Documento Técnico

Pruebas de rendimiento bases de datos columnares vs bases de datos orientadas a filas.

Fecha de Creación: 04/05/2012

info@stratebi.com

@stratebi

www.stratebi.com - www.todobi.com

stratebi
open business intelligence

1. Introducción a las bases de datos columnares

Como su nombre lo indica, las bases de datos están organizadas de columna por columna en lugar de la fila: es decir, todos los casos de un solo elemento de datos (por ejemplo, Nombre de Persona) se almacenan de modo que se puede acceder como una unidad. Esto las hace especialmente eficaces en las consultas analíticas, como la lista de selecciones, que a menudo lee unos pocos elementos de datos, pero necesitamos ver todas las instancias de estos elementos. En contraste, en una base de datos relacional convencional los datos se almacenan por filas, por lo que toda la información de un registro (fila) es inmediatamente accesible. Esto tiene sentido para las consultas transaccionales, que suelen referirse a todo el contenido de un registro.

Hoy los sistemas columnares combinan su estructura columnar con técnicas que incluyen la indexación, compresión y paralelización.

- ***Tiempo de carga:*** ¿Cuánto tiempo se necesita para convertir datos de origen en el formato de columna? Esta es la pregunta más básica de todas. Tiempos de carga son a menudo medidos en gigabytes por hora, que puede ser extremadamente lento, cuando de decenas o cientos de gigabytes de datos se trata. La cuestión a menudo carece de una respuesta sencilla, porque la velocidad de carga puede variar en función de la naturaleza de los datos y las elecciones realizadas por el usuario. Por ejemplo, algunos sistemas pueden almacenar varias versiones de los mismos datos, ordenados en diferentes secuencias o en los diferentes niveles de agregación. Los usuarios pueden construir un menor número de versiones a cambio de una carga rápida, pero puede pagar un precio más adelante con consultas más lentas.

- ***Carga Incremental:*** Una vez que un conjunto de datos se ha cargado, todo debe ser recargado cada vez que hay una actualización. Muchos sistemas columnares permiten carga incremental, teniendo sólo los registros nuevos o modificados y la fusión de los datos anteriores (LucidDB permite cargas incrementales, mientras que InfoBright no dispone de esta funcionalidad en su versión community). Pero la atención al detalle es fundamental, ya que las funciones de carga incremental varían ampliamente. Algunas cargas incrementales tardan hasta una completa reconstrucción y algunos resultados son el rendimiento más lento, algunos pueden agregar registros, pero no cambiar o suprimirlos. Las Cargas incrementales a menudo deben completarse periódicamente con una reconstrucción completa.

- ***Compresión de datos***: Algunos sistemas columnares pueden comprimir mucho la fuente de datos y archivos resultantes a fin de tomar una fracción de espacio en el disco original. Puede ocasionar en estos casos un impacto negativo en el rendimiento por la descompresión de datos a realizar la lectura. Otros sistemas

utilizan menos compresión o almacenan varias versiones de los datos comprimidos, teniendo más espacio en disco, pero cobrando otros beneficios a cambio. El enfoque más adecuado dependerá de sus circunstancias. Tenga en cuenta que la diferencia de los requisitos de hardware pueden ser sustanciales.

- ***Técnicas de acceso:*** Algunas bases de datos de columnares sólo se pueden acceder utilizando su propio proveedor de lenguaje de consultas y herramientas. Estos pueden ser muy poderosos, incluyendo capacidades que son difíciles o imposibles usando el estándar SQL. Pero a veces faltan funciones especiales, tales como las consultas que comparan valores con o en los registros. Si necesita acceder al sistema con herramientas basadas en SQL, determine exactamente qué funciones SQL y dialectos son compatibles. Es casi siempre un subconjunto completo de SQL y, en particular, rara vez se dispone de las actualizaciones. También asegúrese de encontrar si el rendimiento de las consultas SQL es comparable a los resultados con el sistema de la propia herramienta de consulta. A veces, el ejecutar consultas SQL mucho más lento.

