

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 : Alfredo Alex Manns Arce
Profesor Guía : Roberto Caru Cisternas
Profesor Integrante : Jorge Tapia Castillo

Santiago – Chile

Junio, 2015

Agradecimientos

A Dios, padre eterno, por ser mi fuente de energía y entendimiento,

A mis padres: ... por impulsarme a esforzarme para ser una profesional

A mis familiares que me dieron apoyo y ánimos para terminar la carrera

A mi profesor guía que me aconsejó y guió en la realización de esta

...A todos muchas Gracias...

Índice de Contenido

I INTRODUCCIÓN	2
I.1 ¿Qué es una Base de Datos	2
I.2 Modelos principales de bases de datos:	3
I.2.1 Modelo Jerárquico:.....	3
I.2.2 Modelo en Red :	4
I.2.3 Modelo Relacional:	5
I.2.4 Modelo de bases de datos Deductivas:.....	6
I.3 Otros modelos que se están implementando son:	7
I.3.1 Bases de datos Orientadas a Objeto:	7
I.3.2 Modelo de base de datos orientado a la Columna.....	9
II MARCO TEORICO	10
II.1 Hipótesis	10
II.2. Objetivo General	10
II.3. Definición base de datos orientado a la columna	11
II.4 Variables de funcionamiento a considerar	12
II.4.1 Tiempo de carga.....	12
II.4.2 Carga Incremental	13
II.4.3 Compresión de datos.....	13
II.4.4 Técnicas de acceso.....	14
II.4.5 Rendimiento.....	15
II.4.6 Escalabilidad.....	15
III DESARROLLO	16
III.1. BD orientados a la fila versus BD orientada a la columna	16
III.2. Principales Bases de Datos Columnares Pagadas	20
III.2.1. Teradata Corporation (NYSE: TDC).....	20

III.2.2. Oracle Exadata Hybrid Columnar Compression	25
III.2.3. IBM DB2 10.5 con aceleración BLU	37
III.2.4. Base de Datos SYBASE	47
III.3. Principales Bases de Datos Columnares Open Source	51
III.3.1. Base de Datos Cassandra	51
III.3.2. Base de Datos LUCIDDB	57
III.3.3. <i>Base de Datos</i> INFOBRIGHT	60
III.3.4. Base de Datos VERTICA	62
III.3.5. Base de Datos QD TECHNOLOGY	65
III.3.6. Base de Datos ParAccel	69
III.4. Visión según Gartner	72
III.5. Pruebas realizadas	77
III.5.1 Entorno de la Prueba	77
III.5.2 Características Hardware del Sistema:	77
III.5.3 Instalación de LucidDB	77
III.5.4 <i>Carga de Datos en LucidDB</i>	82
III.5.5 Instalación de InfoBright Community Edition	83
III.5.6 Carga de datos en InfoBright Community Edition	84
III.5.7 Consultas realizadas a la BD con los distintos motores de búsqueda de datos.	85
IV CONCLUSION	89
GLOSARIO	93
BIBLIOGRAFIA	98

I INTRODUCCIÓN

I.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

I.2 Modelos principales de bases de datos:

I.2.1 Modelo Jerárquico:

Un modelo de datos jerárquico es un modelo de datos en el cual los datos son organizados en una estructura parecida a un árbol. La estructura permite a la información que repite y usa relaciones padre/Hijo: cada padre puede tener muchos hijos pero cada hijo sólo tiene un padre. Todos los atributos de un registro específico son catalogados bajo un tipo de entidad.

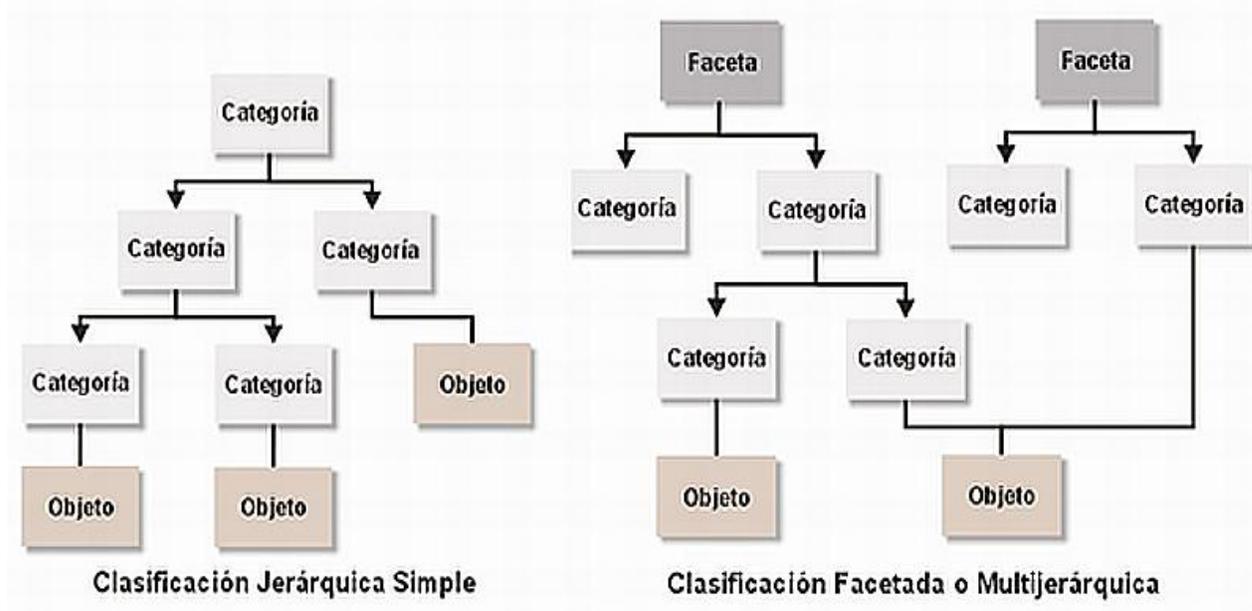


Figura 2: ejemplo modelo Jerárquico

I.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.

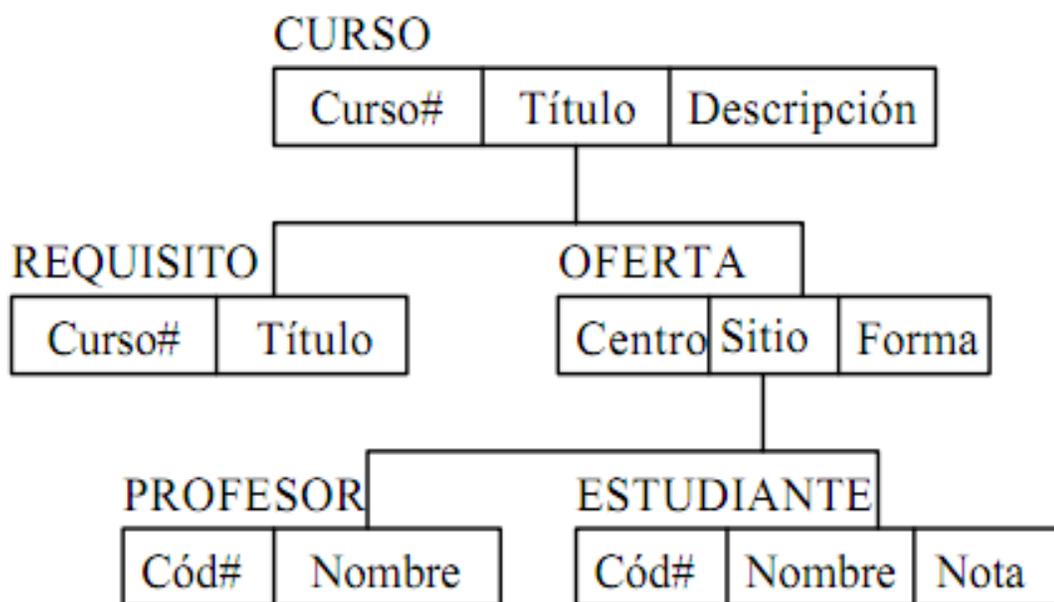


Figura 3: ejemplo modelo en Red.

I.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”.

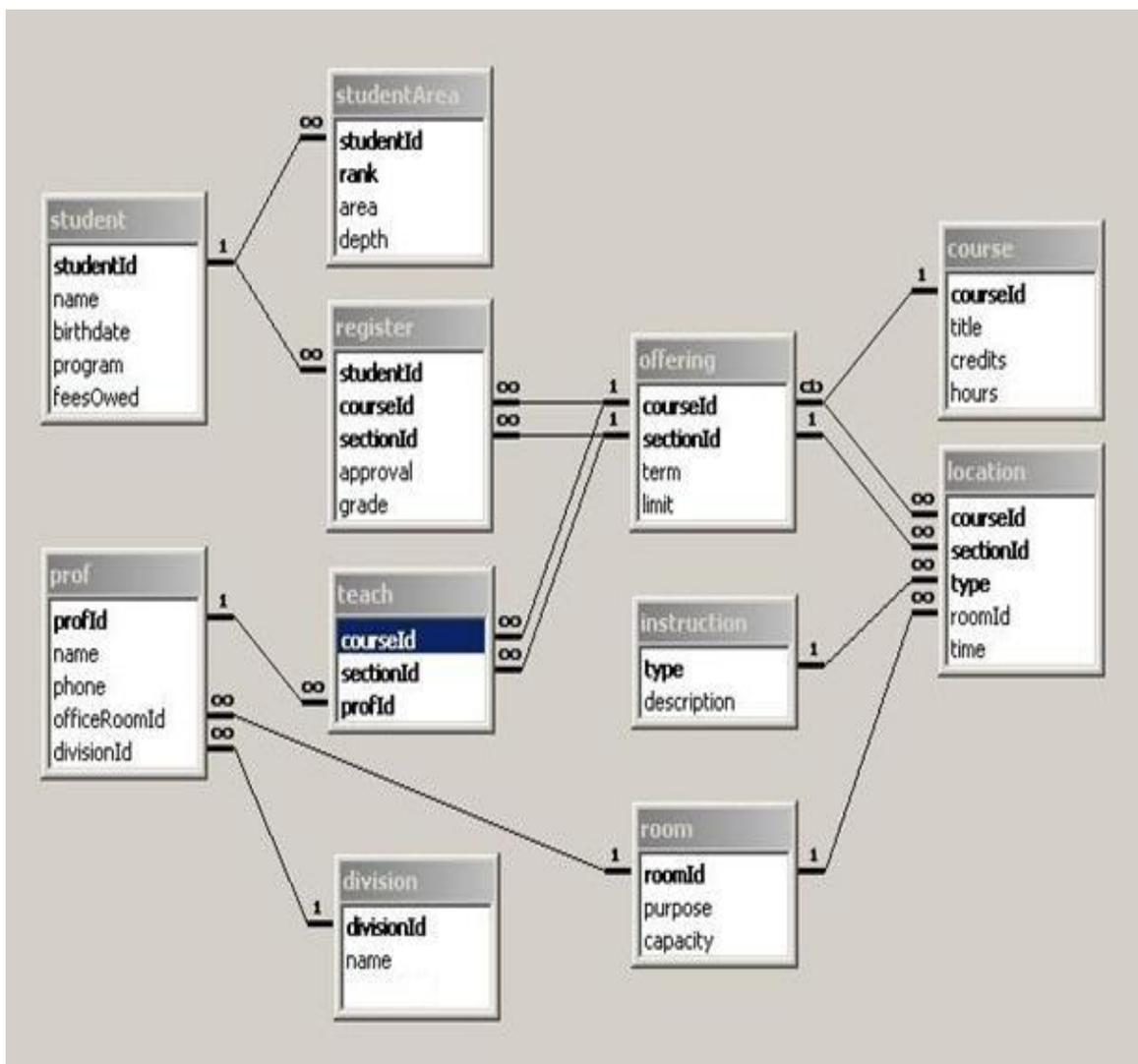


Figura 4: ejemplo modelo Relacional.

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.

I.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.

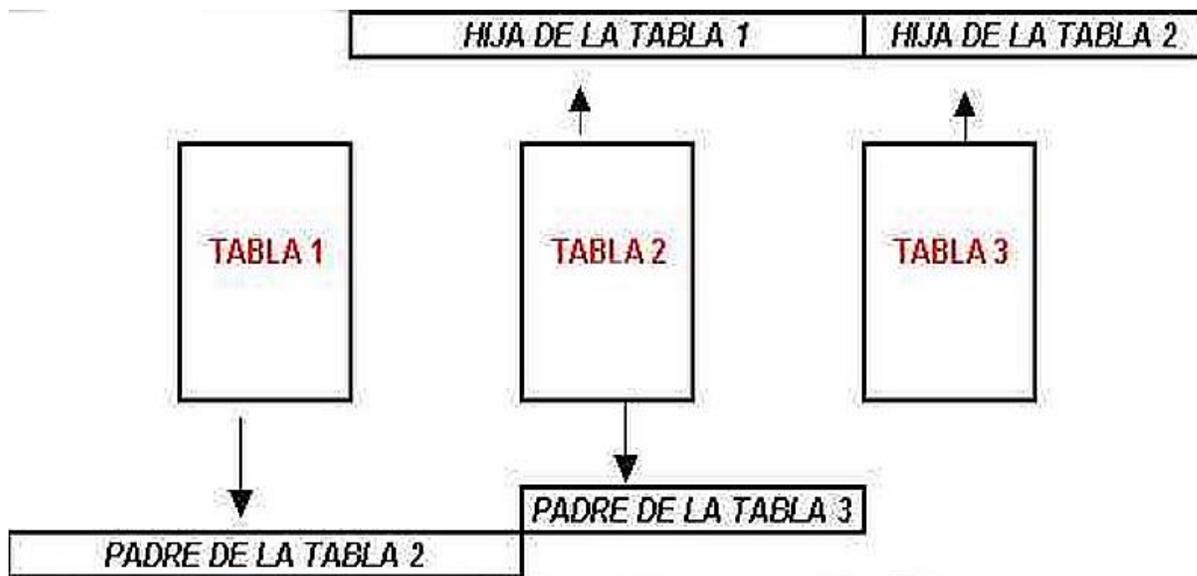


Figura 5: ejemplo modelo Deductivo.

I.3 Otros modelos que se están implementando son:

I.3.1 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.

