

**UNIVERSIDAD GABRIELA MISTRAL
FACULTAD DE INGENIERÍA**

**BASES DE DATOS ORIENTADAS A LA
COLUMNA**

Memoria para optar al título de Ingeniero de Ejecución en Informática

Autor : Juan Edmundo Teke Fuentes.

Profesor Guía : Roberto Caru Cisternas

Profesor Integrante : Jorge Tapia Castillo

Puerto Montt – Chile

Agosto, 2013

Índice de contenido

I. HIPÓTESIS	6
II. OBJETIVO GENERAL.....	6
III. OBJETIVO ESPECIFICO	6
IV. INTRODUCCIÓN	7
IV.1 ¿Qué es una Base de Datos?.....	7
V.2 Hay cuatro modelos principales de bases de datos:	8
IV.2.1 Modelo jerárquico:	8
IV.2.2 Modelo en red :	9
IV.2.3 Modelo relacional:	10
IV.2.4 Modelo de bases de datos deductivas:	11
IV.2.4 Modelo de bases de datos deductivas:	11
V. Otros modelos que se están implementando son:	12
V.1 Bases de datos orientadas a objeto:.....	12
V.2 Modelo de base de datos orientado a la columna.	13
VI. Definición base de datos orientado a la columna.....	14
VII Modelo de Almacenamiento Descompuesto	16
VIII DSM vs NSM	18
VIII.1 Elimina valores nulos	18
VIII.2 Método de Acceso Uniforme.....	19
VIII.3 Soporte de atributos multivaluados.....	20
VIII.4 Soporte de Entidades	21
VIII.5 Soporte de múltiples relaciones padres	21
VIII.6 Soporte de registros heterogéneos.....	22
IX Variables de funcionamiento a considerar	23
IX.1 Tiempo de carga	23
IX.2 Carga Incremental	23
IX.3 Compresión de datos.....	24

IX.4 Técnicas de acceso	25
IX.5 Rendimiento.....	25
IX.6 Escalabilidad.....	25
X Principales Bases de Datos Columnares.....	26
X Principales Bases de Datos Columnares	26
X.1 Base de Datos Cassandra	26
X.1 Base de Datos Cassandra	26
X.2 PROYECTO GEMINI.....	29
X.3 LUCIDDB	30
IX.4 NFOBRIGHT.....	32
X.5 VERTICA	34
X.6 QD TECHNOLOGY	37
X.7 SYBASE	40
X.8 ParAccel	43
XI BDOC vs BDOF	46
XII Teradata Corporation (NYSE: TDC)	48
XIII Funcionamiento de la base de datos orientada a la columna.....	53
XIV Pruebas realizadas.....	57
XIV.1 Entorno de la Prueba	57
XIV.2 Características Hardware del Sistema:	57
XIV.3 Instalación de LucidDB	58
XIV.4 Carga de Datos en LucidDB.....	62
XIV.5 Instalación de InfoBright Community Edition	63
XIV.6 Carga de datos en InfoBright Community Edition	65
XIV.7 Consultas realizadas a la base de datos con los distintos motores de búsqueda de datos.....	66
XIV.8 Resultados	70
XV Conclusiones.....	71
XV.1 Ventajas.....	71
XV.1 Ventajas	71

XV.2 Desventajas.....	73
XVI Glosario	74
XVII Bibliografía.....	77

I. HIPÓTESIS

Esta investigación pretende demostrar que la tecnología de base de datos orientada a la columna resuelve el problema de los grandes volúmenes de datos y velocidad de las consultas siendo una alternativa valida a soluciones como Teradata.

II. OBJETIVO GENERAL

Mostrar la tecnología de base de datos orientadas a columnas y sus diferentes sistemas de manejo de base de datos tanto open source como pagados y las ventajas de este sobre el actual modelo de base de datos relacional.

III. OBJETIVO ESPECIFICO

Evaluar el rendimiento, la portabilidad y usabilidad de las tecnologías que aplican el uso de la de base de datos orientadas a columna.

IV. INTRODUCCIÓN

IV.1 ¿Qué es una Base de Datos?

Las bases de datos son cualquier conjunto de datos organizados para su almacenamiento en la memoria de un computador, diseñado para facilitar su mantenimiento y acceso de una forma estándar. Los datos suelen aparecer en forma de texto, números o gráficos. Desde su aparición en la década de 1950, se han hecho imprescindibles para las sociedades industriales.



Figura 1: ejemplo base de datos

IV.2 Hay cuatro modelos principales de bases de datos:

IV.2.1 Modelo jerárquico: Una base de datos jerárquico, como su nombre indica, almacenan la información en una estructura jerárquica que enlaza los registros en forma de estructura de árbol, en donde un nodo padre de información puede tener varios nodos hijo.

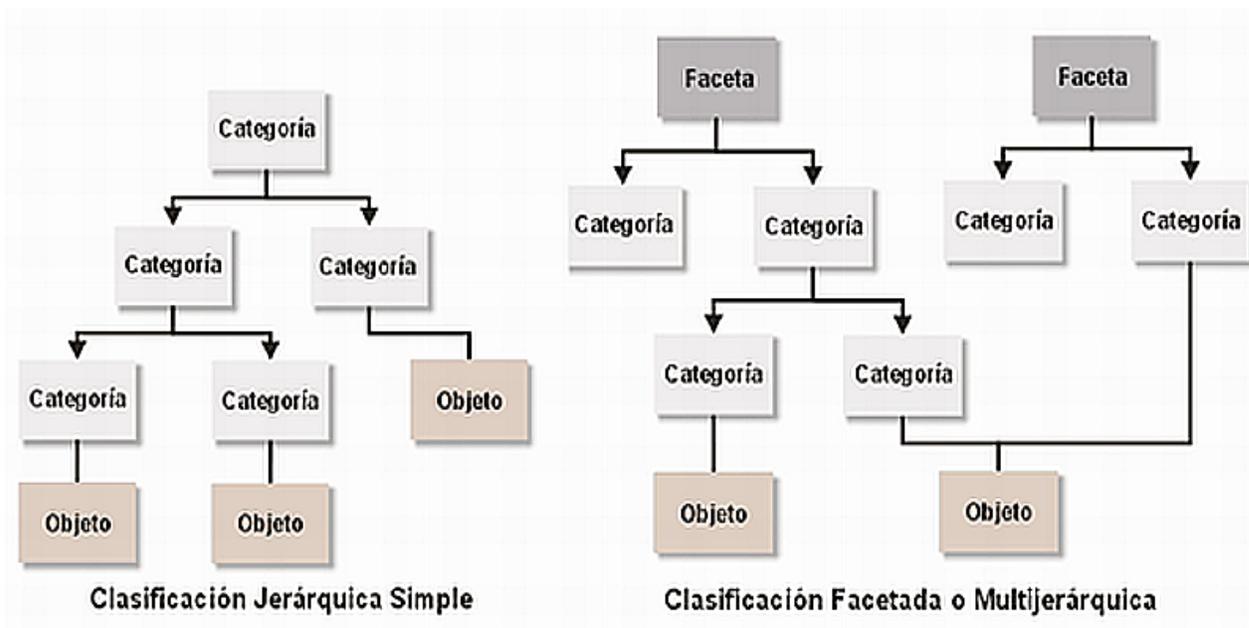


Figura 2: ejemplo modelo jerárquico

IV.2.2 Modelo en red : Éste es un modelo ligeramente distinto del jerárquico; su diferencia fundamental es la modificación del concepto de nodo: se permite que un mismo nodo tenga varios padres (posibilidad no permitida en el modelo jerárquico). Fue una gran mejora con respecto al modelo jerárquico, ya que ofrecía una solución eficiente al problema de redundancia de datos; pero, aun así, la dificultad que es administrar la información en una base de datos de red ha significado que sea un modelo utilizado en su mayoría por programadores más que por usuarios finales.

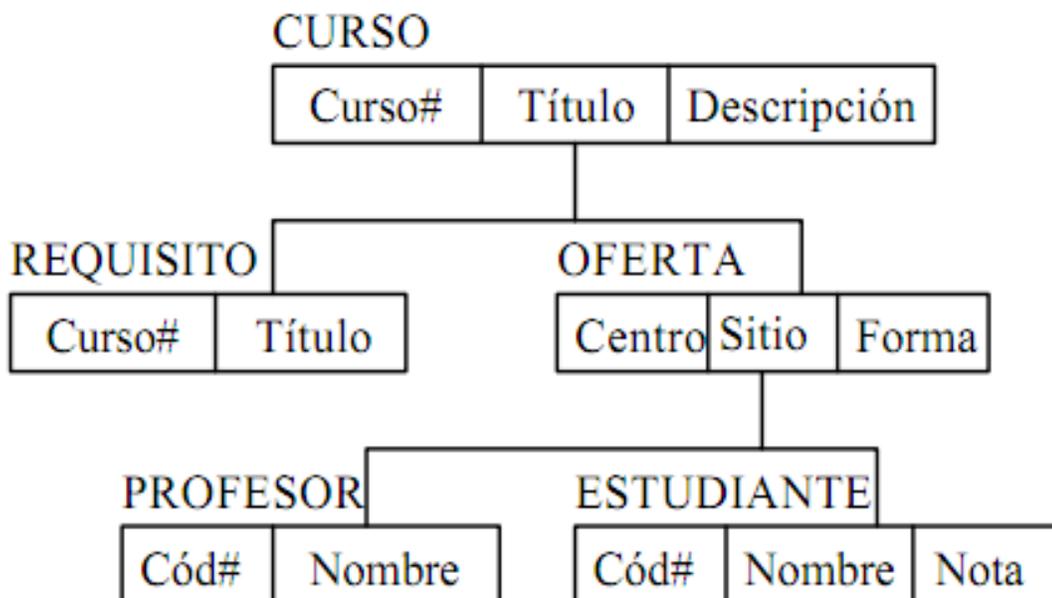


Figura 3: ejemplo modelo en red.

IV.2.3 Modelo relacional: Es el modelo más utilizado en la actualidad para implementar bases de datos ya planificadas. Permiten establecer interconexiones (relaciones) entre los datos, y a través de dichas conexiones relacionar los datos de 2 o mas tablas, de ahí proviene su nombre “Modelo Relacional”.

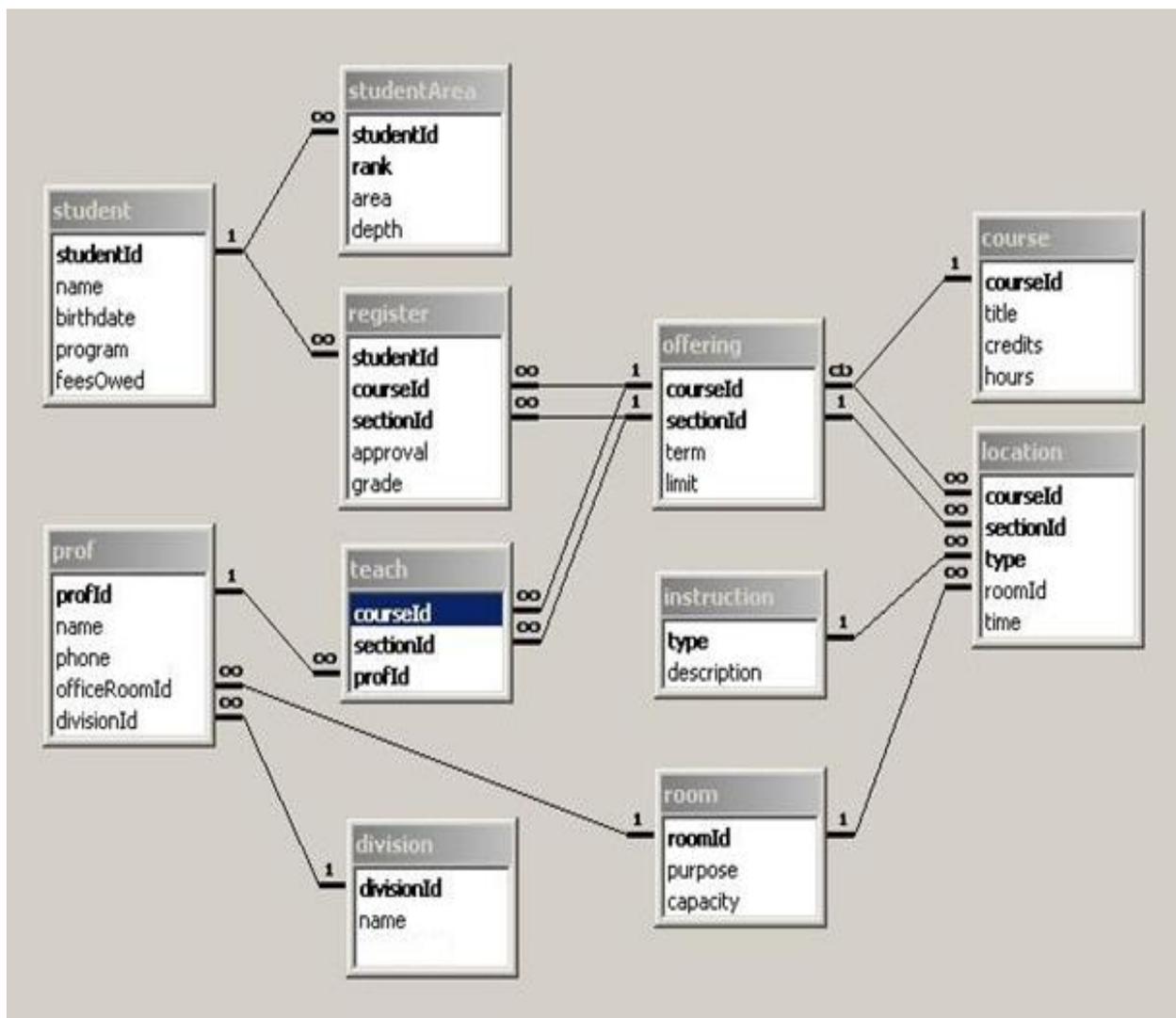


Figura 4: ejemplo modelo relacional.

Características:

- * Una base de datos relacional se compone de varias tablas o relaciones.
- * No pueden existir dos tablas con el mismo nombre ni registro.
- * Cada tabla es a su vez un conjunto de registros (filas y columnas).
- * La relación entre una tabla padre y un hijo se lleva a cabo por medio de las claves primarias y foráneas.
- * Las claves primarias son la clave principal de un registro dentro de una tabla y éstas deben cumplir con la integridad de datos.

IV.2.4 Modelo de bases de datos deductivas: es un sistema de base de datos pero con la diferencia que permite hacer deducciones a través de inferencias. Se basa principalmente en reglas y hechos que son almacenados en la base de datos. También las bases de datos deductivas son llamadas base de datos lógica, a raíz de que se basan en lógica matemática.

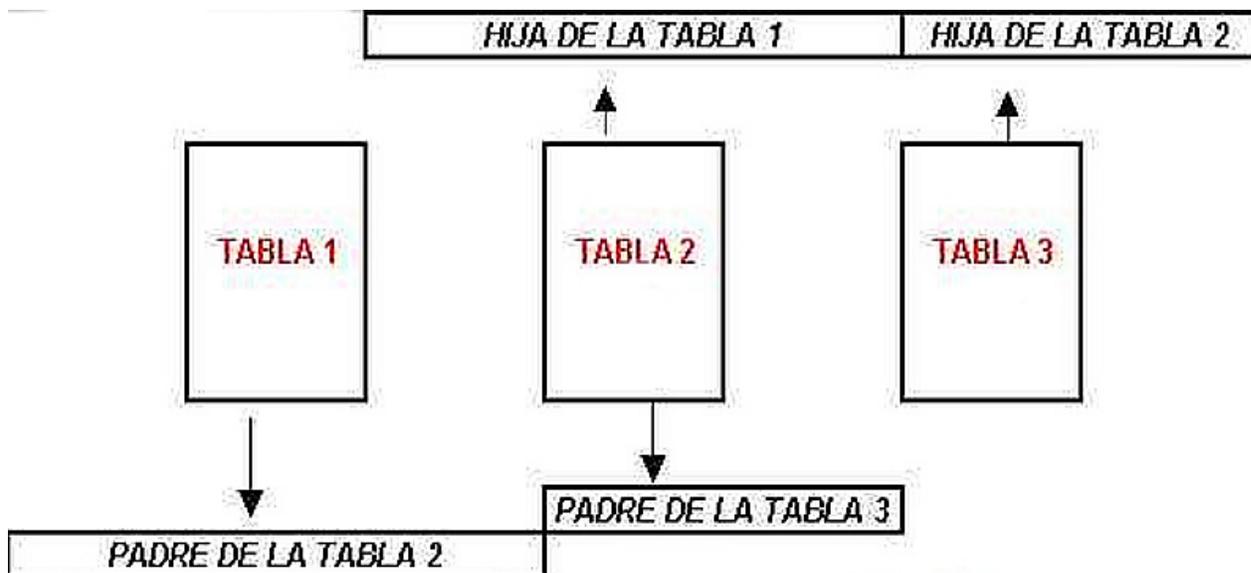


Figura 5: ejemplo modelo deductivo.

