Datalytics

Intelligence Consulting

Ing. Diego Robledo

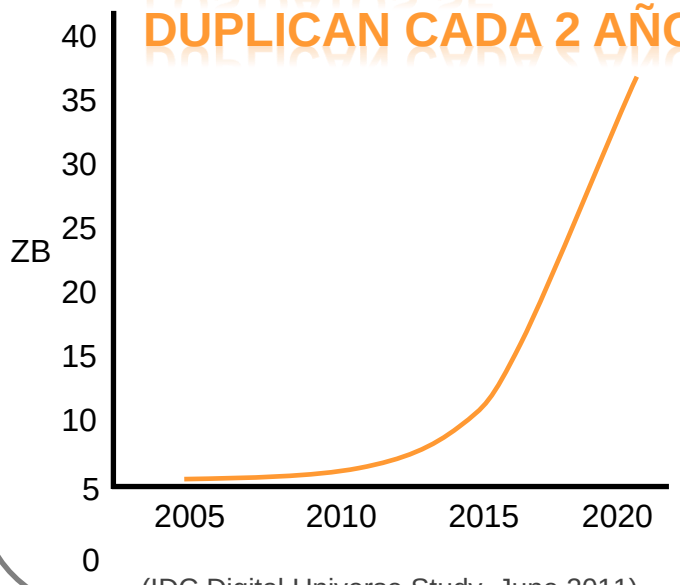
@DatalyticsBI

www.datalytics.com

Big Data & NoSQL – Ceiba DevFest 2013

Un punto de inflexión

LOS DATOS SE DUPLICAN CADA 2 AÑOS



(IDC Digital Universe Study, June 2011)

80%

DE LA INFORMACIÓN ES NO ESTRUCTURADA

(Gartner, December 2011)

LAS EMPRESAS MANEJARÁN 50X MÁS DATOS

EN LA PRÓXIMA DÉCADA

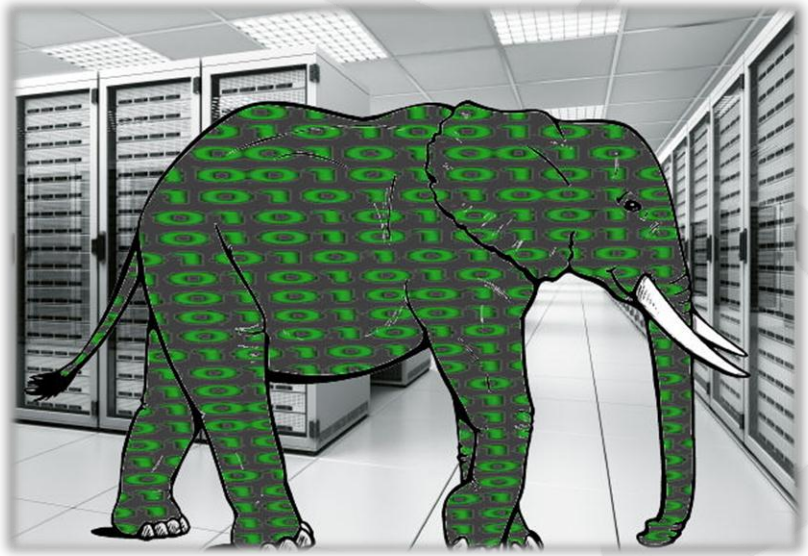
(IDC Digital Universe Study, June 2011)

"El aumento exponencial en la cantidad de información disponible ha hecho que las empresas recurran a nuevas herramientas y procesos para recopilar datos (tanto estructurados como no estructurados) y para almacenar, administrar, manipular, analizar, e integrar datos. "

J|M|P Securities, Big Data and How BI Got Its Groove Back, Nov. 2011

Qué es Big Data?