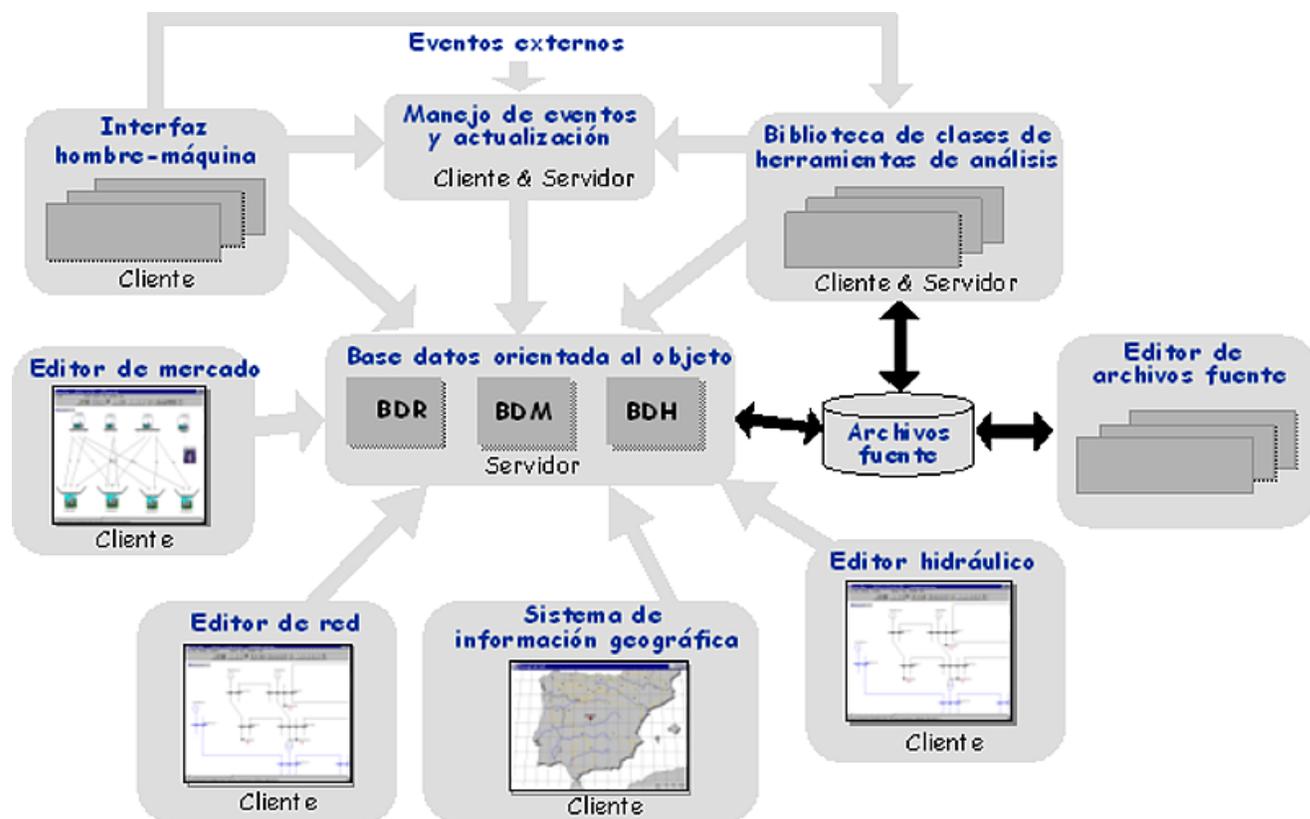


Figura 6: Ejemplo Modelo Base de Orientado a Objetos.

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.

I.3.2 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.

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
Data	Data	Data	Data	Data

Figura 7: Ejemplo Modelo Base de Orientado a Columna.

II MARCO TEORICO

II.1 Hipótesis

La investigación se orientara a demostrar como la estructura de las bases de datos orientado a la columna, logran mejorar el acceso de la información sobre grandes volúmenes de datos optimizando los tiempos de consulta, como se puede ver en el caso de twitter y facebook que usa la base de datos Apache Cassandra.

II.2. Objetivo General

Dar a conocer los avances de la tecnología de base de datos orientadas a columnas a través de 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, lo cual se realizara mediante la evaluación del rendimiento, la portabilidad y usabilidad de las tecnologías que se aplican.

II.3. 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.

II.4 Variables de funcionamiento a considerar

II.4.1 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 Terabytes por hora, que puede ser extremadamente lento, cuando se trata de decenas o cientos de Terabytes 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.

II.4.2 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.

II.4.3 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.

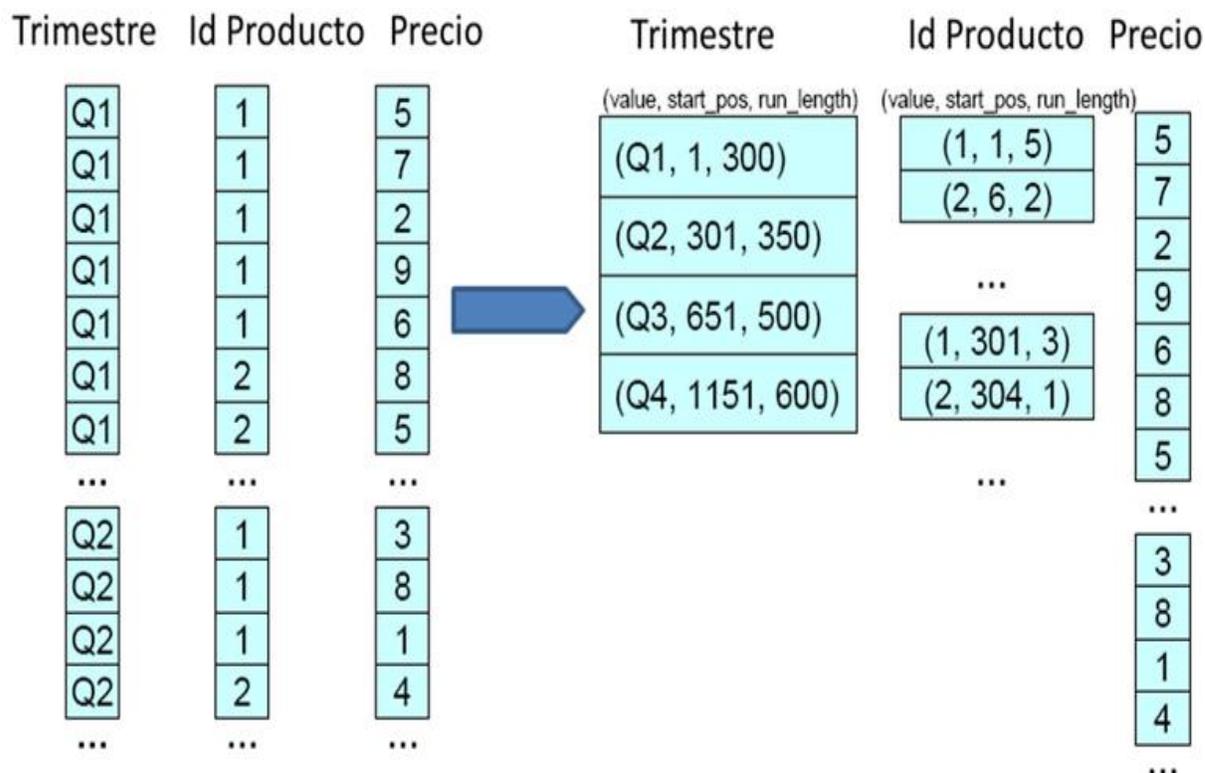


Figura 8: ejemplo de compresión de datos.

II.4.4 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.

II.4.5 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.

II.4.6 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.

III DESARROLLO

III.1. BD orientados a la fila versus BD orientada a la columna.

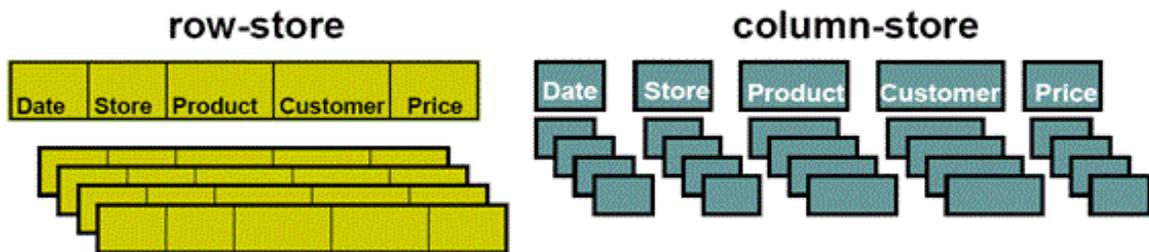


Figura 9: Ejemplo de almacenamiento a la fila y almacenamiento a la columna,

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 10: ejemplo de lectura entre BDOF y BDOC

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

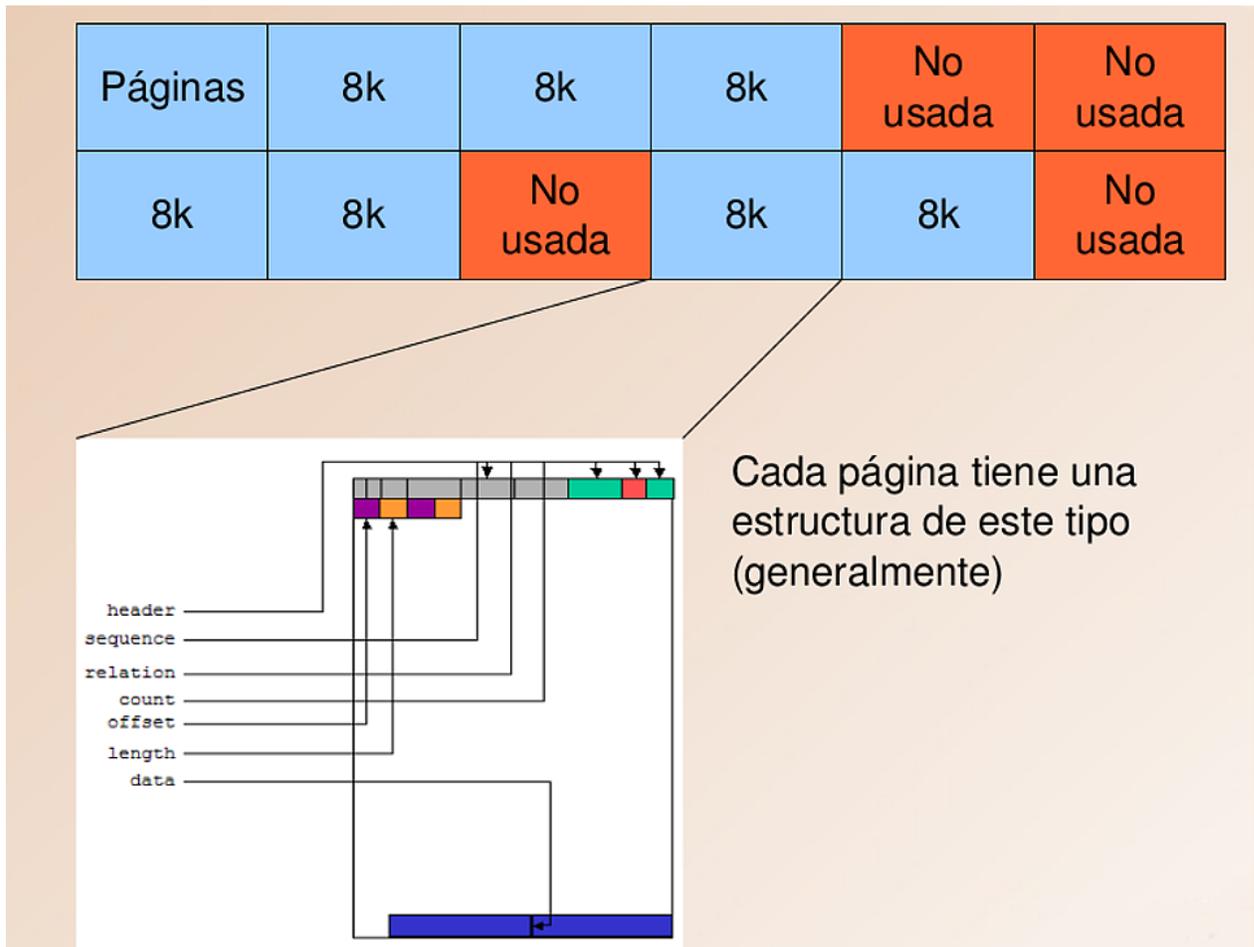


Figura 11: ejemplo de almacenamiento de datos orientado a la fila.

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

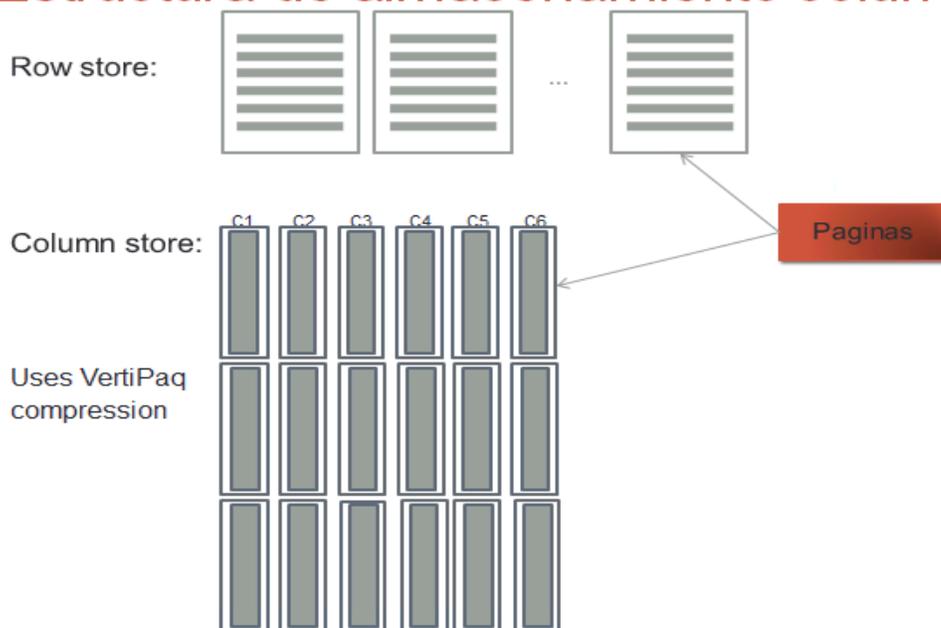


Figura 12: Ejemplo de almacenamiento columnar.

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 rendimiento de las consultas.

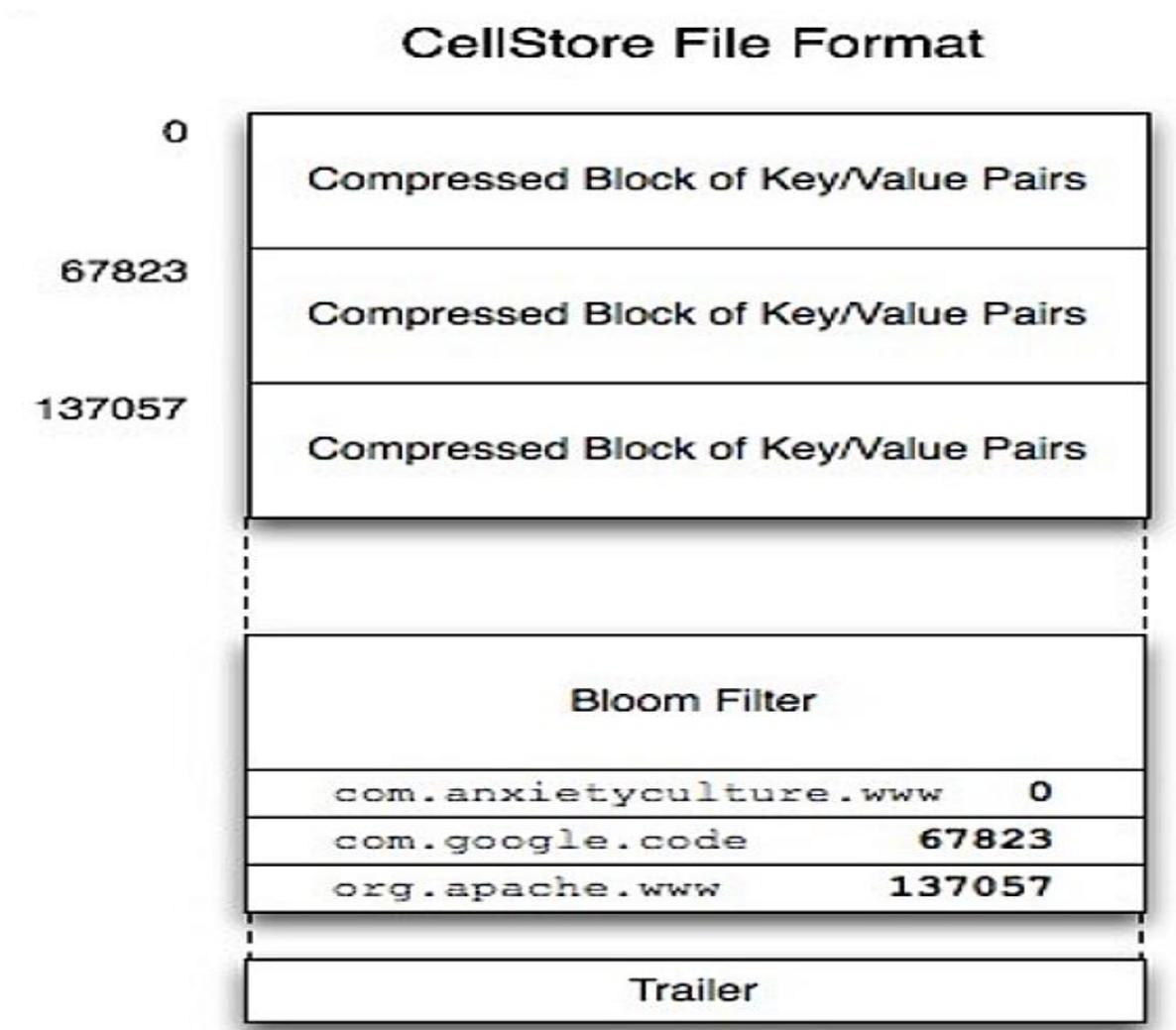


Figura 13: formato de almacenamiento en columnar.