V. Otros modelos que se están implementando son:
V.1 Bases de datos orientadas a objeto: Las bases de datos orientadas a objetos (BDOO) son aquellas cuyo modelo de datos está orientado a objetos y almacenan y recuperan objetos en los que se almacena estado y comportamiento. Su origen se debe a que en los modelos clásicos de datos existen problemas para representar cierta información, puesto que aunque permiten representar gran cantidad de datos, las operaciones que se pueden realizar con ellos son bastante simples. Las clases utilizadas en un determinado lenguaje de programación orientado a objetos son las mismas clases que serán utilizadas en una BDOO; de tal manera, que no es necesaria una transformación del modelo de objetos para ser utilizado por un SGBDOO. De forma contraria, el modelo relacional requiere abstraerse lo suficiente como para adaptar los objetos del mundo real a tablas.

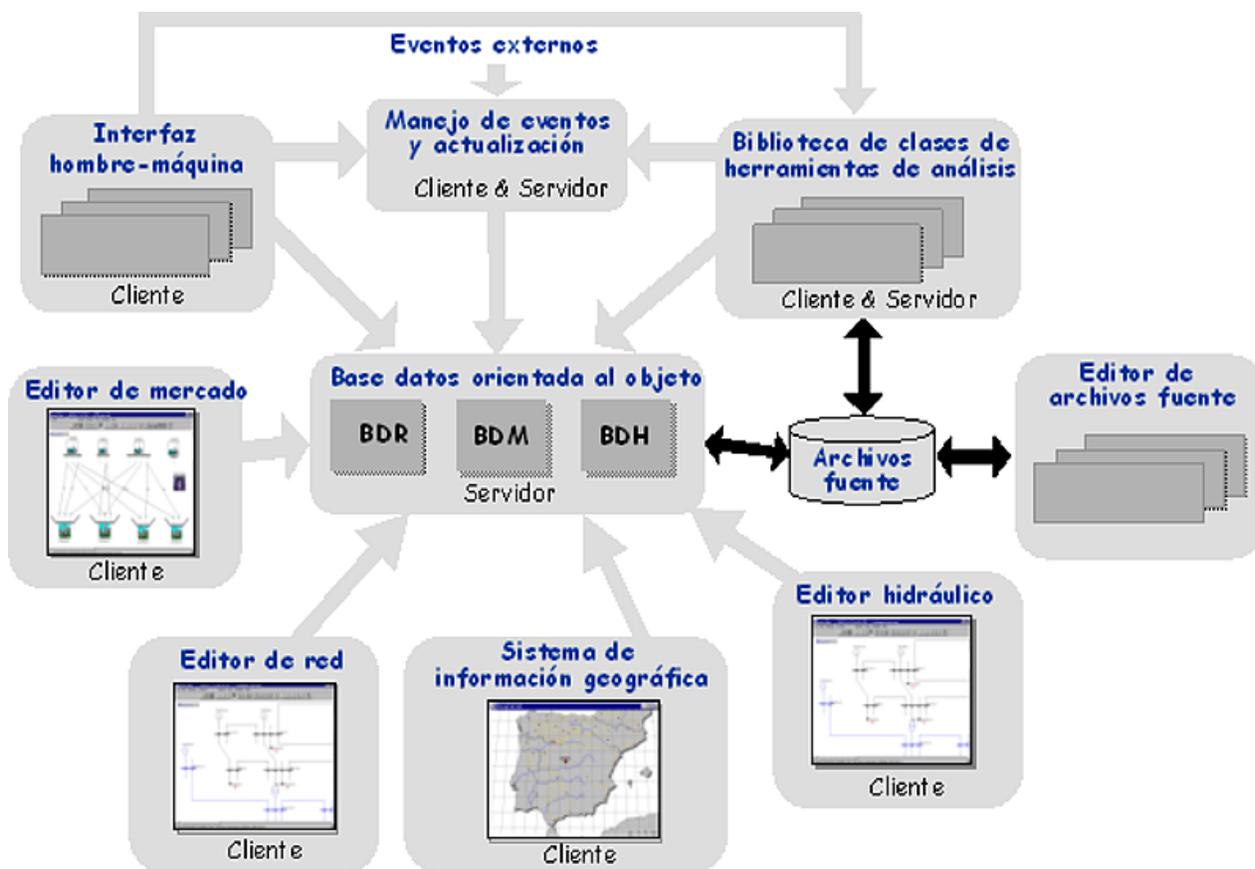


Figura 6: ejemplo modelo orientado a objetos.

Las bases de datos orientadas a objetos aparecen para evitar los problemas que surgen al tratar de representar cierta información, aprovechar las ventajas del paradigma orientado a objetos en el campo de las bases de datos y para evitar transformaciones entre modelos de datos.

V.2 Modelo de base de datos orientado a la columna.

Las Bases de Datos orientadas a la columna se introdujeron por primera vez en 1970 en productos como Model 204 y ADABAS, cabe resaltar que parte del auge actual que está provocado por NoSQL se debe a la adopción de Cassandra (originalmente desarrollada por y para Facebook, luego donada a la fundación Apache) por parte de Twitter y Digg. Apache Cassandra es la base de datos orientada a columnas más conocida y utilizada actualmente.

VI. Definición base de datos orientado a la columna.

Son sistemas de bases de datos que tienen la característica de almacenar los datos en forma de columna. La ventaja principal de este tipo de sistema es que permite el acceso a grandes volúmenes de datos de forma rápida porque se puede acceder como una unidad a los datos de un atributo particular en una tabla. Una base de datos orientada a columnas es un sistema de gestión de bases de datos que almacena su contenido por columnas (atributos) y no por filas (registros) como lo hacen las bases de datos relacionales.

Cada columna es almacenada contiguamente en un lugar separado en disco, usando generalmente unidades de lectura grandes para facilitar el trabajo al buscar varias columnas en disco. Para mejorar la eficiencia de lectura, los valores se empaquetan de forma densa usando esquemas de compresión ligera cuando es posible. Los operadores de lectura de columnas se diferencian de los comunes (de filas) en que son responsables de traducir las posiciones de los valores en locaciones de disco. Con este cambio ganamos mucha velocidad en lecturas, ya que si se requiere consultar un número reducido de columnas, es muy rápido hacerlo pero no es eficiente para realizar escrituras. Por ello este tipo de soluciones es usado en aplicaciones con un índice bajo de escrituras pero muchas lecturas. Típicamente en data warehouses y sistemas de inteligencia de negocios, donde además resultan ideales para calcular datos agregados.

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

Title	Title	Title	Title	Title
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data
Data	Data	Data	Data	Data

Figura 7: Ejemplo de base de datos orientada a la columna.

VII Modelo de Almacenamiento Descompuesto

(Decomposition Storage Model DSM)

Este fue un modelo propuesto por Copeland y Khoshafian en 1985, el cual fue desechado en un principio por los administradores de bases de datos de aquella época. La mayoría de los sistemas de bases de datos usan un Modelo de Almacenamiento N-ario (NSM) para un conjunto de registros. Este enfoque almacena los datos como se pueden ver en el modelo conceptual. También, varios archivos invertidos o índices agrupados podrían ser agregados para mejorar velocidad de accesos. El concepto central en el NSM es que todos los atributos de un registro en el esquema conceptual son almacenados de manera contigua. Por ejemplo, la relación conceptual siguiente contiene una clave para identificar cada registro y tres atributos por registro.

R	SUR	TYPE	a1	a2	a3	a4
	s1	t1	v11	v21	v31	NA
	s2	t2	v12	v22	NA	v42
	s3	t2	v13	v23	NA	v43
	s4	t1	v14	v24	v34	NA

a1	sur	val
	s1	v11
	s2	v12
	s3	v13
	s4	v14

a2	sur	val
	s1	v21
	s2	v22
	s3	v23
	s4	v24

a3	sur	val
	s1	v31
	s4	v34

a4	sur	val
	s2	v42
	s3	v43

Figura 8: El NSM guardaría s1, v1i, v2i y v3i juntos para cada registro i.

Algunos sistemas de bases de datos usan un modelo de almacenamiento completamente transpuesto. Este enfoque almacena todos los valores del mismo atributo de una relación juntos. Varios estudios han comparado el desempeño de modelos de almacenamiento DSM con el NSM. Las ventajas de un modelo de almacenamiento altamente descompuesto (DSM) el cual es un modelo de almacenamiento con claves sustitutas incluidas.

DSM empareja cada valor de atributo con la clave sustituta de su registro en una relación binaria. Adicionalmente, el DSM guarda dos copias de cada atributo de relación. Una copia esta agrupada con el valor mientras la otra esta agrupada con la clave sustituta.

VIII DSM vs NSM

VIII.1 Elimina valores nulos

En el siguiente ejemplo podemos observar como el NSM y el DSM se comportan respecto a los valores nulos.

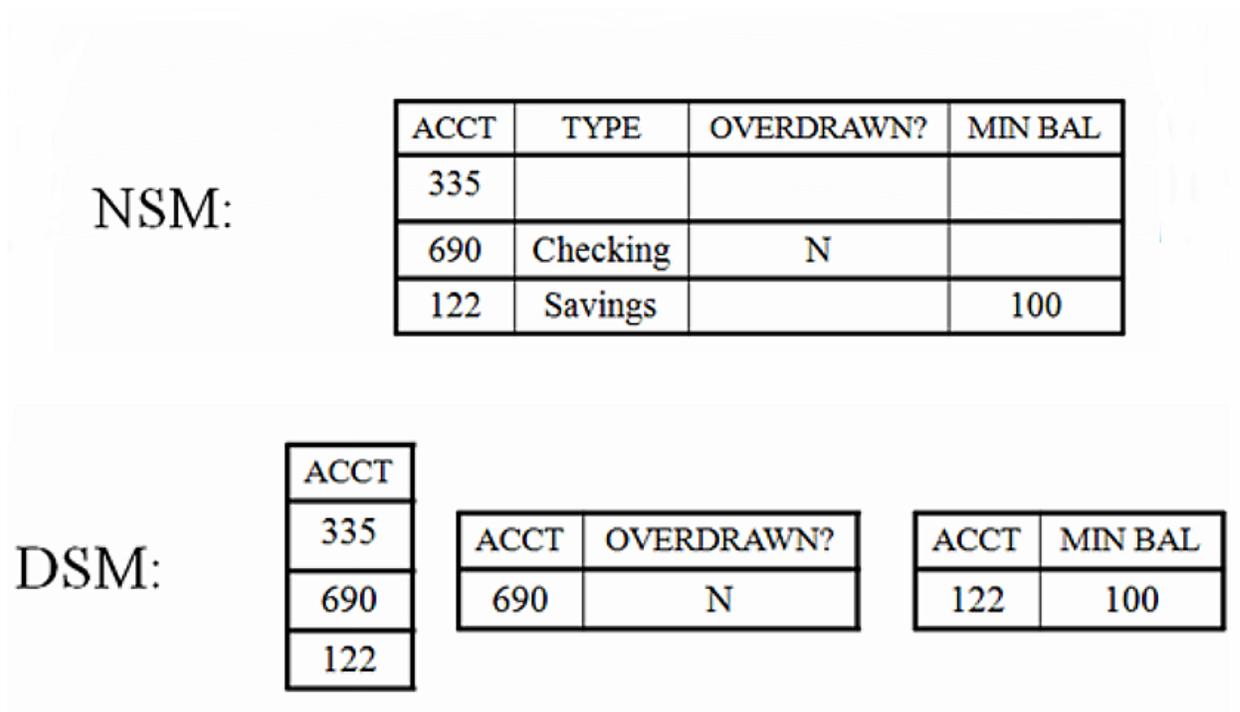


Figura 9: ejemplo de DSM eliminando nulos.

VIII.2 Método de Acceso Uniforme

Registros en NSM están organizados de variadas formas:

- **Secuencial:** es la lectura o escritura de datos en forma secuencial, o sea, uno tras otro.
- **Pila:** es el modo de acceso a elementos de tipo LIFO (del inglés *Last In First Out*, último en *entrar*, primero en *salir*) que permite almacenar y recuperar datos.
- **Indexada:** este método el usuario obtiene acceso secuencial a los registros clasificados por llaves y a los mismos registros.

En cambio DSM casi siempre usa el mismo método: una instancia agrupada por clave, y atributo valor. Actualmente, los motores columnares aceptan índices, pero el esquema básico se mantiene.

VIII.3 Soporte de atributos multivaluados

En NSM esto se complica, ya que implica re-estructuraciones de campo para tipos de datos especiales (tipo arreglo) que soporten los diversos valores posibles. En DSM este problema no existe, ya que internamente, el motor se encarga de “normalizar” este atributo, de manera que sea transparente para el usuario de la base de datos.

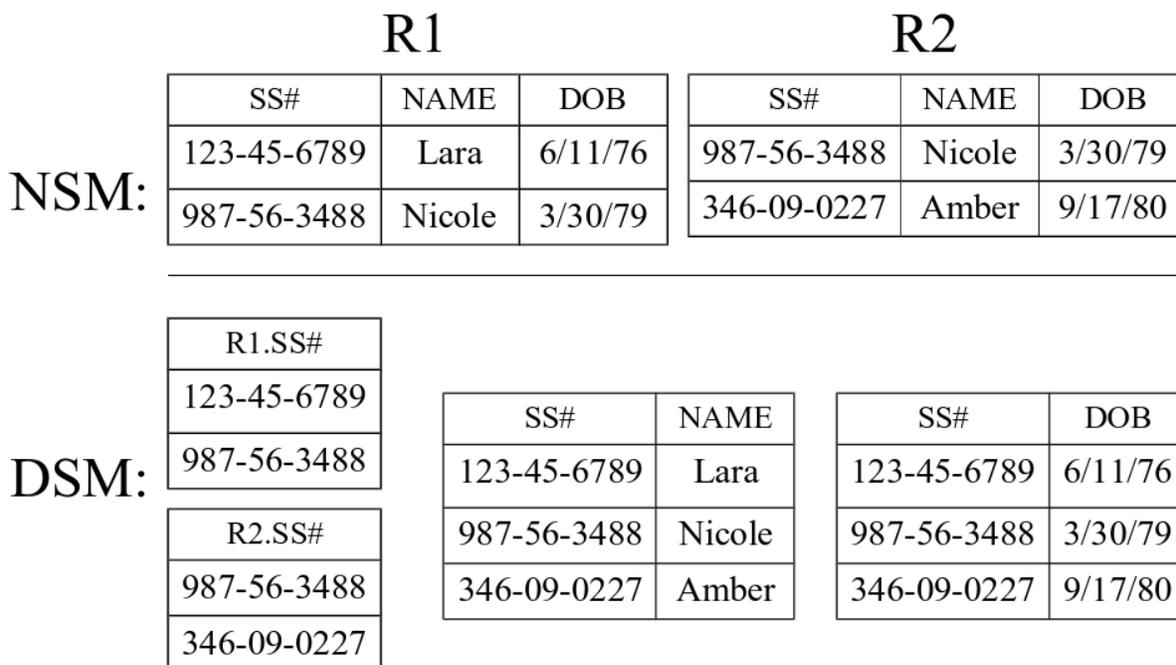


Figura 10: ejemplo DSM multivaluados.

VIII.4 Soporte de Entidades

Este concepto es novedoso en DSM. Esta característica implica que si tenemos un registro, digamos empleado, pero no conocemos todavía el resto de su información, entonces, no tendríamos ninguna entrada en las correspondientes relaciones binarias de DSM.

