columnas Varchar2, date, timestamp, es decir las no numericas.

2 Con los datos numéricos una consulta con cada función de acumulación y en las preferencias se podría definir una funcion por defecto y así podría crear solamente una consulta separada del resto.

3 Una consulta con los datos numéricos con la función de acumulación definida en las preferencias, agrupadas por cada columna con constraints de claves foráneas, estas columnas habitualmente son definidas como dimensiones en DataWarehousing.

4 Una consulta con los datos numéricos con la función de acumulación en las preferencias agrupadas por las columnas claves foráneas definidas como dimensiones y como claves primarias.

5 Todas las consultas con join contra las tablas referenciadas por todas las columnas claves foráneas.

6 En caso de que las columnas tengan claves foráneas sobre si misma escribir las consultas de self joins y ver, si se implementan, consultas jerárquicas top-bottom y bottom-top.

7 Procedimientos de actualización de cada columna, de inserción y de borrado de registros, mas la posibilidad de generar en Java, los metodos constructores y los get y sets de cada columna.

8 Si el desarrollador lo indicara, 'Prometeo' debería traducir a todas estas consultas (hasta el punto 6) vistas recibiendo el nombre cada una de ellas con un criterio propio 'Patron' o definido uno a uno por el usuario.

9. Mostrar la información residente en las vistas ALL\_TAB\_COMMENTS y ALL\_COL\_COMMENTS, ALL\_INDEXES, ALL\_CONSTRAINTS y ALL\_COL\_CONSTRAINTS y generar un archivo html y/o documento para Reunir y documentar lo registrado para esas tablas.

10. Prometeo podría generar datos de prueba para llenar esta tabla observando los constraints para insertar 1, 10 o la cantidad que se definar en las preferencias.

## **Experiencia con herramientas similares**

En la experiencia laboral como usuarios-desarrolladores de aplicaciones con herramientas de desarrollo, hemos tenido particularmente en contacto con el generador de Consultas 'Query Builder' de la suite Oracle Developer que es utilizada como herramienta autónoma para crear archivos de textos con la consulta generada tanto como integrante del proceso de creación de Record Groups dentro de Form Builder o de la consulta principal de un reporte en Report Builder.

También hemos tenido contacto con una herramienta de desarrollo de consultas Discoverer de Oracle, cuya complejidad y alcance constituye un nivel de abstracción mayor, dirigido a los tomadores de decisiones. Con esta herramienta es mayor aún el valor que agrega una herramienta de construcción de sentencias SQL, en este caso con orientación a la consulta y no a la actualización vía DML o DDL.

Se destaca que el nivel de satisfacción con las herramientas nombradas es muy alto. Cabe destacar que estas herramientas son diseñadas y construidas para trabajar sobre bases de datos Oracle y vía ODBC se conectan a otras Bases de Datos.

El usuario, lego en SQL, pero diestro en herramientas como Excel de Microsoft, enseguida valora el acceso a información que antes tenía que cargar por su cuenta y sin necesidad de conocer SQL ni la complejidad del modelo de datos subyacente en la base de datos.

En este caso, es para destacar el desarrollo alcanzado por la herramienta de Oracle en el uso de un modelo de datos (llamado End User Layout - EUL) que almacena los metadatos que ayudan a construir las consultas.

Este modelo de datos constituye un basamento para decidir sobre las características de los Metadatos en la construcción de Nuestra Herramienta.

También es muy difundido el contacto con herramientas que cuentan con asistentes de escritura de SQL, como Microsoft Visual Fox, Cristal Reports, etc.

En las reuniones de trabajo para la enunciación de este proyecto destacamos que en todos los paquetes de software nombrados anteriormente, encontramos asistentes de generación de consultas, con distintos alcances y resultados. En ellos encontramos la posibilidad de ver el resultado final en una sentencia SQL, indentada y alineada apropiadamente, para después guardarlo en un archivo plano o en una columna adecuada de la base de datos y así poder recuperarla para usarla mas tarde. También queremos esta desarrollar esta característica para que lo tenga Prometeo.