III.2. Principales Bases de Datos Columnares Pagadas

III.2.1. Teradata Corporation (NYSE: TDC)



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.

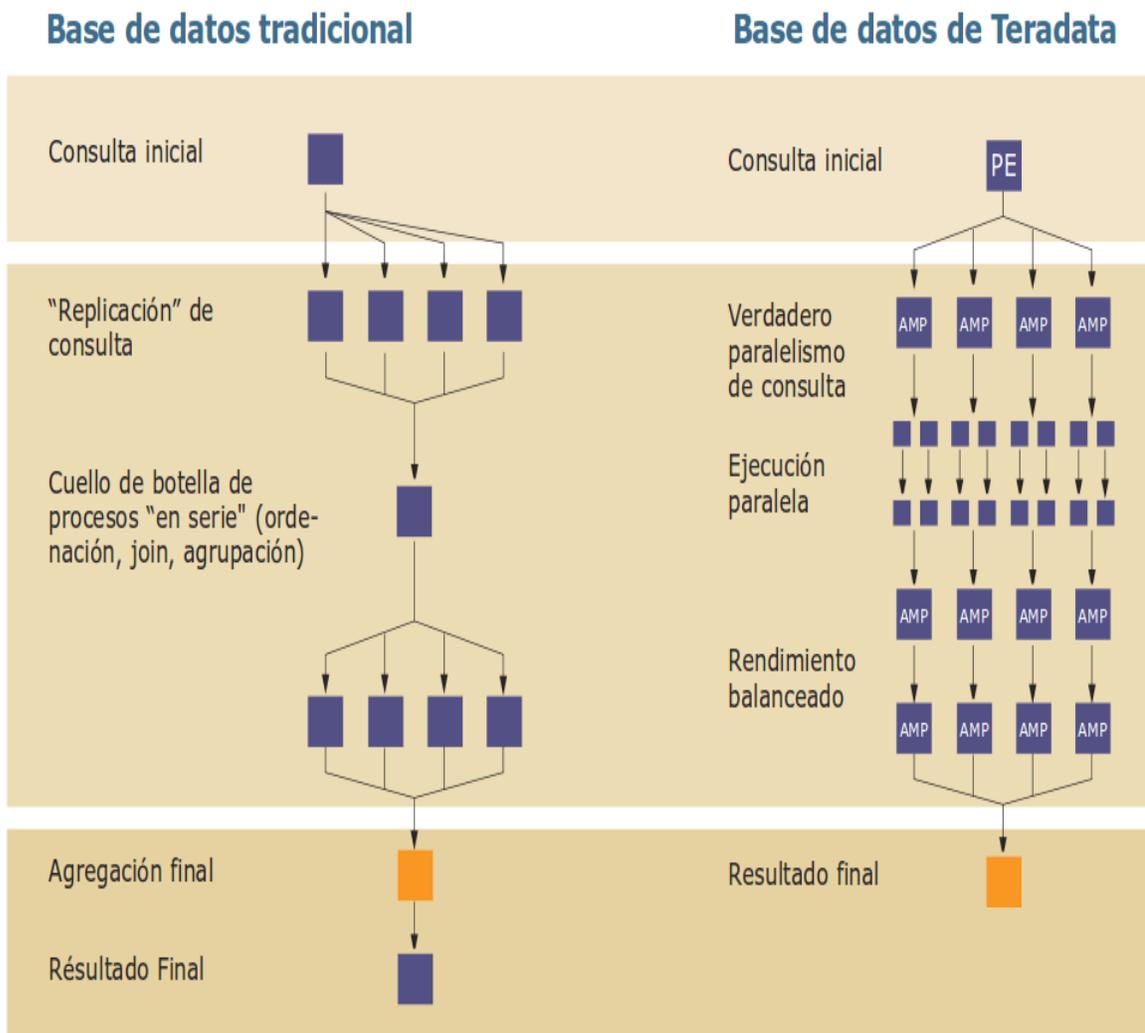


Figura 14: funcionamiento de Teradata.

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 Access Module Processors (AMP).

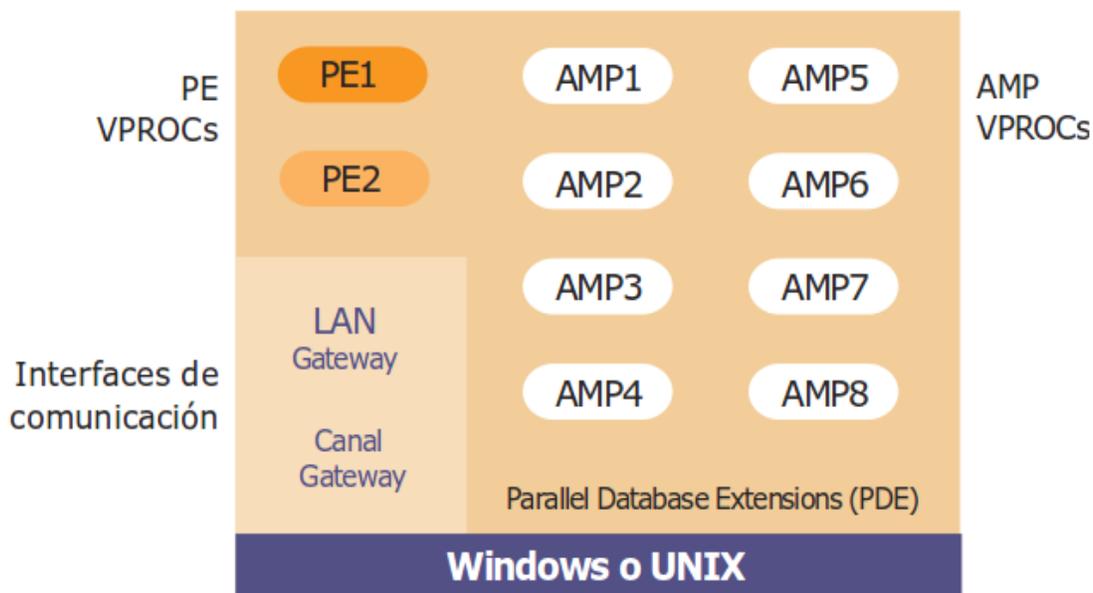
A cada AMP le corresponde una parte del espacio y de los datos en la base de datos. En cada nodo residen varios AMPs. Por consiguiente Teradata no está condicionada por la plataforma hardware para soportar paralelismo, escalabilidad o alta disponibilidad. Estas características son inherentes a su arquitectura 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 Parsing Engine (PE), 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 Parallel Database Extensions (PDE). 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

Figura15: 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 16: 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.

III.2.2. Oracle Exadata Hybrid Columnar Compression



Una de las características que Oracle nos provee para manejar adecuadamente este tipo de escenarios es la compresión de los datos utilizando "Hybrid Columnar Compression" que en esta ocasión lo estaremos aplicando en un ambiente de "Oracle Exadata Machine". Analizaremos resultados de compresión entre los diferentes tipos que existen.

"Exadata Hybrid Columnar Compression" es una característica incluida dentro de los "Exadata Storage Server". Esta funcionalidad provee un alto nivel de compresión de datos en objetos de una base de datos Oracle y tiene la posibilidad de personalizar el nivel de compresión dependiendo si el ambiente es de tipo OLTP (escrituras y lecturas frecuentes hacia datos no secuenciales) o del tipo "Datawarehousing" (consultas frecuentes hacia grandes cantidades de datos).

Esta característica permite a la base de datos reducir el número de lecturas y escrituras físicas requeridas para analizar una tabla, de esta manera grandes cantidades de información pueden ser procesadas rápidamente sin caer en altas tasas de I/O.

Es posible utilizar "Exadata Hybrid Columnar Compression" en varios niveles:

- Nivel de tabla
- Nivel de partición
- Nivel de Tablespace

Existen dos tipos de "Exadata Hybrid Columnar Compression":

- Warehouse Compression
 - Optimiza el rendimiento de consultas
 - Adecuado para datos de aplicaciones del tipo "Warehouse".
 - Existen dos opciones: Query High y Query Low

- Online Archival Compression
 - Se enfoca en obtener una máxima compresión de los datos.
 - Adecuado para datos que no cambian frecuentemente.
 - Existen dos opciones: Archive High y Archive Low

Sintaxis para comprimir una tabla

Query Compression:

```
CREATE TABLE emp (···) COMPRESS FOR QUERY [LOW | HIGH];
```

Archive Compression:

```
CREATE TABLE emp (···) COMPRESS FOR ARCHIVE [LOW | HIGH];
```

Nota: Si la tabla que se desea comprimir ya existe, se puede utilizar uno de los siguientes métodos:

Utilizando el paquete DBMS_REDEFINITION

```
ALTER TABLE <NOMBRE TABLA> MOVE COMPRESS FOR  
<TIPO COMPRESION>;
```

Organización de los datos

Las tablas son organizadas dentro de unidades de compresión (CU) tal como lo muestra la siguiente imagen:

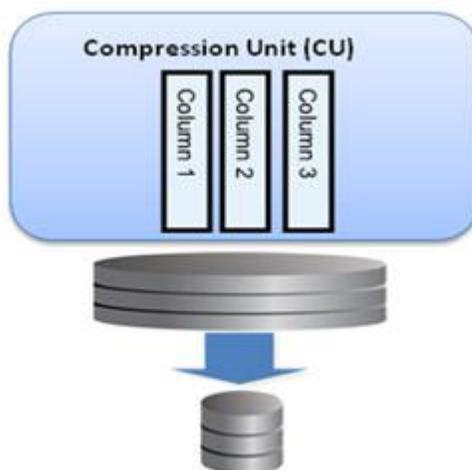


Figura 17: Unidad de Compresión

El tamaño de un CU comúnmente es de 32KB y esta dado por la siguiente fórmula:

Tamaño de un CU=Tamaño de un bloque x No. de Bloques dentro del CU

Ejemplo:

Tamaño de un bloque: 8KB

No. de bloques en un CU: 4

Tamaño de un CU: 8KB x 4=32KB

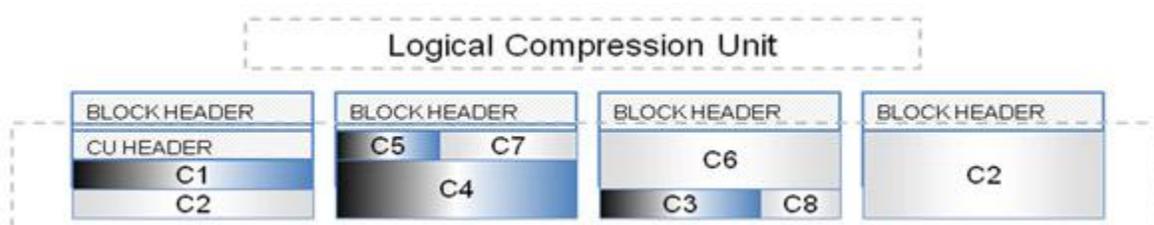


Figura 18: Bloques dentro de la Unidad de Compresión

Nota: En la imagen se muestra cuatro bloques dentro del CU, sin embargo, esto es posible que cambie, ya que un CU puede tener menos o más bloques dentro de él.

Características de "Exadata Hybrid Columnar Compression (EHCC)"

- Las unidades de compresión (CU), siempre son más grandes que los bloques de la base de datos.
- Con las unidades de compresión (CU), los datos son organizados por columnas, no por filas.
- Cada fila es auto contenida en una unidad de compresión (CU).
- Las estructuras lógicas abarcan múltiples bloques de base de datos.
- Cada columna es comprimida separadamente.
- Si se realiza una modificación a una fila, la unidad de compresión (CU) en la que se encuentra dicha fila es bloqueada en su totalidad.
- El beneficio en espacio al utilizar "Hybrid Columnar Compression" puede ser hasta 15 veces menos.
- En algunos estudios se ha logrado un beneficio en espacio hasta de 24 veces menos.
- La descompresión de los datos es realizada por los "Exadata Storage Server".
- Es posible utilizar diferentes formas de compresión en una misma tabla, si y solo si, la tabla está particionada.

- Algunas veces es necesario volver a comprimir la tabla después de algunas actualizaciones, según la nota de metalink No. 1332853.1.
- "Exadata Hybrid Columnar Compression" completa grandes niveles de compresión de datos que son insertados de manera directa.
- No es recomendado para tablas que son modificadas con mucha frecuencia.
- No es permitido en tablas organizadas por índices (IOT)

"Exadata Hybrid Columnar Compression" soporta:

- Índices tipo B-Tree, Bitmap y texto.
- Vistas Materializadas
- Operaciones utilizadas por "Smart Scan"
- Particionamiento
- "Parallel Query", PDML, PDDL
- "Physical Standby" con "Data Guard"
- Modificaciones de esquemas, Online, Metadata-Only, Agregar o eliminar Columnas.
-

"Exadata Hybrid Columnar Compression (EHCC)" está soportado en las siguientes soluciones:

- Oracle Exadata
- Oracle Exadata Expansion Rack
- Pillar Axiom
- ZFS Storage Appliance
- Super Cluster

Beneficios de "Exadata Hybrid Columnar Compression":

- Almacenamiento
- Baja transferencia de datos desde los discos
- Soportado por Flash Cache
- Soportado por DRAM Cache
- Ambientes de pruebas, Ambientes de desarrollos ambientes de recuperación a desastres

Estimación de la Compresión:

Existe un nuevo Consejero para compresión en la versión 11gR2 de una base de datos Oracle:

DBMS_COMPRESSION

El consejero reunirá información relacionada con la compresión de los datos, esto incluye estimaciones del nivel de compresión para una determinada tabla, tanto para tablas particionadas como tablas no particionadas. Esto ayudará al administrador a tomar mejores decisiones de compresión para sus objetos, basándose en la información proporcionada por el consejero.

Nota: Antes de ejecutar el consejero de compresión cree otro "Tablespace" en la base de datos ya que dicho proceso crea tablas temporales y realizará significantes lecturas y escrituras. Se recomienda ejecutar el consejero de compresión en periodos de tiempo en que no hay demasiada carga de trabajo del negocio.