VIII.5 Soporte de múltiples relaciones padres

Supongamos que tenemos un registro que puede estar presente en dos relaciones R y S, esto trae como consecuencia en NSM un incremento de la complejidad al tener que lidiar con duplicación de la información (con el consiguiente riesgo de actualizar o borrar

en varias ubicaciones). Mientras que en DSM esto no sucede ya que con el uso de entidades, esto se evita.

VIII.6 Soporte de registros heterogéneos

Registros de una misma relación pueden tener atributos diferenciables. En NSM necesitaríamos un indicador o campo tipo para diferencias registros de uno u otro tipo. El enfoque de DSM no requiere almacenamiento en cada registro de atributos tipo o algún indicador de atributos aplicables.

SS#	NAME	PHONE
123-45-6789	Lara	1112222
987-56-3488	Nicole	3334444

Base table

Change Lara's
phone to 5556666

Update

NSM differential file:

SS#	NAME	PHONE
123-45-6789	Lara	5556666

DSM differential file:

SS#	PHONE
123-45-6789	5556666

Fig
ura
11:
eje
mpl
o
DS
M
sop
orte

de registro heterogéneos.

IX Variables de funcionamiento a considerar IX.1 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 se trata de decenas o cientos de gigabytes de datos. La pregunta 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. ***IX.2 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.

IX.3 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. Teniendo en cuenta que la diferencia de los requisitos de hardware pueden ser sustanciales.

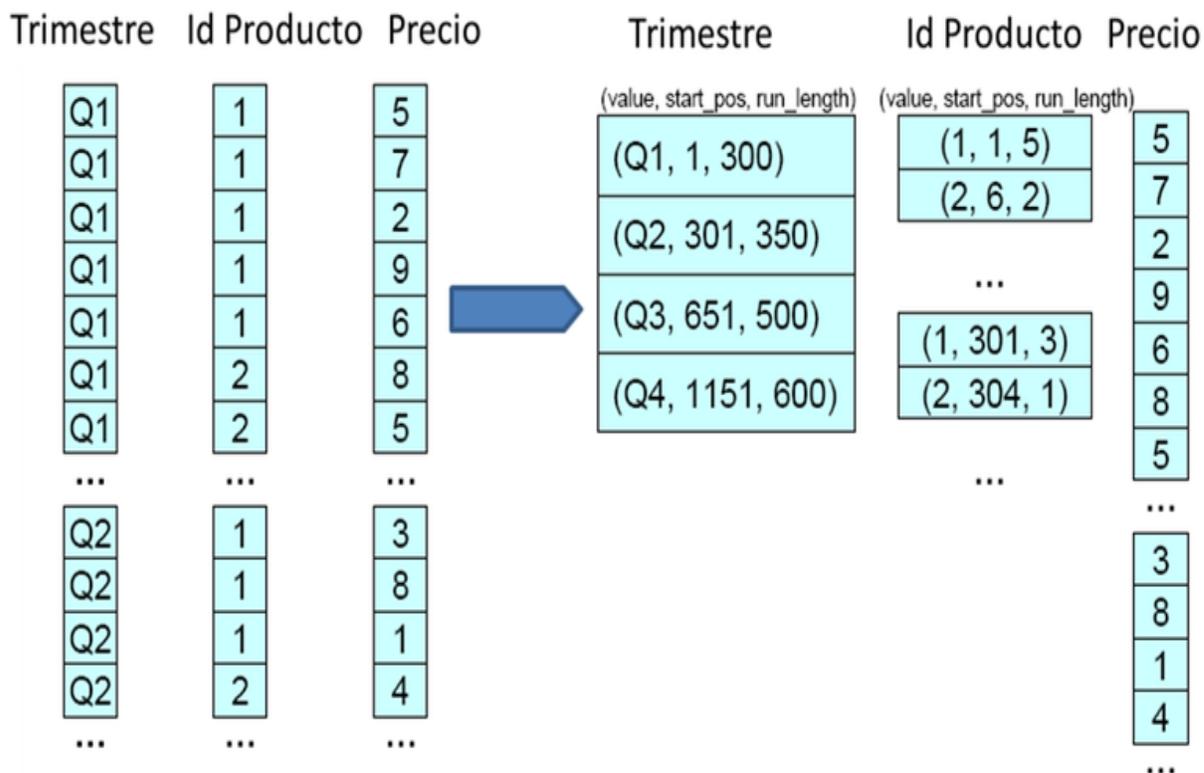


Figura 12: ejemplo de compresión de datos.

IX.4 Técnicas de acceso

Algunas bases de datos 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 se necesita acceder al sistema con herramientas basadas en SQL, se determinara 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 asegurarse de encontrar si el rendimiento de las consultas

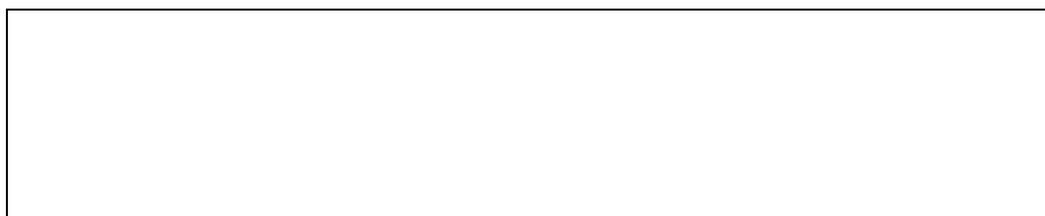
SQL es comparable a los resultados con el sistema de la propia herramienta de consulta. A veces, el ejecutar consultas SQL es mucho más lento. **IX.5 Rendimiento**

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. **IX.6 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 que 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 el equipo debe tener memoria suficiente para hacer esto.

X Principales Bases de Datos Columnares

X.1 Base de Datos Cassandra



Apache Cassandra es la base de datos NoSQL que, en sus inicios, fue desarrollada por Facebook, es una base de datos no relacional distribuida y basada en un modelo de almacenamiento de clave-valor, escrita en Java. Permite grandes volúmenes de datos en forma distribuida. Hoy la utilizan otros grandes usuarios como Twitter y Digg, aunque también se quiere ir hacia los ambientes empresarios. Ahora que Oracle adquirió a la

base de datos open source líder, MySQL, los esfuerzos para impulsar a Cassandra se han redoblado.

De hecho, acaba de aparecer una primera empresa ofreciendo soporte comercial para Cassandra. Se trata de Datastax, que además está creando herramientas para usar sobre el core de Cassandra. Datastax es apoyada con capital de Rackspace, la empresa a la que pertenecieron sus fundadores. Cassandra es una base de datos sumamente escalable y la escalabilidad es el mantra que se escucha ya no sólo en empresas Web 2.0. La empresa está comenzando a ser terreno propicio para sus propiedades.

El plan de Datastax consiste en usar a Cassandra como plataforma central e ir agregándole porciones propietarias que hagan a su versión comercial. Uno de sus primeros servicios podría ser el de migración para que usuarios de otras bases de datos puedan pasar su información a Cassandra. Datastax agregará software propietario de administración, pero sin dejar de contribuir al proyecto open source. Cassandra se encuentra actualmente en su versión 1.2.8 (lanzada el 28-07-2013), El proyecto Apache Cassandra lanza un nuevo release cada tres a cuatro meses.

Entre los elementos clave de Cassandra 1.2.8 está el soporte a un índice secundario. Por ejemplo, en la búsqueda por filas de las bases de datos, se arranca de una clave de fila conocida, pero no se puede, por ejemplo, buscar a todos los usuarios que nacieron en un determinado año, sin tener que manipular un índice a nivel de aplicación para tener esa información. Un índice secundario como el que soportará Cassandra, facilitará esa clase de búsquedas.

Cassandra fue creada para manejar grandes volúmenes de datos distribuidos en numerosos servidores estándar, ofreciendo alta disponibilidad sin ningún punto único de falla. Tiene un almacén de valores de claves manejado con consistencia eventual (modelo de consistencia usado en programación paralela). Las claves se dirigen hacia múltiples valores que se agrupan en familias de columnas. Esas familias se definen cuando se crea una base Cassandra, pero luego se les puede agregar columnas a las diferentes familias. Las columnas se pueden agregar a claves específicas y así

diferentes claves tendrán diferentes cantidades de columnas dentro de una misma familia. Los valores de una familia de columnas para cada clave se almacenan juntos, haciendo de Cassandra un híbrido entre una DBMS (sistema de gestión de base de datos) orientada a columnas y un almacén de datos orientado a filas. Cassandra fue desarrollada por Facebook para impulsar su dispositivo Inbox Search. Trabajaron Avinash Lakshman (uno de los autores de Dynamo de Amazon) y Prashant Malik, ingeniero de Facebook. Fue liberada como proyecto open source en julio 2008 en código Google y en marzo de 2009 se convirtió en un proyecto de Apache Incubator. En febrero de este año, pasó a ser un proyecto de primer nivel.

Soporta arquitecturas de 32/64 Bit

Sistemas Operativos:

Macintosh OS X

Windows 7

Windows Server 2008

Centos 5.x, 6.x

Debian 5.x, 6.x

Red Hat Linux Enterprise 5.x, 6.x

Ubuntu 10.x, 11.x, 12.x

X.2 PROYECTO GEMINI

(Incluido Versión Beta en SQL Server 2008 R2)

Ese es el nombre que recibe la propuesta que nos hacen desde Microsoft, para renovar su base de datos OLAP. Se trata de un almacenamiento en memoria y orientado a columnas de Analysis Services para ser explotado desde Excel, entre otras opciones. Se trata de empezar a jugar fuerte en el campo de la inteligencia de negocios, aprovechando la herramienta basada en la popular hoja de cálculo que se incluye dentro del portafolio ofrecido actualmente por Microsoft.

De esta forma se consigue que los usuarios no técnicos puedan acceder al mundo de la inteligencia de negocios.

X.3 LUCIDDB



Ha sido diseñado especialmente como un propósito de RDBMS de código abierto construido enteramente para almacenamiento de datos e inteligencia de negocios. Se basa en pilares arquitectónicos como almacén de columna, indexación de mapa de bits, unión/agregación de hash y versiones múltiples de nivel de página. La mayoría de los sistemas de bases de datos comienzan su vida con un enfoque en las capacidades de procesamiento de transacciones y, a continuación, obtienen capacidades analíticas agregadas como una ocurrencia tardía. Por el contrario, todos los componentes fueron diseñados con los requisitos de integración de datos flexibles y de alto rendimiento y procesamiento de consulta sofisticada en la mente.

Además, la exhaustividad centrada dentro de su arquitectura significa simplicidad para el usuario: sin DBA requerido.

En lugar de lanzar hardware en los problemas de almacenar datos por depender de clústeres costosos o "aparatos" especializados, la escalabilidad ofrecida por la arquitectura única de LucidDB permite obtener gran rendimiento utilizando sólo un único estándar de servidor Linux o Windows.

Además de mantener los costos bajos, esto también minimiza los problemas de mantenimiento y administración.

Esta basada en EigenBase <http://www.eigenbase.org/> un software base que permite crear sistemas administradores de datos.

LucidDB esta pensada con el propósito de hacer data warehousing y business intelligence.

Esta pensada para ser básicamente solo read only, las actualizaciones crean nuevas páginas que reemplazan a las existentes y se guardan versiones de estas.

Las páginas miden 32K, se maneja un buffer de 5,000 páginas con la información mas leída.

Se usa una técnica de indexación conocida como "bitmap", índices y data son comprimidos y se utiliza la técnica del "semi join" para determinar la data que es únicamente necesaria acceder por las consultas.

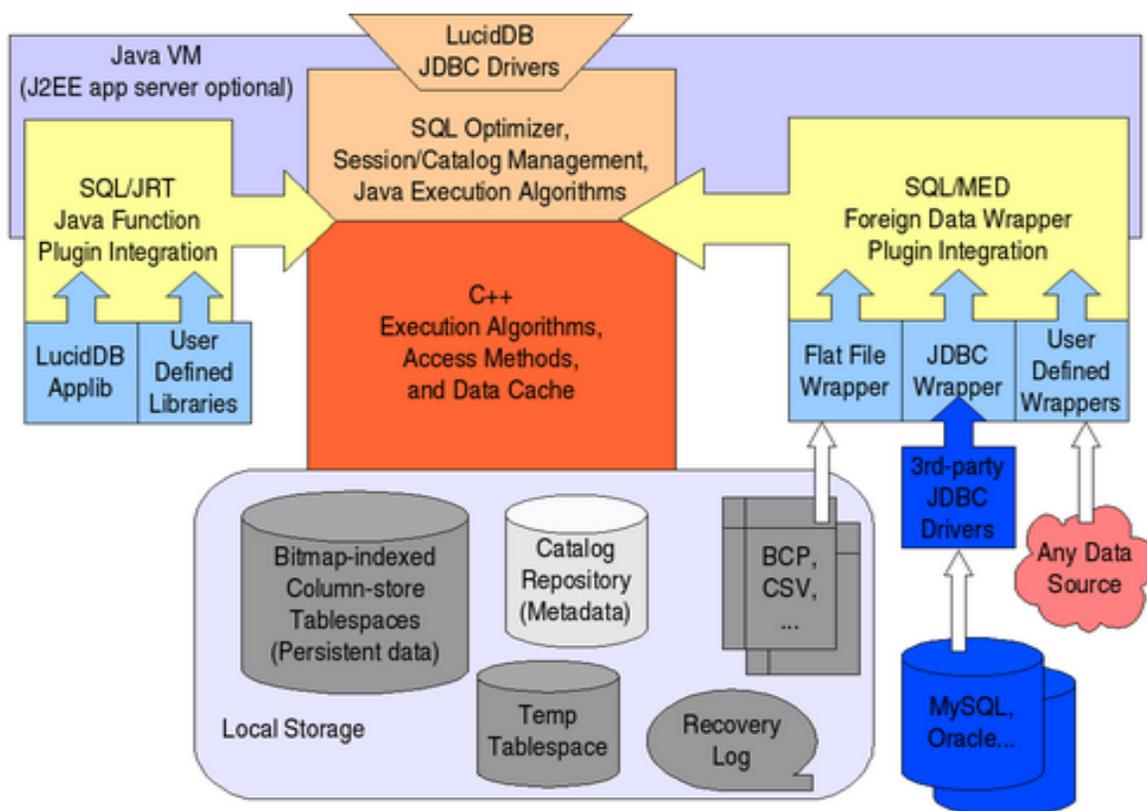


Figura 13: Funcionamiento de LucidDB.***IX.4 NFOBRIGHT***

Infobright combina una base de datos orientada a la columna con la red de conocimiento para ofrecer una arquitectura de auto-gestión de Data Warehouse optimizado para el análisis. Este software sofisticado elimina el tiempo y el esfuerzo que suelen participar en la ejecución y la gestión de un Data Warehouse, liberando su tiempo y su presupuesto.

Infobright Analytic Data Warehouse está basado en los siguientes conceptos:

- Orientación a Columnas.
- Paquetes de Datos.
- Conocimiento de Red.
- La optimización.

Infobright es, en su núcleo, un comprimido de bases de datos orientadas a la columna. Esto significa que en lugar de los datos que se almacenan fila por fila, se almacena columna por columna. Hay muchas ventajas en la columna-orientación, incluyendo la capacidad de hacer más eficiente la compresión de datos, ya que cada columna guarda un solo tipo de datos (por oposición a las filas que normalmente contienen varios tipos de datos), y que permite la compresión de ser optimizados para cada tipo de datos. Infobright, organiza cada columna en paquetes de datos, tiene más compresión que otras bases de datos orientadas a la columna, ya que se aplica un algoritmo de compresión basado en el contenido de cada paquete de datos, no sólo cada columna.