- ***Rendimiento:*** Los sistemas columnares por lo general superan a los sistemas de relaciones en casi todas las circunstancias, pero el margen puede variar ampliamente. Las consultas que incluyen cálculos o acceso individual a los registros puede ser tan lento o más que un sistema relacional adecuadamente indexado. Aquí podemos ver la potencia de estos sistemas de bases de datos cuando están aplicados a análisis.

- ***Escalabilidad:*** Uno de los principales objetivos de las bases de datos columnares es obtener buenos resultados en grandes bases de datos. Pero no puede asumir todos los sistemas pueden escalar a decenas o centenares de terabytes. Por ejemplo, el rendimiento puede depender de determinados índices de carga en la memoria, de modo que su equipo debe tener memoria suficiente para hacer esto. Como siempre, en primer lugar preguntar si el vendedor tiene en ejecución los sistemas existentes a una escala similar a la suya y hablar con las referencias para obtener los detalles. Si el suyo sería más grande que cualquiera de las instalaciones existentes, asegúrese de probar antes de comprar.

2. Entorno de la Prueba

Para la realización de los test de Rendimiento vamos a utilizar un esquema en estrella con una tabla de hechos (H_RRHH) con unos 4.300.000 registros que cuenta con 12 dimensiones asociadas, siendo la dimensión de personas (DIM_PERSONA) la más numerosa y que cuenta con 27.000 registros. Todas las tablas tienen indexados tanto los campos clave primaria como los que son ajena para buscar lograr una mejor eficiencia en los accesos.

Características Hardware del Sistema:

Procesador: Intel Core i3-2330M CPU @ 2,20 GHz @ 2,20 GHz

Memoria RAM instalada: 4,00 GB

Operativo: Windows 7 Home Premium 64 bits (Service Pack 1)

3. Instalación de LucidDB

1-) Como prerrequisito es necesario tener configurado el entorno virtual de Java (JRE)

2-) Descargarse de http://www.luciddb.org en la sección de descargas la versión que mejor se ajuste al sistema operativo en el que deseamos instalar lucidDB.

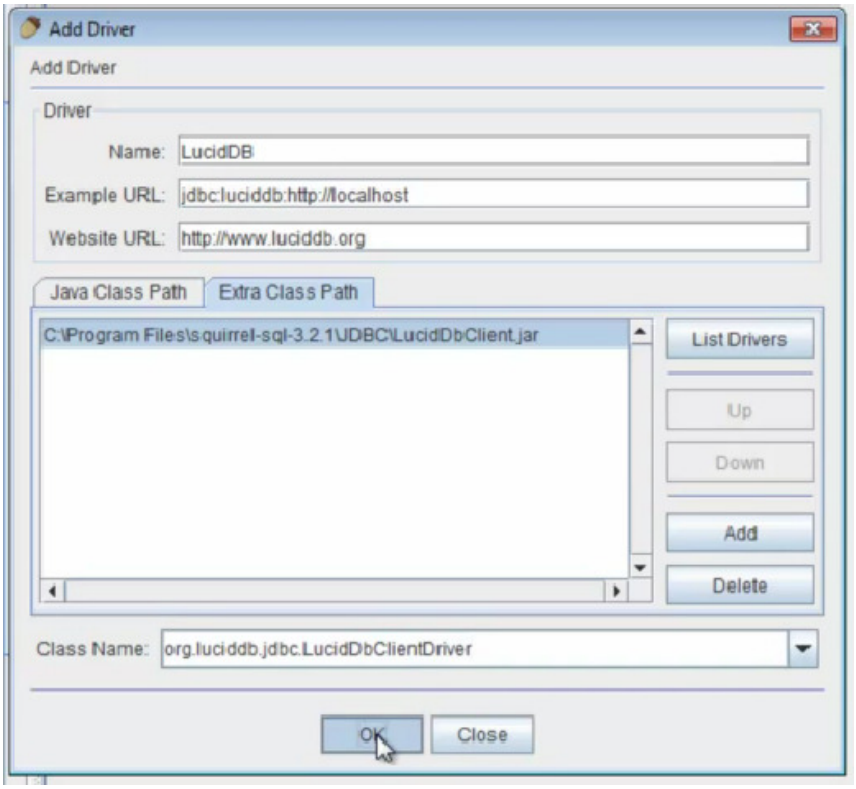
3-) Descomprimir el paquete y ejecutar desde línea de comandos el script install.bat que está dentro de la carpeta **/luciddb/install**