Ejemplo de "Exadata Hybrid Columnar Compression"

A continuación se realizará un ejemplo paso a paso sobre compresión y descompresión de objetos dentro de una base de datos utilizando "Exadata Hybrid Columnar Compression". El ambiente utilizado para realizar el ejemplo es el siguiente:

Base de Datos: Oracle Enterprise Edition 11gR2 (11.2.0.3.0)

Tipo de Exadata: Oracle Exadata Quarter Rack

Sistema Operativo: Oracle Linux 5.5 x86-64

Identificar la tabla a comprimir:

```
SELECT segment_name,sum(bytes)/1024/1024 Size_MB
FROM user_segments
WHERE segment_name in ('EHCC_NOCOMPRESS')
GROUP BY segment_name;
```

SEGMENT_NAME	Size_MB
EHCC_NOCOMPRESS	25487.0625

Creación de la tabla utilizando compresión tipo "QUERY" y nivel alto:

```
SQL> CREATE TABLE EHCC_QUERY COMPRESS
FOR QUERY HIGH AS SELECT * FROM EHCC_NOCOMPRESS;
Table created.
```

Creación de la tabla utilizando compresión tipo "ARCHIVE" y nivel alto:

```
SQL> CREATE TABLE EHCC_ARCHIVE COMPRESS FOR ARCHIVE HIGH AS  
SELECT * FROM EHCC_NOCOMPRESS;  
Table created.
```

Verificar resultados:

```
SQL> SELECT segment_name,sum(bytes)/1024/1024 Size_MB  
FROM user_segments  
WHERE segment_name in ('EHCC_QUERY','EHCC_ARCHIVE')  
GROUP BY segment_name;
```

SEGMENT_NAME	Size_MB
EHCC_ARCHIVE	848
EHCC_QUERY	1280

Comparación entre EHCC y no-EHCC:

```
SQL>select table_name,compression,compress_for  
from user_tables  
where table_name in ('EHCC_ARCHIVE', 'EHCC_QUERY',  
'EHCC_NOCOMPRESS');
```

TABLE_NAME	COMPRESS	COMPRESS_FOR
EHCC_ARCHIVE	ENABLED	ARCHIVE HIGH
EHCC_QUERY	ENABLED	QUERY HIGH
EHCC_NOCOMPRESS	DISABLED	

Descomprimir una tabla:

```
SQL> ALTER TABLE EHCC_QUERY MOVE NOCOMPRESS PARALLEL;  
Table altered.
```

```
SQL> ALTER TABLE EHCC_ARCHIVE MOVE NOCOMPRESS PARALLEL;  
Table altered.
```

Comprobando resultados de descompresión:

```
SQL> SELECT segment_name,sum(bytes)/1024/1024 Size_MB  
FROM user_segments  
WHERE segment_name in ('EHCC_QUERY','EHCC_ARCHIVE')  
GROUP BY segment_name;
```

SEGMENT_NAME	Size_MB
EHCC_ARCHIVE	22947

EHCC_QUERY

22942

Verificar que el número de registros dentro de las tablas no se hayan alterado:

```
SQL>SELECT COUNT(*) FROM EHCC_NOCOMPRESS;
```

```
COUNT(*)
```

```
-----
```

```
102938023
```

```
SQL> SELECT COUNT(*) FROM EHCC_QUERY;
```

```
COUNT(*)
```

```
-----
```

```
102938023
```

```
SQL> SELECT COUNT(*) FROM EHCC_ARCHIVE;
```

```
COUNT(*)
```

```
-----
```

```
102938023
```

Resultados

Compresión:

Método de Compresión	Nombre de Tabla	Tamaño (MB)	Tiempo de creación (HH:MM:SI)
Sin Compresión	EHCC_NOCOMRESS	25487.0625	
Query High	EHCC_QUERY	1280	00:10:45.01
Archive High	EHCC_ARCHIVE	848	00:40:32.26

Descompresión:

Nombre de Tabla	Tamaño (MB)	Tiempo de descompresión (HH:MM:SI)
EHCC_QUERY	22942	00:03:08.42
EHCC_ARCHIVE	22947	00:02:57.58

III.2.3. IBM DB2 10.5 con aceleración BLU



BLU es una nueva tecnología desarrollada por IBM e integrada directamente dentro del motor de DB2. BLU es un nuevo motor de almacenamiento que permite guardar la información en forma columnar y así hacer la explotación de la información de una manera fácil y rápida. Esta característica incrementa el performance, ahorros en almacenamiento, optimización del uso de memoria, mejorar el I/O de manera eficiente y explotando las capacidades del CPU.

DB2 BLU esta diseñado bajo los siguientes conceptos

Simple y fácil de usar

DB2 10.5 con Aceleración BLU esta diseñado para bases de datos de tipo OLAP (On Line Analytics Process), es decir para la parte de analíticos. Es simple y fácil porque solo se requiere cambiar el valor de la variable de registro DB2_WORKLOAD antes de crear la base de datos. Al crear la base de datos DB2 configurará los parámetros necesarios para el uso óptimo de analíticos y también tomará en cuenta las características del Hardware.

Para verificar el valor de la variable de registro DB2_WORKLOAD se ejecuta el siguiente comando:

```
$ db2set -all
```

```
[i] DB2RSHCMD=/usr/bin/ssh  
[i] DB2COMM=TCPIP  
[i] DB2AUTOSTART=YES  
[g] DB2SYSTEM=db2awse  
[g] DB2INSTDEF=db2inst1
```

Figura 19 parametrización de variables ambientes

En la Figura 19 observamos que la variable DB2_WORKLOAD no esta configurada. Para configurar el valor de la variable de registro DB2_WORKLOAD a ANALYTICS y verificar que la variable de registro se haya cambiado ejecutamos el siguiente comando:

```
$ db2set DB2_WORKLOAD=ANALYTICS
```

```
$ db2set -all
```

```
[i] DB2_WORKLOAD=ANALYTICS  
[i] DB2RSHCMD=/usr/bin/ssh  
[i] DB2COMM=TCPIP  
[i] DB2AUTOSTART=YES  
[g] DB2SYSTEM=db2awse  
[g] DB2INSTDEF=db2inst1
```

Figura 20 visualización de variables ambientes

Para que el valor de la variable de registro tenga efecto hay que dar de baja la Instancia y levantarla nuevamente.

```
$ db2stop
```

```
$ db2start
```

Ahora solo se necesita crear la base de datos, crear las tablas y realizar la carga de datos. En este ejemplo, vamos a crear una base de datos llamada "pbacol".

```
$ db2 CREATE DATABASE PBACOL COLLATE USING IDENTITY
PAGE SIZE 32K
```

Conectarse a la base de datos para verificar que los parámetros que se configuraron al momento de crear la base de datos sean correctos.

```
$ db2 connect to pbacol
```

Los parámetros que DB2 configura automáticamente son los siguientes:

```
$ db2 get db cfg | grep -Ei --color "Database page
size|dft_table_org|dft_degree|dft_extent_sz|catalogcache_sz|sortheap|util_hea
p_sz|auto_reorg|sheapthres_shr"
```

```
Database page size = 32768
Degree of parallelism (DFT_DEGREE) = ANY
Sort heap thres for shared sorts (4KB) (SHEAPTHRES_SHR) = 55188
Sort list heap (4KB) (SORTHEAP) = 32768
Catalog cache size (4KB) (CATALOGCACHE_SZ) = 360
Utilities heap size (4KB) (UTIL_HEAP_SZ) = AUTOMATIC(31733)
Default tablespace extentsize (pages) (DFT_EXTENT_SZ) = 4
Automatic reorganization (AUTO_REORG) = ON
Default table organization (DFT_TABLE_ORG) = COLUMN
```

Nota: El valor de cada uno de los parámetros dependerá de la capacidad de memoria y de los recursos del sistema. Cada uno de estos parámetros se configura durante la creación de la base de datos.

Una vez que los parámetros se han configurado y se ha creado la base de datos solo es necesario crear las tablas y cargar los datos. Una de las características más importantes de DB2 con aceleración BLU es que no se requiere la creación o el uso de:

- Índices
- Particionamiento de tablas
- MQT (Material Query Tables) o Vistas materializadas
- MDC (Multi Dimensional Cluster)
- Hints
- Ejecución estadísticas
- Vistas

Para la migración de la información a tablas organizadas por columna se recomienda utilizar el comando LOAD.

Compresión extrema

Adicional a las características de compresión de versiones anteriores, como por ejemplo DB2 10.1 Adaptive Compression, DB2 10.5 ha implementado nuevos algoritmos de compresión con la cual se incrementa la tasa de compresión y, por consiguiente, el performance. El nuevo algoritmo de compresión se llama codificación Huffman. Consiste en una cadena de datos con una serie de 0 a 1, en el cual cada carácter es asignado a 8 bits y a su vez podemos separar un carácter de otro por la ruptura de 8 trozos de bits. Sin embargo, la codificación Huffman es un número variable de bits de cada carácter dependiendo de la frecuencia.

DB2 utiliza secuencias de caracteres de acuerdo con la probabilidad de ocurrencia para utilizar menos bits y así lograr una compresión extrema.

La siguiente figura se muestra para efectos ilustrativos.

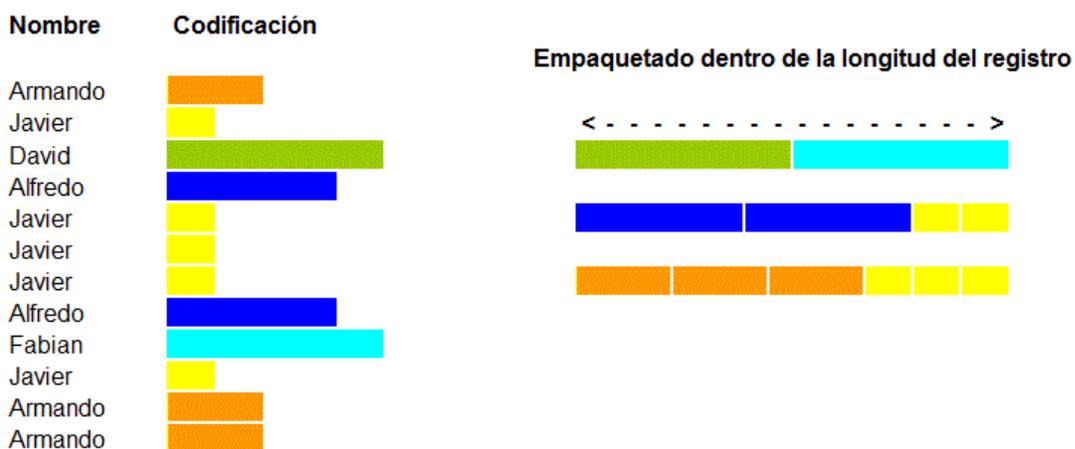


Figura 21 Ejemplo de empaquetado

La codificación es siempre un valor de la columna entera y ayudará a tener menos I/O, mejor uso de la memoria, un uso más eficiente del CPU y por supuesto un mejor rendimiento. Otra característica es que permite el uso de predicados y Joins sobre los datos codificados, es decir sin tener que descomprimir los datos.

Para hacer la compresión de la información en tablas organizadas por columnas se crea un diccionario por columna. A continuación se muestra una comparativa con tablas organizadas por renglón y columna.

Tabla 2. Compresión del Diccionario

	Renglón	Columna
Compresión a nivel tabla	ALTER TABLE ... COMPRESS YES	Siempre esta la compresión en ON
Compresión a Nivel Página	ALTER TABLE ... COMPRESS YES ADAPTIVE	Siempre esta la compresión en ON
Número de Diccionarios	Uno	N = Número de columnas
Estático	Una vez construido – nunca actualizado	Una vez construido – nunca actualizado
Adaptado	A nivel página si es requerido	Nivel página por cada columna si es requerido
Quitar compresión	ALTER TABLE ... COMPRESS NO	No puede deshabilitarse
Reorganización	Utilizar KEEPDICTIONARY o RESETDICTIONARY	No cambia el diccionario a nivel columna
LOAD REPLACE	KEEPDICTIONARY es el default si el diccionario existe	RESETDICTIONARY es el default si el diccionario existe
LOAD INSERT	KEEPDICTIONARY es el default	KEEPDICTIONARY es el default

Figura 22 Compresión del Diccionario

Almacenamiento de la información por columnas

En conjunto con la codificación y el almacenamiento por columnas nos permite obtener mayor compresión pero también minimizar la Entrada/Salida al disco para realizar consultas a grandes volúmenes de información. Se mejora la Entrada/Salida porque solo se leen las columnas que desea consultar. DB2 trabaja directamente sobre los datos de las columnas y a medida que la consulta avanza menos registros se clasificarán, por lo tanto, el trabajo para acceder al conjunto de páginas para la siguiente columna se reducirá.

Los registros y las columnas no están juntos sino hasta que el resultado de la consulta se devuelve al usuario final o cuando se realiza un Join con una tabla organizado por renglón.

Ventajas

- Minimiza la E/S porque solo ejecuta las operaciones de E/S en columnas y valores que utiliza la consulta.
- Extrema la compresión porque empaqueta más valores de datos dentro de pequeños montos de memoria o disco.
- El uso de Predicados, Join y Scan trabaja directamente sobre las columnas y sobre datos codificados lo que incrementa el performance. Los datos son descomprimidos hasta el momento en que son devueltos al usuario final.
- Mejora la densidad de la memoria porque los datos almacenados por columnas se mantienen comprimidos en la memoria.
- Eficiencia el cache debido a que no hay necesidad de utilizar más memoria y ancho de banda para columnas que no son necesarias.
- Generalmente, ocupan menos espacio que las tablas organizadas por renglón. Sin embargo, si una tabla organizada por renglón tiene pocos registros ocupara más espacio.

- Cada tabla organizada por columna tiene una tabla auxiliar llamada Synopsis. Esta tabla ocupa el 0.1% del tamaño actual.

Tabla 3. Diferencias entre Almacenamiento por Columna y Renglón

Renglón	Columna
Comando para Crear la base de datos	Mismo
Sentencia para crear el <u>Bufferpool</u>	Mismo
Sentencia para crear el Tablespace	Mismo
Comando para Crear una Tabla	Especificar ORGANIZE BY COLUMN en la cláusula CREATE TABLE.
Comando para cargar datos con LOAD	Mismo
Sentencia para crear índices	No es requerido
Definición de Constraints	Mismo
Definir o Refrescar MQT's	No es requerido porque el performance ya está optimizado
Comando para ejecutar las Estadísticas	No es requerido porque la ejecución de las estadísticas en tablas son ejecutadas como parte del proceso de LOAD.