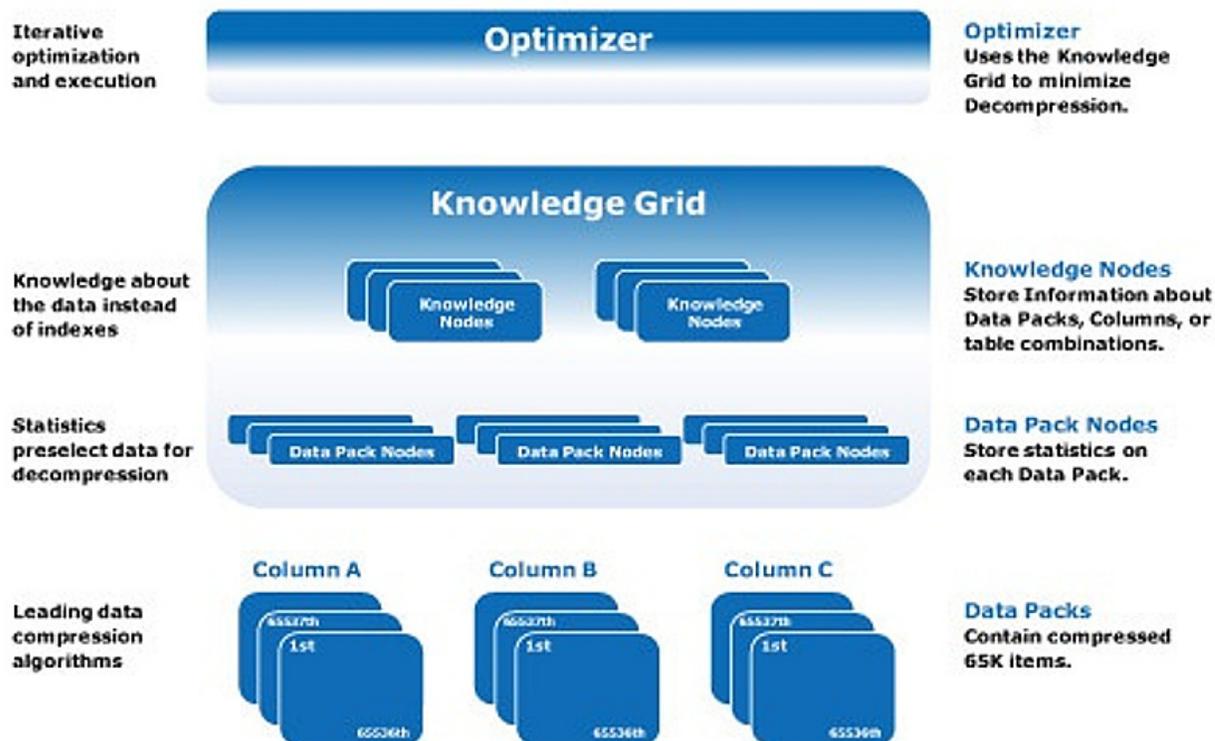


Figura 14: Funcionamiento Infobright.

La mayoría de las preguntas sólo implican un subconjunto de las columnas de las tablas y por lo que una base de datos orientada a la columna se centra solo en recuperar los datos que se requieren.

Los datos se almacenan en 65 mil Paquetes de Datos. Paquete de nodos de datos contienen un conjunto de estadísticas sobre los datos que se almacenan y comprimen en cada uno de los Paquetes de Datos. Los Nodos de conocimiento proporcionan una nueva serie de metadatos relacionados con paquetes de datos o relaciones columnares.

El optimizador es el más alto nivel de inteligencia en la arquitectura. Utiliza la red de conocimientos para determinar el conjunto mínimo de paquetes de datos que necesitan ser descomprimidos con el fin de satisfacer una consulta en el menor tiempo posible.

X.5 VERTICA

Vertica es el único DBMS habilitado para gestionar terabytes de datos más rápido y más fiable que cualquier otro producto de almacenamiento de datos. Obtiene rápidamente inteligencia de negocios con las siguientes características:

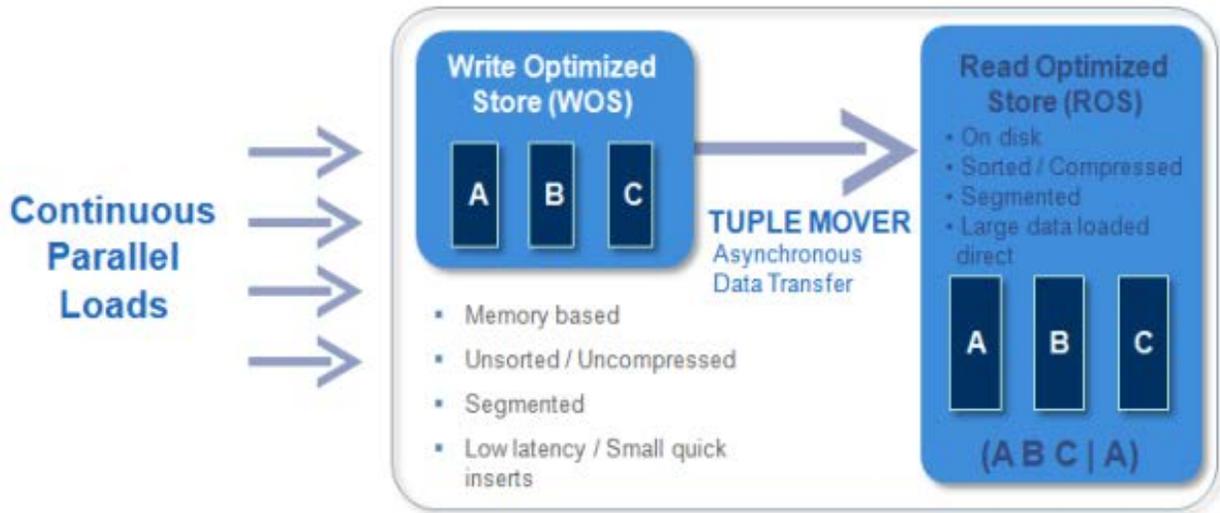
- Orientación a columnas. 50x – 200x mas rápido, eliminando los costos de input - output
- Escala a arquitectura MPP (Procesadores Masivamente Paralelos). Escala ilimitadamente solo por la adición de nuevos servidores a la red.
- Agresiva compresión de datos. Reduce los costos de almacenamiento hasta en un 90%.
- Alta disponibilidad inmediata. Corre sin parar con replicación automática, resistente a fallos y recuperación.
- Flexibilidad de despliegue. Despliegue en Linux, VMwire o en amazon cloud para manejar variedad de proyectos.

Figura 15: cargas continuas paralelas de Vertica.

Figura 16: análisis en tiempo real.

Vertica cambia completamente la economía de la inteligencia de negocios, que permite rápidamente iniciar un espectro mucho más amplio de análisis del negocio:

- Ver mayores volúmenes de datos históricos.
- Analizar los datos en cualquier nivel de detalle.
- Realizar análisis en tiempo real.
- Conducta ad-hoc y de corta duración de análisis de proyectos de negocios.
- Construir Análisis de Negocio con Software as a Service (SaaS).



X.

6 QD TECHNOLOGY

QD Technology Base de Datos de respuesta rápida (QD™) es una solución de base de datos relacional que permite a los ejecutivos de negocios y analistas de datos fácil y cómodamente obtener respuestas rápidas a consultas de base de datos de copias locales de su base de datos. Las consultas se ejecutan más rápido que con otras soluciones y ya que se ejecutan en los computadores del usuario, que no interfieran con otros las actividades de los usuarios, y se puede ejecutar en cualquier lugar.

Beneficios del Usuario:

- Acceso a demanda para datos localizados. Los usuarios pueden obtener respuestas a sus preguntas sobre sus PC sin impactar negativamente las operaciones de IT, ya que la base de datos entera está comprimida, encriptada y es copiada a su PC local.
- Rápida respuesta a consultas a través de una compresión inteligente. Dependiendo de la naturaleza de los datos, puede ser hasta cinco veces más rápido que las tradicionales bases de datos relacionales.
- Actualizaciones automáticas entregadas al usuario. Cuando la base de datos master cambia, las actualizaciones pueden ser enviadas a los usuarios, sin la necesidad de la intervención del usuario final.

Beneficios para la empresa.

- Facilita la innovación. Puede reducir dramáticamente el tiempo de respuesta de las consultas.
- Reducción de la infraestructura informática. Pone los derechos de la base de datos directamente en la PC de los usuarios, no requiere de varios niveles de infraestructura para las consultas, los datos del cubo de gestión o de depósito.
- Mejora el rendimiento del usuario. Los analistas de datos y los usuarios pueden ejecutar sus consultas cuando quieran, sin las restricciones que les imponen las bases de datos de administradores u otros usuarios de negocios.
- **Solución completa.**
 - Alta velocidad de compresión.

- Tecnología para almacenar una copia local de la base de datos en PC del usuario.
- Capacidad de consulta a la base de datos sin estar en red.
- Herramientas para reducir la demanda en el sistema y base de datos de administradores.

Compresión.

- Mejora del rendimiento de consultas a través de una compresión inteligente.
- Rápido acceso a través de una compresión optimizada.
- Aplica una técnica de compresión especialmente seleccionada de una biblioteca para cada columna en cada tabla.
- Comprime los datos y tablas basándose en patrones.

Compatibilidad.

- Compatible con ODBC.
- Consultas con el estándar SQL-92.
- Datos almacenados en filas y columnas.
- Solamente datos basados en ASCII.

Tecnología.

Tablas antiguas nunca mueren. La compresión de QD es compatible con todas las versiones anteriores de QD.

Plataforma y Despliegue.

- Plataforma. La configuración mínima para QD server es:
 - Windows server 2003 o 2000, XP o Vista.
 - 2 GB RAM
 - Espacio en disco suficiente para mantener la base de datos.

La mínima configuración para el sistema de consulta QD (desktop o laptop)

- Windows server 2003, 2000, XP o Vista.
- 1 GB RAM.
- Espacio suficiente para mantener la base de datos comprimida.

Tiempo de implementación.

- Instalación y configuración inicial de QD, en general se puede completar en 30 min.

Habilidades requeridas.

- QD es compatible con ODBC.

Seguridad.

- Datos en la PC
- Encriptación.

X.7 SYBASE

Sybase es una base de datos relacional basada en columnas que es intrínsecamente más apropiado para el adecuado procesamiento de consultas que un enfoque basado en filas. Debido a que está basado en columnas, Sybase IQ aprovecha las características de cada columna en la tabla, en un número de diferentes caminos. Sybase soporta los esquemas relacionales tradicionales, incluyendo la normalización de esquemas usados para procesos de transacción.

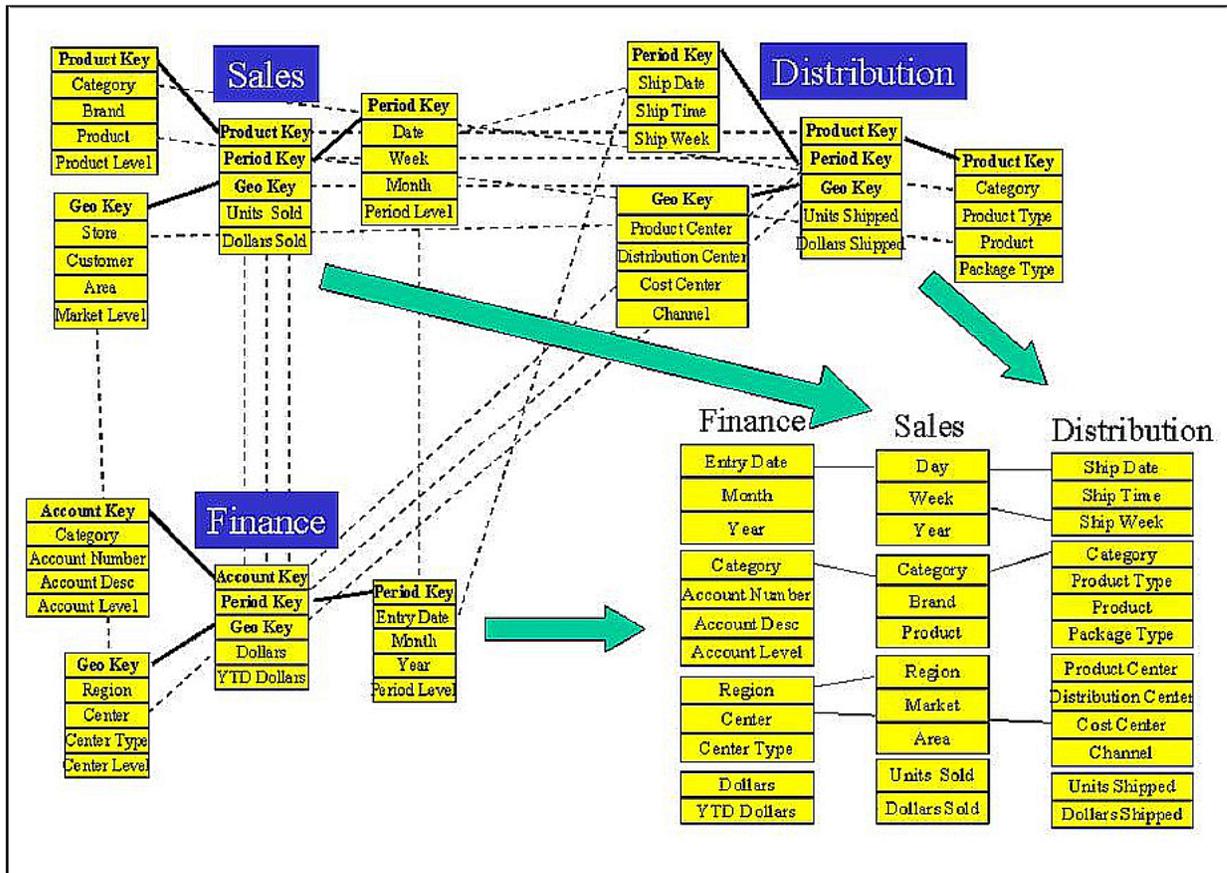


Figura 17: Ejemplo de almacenamiento de Sybase.

Como se puede ver Sybase incluye una API SQL que permite el acceso a SQL, también incluye ODBC, JDBC y XML, provee java para que puede ser usado para escribir procedimientos almacenados y funciones de usuario.

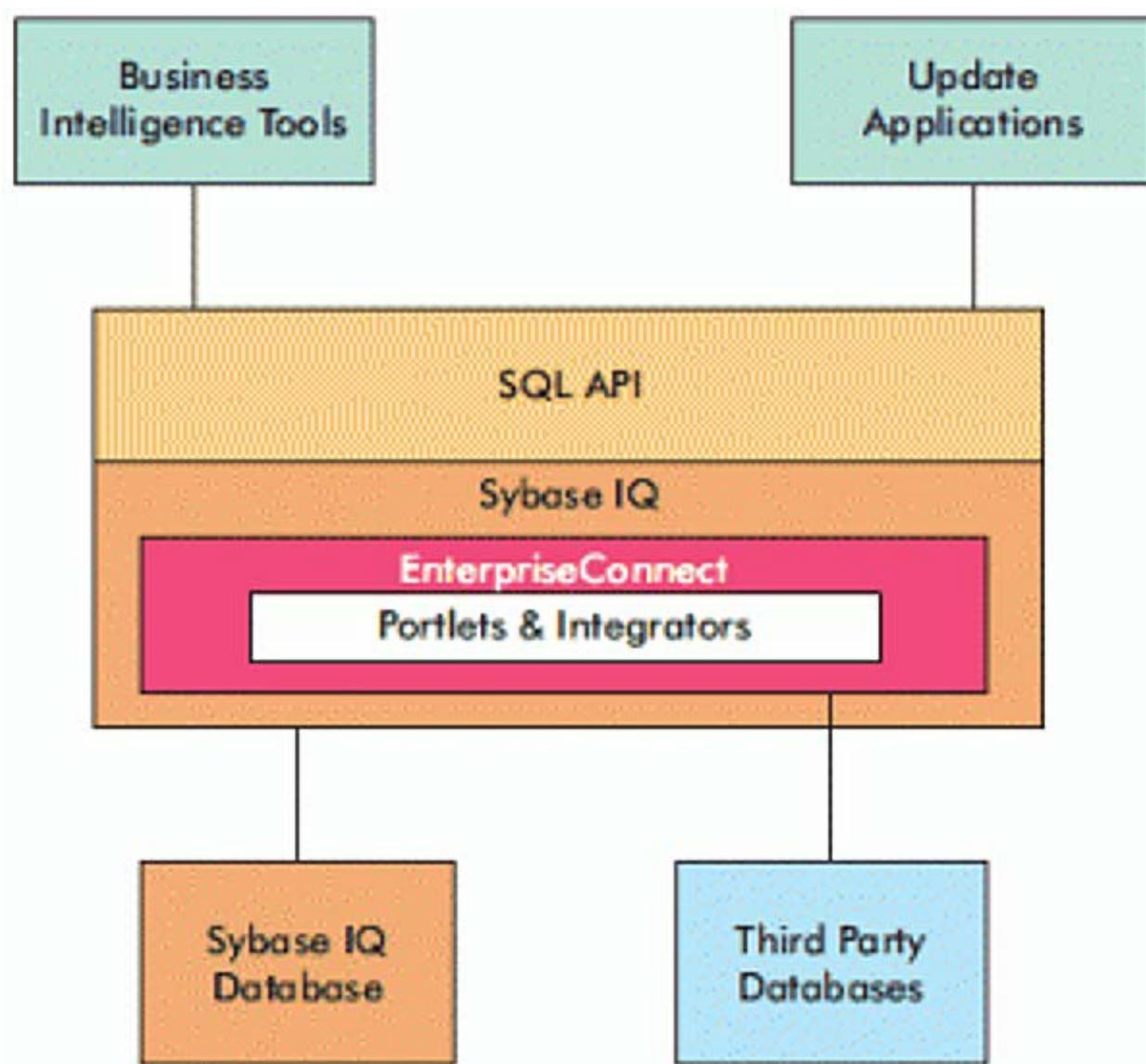


Figura 18: muestra como Sybase incluye una API SQL,

- Ofrece una serie de índices especializados para el adecuado rendimiento de las consultas.
- Una consecuencia de utilizar el almacenamiento columnar en conjunción con la indexación de Sybase IQ Bit Wise es que las agrupaciones pueden hacerse en la

producción. Dado que una parte significativa de extraer, transformar y cargar es la anterior agrupación de transacciones.