También es destacable que los productos utilizados acceden al diccionario de datos o interpretan las cabeceras de los archivos de datos, de donde obtiene los Metadatos. 'Prometeo' Herramienta debería realizar lo mismo y también ser capaz de almacenar esa información en nuestro modelo de datos en forma de metadatos, pero en tablas que pertenezcan a un usuario definido en la instalación.

El acceso a esta información será muy valiosa porque ayuda a tener precisión en los nombres de columnas y tablas, que utilizaremos en la consulta, el uso de alias de tablas y de columnas para no recargar la tarea de elaborar la consulta propiamente dicha sin distracción por los detalles menores de la construcción.

Encontramos como un defecto, en la mayoría de estos productos, la ausencia de la posibilidad de construir subconsultas, es decir sentencias select dentro de otra sentencia select. Esta característica que no encontramos en los productos utilizados, va a ser evaluada para ser incorporada en Prometeo.

### **Lenguajes a utilizar**

Los objetivos de Multiplataforma de ejecución, nos orientan a la familia de C, es decir C, C++ y Java como los lenguajes que permitirán construir Nuestra Herramienta, con características

visuales, con conexión a las bases de datos y que puedan ejecutarse en cualquier Sistema operativo con la instalación adecuada, es decir previendo la instalación de las librerías C, o de la JavaVirtual Machine correspondiente.

### **Estado Actual**

Se definieron los Objetivos Iniciales, se enunciará un plan de trabajo para alcanzar el objetivo, una vez completado el grupo de desarrolladores que integraremos como Becarios.

Estamos iniciando es estudio de las herramientas de objetivos similares a Prometeo e inmediatamente después, tercera semana de Junio aproximadamente, iniciaremos el tronco principal de escribir las sentencias SQL, ya que la obtención de metadatos la realizaremos por separado para hacer componentes especializados en cada motor de base de datos a estudiar, comenzando con Oracle.

Estamos estudiando que 'Prometeo' pueda generar código XML, ya que conocemos productos 'Reports de Oracle Developer', que aceptan definiciones escritas en este lenguaje y que se podría obtener resultados ya formateados con templates de finidos que le ayudarían al desarrollador a finalizar el trabajo con terminación profesional sin mayor esfuerzo que el enunciado anteriormente.

### **Conclusión**

Por varios motivos, ya sea de la herramienta en si, como el aporte que le hará a TecnoDBA, consideramos valioso el desarrollo de 'Prometeo', pero quisimos buscar antecedentes comparables para poder mejorar los requerimientos del producto.

Por esta definición a priori, en el Plan de tareas esta previsto construir un método, para formalizar las características y plantear el funcionamiento del programa de generación de código.

Luego de los estudios tenemos previsto el desarrollo de un Prototipo en Java como un objetivo para Septiembre-Octubre para tener la versión Alfa del producto.

El pulso del trabajo esta sostenido por ver, en las primeras búsquedas en Internet, que son pocos los productos que se lanzan al mercado y que no cubren aparentemente los objetivos planteados. En las fases avanzadas del estudio inicial esto se analizó como una fortaleza.

## Bibliografía y Referencias:

### Bibliografía

Documentos ACM

Formal semantics of SQL queries

M. Negri, G. Pelagatti, L. Sbatella

Quickly generating billion-record synthetic databases

Jim Gray, Prakash Sundaresan, Susanne Englert, Ken Baclawski, Peter J. Weinberge

SQL/XML and the SQLX Informal Group of Companies, Andrew Eisenberg and Jim Melton,

ACM SIGMOD Record, Vol. 30 No. 3, Sept. 2001,

<http://www.acm.org/sigmod/record/issues/0109/standards.pdf>.

Database Languages---SQL-Part 14: XML-Related Specifications (SQL/XML)---Final Committee

Draft, H2-2002-063, WG3:ICN-011, Jim Melton (Editor), March 2002,

<ftp://sqlstandards.org/SC32/WG3/Progression Documents/FCD/4FCD1-14-XML-2002-03.pdf>.