Figura 23 Diferencia entre almacenamiento Columna y Fila

Tabla 4. Hardware y Software Soportados

Hardware
Intel Nehalem (Core i7 para Desktop y Xeon para Server) o superior
POWER7 – <u>Superscalar Symmetric Multiprocessor</u> o superior
Software
Linux 64-bit on Intel/AMD (RHEL 6 o superior, SLES 10 SP2, SLES 11 SP2)
AIX en hardware POWER7 (AIX 6.1 TL7 SP6, AIX 7.1 TL1 SP6)

Figura 24 Hardware y software soportado

Tabla 5. Recomendaciones para Cargas de trabajo

	Pequeño	Mediano	Grande
Tamaño de datos	1 TB	5 TB	10 TB
Mínimo Recomendado			
Cores	8	16	32
Memoria	64 GB	256 GB	512 GB
Recomendado			
Cores	16	32	64
Memoria	128-256 GB	384-512 GB	1024-2048 GB

Figura 25 Recomendaciones para carga de trabajo

Tabla 6. Cargas de trabajo Beneficiados

Beneficiados	No beneficiados
Analíticos y Data Marts	Consultas con pocos Insert, Update y Delete
Estrella o Esquemas Dimensionales	Bases de datos totalmente normalizadas para OLTP
SAP Business Warehouse	Insertar, Actualizar y/o Eliminar pocos registros por transacción
Agregaciones, Agrupaciones, Búsquedas por rango, Joins	Consultas accediendo a muchas o todas las columnas de una tabla
Consultas que acceden a pocas columnas en una tabla	El uso intensivo de LOB, XML, datos estructurados, temporales y columnas generadoras
Consultas que obtienen más de 1% de los datos	Alta disponibilidad (HADR), Seguridad basado en etiquetas (LBAC) y Seguridad por renglón y columna (RCAC)
Bases de datos de 1, 5 y 10 TB	Si el sistema operativo no es AIX o Linux
Cantidad moderada de datos <20TB	

Figura 26 Carga de Trabajo Beneficiados

Paralelismo Multiprocesamiento

Otra de las características de DB2 con Aceleración BLU es que al ejecutar las consultas automáticamente son paralelizadas con los procesadores del equipo logrando así una excelente escalabilidad de multiprocesamiento. Los datos son cuidadosamente colocados y alineados poniendo atención en los atributos físicos del servidor sin la necesidad de particionar los datos en diferentes servidores.

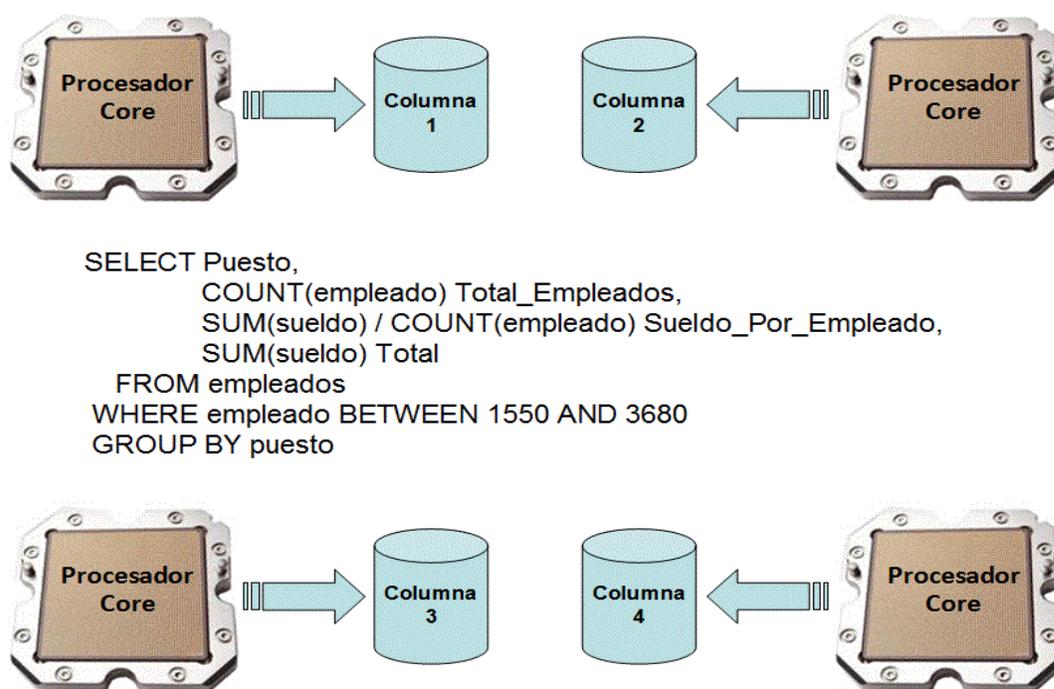


Figura 27 Ejemplo de Procesamiento

Un agente independiente por cada núcleo o procesador puede recuperar datos de columnas individuales. Cada agente trabaja en diferentes funciones de la consulta. Antes de llegar al operador CTQ los datos son procesados por núcleo y por último en un plan de ejecución el operador REBAL indica que múltiples procesadores se están utilizando.

III.2.4. Base de Datos 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.

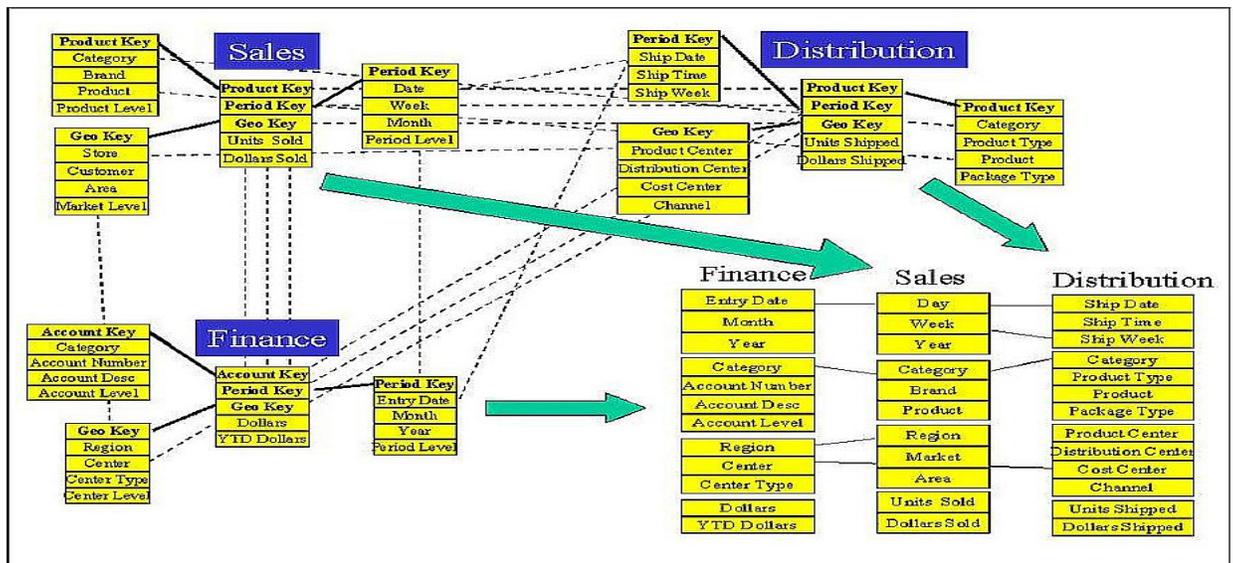


Figura 28: Ejemplo de almacenamiento de Sybase

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.

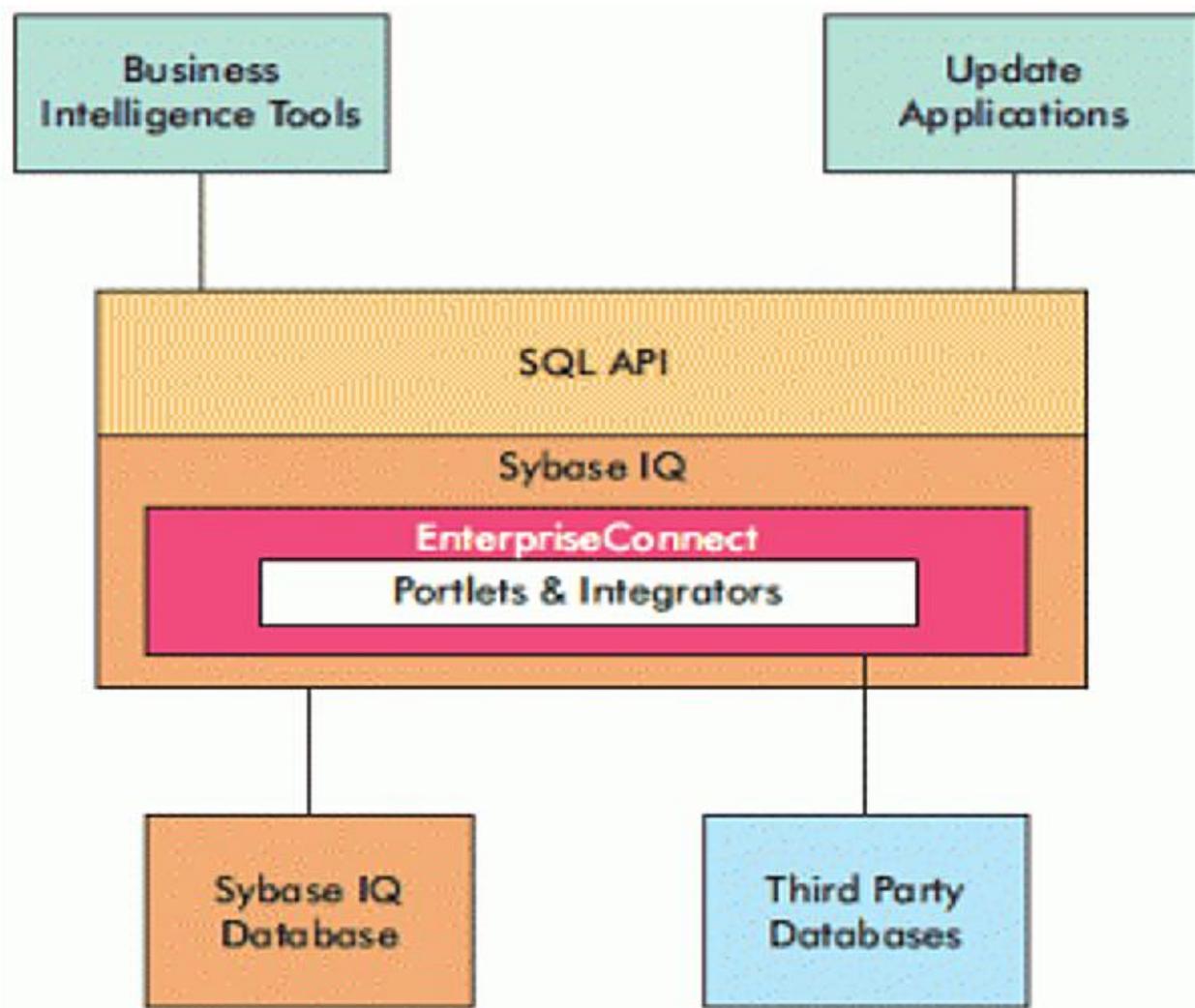


Figura 29: Muestra como Sybase incluye una API SQL

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.

III.3. Principales Bases de Datos Columnares Open Source

III.3.1. Base de Datos Cassandra



Apache Cassandra es un motor de bases de datos NoSQL, Open Source e implementado en Java. Fue originalmente creada por Facebook y donada a Apache como software libre en 2009. Es una de las base de datos NoSQL más relevantes a nivel mundial: Netflix, eBay, Twitter, Urban Airship, Constant Contact, Reddit, Cisco, OpenX, Digg, CloudKick, Hóyala, entre otras. Cassandra puede manejar varios terabytes de datos si lo necesita y puede, fácilmente, manejar millones de ficheros, incluso en un clúster pequeño (Big Data). La información en las bases de datos relacionales, se almacenan en forma de filas, pero en Cassandra la información se almacena en columnas con pares key-value.

Esquema dinámico. El esquema que define la estructura de los datos puede cambiar en tiempo de ejecución. No hay un único punto de fallo. Los datos se replican automáticamente a varios nodos. Perder un nodo no causa la baja del clúster. Alta disponibilidad. Los datos están disponibles la mayor parte del tiempo gracias a la redundancia que introduce la replicación de datos. Particionamiento de los datos. La topología de Cassandra es la de un anillo a través del cual se distribuyen los datos para minimizar cuellos de botella en el acceso a los mismos. Escalabilidad horizontal. Hasta un alto número de máquinas la capacidad de cómputo aumenta linealmente con el número de máquinas. Capacidad para manejar cientos de Terabytes de datos.

Terminología Utilizada

- **Column**

Es la unidad mas básica en el modelo de datos de Cassandra. Una column es un triplete de un key (un nombre) un value (un valor) y un timestamp. Los valores son todos suministrados por el cliente. El tipo de dato del key y el value son matrices de bytes de Java, el tipo de dato del timestamp es un long primitive.

Las column son inmutables para evitar problemas de multithreading.

Las column se organizan dentro de las column families.

Las column se ordenan por un tipo, que pueden ser uno de los siguientes:

AsciiType

BytesType

LongType

TimeUUIDType

UTF8Type

- **SuperColumn**

Es una column cuyos values son una o más columns, que en este contexto se llaman subcolumns. Las subcolumns están ordenadas, y el numero de columns que se puede definir es ilimitada. Las Super columns, a diferencias de las columns, no tienen un timestamp definido.

- **Column Family**

Es mas o menos análogo a una tabla en un modelo relacional. Se trata de un contenedor para una colección ordenada de columns. Cada column family se almacena en un archivo separado

- **Keyspace**

Es el contenedor para las column family. Es mas o menos análogo a una base de datos en un modelo relacional, usado en Cassandra para separar aplicaciones. Un keyspace es una colección ordenada de columns family.

- **Clúster**

Conjunto de máquinas que dan soporte a Cassandra y son vistas por los clientes como una única máquina.

Estructura

Row key1	Column Key1	Column Key2	Column Key3	...
	Column Value1	Column Value2	Column Value3	
⋮				

Row key1	Super Column key1			Super Column key2			...
	Subcolumn Key1	Subcolumn Key2	...	Subcolumn Key3	Subcolumn Key4	...	
	Column Value1	Column Value2	...	Column Value3	Column Value4	...	
⋮							

Relational Model	Cassandra Model
Database	Keyspace
Table	Column Family (CF)
Primary key	Row key
Column name	Column name/key
Column value	Column value

Figura 30 Modelo de Columna

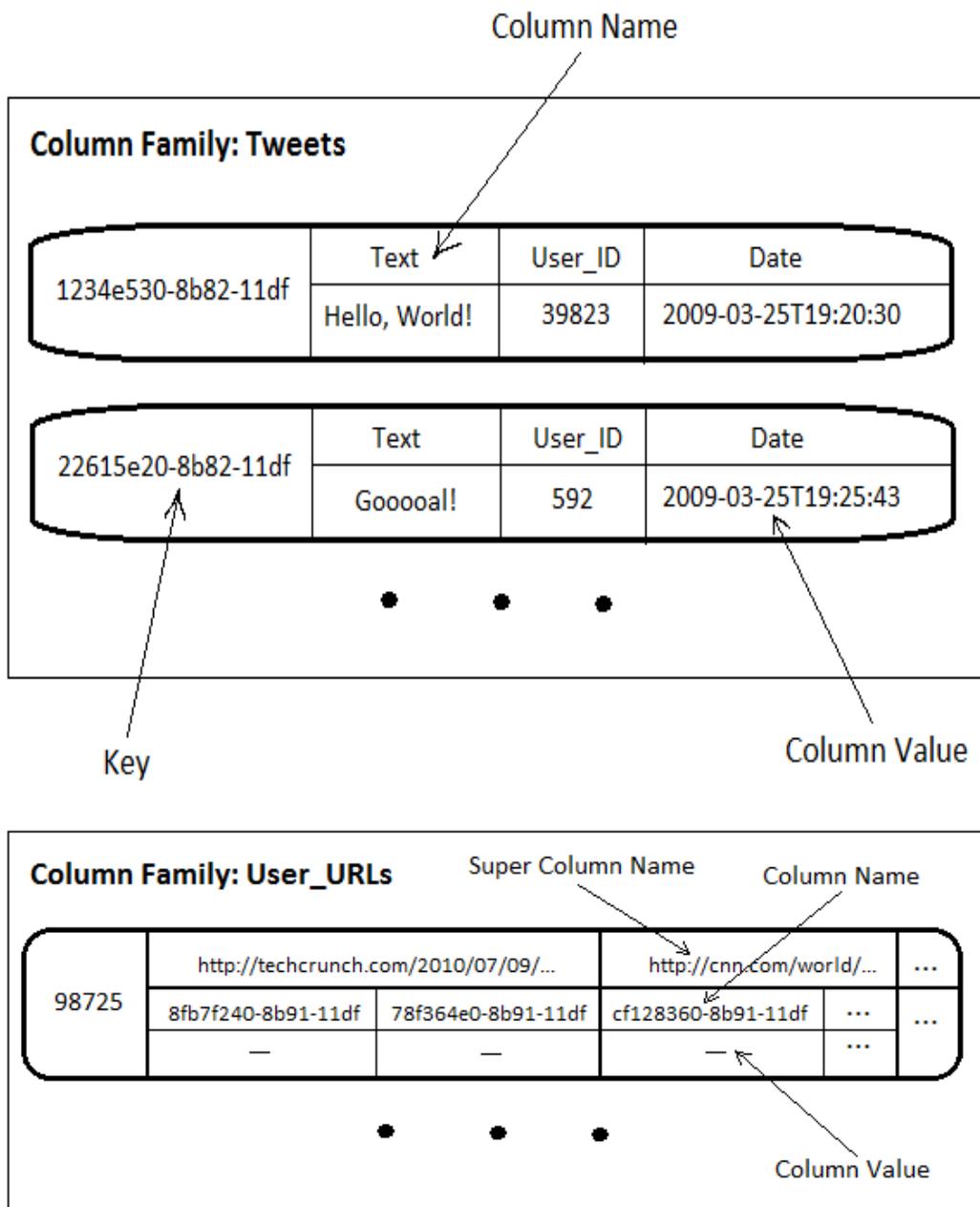


Figura 31 Modelo de Columna

■ Entidad

```

▷ People: {
  John Dow: {
    twitter: jdow,
    email: jdow@example.com,
    bio: bla bla bla,
    ...
  },
}
    
```

People				
John Dow	Twitter	Email	bio	...
	jdow	jdow@	Bla, bla	...