4-) LucidDB cuenta con 2 componentes principales, por un lado está el servidor y un cliente en consola. Primeramente debemos poner a ejecutarse el servidor, esto es muy sencillo basta con ejecutar en línea de comandos el script lucidDbServer.bat que se encuentra ubicado dentro del directorio **/luciddb/bin**. El servidor comenzará a escuchar peticiones de conexión y a prestar servicios en el puerto HTTP 8034

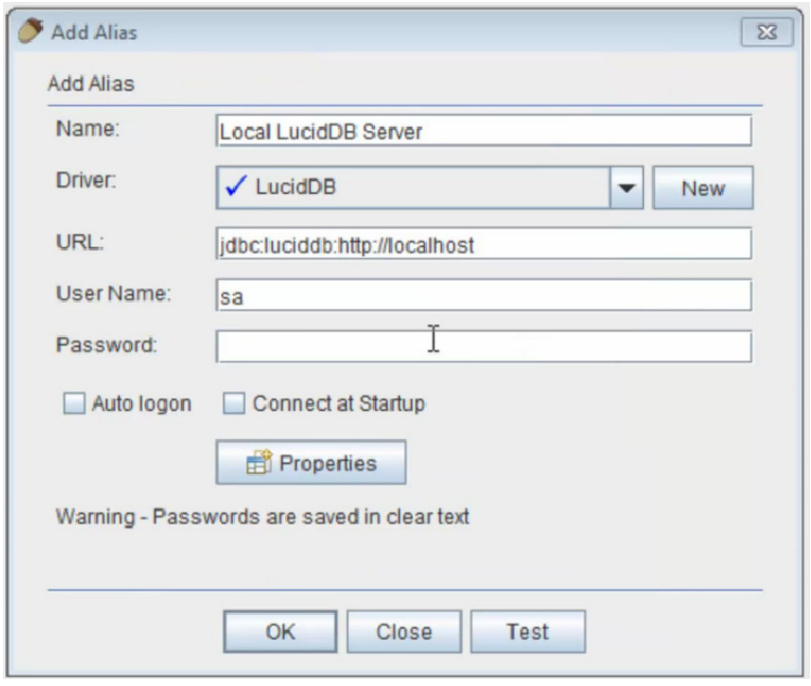
5-) Ahora vamos a instalar un cliente sql para trabajar más cómodos. Elegimos squirrel-sql por su integración con LucidDB, para ello nos descargamos el último .jar de su página de sourceforge sourceforge.net/projects/squirrel-sql.