- Compresión de datos es mucho más fácil de implementar en un enfoque basado en columnas que cuando se utilizan los métodos convencionales. Es significativamente más eficiente. En la práctica Sybase IQ ha demostrado una compresión de datos de un 50% a un 70% del conjunto de datos original.
- Es fácil agregar y cargar una columna de datos a una tabla como sería agregar una fila a una base de datos relacional convencional.
- Un enfoque basado en columnas es mucho más fácil de mantener y requiere menos sintonización que un datawarehouse convencional.
- Multihilo y alta disponibilidad 24 x 7.
- A parte de las características ya mencionadas, también apoya RCube, estructura plana que puede proveer importantes beneficios en comparación con los esquemas convencionales. En particular Rcube puede acelerar significativamente la implementación, así como el rendimiento en tiempo de ejecución y proporcionar una mayor flexibilidad.
- Sybase ha sido creado para soportar el mayor número de consultas posible corriendo en paralelo en lugar de concentrarse en el uso del paralelismo para optimizar el rendimiento de una consulta en particular.

X.8 ParAccel

PADB tiene el propósito de construir Data Warehousing y manejadores analíticos de sistemas de bases de datos. PADB base de datos columnar es el más eficiente DBMS para DWH and BI y algunas otras consultas analíticas que se necesitan. Diseñado para velocidad y escalabilidad con simplicidad, se ajusta fácilmente dentro de un ambiente de DWH.

ParAccel es una base de datos robusta, relacional y totalmente transaccional DBMS para Data Warehousing, analítica y con aplicaciones BI. Su impresionante velocidad se construye en la sintonización manual de estructuras como índices, vistas materializadas y tablas de resumen no son necesarias (junto con las necesidades de mantenimiento). Este versátil rendimiento no depende de un diseño de esquema estrella, su flexible esquema neutral permite también desencadenar el poder de ParAccel para normalizar y desnormalizar los datos. Las características de rendimiento incluyen:

- Almacenamiento de Datos Columnar.
- Cargas de Alto rendimiento y actualizaciones.
- Procesamiento Masivo paralelo.
- Compresión adaptada.
- Optimizador de consultas basado en el costo.
- Consulta de compilación analítica.
- Protocolo de interconexión con el cliente.
- Todo en memoria con capacidad de diseño.

Escalabilidad:

La escalabilidad le proporciona protección a su inversión de Data warehouse. PADB arquitectura MPP ofrece escalabilidad incremental lineal, servidor por servidor, por lo que no ocupara gran espacio en el centro de datos antes de que realmente se necesite.

- Provee escalabilidad lineal lo que aumenta la capacidad y concurrencia.
- Los nodos en momento de expansión se adaptan fácilmente a los ciclos predefinidos.
- El hardware estándar asegura la eficiencia de precios a medida que crecen los datos al automáticamente ser balanceados y agregados como nodos.

Simplicidad:

ParAccel carga y reparte diseños de Data Warehouse de fácil configuración, así que el administrador puede rápidamente solucionar analíticamente problemas de rendimiento.

Características de manejabilidad que simplifican las operaciones de DWH en curso:

- Diseño para una rápida instalación
- Interfaz estándar de apoyo (ODBC, JDBC, ANSI SQL 92)
- Construidos para alta disponibilidad con ambientes SAN y no SAN.
- SAN aprovecha la gestión de datos empresariales.
- Se ejecuta en hardware estándar para apoyar los requerimientos operacionales.
- Incluye la consola de administración de sistema, incorporado con un sistema manejador de reportes.
- La Sintaxis “Amigo” tiene la capacidad de apoyar en la migración de Oracle y SQL server.

Procesamiento de Consultas Analítico y Eficaz

CPU por CPU, PADB MPP- arquitectura columnar, ofrece naturalmente un mayor rendimiento de DWH que otras bases de datos. El alto rendimiento orgánico es generado porque se necesitan menos CPU para alcanzar el rendimiento del DWH que se necesita. Inherente al alto rendimiento también significa una mayor productividad para la parte analítica de negocios y de TI.

XI BDOC vs BDOF

En principio, los microprocesadores de 64 bits que se encuentran en muchos computadores pueden abordar 16 exabytes de memoria.

Capacidad tecnológica del mundo para almacenar información creció de 2,6 exabytes en 1986 a 15,8 en 1993, más de 54,5 en 2000, y 295 exabytes en 2007 - lo que equivale a menos de un CD de 730 MB por persona en 1986, alrededor de 4 CDs por persona de 1993, 12 CDs por persona en el año 2000, y casi 61 CDs por persona en 2007 - apilando los CDs a partir de 2007 se podría crear una pila de la tierra a la luna y una cuarta parte de la distancia más allá.

Los mundos de la capacidad tecnológica para recibir información a través de redes de difusión unidireccionales era 432 exabytes de información en 1986, 715 exabytes en 1993, 1.200 exabytes en 2000 y 1.900 en 2007.

Capacidad efectiva del mundo para el intercambio de información a través de redes de telecomunicaciones bidireccionales era 0.281 exabytes de información en 1986, 0.471 en 1993, 2,2 en 2000, y 65 exabytes en 2007.

En el 2004, el tráfico global de Internet mensual pasó 1 Exabyte por primera vez. En enero de 2007, Bret Swanson del Discovery Institute acuñó el término exaflood para una inundación supuestamente inminente de exabytes que causaría el colapso congestivo del Internet. Sin embargo, el tráfico global de Internet ha continuado su crecimiento exponencial, sin ser molestado, y en marzo de 2010 se estimó en 21 exabytes por mes.

Según el pronóstico de Cisco Visual Networking Index tráfico IP, el tráfico global de datos móviles crecerá a una tasa compuesta anual de 131 por ciento entre 2008 y 2013, llegando a más de dos exabytes por mes en el 2013.

Según la actualización del 2011 de las previsiones de tráfico IP de Cisco VNI, el año 2015, el tráfico IP global anual alcanzará los 966 exabytes o casi un zettabyte completo. Video por Internet representará el 61% del total de datos en internet.

Según la actualización de febrero de 2013 Cisco VNI Forecast para 2012-2017, el tráfico IP global anual pasará el umbral zetabyte a finales de 2016 - En 2016, el tráfico IP mundial llegará a 1,3 zettabytes por año o 110,3 exabytes por mes. En 2017, el tráfico mundial de datos móviles alcanzará los 11,2 exabytes por mes, creciendo 13 veces desde 2012 hasta 2017

La información sigue creciendo nadie va a parar eso, es mas va a ser peor.

Actualmente el porcentaje de usuarios que provee información a la red es mucho menor de los que lo usan.

Por esta razones se busca la optimización de grandes volúmenes de datos y velocidad de las consultas siendo una alternativa valida a soluciones como Teradata.

***XII Teradata Corporation
(NYSE: TDC)***

The logo for Teradata, featuring the word "TERADATA" in a bold, orange, sans-serif font. The letter "T" is significantly larger than the other letters. A registered trademark symbol (®) is located at the bottom right of the word.

Es una empresa estadounidense especializada en herramientas de data warehousing y herramientas analíticas empresariales.

Teradata Database Software: Motor de bases de datos relacional orientado al procesamiento de grandes volúmenes de datos.

Teradata Platform: Plataforma de hardware sobre la que puede correr su motor relacional.

Teradata Analytic Applications: Conjunto de herramientas y aplicaciones de acceso y gestión de datos.

Teradata Professional Consulting Services: Servicios profesionales de Teradata.

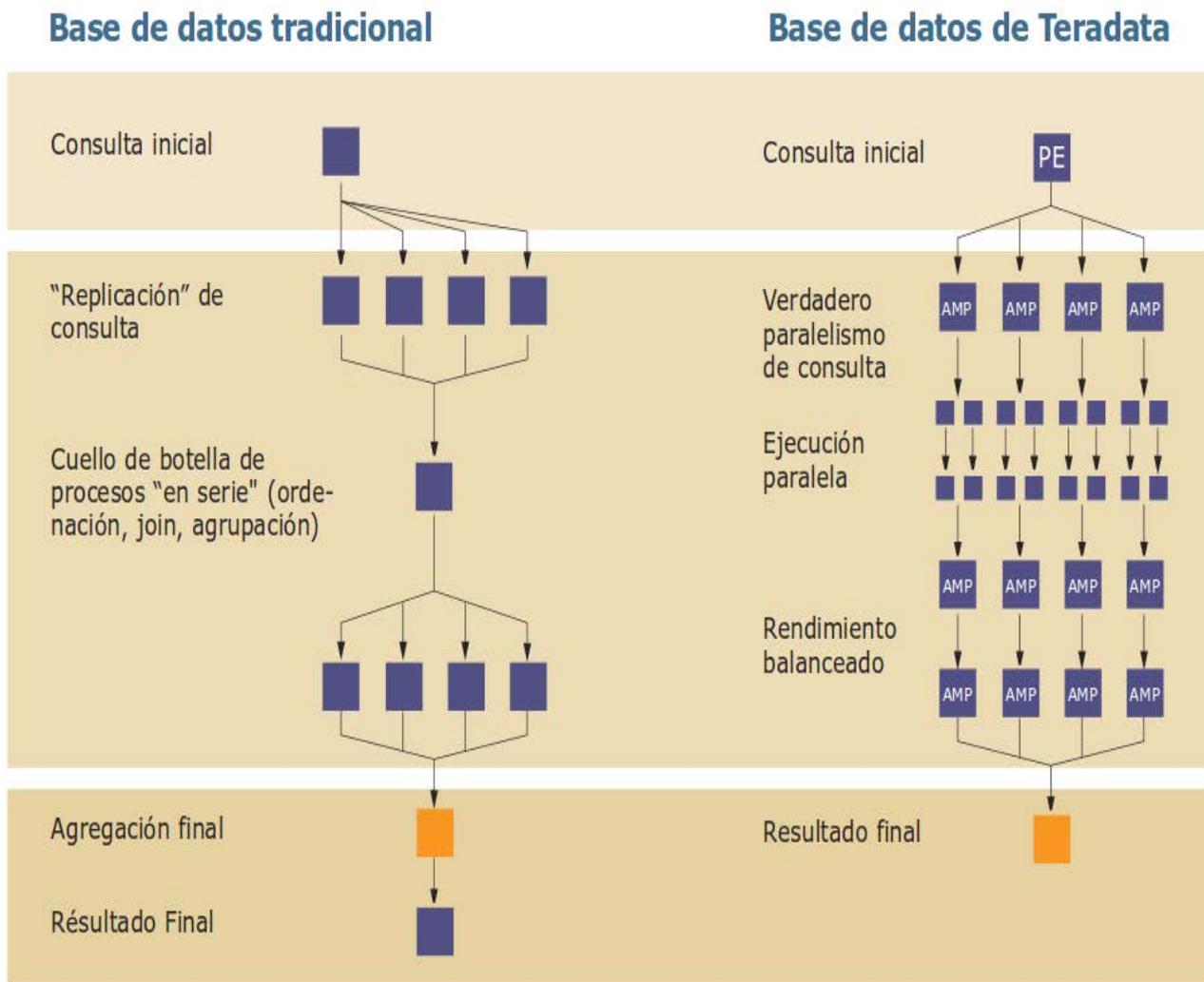


Figura 19: funcionamiento de Terada.

Teradata es un gestor de base de datos relacional específicamente diseñado para soportar paralelismo. Su arquitectura patentada permite descomponer las preguntas complejas entre múltiples unidades de trabajo paralelas en el software de la base de datos, cada una denominada AMP (Access Module Processors).

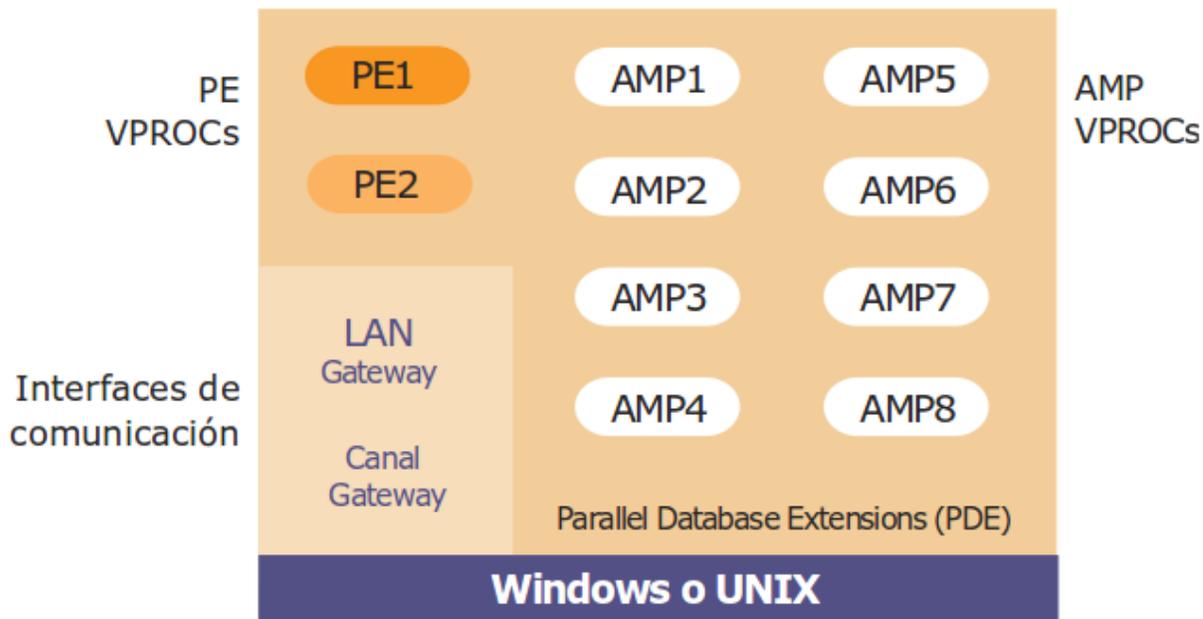
A cada AMP le corresponde una parte del espacio y de los datos en la base de datos. En cada nodo residen varios AMPs. Por consiguiente Teradata no está condicionada por la plataforma hardware para soportar paralelismo, escalabilidad o alta disponibilidad. Estas características son inherentes a su arquitectura de software e independientes del sistema operativo o de la configuración de hardware.