Figura 32 Modelo Relaciones

■ One to one

```

▷ People: {
  John Dow: {
    twitter: jdow,
    email: jdow@example.com,
    bio: bla bla bla,
    marriedTo: Mary Kay,
    ...
  },
  Mary Kay: {
    marriedTo: John Dow,
    ...
  },
}
    
```

■ One to many

```

▷ Children: {
  John Dow: {
    01/18/1976: John Dow Jr,
    05/27/1982: Kate Dow
  },
  ...
}

▷ People: {
  John Dow Jr: {
    father: John Dow,
    ...
  },
  ...
}
    
```

■ Many to many

```

▷ Friendship: {
  John Dow: {
    10: Mark Seldon,
    8: Julian Hendrix,
    ...
  },
  Mark Seldon: {
    9: John Dow,
    ...
  },
}
    
```

Figura 33 Modelo Relaciones

III.3.2. Base de Datos 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.

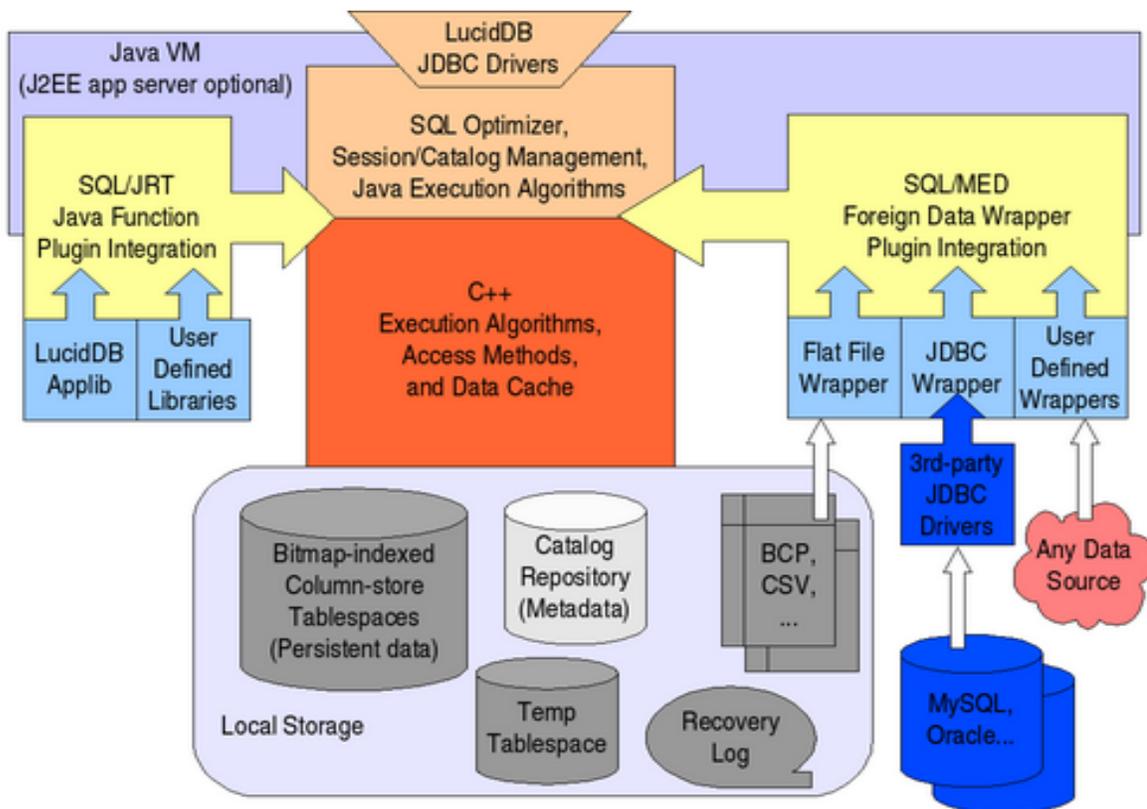


Figura 34 Funcionamiento de LucidDB

III.3.3. Base de Datos INFOBRIGHT



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.

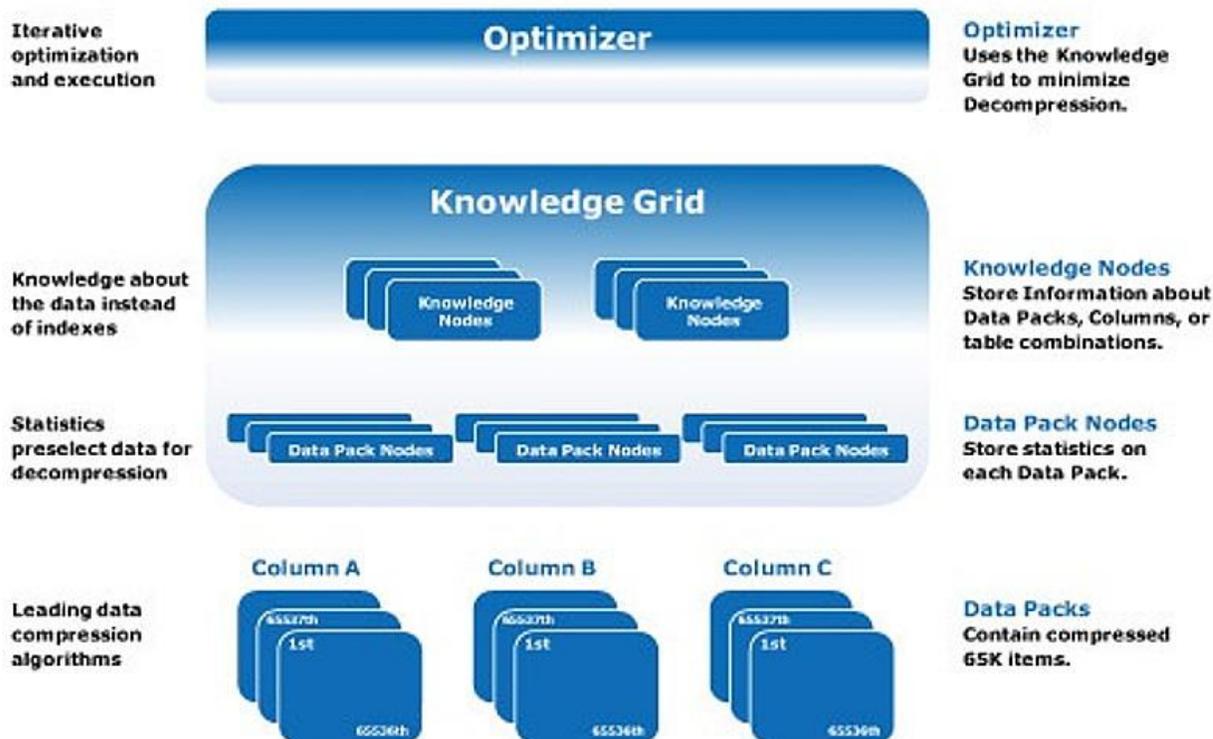


Figura 35: Funcionamiento Infobright.

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.

III.3.4. Base de Datos 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.

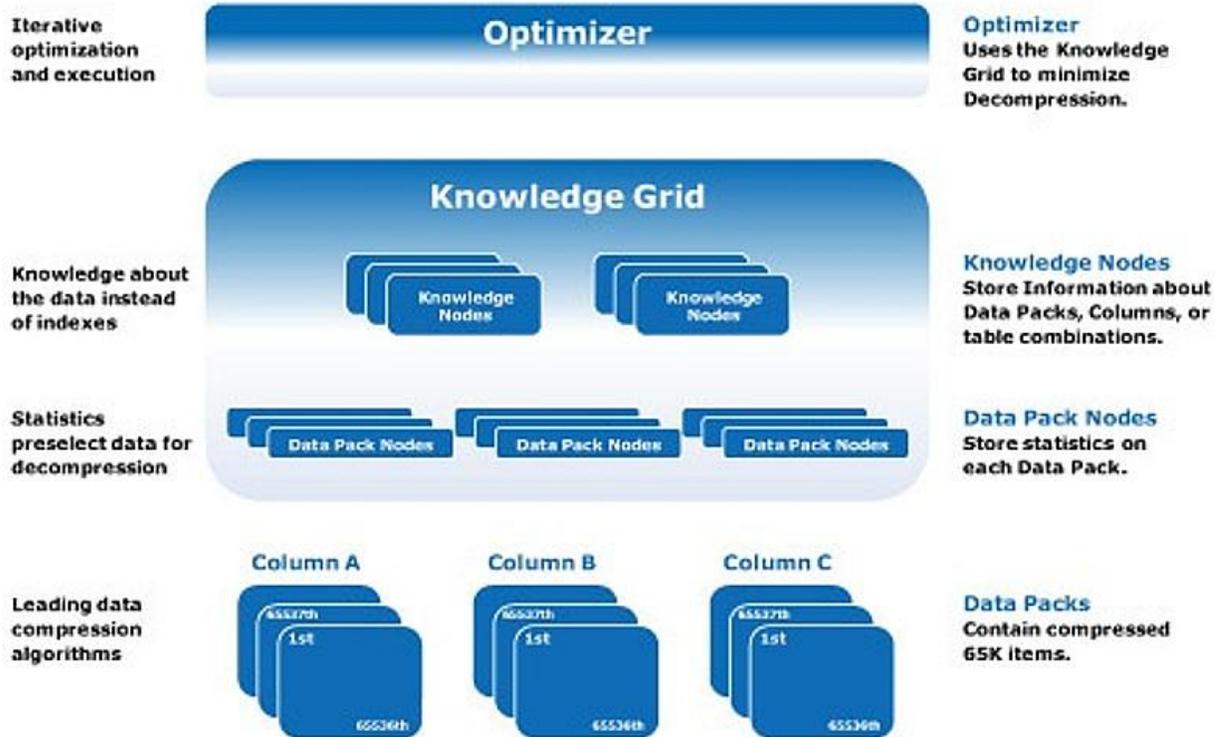
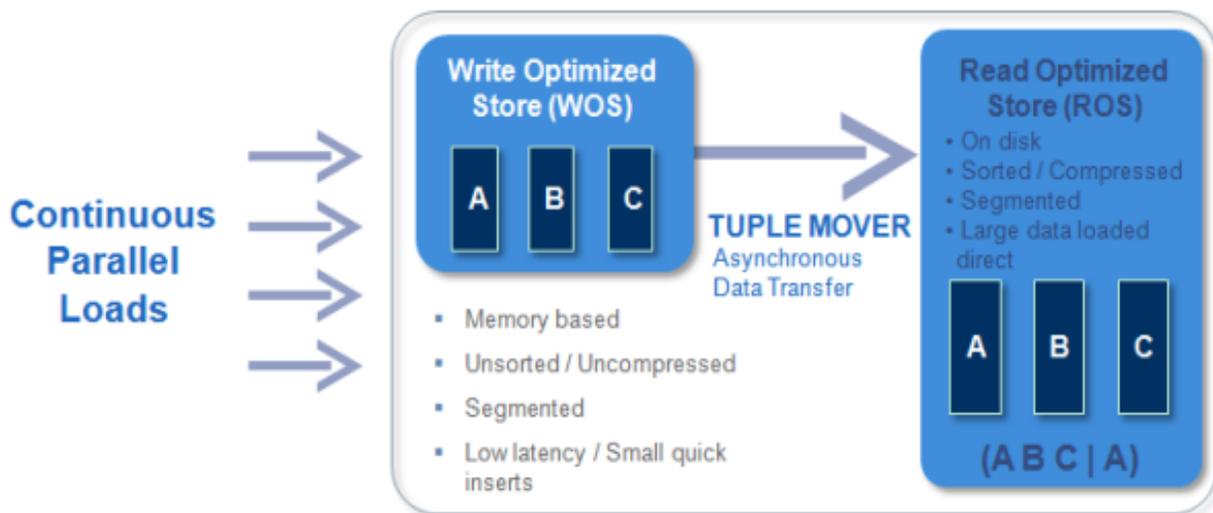


Figura 36: Cargas continuas paralelas de Vertica.



Figura 37: 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).

III.3.5. Base de Datos 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 otras 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.

III.3.6. Base de Datos 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.

III.4. Visión según Gartner

Cada año, Gartner analiza a cada proveedor del mercado de inteligencia de negocios y análisis. Su investigación tiene una importancia particular porque a menudo identifica las innovaciones que dirigen el mercado. Como parte de ese informe, Gartner también lanza el Magic Quadrant, que muestra las posiciones relativas de los competidores en el mercado.