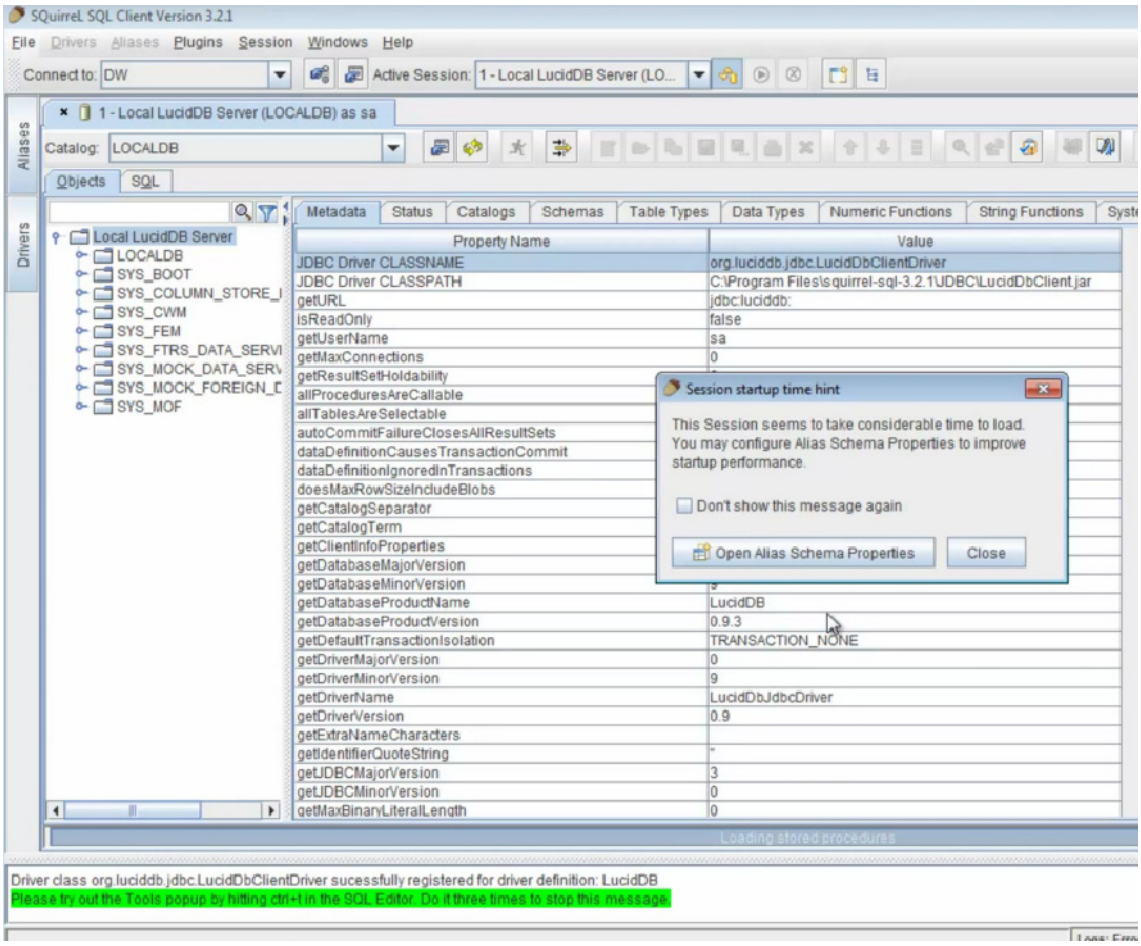
6-) Para la instalación de este cliente es necesario abrir la línea de comandos en modo administrador y ejecutar el comando **java –jar squirrel-sql-X.X.X.jar** que nos abrirá un breve asistente de instalación. A continuación debemos de crear una carpeta en el directorio raíz de nuestra instalación de squirrel (ej: C:\Program Files\squirrel-sql-3.3.0\JDBC) que llamaremos JDBC, en ella debemos copiar el driver JDBC (LucidDBClient.jar) de LucidDB, ubicado en la carpeta plugin de la instalación de LucidDB.

7- En este punto abrimos squirrel a través del script **squirrel-sql.bat** y hacemos click en la pestaña de la izquierda correspondiente a Drivers, y añadimos el driver de Lucid con la siguiente configuración que vemos en pantalla, recordar escoger el driver LucidDBClient.jar que hemos alojado en la carpeta JDBC.