El AMP es un tipo de procesador virtual (VPROC). Existe otro VPROC, el PE (Parsing Engine), que descompone las consultas (sentencias SQL) en sentencias más sencillas, distribuyendo a continuación la tarea entre los AMPs. Varios PEs pueden convivir en un único nodo.

Teradata puede ejecutar múltiples AMPs y PEs en un único nodo gracias a la existencia de las PDE (Parallel Database Extensions). Las PDEs conforman la infraestructura necesaria para que la base de datos pueda trabajar en paralelo en entornos NCR UNIX SVR4 MP-RAS o Microsoft Windows. La base de datos funciona independientemente del sistema operativo.

Teradata Database siempre es Teradata, independientemente de del sistema operativo o plataforma hardware.

Paralelismo intranodo: En la Figura se presenta un ejemplo de paralelismo intranodo.



Arquitectura software de un nodo Teradata

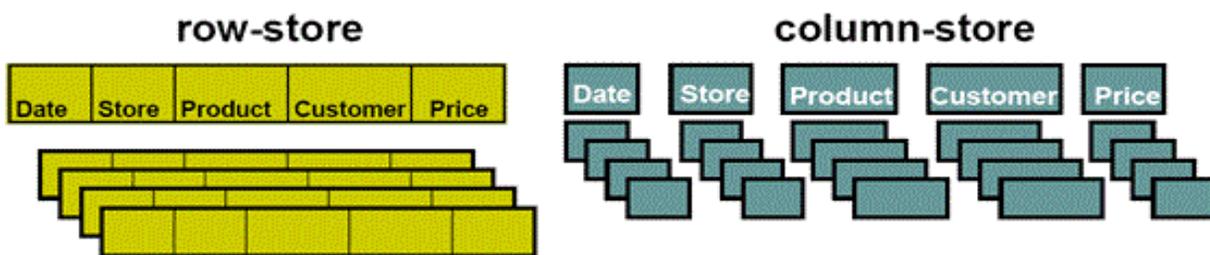
Figura 20: presenta un ejemplo de paralelismo intranodo de Teradata.

La configuración representada es un nodo SMP de 2 CPUs con diez VPROCs. En esta configuración hay ocho procesadores virtuales AMP y dos PEs. Cada PE tiene acceso a cada AMP, y esto permite que cada solicitud puede procesarse totalmente en paralelo. Los datos se distribuyen uniformemente entre todos los AMPs. Procesamiento paralelo masivo (MPP) Pero Teradata saca mucho más partido de esta arquitectura en aspectos de escalabilidad y disponibilidad.

Teradata Warehouse Especificaciones	
Capacidad de la base de datos	<ul style="list-style-type: none"> > 1.023 Petabytes (PB) > 512 nodos
Modelos de datos	<ul style="list-style-type: none"> > Relacional > Compatible con ANSI SQL-92 > Completamente paralela
Arquitecturas soportadas	<ul style="list-style-type: none"> > Multiproceso simétrico (SMP) > Proceso paralelo masivo (MPP)
Sistemas operativos soportados	<ul style="list-style-type: none"> > UNIX SVR4 MP-RAS de NCR > Microsoft Windows
Plataformas cliente soportadas	<ul style="list-style-type: none"> > Microsoft MS-DOS > Microsoft Windows > IBM VM, MVS, y OS/2 > NCR UNIX SVR4 MP-RAS, y la mayoría de las demás plataformas UNIX
Integración con Mainframe	<ul style="list-style-type: none"> > IBM > Amdahl > Bull
Preprocesadores	<ul style="list-style-type: none"> > PL/1, C, COBOL
Herramientas y utilidades Teradata Utility Pack	<ul style="list-style-type: none"> > BTEQ (consultas/redacción de informes) > Teradata SQL Assistant (consultas/DBA) > Teradata SQL Assistant Web edition (consultas/DBA) > ODBC > JDBC > OLE DB provider > Teradata Call Level Interface (múltiples plataformas) > Teradata MultiTool
Base de datos de Teradata Utilidades de administración	<ul style="list-style-type: none"> > Teradata Manager (administración del sistema) > Teradata Dynamic Query Manager (gestión de carga de consultas) > Teradata Metadata Services > Teradata Analyst Pack, compuesto por Teradata Visual Explain, Teradata System Emulation Tool, Teradata Index Wizard y Teradata Statistics Wizard
Utilidades de carga paralela	<ul style="list-style-type: none"> > TPump (actualización de datos) > FastLoad (carga de datos) > MultiLoad (carga/actualización de datos) > FastExport (extracción de datos)
Otras herramientas y utilidades	<ul style="list-style-type: none"> > ARCMAN (back-up en Mainframe) > TS/API

Figura 21: Especificaciones de Teradata Warehouse.

Teradata es una excelente aplicación para el manejo de bases de datos de gran tamaño pero su alto costo la limita solo a la aplicación en grandes empresas.^{XIII}



Funcionamiento de la base de datos orientada a la columna.

Figura 22: Ejemplo de almacenamiento a la fila y almacenamiento a la columna,

La base de datos orientada a filas debe leer toda la fila con el fin de acceder a los atributos necesarios. Como resultado, las consultas analíticas y de inteligencia de negocios terminan leyendo data más de lo necesario para satisfacer su consulta.

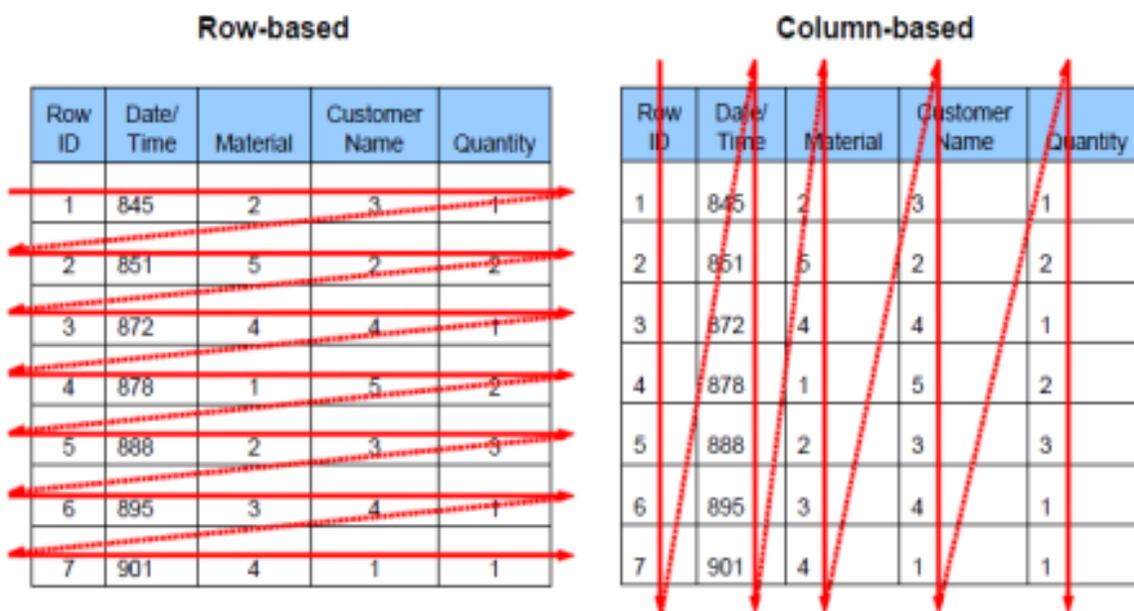


Figura 23: ejemplo de lectura entre BDO y BDO

C

Tradicionalmente las bases de datos hacen esto para guardar los datos, Esto es rápido para operaciones de escritura pero no de lectura.

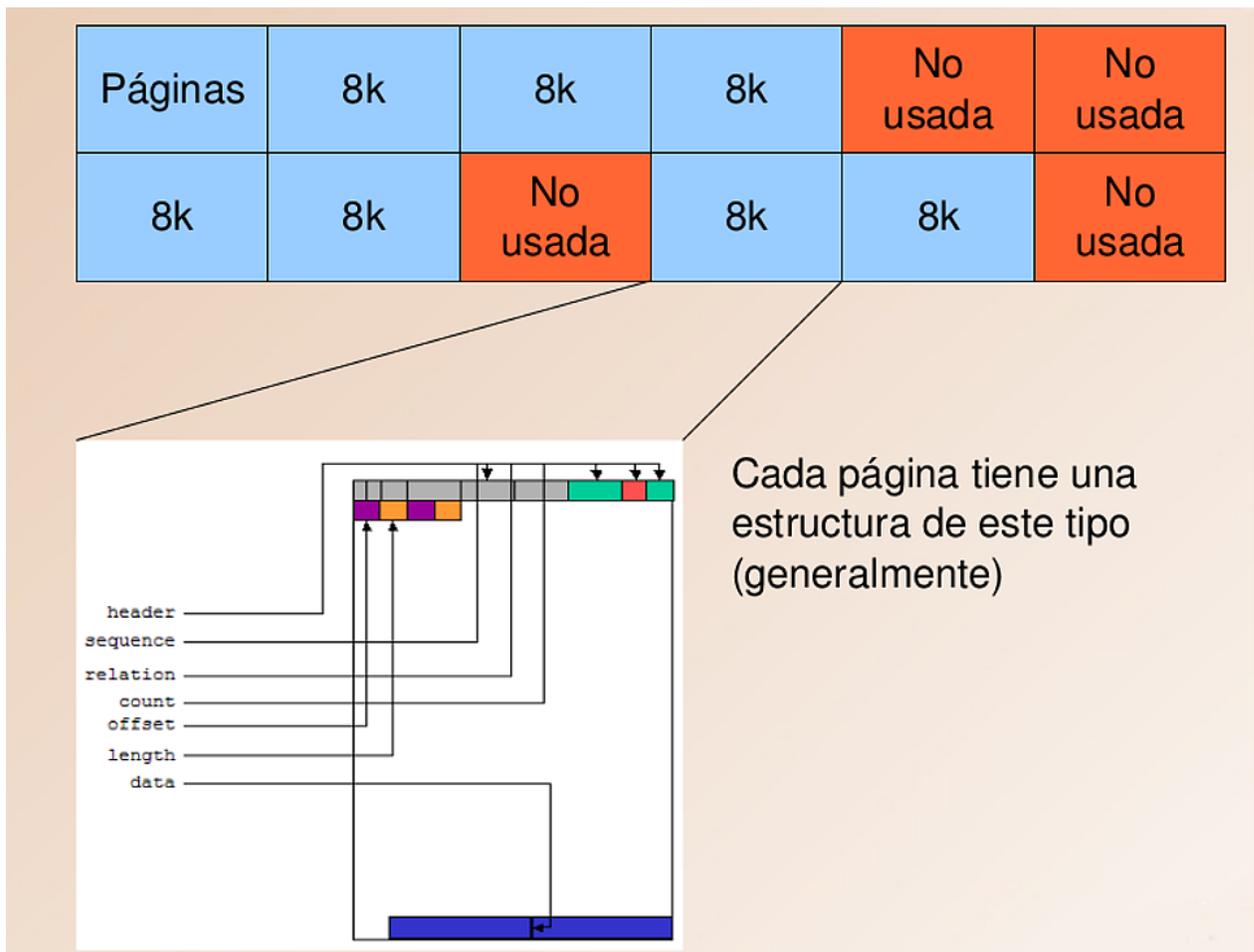


Figura 24: ejemplo de almacenamiento de datos.

Además este tipo de bases de datos habiendo sido diseñada para actividades transaccionales, es a menudo construida para la recuperación óptima y unión de conjunto de datos pequeños en lugar de grandes cargando así los subsistemas de entrada y salida que soportan el almacenamiento analítico. En respuesta, los administradores de base de datos tratan de ajustar el entorno de las diferentes consultas mediante la construcción de índices adicionales así como la creación de vistas especiales. Esto requiere mayor tiempo de procesamiento y consumo adicional de almacenamiento de datos.

Al contrario de estas bases de datos, están las orientadas a columnas, que como su nombre lo indica sus datos están organizados y almacenados en columnas. Debido a que cada columna puede ser almacenada por separado, para cualquier consulta, el sistema puede evaluar las columnas que se está accediendo y recuperar sólo los valores solicitados en las columnas específicas.

Estructura de almacenamiento columnar

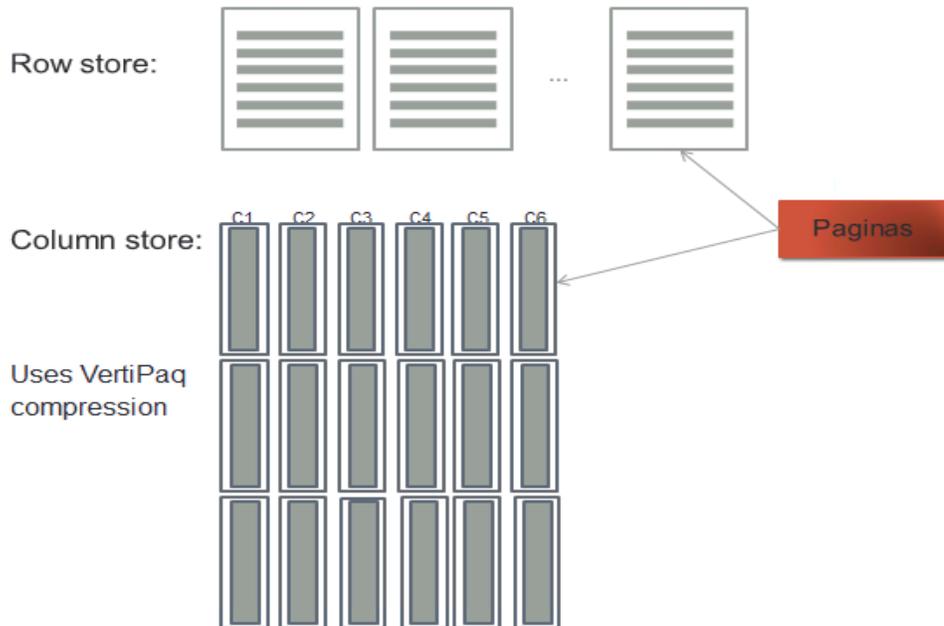
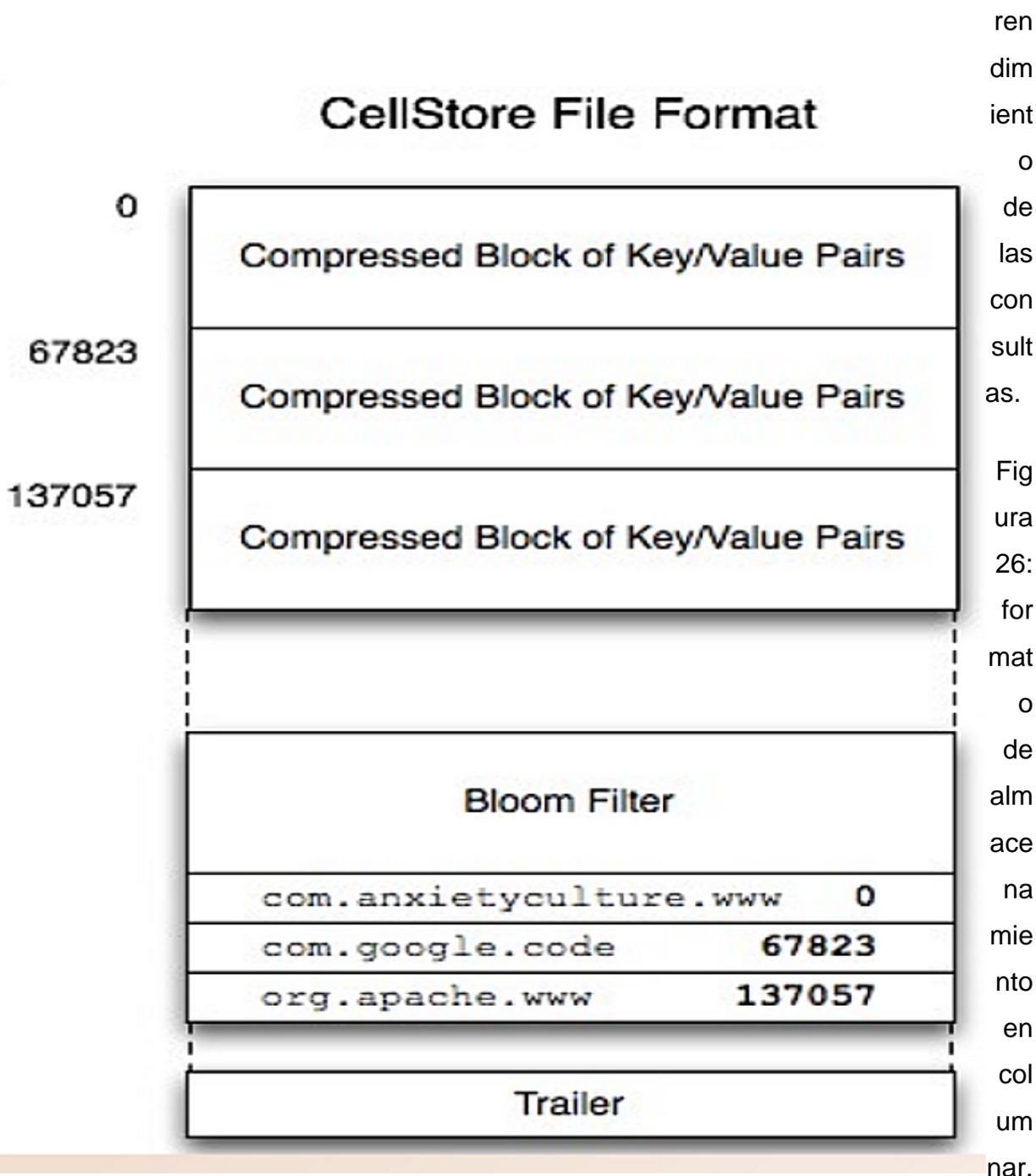


Figura 25: Ejemplo de almacenamiento columnar.