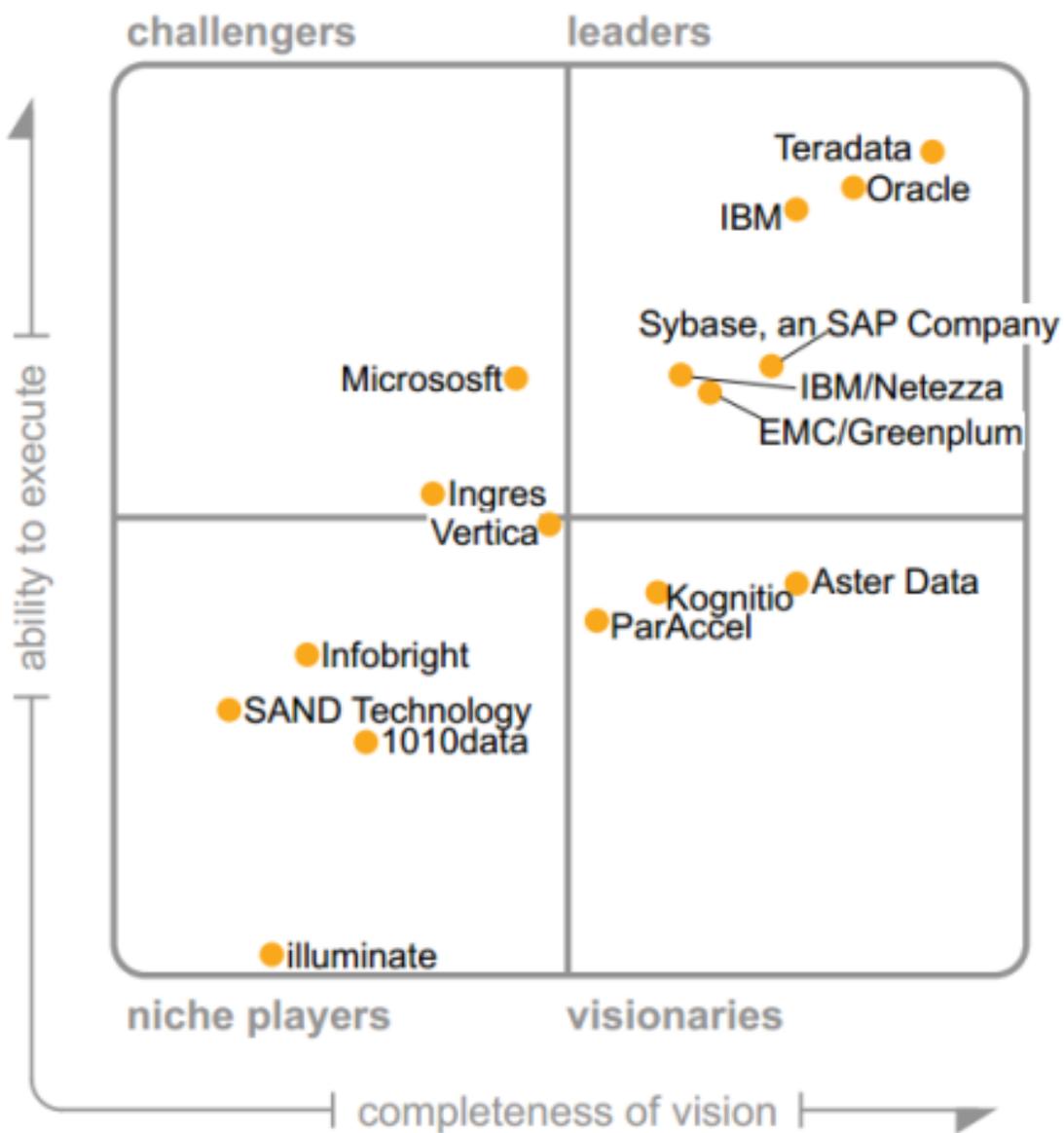
Para entender mejor estos cuadros debemos explicar que significa cada uno de los cuadrantes:

Leaders . Son proveedores que pueden ofrecer un plataforma de BI a toda la organización. Tienen una propuesta comercial de alcance global.

Challengers . Empresas que ofrecen una plataforma BI con buenas funcionalidades, que pueden limitarse a usos específicos; pero fallan en una estrategia coordinada entre todos sus productos, tienen una carencia en canales de ventas o tienen una presencia geográfica limitada.

Visionaries . Innovadores que se caracterizan por la apertura y flexibilidad de las arquitecturas de sus aplicaciones, que aún no han alcanzado un nivel adecuado o existen dudas sobre su capacidad.

Niche players . Empresas que abordan bien un segmento específico de una plataforma de BI, el dominio específico hace que pierdan funcionalidad en otros aspectos



Source: Gartner (January 2011)

As of January 2011

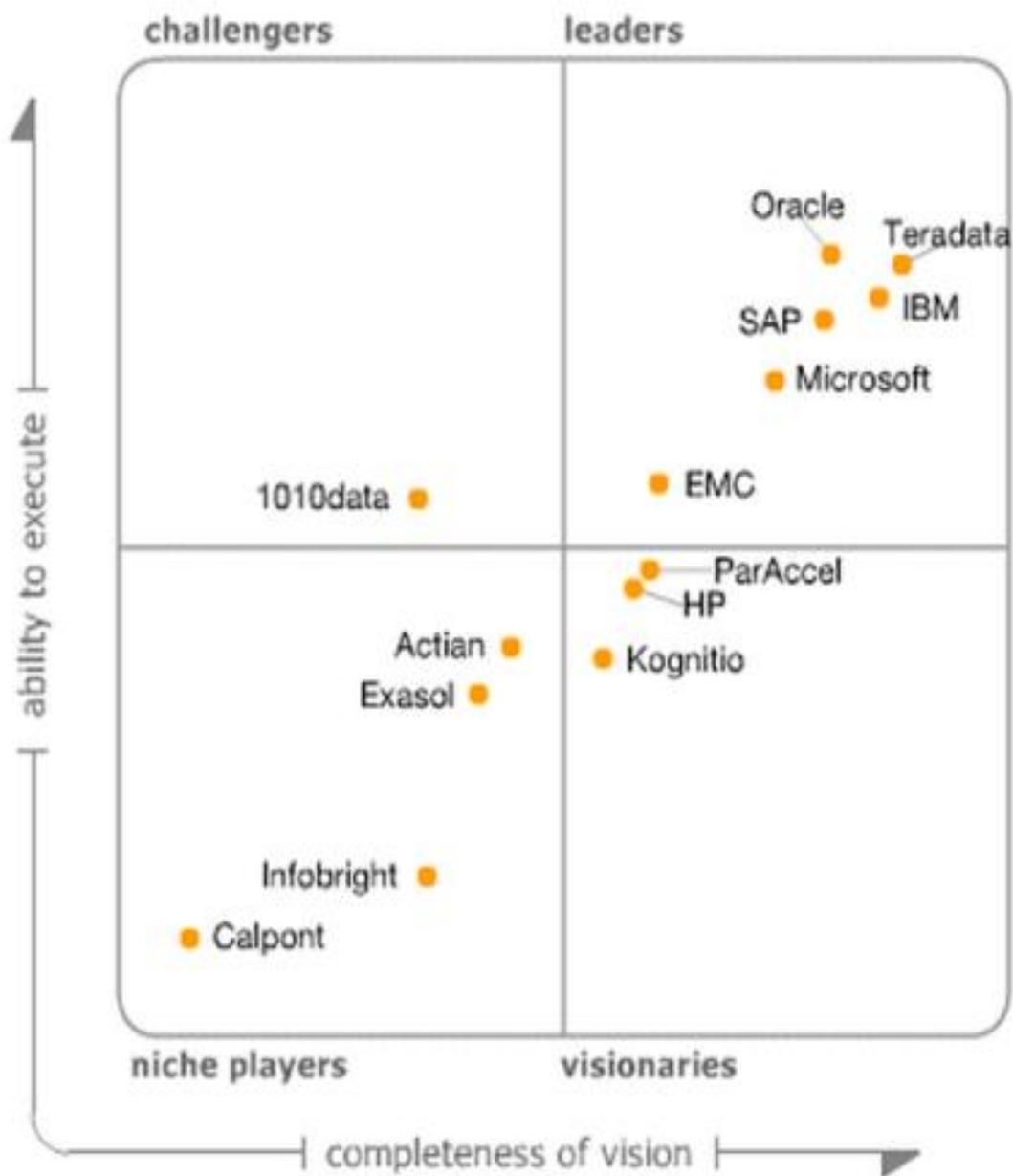
Figura 38: Cuadro Mágico Gartner año 2011



Source: Gartner (February 2012)

As of February 2012

Figura 39: Cuadro Mágico Gartner año 2012



Source: Gartner (January 2013)

As of January 2013

Figura 40: Cuadro Mágico Gartner año 2013

Gartner en los 3 últimos años ha colocado en el cuadrante de los líderes a las empresas Teradata, Oracle e IBM, como las principales empresas innovadoras en el manejo de bases de datos siendo Teradata quien llevaba el liderazgo por mucho tiempo en este tema hasta el 2012, siendo sobrepasado en el 2013, por su principal competidor Oracle.

III.5. Pruebas realizadas

III.5.1 Entorno de la Prueba

Para la realización de los test de Rendimiento se utilizó 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.

III.5.2 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)

III.5.3 Instalación de LucidDB

- 1.- Como prerequisite 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.

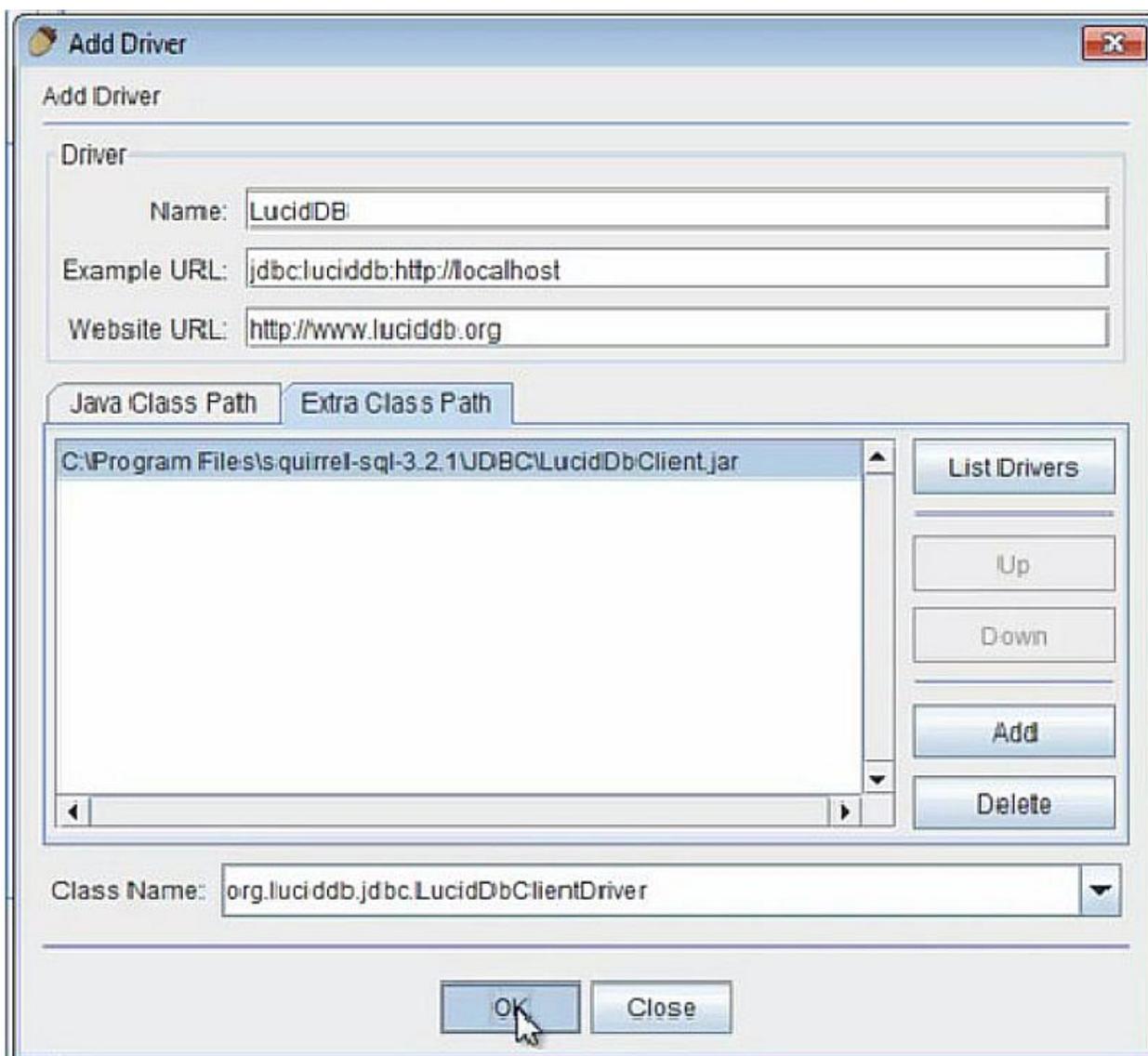


Figura 41: Selección de driver de LucidDB.

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

Add Alias

Name: Local LucidDB Server

Driver: LucidDB

URL: jdbc:luiddb:http://localhost

User Name: sa

Password:

Auto logon Connect at Startup

Warning - Passwords are saved in clear text

Figura 42: Creación de un alias.

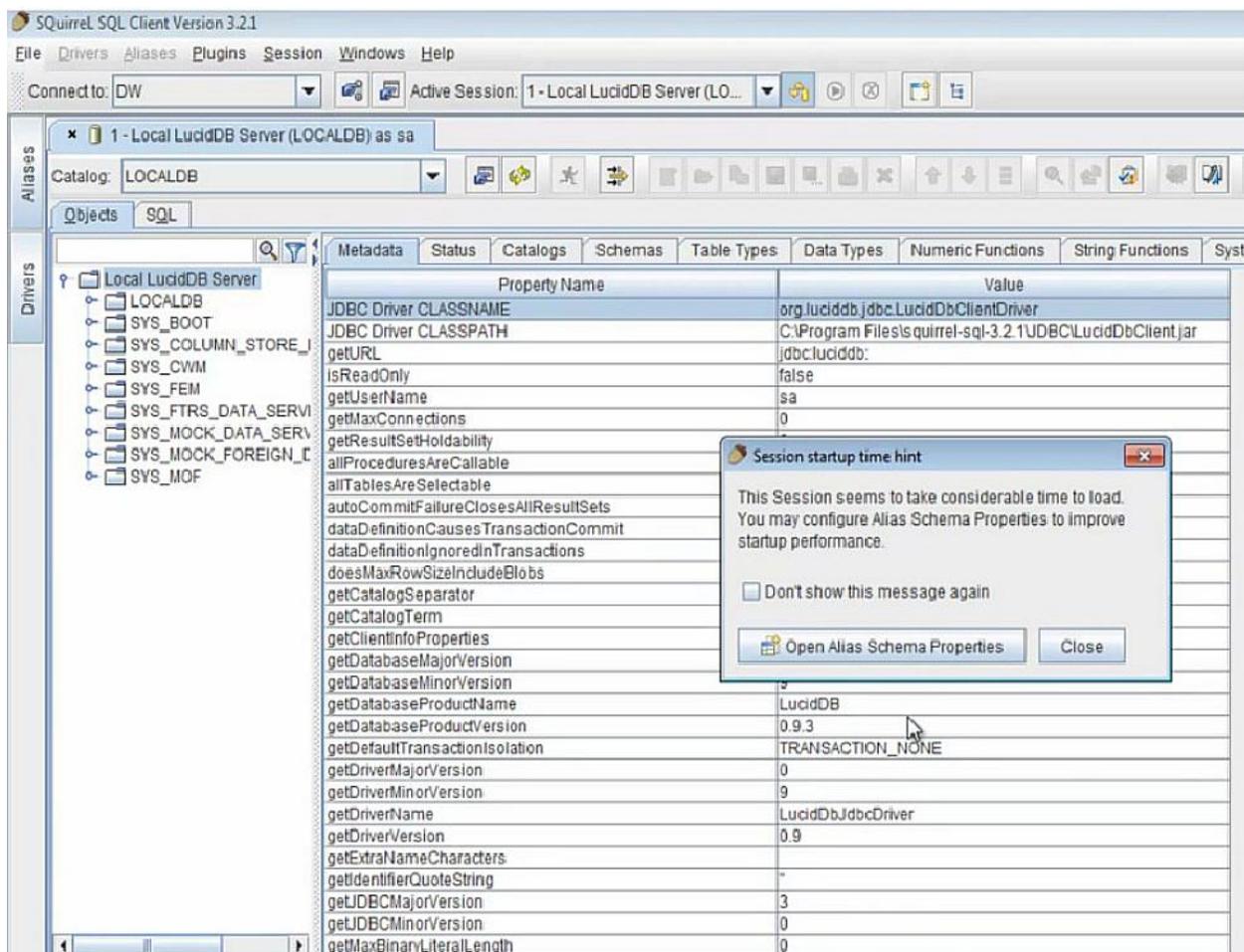


Figura 43: Catalogo y esquemas de la base de datos.