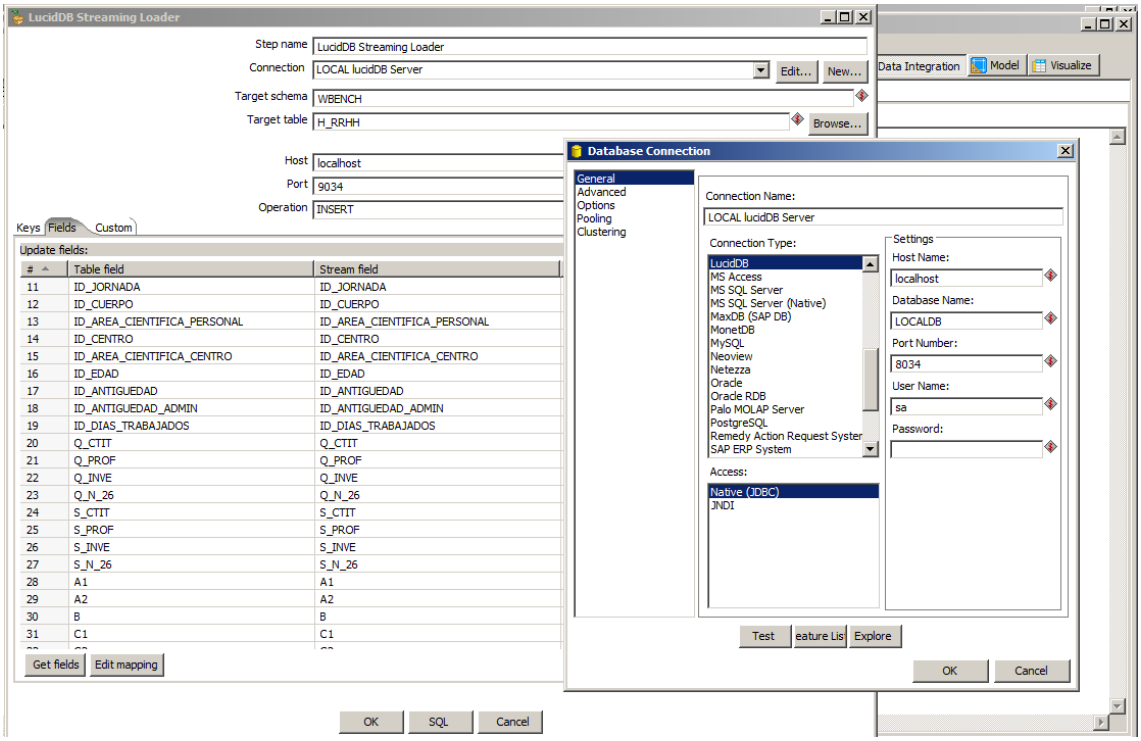
8 -) El siguiente paso sería crear la conexión desde squirrel, añadimos un alias como se muestra a continuación y nos conectamos, con lo que podremos ver los catálogos y esquemas de la base de datos





Carga de Datos en LucidDB

Para la carga de los datos en LucidDB, hemos empleado el paso de Kettle (versión 4.2.1) de carga a través de streaming. En las capturas de pantalla que se acompañan se pueden ver los parámetros necesarios para una correcta ejecución de la secuencia de carga. Comentar que el principal problema de la carga en este motor de bases de datos columnar ha sido la lenta velocidad de carga saliendo de media unos 50 registros/segundo, lo que alargo el periodo de carga en el caso de nuestra tabla de hechos a un día. Una de sus principales ventajas por otro lado es que permite pese a ser open source la realización de cargas incrementales y actualizaciones.



4. Instalación de InfoBright Community Edition

1-) Descargar de dentro de la sección Community la última versión de Infobright, desinstalar el zip y ejecutar el .exe con el instalador. En nuestro caso hemos utilizado la versión 4.0.5 en su versión de 64 bits. El instalador nos creará InfoBright como un servicio de Windows, que debemos si necesitamos cambiarlo de Automático a Manual para que no se ejecute permanentemente con el inicio del Sistema Operativo y así ahorrar recursos.