En lugar de exigir los índices separados para las consultas de forma óptima los datos se valora dentro de cada forma de columna del índice, reduciendo los sistemas de entrada y salida lo que permite un acceso rápido a los datos mejorando el tiempo y el



ren
dim
ient
o
de
las
con
sult
as.
Fig
ura
26:
for
mat
o
de
alm
ace
na
mie
nto
en
col
um
nar.

Para la realización de los test de Rendimiento se utilizo 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.

XIV.2 Características Hardware del Sistema:

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

Memoria RAM instalada: 4,00 GB

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

XIV.3 Instalación de LucidDB

1.- Como prerrequisito es necesario tener configurado el entorno virtual de Java

(JRE)

2.- Descargamos 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.- Descomprimos el paquete y se ejecuta 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.- Se debe instalar un cliente sql para trabajar. Se eligió `squirrel-sql` por su integración con LucidDB, para ello se descarga 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.

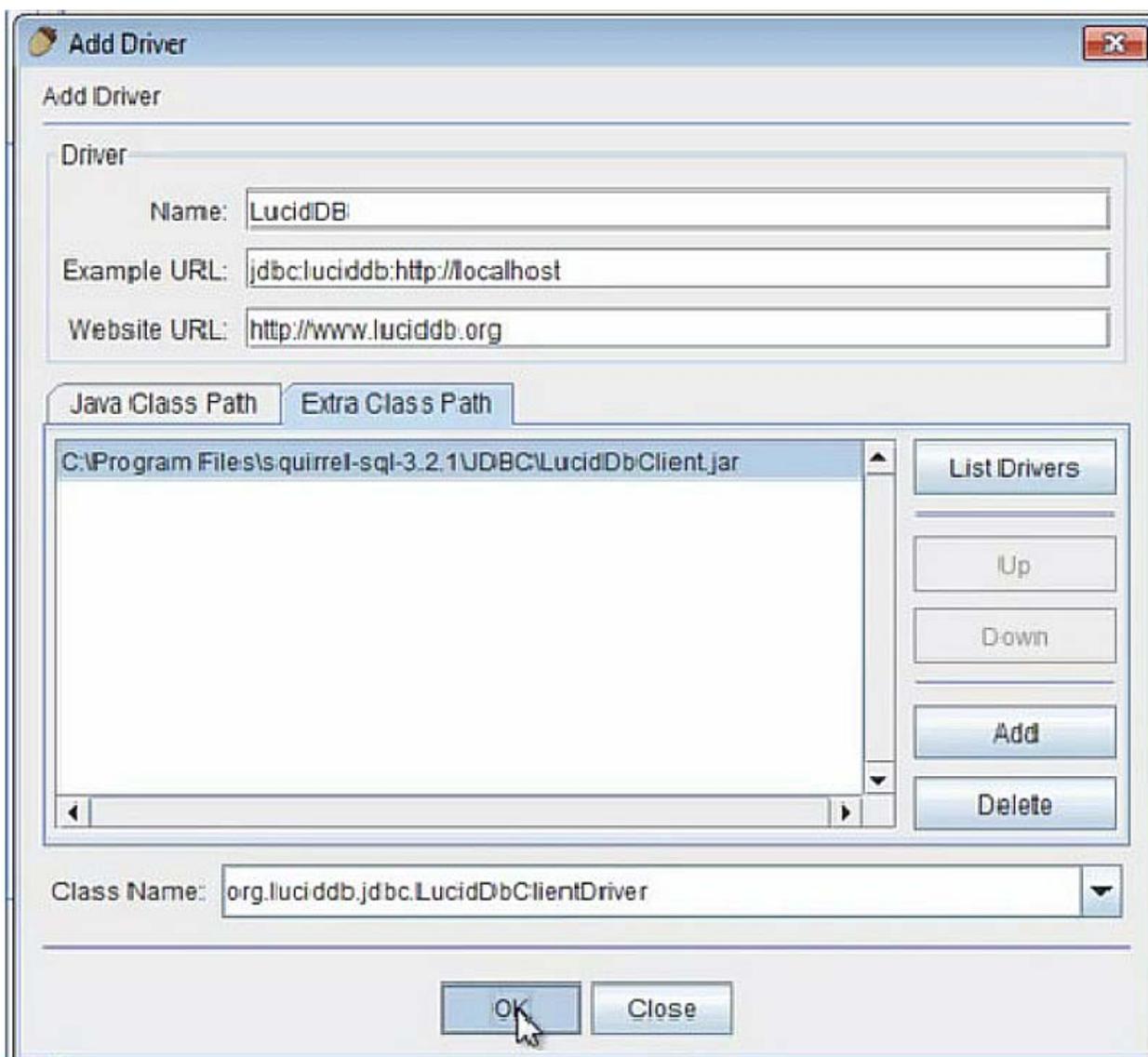


Figura 27: selección de driver de LucidDB.

El siguiente paso sería crear la conexión desde squirrel, se añade un alias como se muestra a continuación y se conecta, con lo que se puede ver los catálogos y esquemas de la base de datos

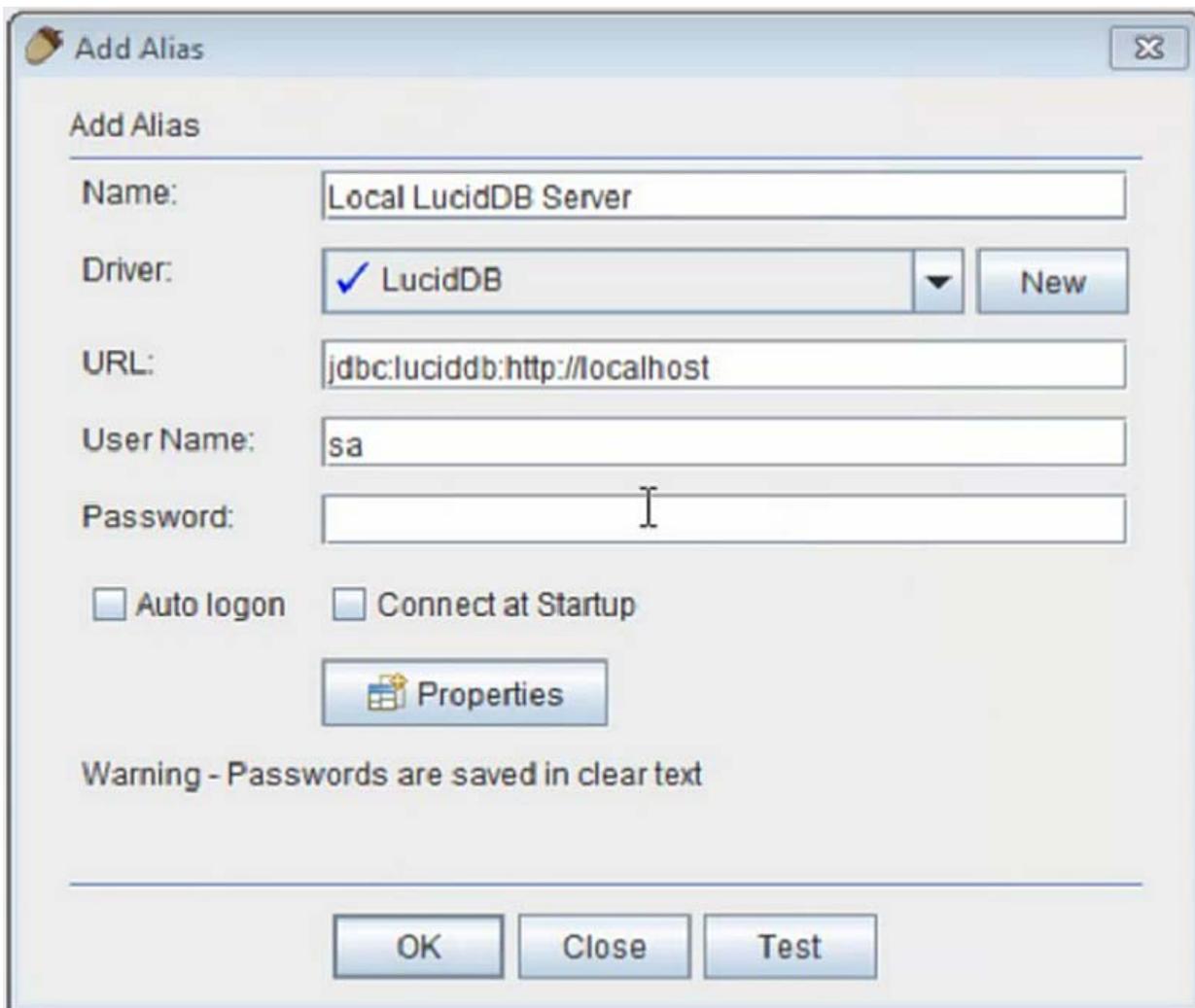


Figura 28: Creación de un alias.

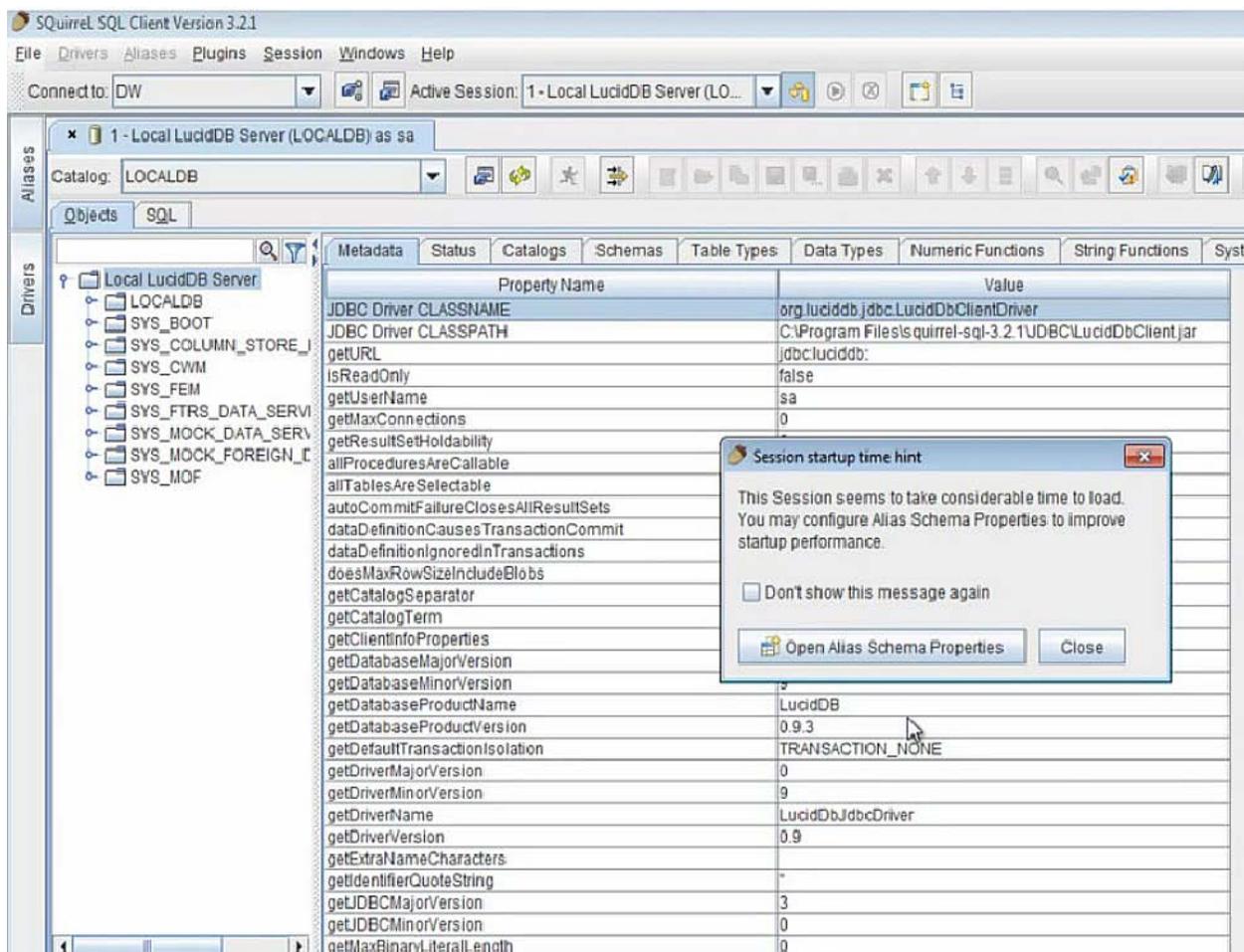


Figura 29: Catalogo y esquemas de la base de datos.

XIV.4 Carga de Datos en LucidDB

Para la carga de los datos en LucidDB. En las capturas de pantalla que se acompañan se pueden ver los parámetros necesarios para una correcta ejecución de la secuencia de carga. 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 pese a ser open source la realización de cargas incrementales y actualizaciones.

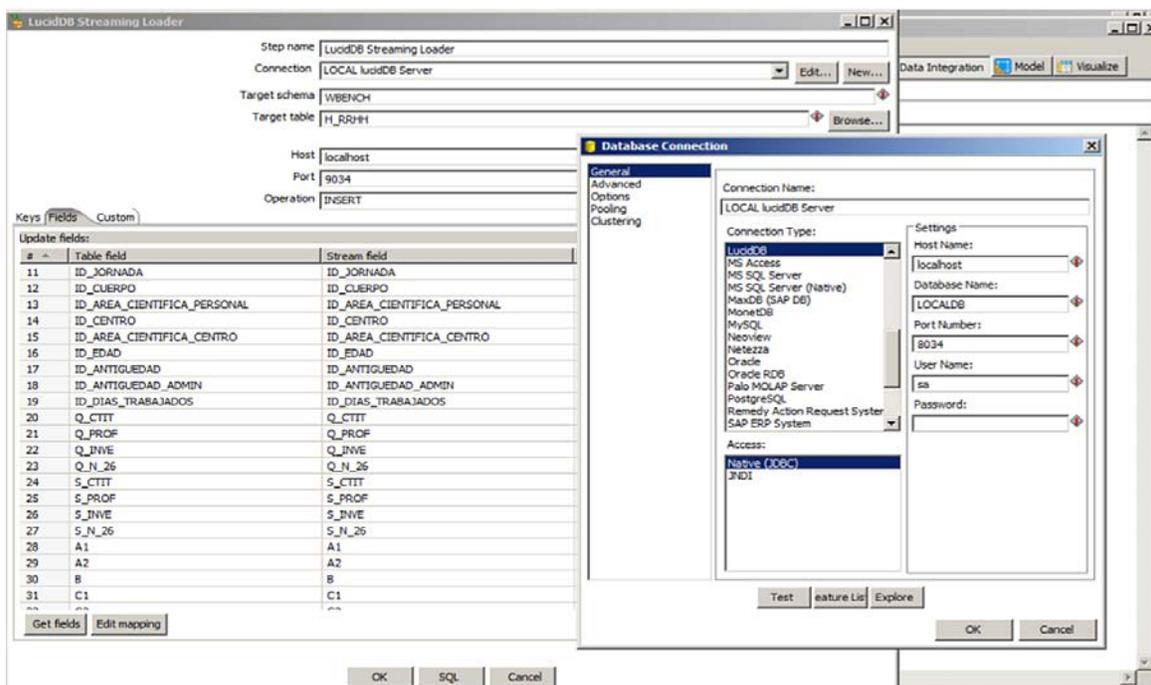


Figura 30: Ejemplo de carga de datos.