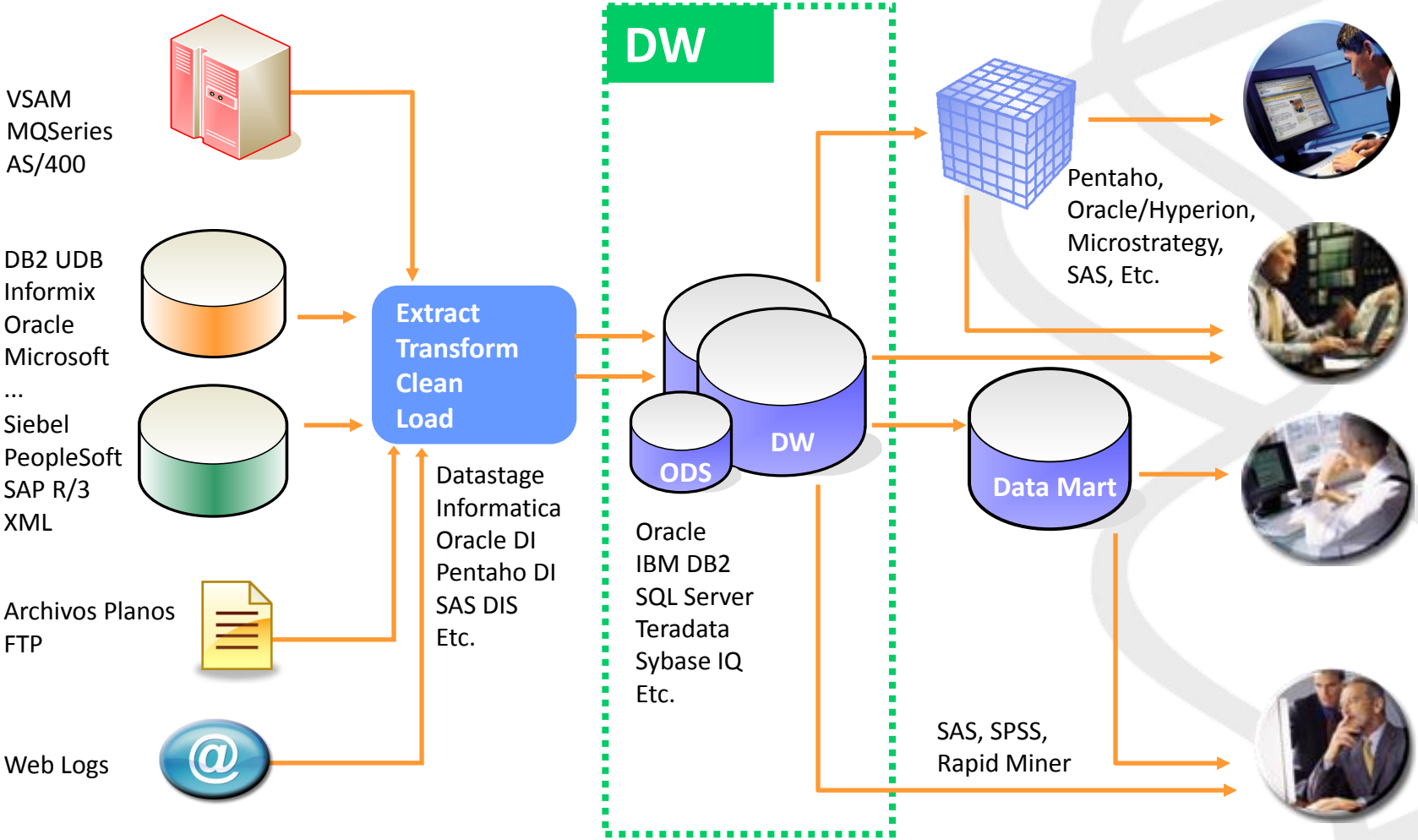
- Big Data es el nombre que se le da a conjuntos de información que crecen de una manera tan exponencial que resulta prohibitivo almacenarlos y/o procesarlos con métodos o técnicas tradicionales del mundo de base de datos relacionales.
- Big Data son tres V's y una C:
  - ***Velocidad***: los datos se generan a un ritmo exponencial.
  - ***Volumen***: la irrupción de Big Data dejó en el pasado el Terabyte para hablar de Petabytes y Zetabytes.
  - ***Variedad***: datos estructurados y no estructurados, proveniente de la web 2.0, sensores, logs, etc.
  - ***Complejidad***: volumen de datos tal que no permite procesarlo con técnicas tradicionales



De dónde viene?

- Información transaccional
  - Operaciones bancarias
  - Servicios financieros, portales de bolsa
  - Tiendas virtuales / e-commerce
- La nube
  - World Wide Web / Blogs
  - Redes sociales: Twitter, Facebook, LinkedIn
- Machine-generated data (MGD)
  - Weblogs
  - Centrales telefónicas
  - Sensores de todo tipo
  - Logs de aplicaciones

DW: el corazón de toda solución de BI



Bases de Datos de Transaccionales

- La mayoría de las empresas utilizan motores de bases datos tradicionales
  - IBM DB2, Oracle, MS SQL Server, PostgreSQL, MySQL, etc.
- Todos muy buenos motores, pero todos **transaccionales**, diseñados y desarrollados para soportar transacciones y trabajar con pocos registros por operación.
- Las soluciones de Business Intelligence en cambio involucran cientos de miles (e inclusive millones) de registros en una única operación, y deben responder en un tiempo adecuado.
- Las bases de datos transaccionales **no fueron diseñadas para responder a consultas analíticas** sobre grandes volúmenes de información.
- La irrupción de Big Data comienza a dejar en evidencia estas falencias, debiendo utilizar complejos índices, tablas agregadas, tablas particionadas, etc., aumentando el costo de desarrollo y mantenimiento, obteniendo tiempos

Bases de Datos Analíticas

- Pero existen en el mercado una variedad de bases de datos analíticas, diseñadas específicamente para ser utilizadas como motores de Data Warehouse:

- Estas bases de datos logran procesar grandes volúmenes de información a velocidades asombrosas, gracias a la aplicación de diferentes conceptos y tecnologías:
  - Almacenamiento en columnas en lugar de filas (registros)
  - Massively parallel processing (MPP)
  - In-Memory Analytics
- Históricamente estas bases de datos tan especializadas tenían un costo muy elevado, pero hoy el mercado nos ofrece varias alternativas que se adaptan al presupuesto de cada organización.