Learning SQL with a computerized tutor

Antonija Mitrovic

March 1998

ACM SIGCSE Bulletin , Proceedings of the twenty-ninth SIGCSE technical symposium on

Computer science education, Volume 30 Issue 1

Efficiently publishing relational data as XML documents

September 2001

The VLDB Journal The International Journal on Very Large Data Bases, Volume 10 Issue 2-3

Efficient evaluation of XML middle-ware queries

Mary Fernandez, Atsuyuki Morishima, Dan Suciu

May 2001

ACM SIGMOD Record , Proceedings of the 2001 ACM SIGMOD international conference on

Management of data, Volume 30 Issue 2

An SQL-based query language for networks of relations

Amit Basu, Rafiul Ahad

March 1990

ACM SIGMOD Record, Volume 19 Issue 1

Retrieval effectiveness of an ontology-based model for information selection

Latifur Khan, Dennis McLeod, Eduard Hovy

January 2004

The VLDB Journal The International Journal on Very Large Data Bases, Volume 13 Issue 1

An algebraic approach to static analysis of active database rules

Elena Baralis, Jennifer Widom

September 2000

ACM Transactions on Database Systems (TODS), Volume 25 Issue3

Research sessions: XML I: Storing and querying ordered XML using a relational database system

Igor Tatarinov, Stratis D. Viglas, Kevin Beyer, Jayavel Shanmugasundaram, Eugene Shekita,

Chun Zhang June 2002

Proceedings of the 2002 ACM SIGMOD international conference on Management of data

Rule based database access control—a practical approach

Tor Didriksen

November 1997

Proceedings of the second ACM workshop on Role-based access control

SQL standardization: the next steps  
Andrew Eisenberg, Jim Melton March 2000  
ACM SIGMOD Record, Volume 29 Issue 1

Database & data management: The forest and the trees: using oracle and SQL server together to teach ANSI-standard SQL

Gary B. Randolph  
October 2003

Proceeding of the 4th conference on Information technology curriculum

SQLator: an online SQL learning workbench

Shazia Sadiq, Maria Orlowska, Wasim Sadiq, Joe Lin  
June 2004

ACM SIGCSE Bulletin , Proceedings of the 9th annual SIGCSE conference on Innovation and technology in computer science education, Volume 36 Issue 3

### **Referencia sobre herramientas Similares**

JmSD-Query Generator-- <http://www.jmsdesigner.8m.com/query/>

**ASKME** es una interfaz poderosa y amigable entre el **IQL** (IdeaFix Query Language) y el usuario final para la generación de consultas. <http://www.datafix.com.ar/askme.htm>

Un generador de salidas en excel que se conecta a diferentes bases de datos  
<http://www.apesoft.com/dcreport.asp>

Un interesante proyecto de desarrollo

[http://www.fing.edu.uy/inco/proyectos/linkall/index\\_sp.php](http://www.fing.edu.uy/inco/proyectos/linkall/index_sp.php)

PostgreSQL curso demostración del pgaccess

<http://www.programacion.net/bbdd/tutorial/intropostgresql/5/>

<http://es.tldp.org/Postgresql-es/web/navegable/Howto/PostgreSQL-COMO-11.html>

### **Marcas de Productos Comerciales de generación de Reportes,**

Forest & Trees.

### **Marcas de Productos Comerciales de desarrollo de Aplicaciones**

Se contactó y bajó para instalar dos productos que según su promoción realizarían tareas comparables con los requerimientos definidos para Nuestra Herramienta.

**DB Artisan** de TSG Group y

**Rapid SQL** de Embarcadero Technologies Products

Entre las primeras aproximaciones a los productos detectamos que ofrecían soporte a comunicarse con múltiples bases de datos, Oracle, DB2, Sybase, Sql Server.

Actualmente estamos en fase de Instalación y Prueba de las dos herramientas