III.5.4 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 permite pese a ser open source la realización de cargas incrementales y actualizaciones.



Figura 44: Ejemplo de carga de datos.

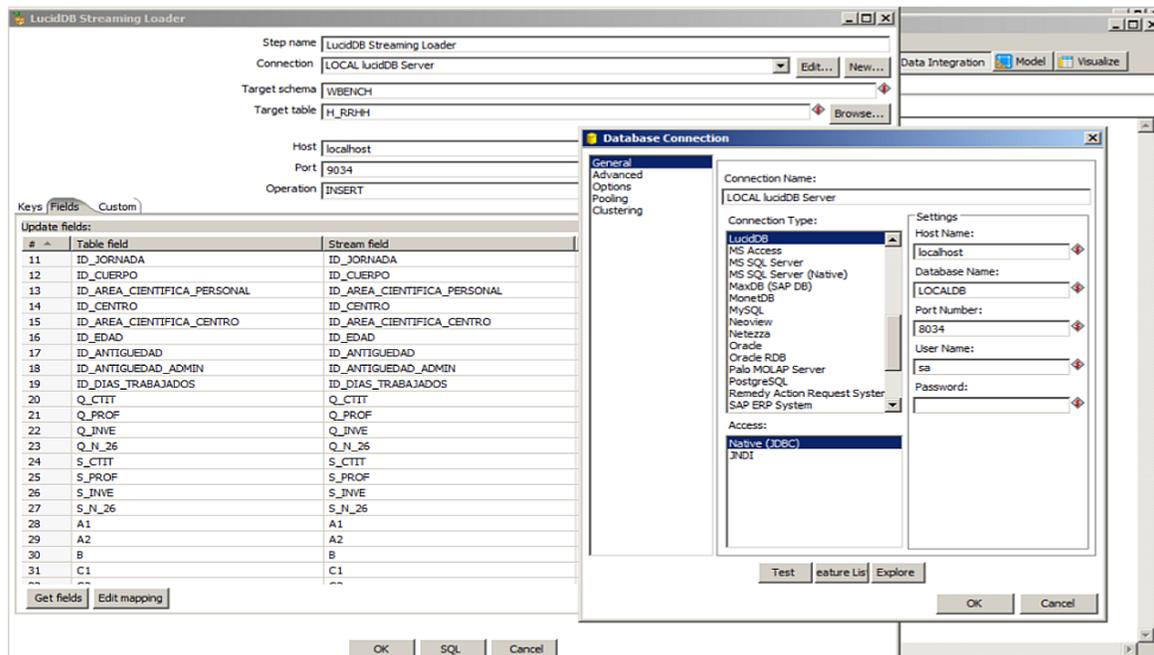


Figura 45: Ejemplo de carga de datos.

III.5.5 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.

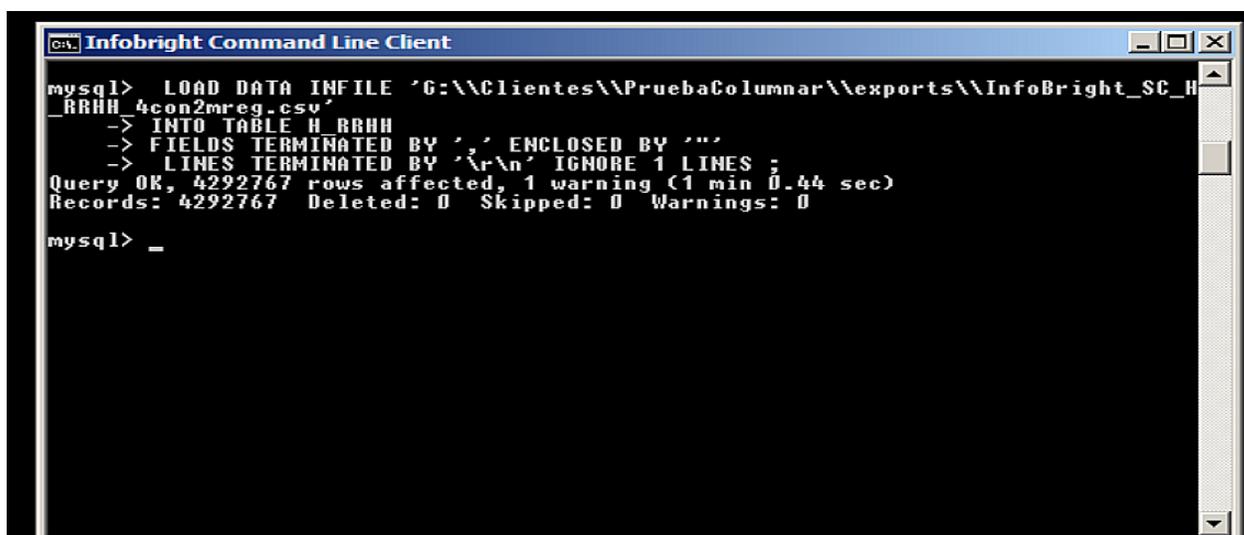
5.- La creación de una base de datos y una tabla es idéntica a MySQL, la principal diferencia es que el motor de InfoBright utiliza el denominado BrightHouse (mysql> create table <nombre_tabla> (<columna(s)>) engine=brighthouse;)

6.- 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.

7.- 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.

III.5.6 Carga de datos en InfoBright Community Edition

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.



```
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 46: Carga de datos en Infobright.

III.5.7 Consultas realizadas a la BD 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 ar.DESC_CATEGORIA, ga.DESC_CORTA,
       cg.DESC_CLASIFICACION ,
       cg.DESC_CATEGORIA_GRUPO , cg.CAT_2 ,
       ac.ID_AREA_CIENTIFICA_ODS ,
       COUNT(DISTINCT hr.ID_PERSONA) as m0
FROM DIM_AREAFUNCIONAL AS af , H_RRHH AS hr,
     DIM_GRADO_ACADEMICO AS ga,
     DIM_CATEGORIA_GRUPO AS cg,
     DIM_AREA_CIENTIFICA AS ac
WHERE hr.ID_AREAFUNC = af.ID_AREA_FUNCIONAL
AND   af.DESC_CATEGORIA = 'Ingenieros'
AND   hr.ID_GRADO_ACADEMICO = gr.ID_GRADO_ACADEMICO
AND   hr.ID_CUERPO = cg.ID_CATEGORIA_GRUPO
AND   cg.DESC_CLASIFICACION = 'Grupo'
AND   cg.DESC_CATEGORIA_GRUPO in ('A', 'B', 'C', 'D')
AND   hr.ID_CUERPO = cg.ID_CATEGORIA_GRUPO
AND   hr.ID_AREA_CIENTIFICA_PERSONAL =
       ac.ID_AREA_CIENTIFICA
GROUP BY af.DESC_CATEGORIA, ga.DESC_CORTA,
         cg .DESC_CLASIFICACION,
         cg.DESC_CATEGORIA_GRUPO , cg.CAT_2,
         af.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;
```

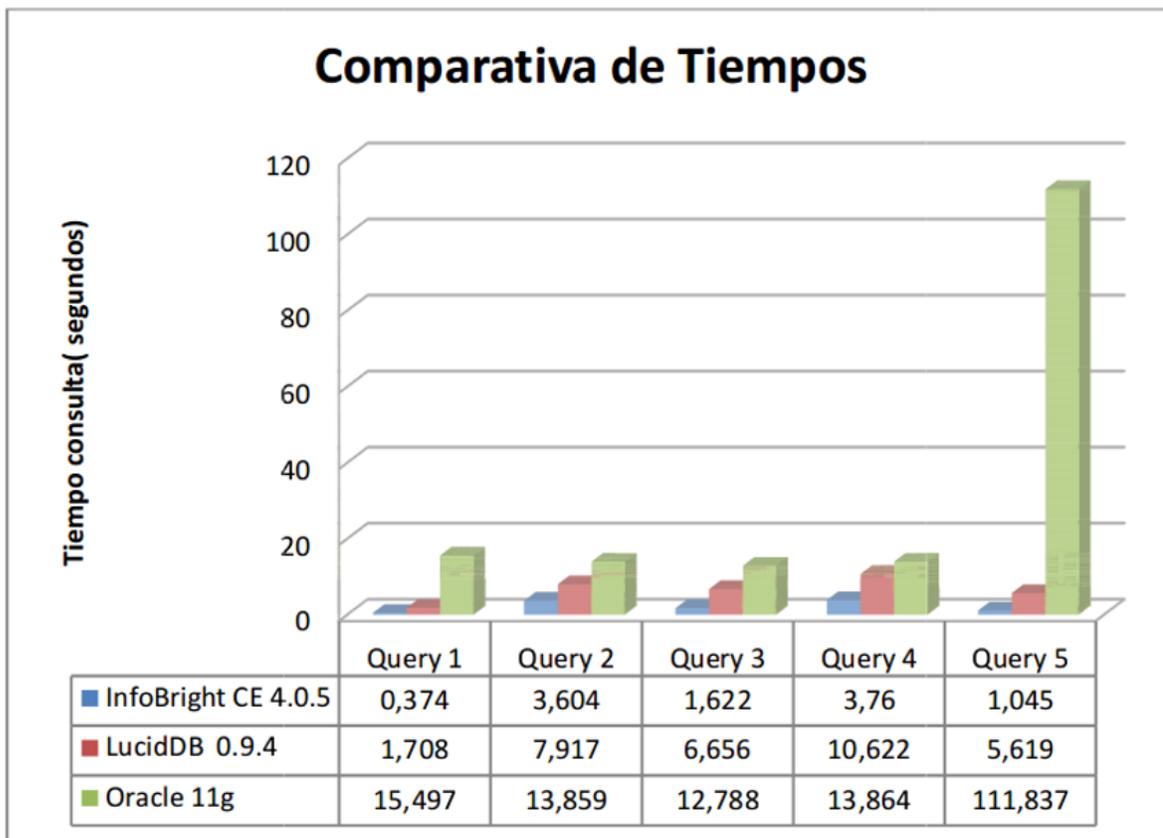


Figura 47: Gráfica comparativa de tiempos.

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.

IV CONCLUSION

A medida que la que el volumen de información siga creciendo dentro de las organizaciones, se verá afectado el rendimiento debido a la actual estructura de almacenamiento utilizada por las bases de datos orientadas a las filas.

En base al estudio realizado se puede concluir que las bases de datos orientados a la columna irán tomando mayor protagonismo dentro del mercado. El crecimiento explosivo de las bases de datos superando los Terabytes de información hace necesario la incorporación de nuevos motores de búsqueda que permitan acceder a esta en forma rápida no importando lo complejo de las consultas.

Exponentes como Oracle, IBM, Teradata y otros ya están en la senda de incorporar estas tecnologías en sus motores comerciales como nuevas prestaciones o simplemente como nuevos productos que se integran a los ya existentes.

El modelo de bases de datos orientado a columnas muestra su potencialidad en contextos como el modelo de gestión de toda la organización, basado en la orientación al cliente (CRM), sistemas electrónicos de biblioteca de catálogos, bases de datos documentales y otros sistemas ad-hoc de investigación y típicamente es usado en data warehouses y sistemas de Business Intelligence (OLAP, etc.), donde además resultan ideales para calcular datos agregados.

Como comentario final se indicaran algunas ventajas y desventajas que tienen por el momento trabajar con las bases de datos orientados a la columna

Ventajas

- **Es de código abierto**

Los productos de código abierto proporcionan a los desarrolladores grandes beneficios, sobre todo por su estado sin costo alguno. Otros beneficios: el software de código abierto tiende a ser más confiable, seguro y rápido de implementar que las alternativas propietarias.

Gestores NoSQL populares son Cassandra, CouchDB, Hbase, MongoDB y Redis.

- **Escalamiento sencillo.**

NoSQL sustituye a la antiguo "escalar" el mantra de los gestores de las bases de datos con una nueva: "manera" en lugar de añadir más servidores para manejar más carga de datos, una base de datos NoSQL permite a una empresa distribuir la carga entre varios hosts a medida que aumenta la carga.

- **Diferentes DBs NoSQL para diferentes proyectos**

MongoDB y Redis son buenas opciones para el almacenamiento de escritura con alta frecuencia, rara vez leen los datos estadísticos, tales como web, contador de visitas.

Cassandra y Riak (clusters automáticos, tiendas redundantes) un buen rendimiento en entornos con aplicaciones de alta disponibilidad, donde el tiempo de funcionamiento máximo es de vital importancia.

- **NoSQL y la nube es un ajuste natural.**

Los servidores de hoy en día son de bajo costo y fácilmente pueden ser ampliados a petición mediante un servicio como Amazon EC2. Al igual que toda la tecnología de la nube, EC2 se basa en la virtualización. El eslabón débil de la virtualización es la E/S, la memoria y CPU que deben ser ágiles.

- **Las bases de datos NoSQL utilizan sobre todo el uso de memoria en vez del disco como la principal ubicación de escritura**

Lo que impide inconsistente rendimiento I/O. Y como los almacenes de datos NoSQL aprovechan típicamente particiones horizontales, son capaces de tomar ventaja en la nube de la elástica del aprovisionamiento.

- **No generan cuellos de botella**

Desventajas

- **No están lo suficientemente maduros para algunas empresas**

A pesar de sus puestas en práctica en algunas grandes empresas, las bases de datos NoSQL aún se enfrentan a un problema de credibilidad importante con muchas empresas. Los críticos señalan la falta de madurez de NoSQL y los posibles problemas de inestabilidad, mientras que citan la madurez, y una gran funcionalidad y estabilidad de los RDBMSes.

- **La falta de experiencia**

La novedad de NoSQL significa que no hay una gran cantidad de desarrolladores y administradores que conocen la tecnología -lo que hace difícil a las empresas encontrar personas con los conocimientos técnicos apropiados. Por el contrario, el mundo RDBMS tiene miles de personas muy cualificadas.

- **Problemas de compatibilidad**

A diferencia de las bases de datos relacionales, que comparten ciertos estándares, las bases de datos NoSQL tienen pocas normas en común. Cada base de datos NoSQL tiene su propia API, las interfaces de consultas son únicas y tienen peculiaridades. Esta falta de normas significa que es imposible cambiar simplemente de un proveedor a otro, por si no quedara satisfecho con el servicio.

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)

Múltiples 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.

BD:

Base de Datos

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.

Paralelismo:

Ejecución de múltiples operaciones .al mismo tiempo con distintos procesadores

Nube

Procesamiento y almacenamiento de datos en servidores externos.

Cluster

Conjunto o conglomerados de computadoras construidos mediante la utilización de hardware comunes y que se comportan como una única computadora

Clúster Homogéneo

Misma configuración de hardware y sistema operativo

Clúster Semihomogéneo

Diferente rendimiento pero con arquitecturas y sistemas operativos similares

Clúster Heterogéneo

Diferente Hardware y sistemas operativos

NoSql (Not Only Sql)

Sistemas de almacenamiento de información que no cumplen con el esquema entidad-relación

Host

Ordenador que funciona como el punto de inicio y final de las transferencias de datos. Descrito como el lugar donde reside un sitio web. Un host de internet tiene una dirección de internet única (dirección IP y un nombre de dominio único).

BIBLIOGRAFIA

Internet

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