XIV.5 Instalación de InfoBright Community Edition

1.- Se descargo de la sección Community la última versión de Infobright, se ejecuta el .exe con el instalador. Se utilizo la versión 4.0.5 en su versión de 64 bits. El instalador crea InfoBright como un servicio de Windows, que se debe cambiar 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.- Se puede 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.

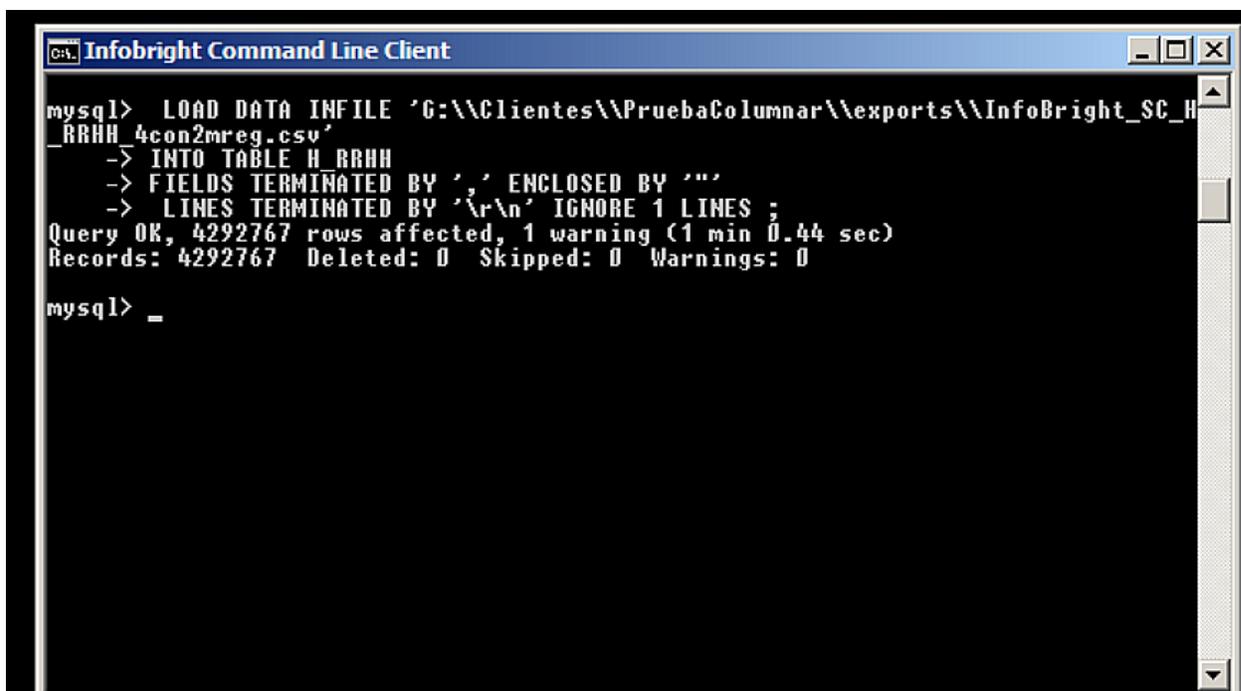
5.- La creación de una base de datos y una tabla es idéntica a MySQL, la principal diferencia es que el motor de InfoBright utiliza el denominado BrightHouse (mysql> create table <nombre_tabla> (<columna(s)>) engine=brighthouse;)

7.- InfoBright 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.- InfoBright 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 autoincrementales e índices. Tampoco están soportados los valores por defecto ni referencias a otras tablas de las columnas de una tabla.

Se realizó 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 dio 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 la 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.

A screenshot of a Windows command prompt window titled "InfoBright Command Line Client". The window shows a MySQL command being executed: "mysql> LOAD DATA INFILE 'G:\\Clientes\\PruebaColumnar\\exports\\InfoBright_SC_H_RRHH_4con2mreg.csv'". The command is followed by several options: "-> INTO TABLE H_RRHH", "-> FIELDS TERMINATED BY ',' ENCLOSED BY ''''", and "-> LINES TERMINATED BY '\\r\\n' IGNORE 1 LINES ;". The output of the command is: "Query OK, 4292767 rows affected, 1 warning (1 min 0.44 sec)", "Records: 4292767 Deleted: 0 Skipped: 0 Warnings: 0". The prompt "mysql> _" is visible at the bottom of the window.

```
mysql> LOAD DATA INFILE 'G:\\Clientes\\PruebaColumnar\\exports\\InfoBright_SC_H
_RRHH_4con2mreg.csv'
-> INTO TABLE H_RRHH
-> FIELDS TERMINATED BY ',' ENCLOSED BY ''''
-> LINES TERMINATED BY '\\r\\n' IGNORE 1 LINES ;
Query OK, 4292767 rows affected, 1 warning (1 min 0.44 sec)
Records: 4292767 Deleted: 0 Skipped: 0 Warnings: 0
mysql> _
```

Figura 32: carga de datos en InfoBright.

XIV.7 Consultas realizadas a la base de datos con los distintos motores de búsqueda de datos.

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.

Consulta 1:

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

Consulta 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;
```

Consulta 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
```

Consulta 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;

Consulta 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;
```

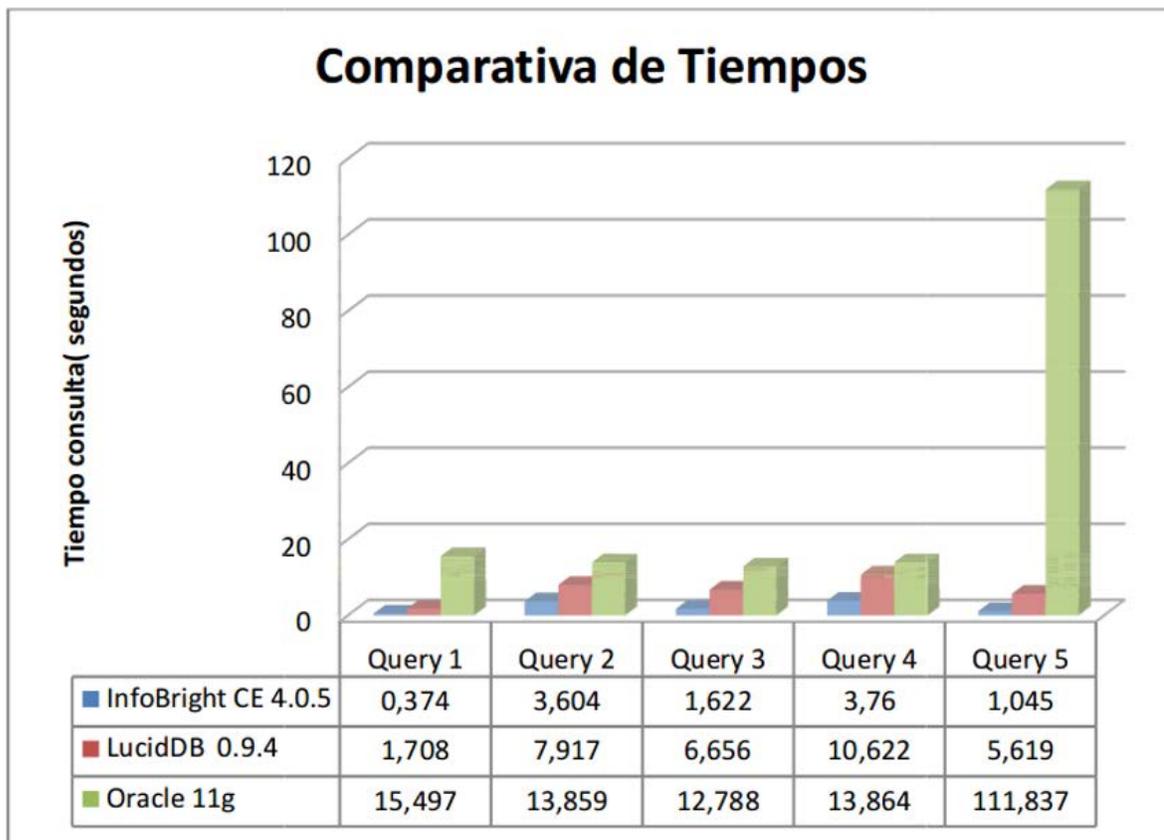


Figura 33: gráfica comparativa de tiempos.

XIV.8 Resultados

A la vista de los resultados vemos como InfoBright CE es la que tiene 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.

XV Conclusiones XV.1 Ventajas

La principal ventaja de este tipo de sistemas es el rápido acceso a los datos: esto ya lo hemos demostrado con el modelo DSM el cual nos permite consultar rápidamente los datos columna a columna, al guardarse físicamente de manera contigua.

El tiempo de acceso al disco, la velocidad del disco suele ser el cuello de botella en los sistemas de almacenamiento ya que es notablemente mas lento que el poder de procesamiento.

Customer Benchmark: 6 months Stock Trade History Data (1.5TB raw data)

	Row-Store	Column Store (Vertica)	Speed Increase
Query 1	23 minutes	44 seconds	30x
Query 2	4.5 hours	1 second	16,200x
Query 3	19 minutes	25 seconds	45x
Query 4	36 minutes	8 seconds	270x
Query 5	23 minutes	6 seconds	230x
Geometric Mean Query Time	39.6 minutes	8.8 seconds	270x
Hardware	Shared storage: 6 servers (each with: 8 CPUs, 16GB RAM); 15TB SAN	Shared Nothing: 6 servers (each with: 2 CPUs, 16GB RAM, 1TB local disk); no SAN	

Figura 34: Pruebas de velocidad de respuesta de consultas entre BDOF y BDOC.

Un DBMS en una base de datos orientada a columnas, lee solo los valores de columnas necesarios para el procesamiento de una consulta determinada por lo cual las bases de datos orientadas a columnas tienen una mayor eficiencia en entornos de almacenes, donde las consultas, típicas incluyen los agregados realizados por un gran número de elementos de datos.

Se comprime la información asignable de cada columna con el fin de mejorar el procesamiento desde el ancho de banda del acceso a disco.

Cambios en el esquema tiene menor impacto y por lo tanto el costo de realizarlo es menor.

XV.2 Desventajas

- No orientado a transacciones: este es el factor más débil de esta tecnología. El hecho de tener los datos guardados columna a columna nos permite retornar las filas más rápidamente, pero al insertar, actualizar o borrar un registro, se deberá hacer en más de una ubicación (al tener que actualizar todos los pares clave-valor asociados a una relación). Por esta razón, este tipo de bases de datos no se recomienda para sistemas de tipo OLTP orientados a transacciones y alta concurrencia aunque cabe mencionar que el concepto de lento ya no es tan válido ya que el algoritmo de la base de datos ejecuta el proceso de escritura en forma muy rápida.

Un ejemplo que posee esta tecnología es el caso de twitter, que con aproximadamente 500 millones de usuario la velocidad de lectura es veloz, así como también lo es en el caso de facebook.

- Reportes operacionales: también llamados reportes de seguimiento en los que se desea ver toda la información de una relación que puede contener muchas tuplas. En algunos casos esto puede resultar ineficiente comparado con los Row-Stores

- No existe un modelo de datos que soporte teóricamente este modelo de base de datos

- No existe un estándar que unifique los criterios de implementación de este modelo de base de datos.

XVI Glosario

Java

Plataforma para el desarrollo de software creada por Sun Microsystems, ampliamente extendida hoy en día, que otorga independencia de plataforma al software creado en ella y lo provee de una gran cantidad de APIs estandarizados.

Open Source

Calificación de software que cumple una serie de requisitos, principalmente aquel que permite una libre redistribución, distribuye el código fuente, y permite modificaciones y trabajos derivados.

Database management system (DBMS)

Es un conjunto de programas que se encargan de manejar la creación y todos los accesos a las bases de datos, esta compuesto por:

DDL: Lenguaje de Definición de Datos

DML: Lenguaje de Manipulación de Datos

SQL: Lenguaje de Consulta.

Inteligencia de negocios (BI)

Conjunto de estrategias y herramientas enfocadas a la administración y creación de conocimiento mediante el análisis de datos existentes en una organización o empresa.

OLTP

Procesamiento de Transacciones En Línea (OnLine Transaction Processing)

Es un tipo de procesamiento que facilita y administra aplicaciones transaccionales, usualmente para entrada de datos y recuperación y procesamiento de transacciones

Data Warehouse (Almacén de datos): es una colección de datos orientada a un determinado ámbito (empresa, organización, etc.), integrado, no volátil y variable en el tiempo, que ayuda a la toma de decisiones en la entidad en la que se utiliza. Se trata, sobre todo, de un expediente completo de una organización, más allá de la información transaccional y operacional, almacenado en una base de datos diseñada para favorecer el análisis y la divulgación eficiente de datos.

MPP: (Procesadores Masivamente Paralelos) es el método de utilización de cientos o miles de microprocesadores coordinados para funcionar en conjunto.

Multihilo:(multithreading) multiples procesos de ejecución independientes para utilizar mejor los recursos aportados por las arquitecturas de procesadores modernos.

AMP (procesador de módulo de acceso): se utiliza para gestionar la base de datos, manejo de archivos, tareas y manipular el subsistema de disco en el entorno multi-tarea y posibilitar el procesamiento paralelo de la base de datos Teradata.

Vproc (Procesador virtual): es un proceso de software que se ejecuta en un entorno SMP (multiprocesamiento simétrico) o un nodo.

PE (motor de análisis): usado para control de la sesión, la tarea de envío y análisis de SQL en el entorno multi-tarea y posibilitar el procesamiento paralelo de la base de datos Teradata.

C-Store: Arquitectura diseñada para maximizar la capacidad y lograr buenos índices de compresión.

Tupla: Es una secuencia de valores agrupados, sirve para agrupar, como si fueran un único valor, varios valores que, por su naturaleza, deben ir juntos.

API: Interfaz de Programación de Aplicaciones.

SQL: lenguaje de consulta estructurado, Es un estándar en el lenguaje de acceso a bases de datos. Originalmente, era un lenguaje de acceso al sistema de gestión de bases de datos denominado DB2 en plataformas 390 de IBM. En la actualidad está adoptado por ISO.

ODBC (Open Database Connectivity): es un estándar de acceso a bases de datos.

DBC (Java Database Connectivity): es una API que permite la ejecución de operaciones sobre bases de datos desde el lenguaje de programación Java.

XML: proviene de eXtensible Markup Language ("Lenguaje de Marcas Extensible"). Se trata de un metalenguaje (un lenguaje que se utiliza para decir algo sobre otro lenguaje).

Exabytes: es una unidad de medida de almacenamiento de datos que equivale a 1018 bytes.

Zettabytes: es una unidad de medida de almacenamiento de datos que equivale a 1021 bytes.

Exaflood: La palabra está compuesta por dos términos: "flood", que en inglés significa diluvio, inundación, plétora, y el prefijo "exa", que se refiere a exabyte.

Drill: modo de búsqueda de datos en la bases de datos.

XVII Bibliografía

Internet

- www.sybase.com
- www.teradata.com
- <http://cassandra.apache.org/>
- <http://www.infobright.com/>
- <http://luciddb.sourceforge.net/>
- <http://www.vertica.com/the-analytics-platform/>
- <http://www.paracel.com/>
- <http://nosql-database.org/>