2-) InfoBright corre en el puerto 5029, con el usuario root y contraseña vacía por defecto.

3-) Podemos utilizar cualquier cliente Mysql, por ejemplo el MySQL Workbench o Toad.

4-) InfoBright comparte sintaxis con MySQL excepto en la carga y actualización de datos INSERT UPDATE y DELETE, que no son soportados.

5-) La creación de una base de datos y una tabla es idéntica a MySQL , la principal diferencia es que el motor que InfoBright utiliza es el denominado BrightHouse (mysql> create table <nombre_tabla> (<columna(s)>) engine=brighthouse;)

7-) Aspectos nuevos: IB incorpora un modificador llamado “lookup” para datos de tipo cadena de caracteres, en las columnas que se incluyen este valor se realiza automáticamente una sustitución por valores enteros. Se pueden crear en columnas CHAR y VARCHAR para incrementar su compresión y mejorar el rendimiento, solo es recomendable incluir este tipo de modificador en campos de texto con un pequeño número de valores distintos por ejemplo: estado, sexo o categoría puesto que todos los valores distintos se cargan en RAM.

8-) IB utiliza una tecnología de auto aprendizaje en lugar de los índices tradicionales por lo que los siguientes parámetros de la creación de las tablas no están soportados: claves, columnas únicas, columnas autoincreméntales e índices. Tampoco están soportados dentro de IB los valores por defecto ni referencias a otras tablas de las columnas de una tabla.

Es posible ver la información del tamaño de las columnas de una tabla en disco a través del siguiente comando.

show **full** columns from nombre_tabla;

Carga de datos en InfoBright Community Edition

Hemos realizado la inserción a través del comando LOAD DATA INFILE que lee registros desde un fichero de texto a una tabla a muy alta velocidad, dado que el paso de Kettle nos ha dado varios problemas de carga. Comentar que las cargas en InfoBright por medio de este comando han resultado extremadamente rápidas pero existe el problema de que en esta versión community no es posible realizar cargas incrementales, algo que resulta de vital importancia en grandes volúmenes de datos.

En la siguiente captura de pantalla se muestra el cliente que InfoBright incorpora y que podemos ejecutarlo desde **Inicio -> InfoBright -> InfoBright Command Line Client**. Destacar la rapidez de la carga (1 minuto) de un fichero csv con los datos de la tabla de hechos con más de 4 millones de registros.

5. Pruebas de Rendimiento

En esta sección adjuntamos las 5 consultas que el servidor OLAP Mondrian generó automáticamente, tras hacer drill a través de tres cubos idénticos que apuntan a diferentes motores de bases de datos. Dos cubos tienen como origen de sistemas de base de datos columnares (InfoBright CE y LucidDB) mientras que el otro tiene como fuente un servidor de bases de datos Oracle 11 g tradicional.

Query 1:

```
SELECT COUNT(DISTINCT ID_PERSONA) as m0
FROM H_RRHH;
```

Query 2:

```
SELECT DIM_PERSONA.NOMBRE_COMPLETO as c0, COUNT(DISTINCT H_RRHH.ID_PERSONA) as m0
FROM DIM_PERSONA, H_RRHH
WHERE H_RRHH.ID_PERSONA = DIM_PERSONA.ID_PERSONA
GROUP BY DIM_PERSONA.NOMBRE_COMPLETO;
```

Query 3:

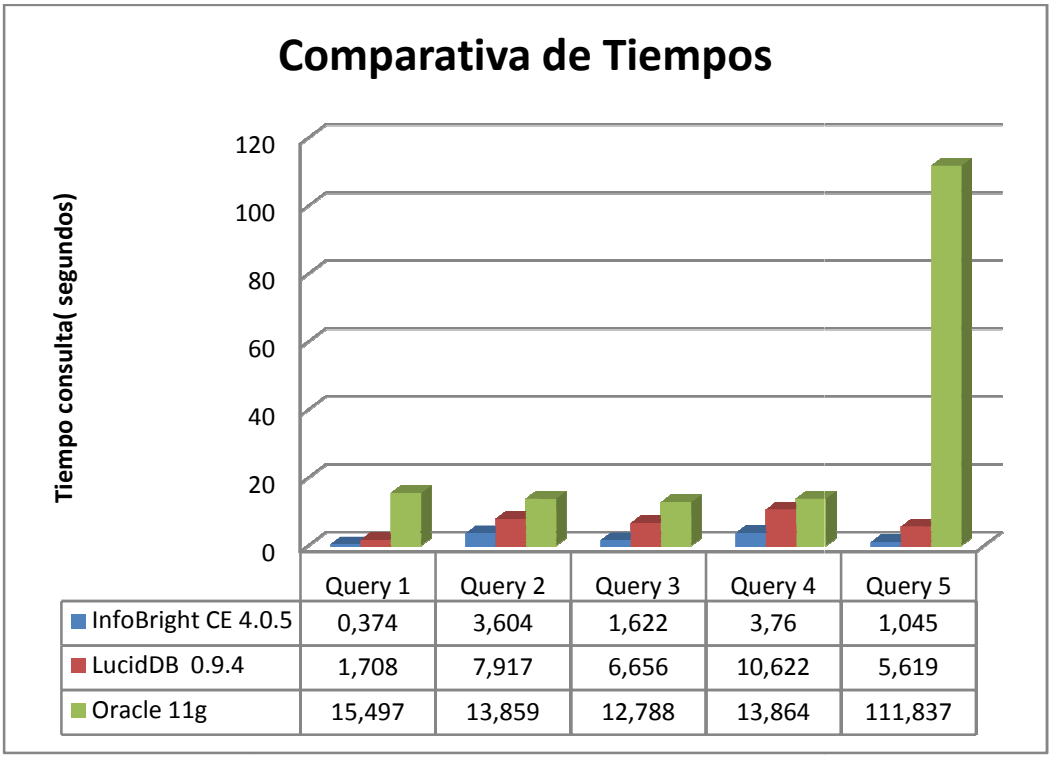
```
SELECT DIM_STRATEBI.DESC_CORTA as c0, COUNT(DISTINCT H_RRHH."ID_PERSONA") as "m0"

FROM DIM_STRATEBI, H_RRHH

 WHERE H_RRHH.ID_STRATEBI = DIM_STRATEBI.ID_STRATEBI

 GROUP BY DIM_STRATEBI.DESC_CORTA
```

Query 4:

```
SELECT DIM_AREAFUNCIONAL.DESC_CATEGORIA
, DIM_GRADO_ACADEMICO.DESC_CORTA , DIM_CATEGORIA_GRUPO.DESC_CLASIFICACION ,
DIM_CATEGORIA_GRUPO.DESC_CATEGORIA_GRUPO ,
DIM_CATEGORIA_GRUPO.CAT_2 , DIM_AREA_CIENTIFICA.ID_AREA_CIENTIFICA_ODS ,
COUNT(DISTINCT H_RRHH.ID_PERSONA) as m0
FROM DIM_AREAFUNCIONAL , H_RRHH , DIM_GRADO_ACADEMICO , DIM_CATEGORIA_GRUPO,
DIM_AREA_CIENTIFICA
WHERE H_RRHH.ID_AREAFUNC = DIM_AREAFUNCIONAL.ID_AREA_FUNCIONAL and
DIM_AREAFUNCIONAL.DESC_CATEGORIA = 'Ingenieros'
and H_RRHH.ID_GRADO_ACADEMICO = DIM_GRADO_ACADEMICO.ID_GRADO_ACADEMICO
and H_RRHH.ID_CUERPO = DIM_CATEGORIA_GRUPO.ID_CATEGORIA_GRUPO
and DIM_CATEGORIA_GRUPO.DESC_CLASIFICACION = 'Grupo'
and DIM_CATEGORIA_GRUPO.DESC_CATEGORIA_GRUPO in ('A', 'B', 'C', 'D')
and H_RRHH.ID_CUERPO = DIM_CATEGORIA_GRUPO.ID_CATEGORIA_GRUPO
and H_RRHH.ID_AREA_CIENTIFICA_PERSONAL = DIM_AREA_CIENTIFICA.ID_AREA_CIENTIFICA
GROUP BY DIM_AREAFUNCIONAL.DESC_CATEGORIA,
DIM_GRADO_ACADEMICO.DESC_CORTA,
DIM_CATEGORIA_GRUPO.DESC_CLASIFICACION,
DIM_CATEGORIA_GRUPO.DESC_CATEGORIA_GRUPO ,
 DIM_CATEGORIA_GRUPO.CAT_2,
DIM_AREA_CIENTIFICA.ID_AREA_CIENTIFICA_ODS;
```

Query 5:

```
SELECT DIM_STRATEBI.DESC_CORTA as c0, DIM_TIEMPO.ANNO4 as c1 , DIM_TIEMPO.ID_MES as c2,

COUNT(DISTINCT H_RRHH.ID_PERSONA) as m0

FROM DIM_STRATEBI, H_RRHH, DIM_TIEMPO

WHERE H_RRHH.ID_STRATEBI= DIM_STRATEBI.ID_STRATEBI

 and DIM_STRATEBI.DESC_CORTA = 'Stratebi_Staff'

and H_RRHH.ID_TIEMPO = DIM_TIEMPO.ID_TIEMPO

and DIM_TIEMPO.ANNO4 = '2011'

and DIM_TIEMPO.ID_MES in (908, 1008, 1108, 1208)

GROUP BY DIM_STRATEBI.DESC_CORTA, DIM_TIEMPO.ANNO4, DIM_TIEMPO.ID_MES;
```



	Query 1	Query 2	Query 3	Query 4	Query 5
InfoBright CE 4.0.5	0,374	3,604	1,622	3,76	1,045
LucidDB 0.9.4	1,708	7,917	6,656	10,622	5,619
Oracle 11g	15,497	13,859	12,788	13,864	111,837

A la vista de los resultados vemos como InfoBright CE es la que mejor rendimiento tiene en todas las pruebas, sin embargo cuenta con el ya mencionado problema de la carencia de cargas incrementales. Decir también que las dos bases de datos columnares poseen menores tiempos de ejecución debido a la naturaleza analítica de las mismas. Gracias a los buenos resultados de las bases de datos orientadas a columnas nos podríamos poner manos a la obra y cambiar entornos de producción tradicionales con motores de bases de datos estables en busca de mejorar su rendimiento. ¿Qué opinas, te gustaría realizar una prueba de concepto con tus datos? No lo dudes contacta con nosotros, los resultados pueden ser asombrosamente positivos.

6. Información Stratebi

Stratebi es una empresa española, ubicada en Madrid y Barcelona, líderes en España en soluciones Business Intelligence Open Source.

En Stratebi nos planteamos como **objetivo** dotar a las compañías e instituciones, de herramientas escalables y adaptadas a sus necesidades, que conformen una estrategia Business Intelligence capaz de rentabilizar la información disponible. Para ello, nos basamos en el desarrollo de soluciones de Inteligencia de Negocio, mediante tecnología Open Source.

Stratebi está compuesto por **profesores y responsables de proyectos del Master en Business Intelligence de la Universidad UOC.**

Los profesionales de Stratebi son los creadores y autores del primer weblog en español sobre el mundo del Business Intelligence, Data Warehouse, CRM, Dashboards, Scorecard y Open Source.

Todo Bi, se ha convertido en una referencia para el conocimiento y divulgación del Business Intelligence en español.

 @TodoBI_OS