NoSQL – Qué es y qué no es

- No es un producto.
- No es una tecnología.
- Cualquier “BD” que no sea relacional.
- No significa que ya no se usa SQL
- Los requerimientos actuales se resuelven de manera relacional
- … y de manera No Relacional.

NoSQL – Qué es y qué no es

Not Only SQL

Categorías de NoSQL

Actualmente existen 4 categorías:

- Clave / Valor (Key Value DB)
- Bases de datos Gráficas – XML (Graph DB)
- Documentos (Document DB)
- Bases de datos Columnares (Columnar DB)

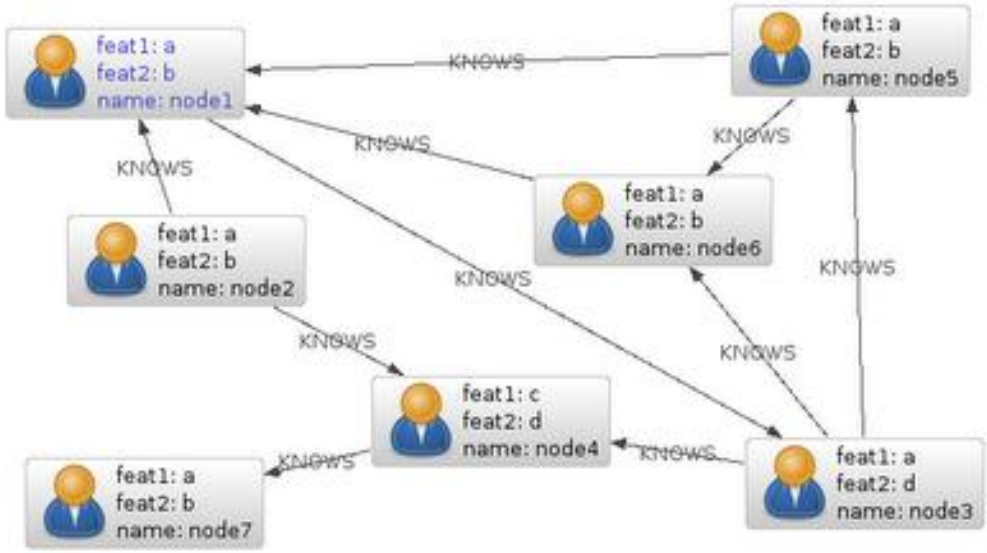
Key Value

- Se concentra en poder escalar volúmenes muy grandes de datos
- Diseñado para manejar altos volúmenes de carga
- El modelo de datos se basa en pares clave/valor
- Basado en el paper Dynamo de Amazon
- Ejemplos:
  - Dynomite
  - Voldemort

key	value
firstName	Bugs
lastName	Bunny
location	Earth

Bases de datos gráficas

- Se concentra en modelar la estructura de los datos y su interconectividad.
- Escala de acuerdo con la complejidad de los datos
- Inspirado en la teoría de grafos
- Nodos y relaciones entre ellos.
- Ejemplos:
  - Neo4j
  - AllegroGraph

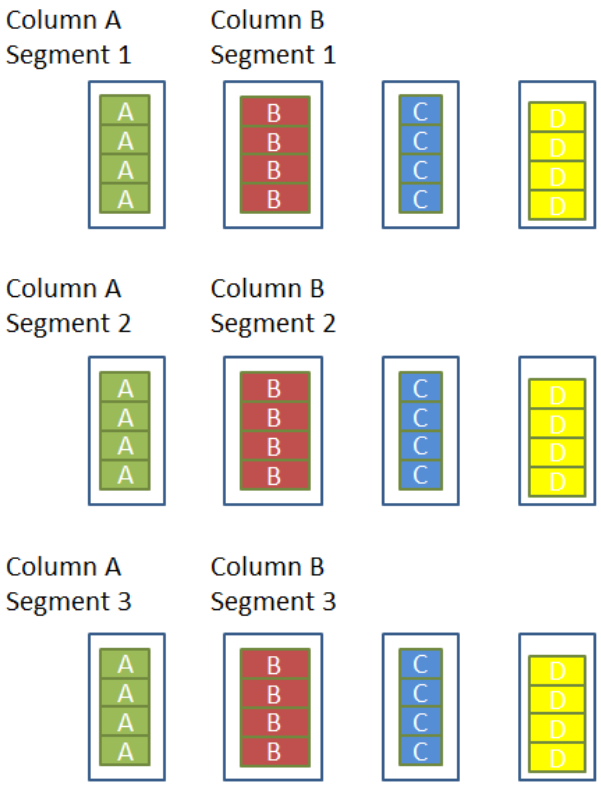


BDs de Documentos

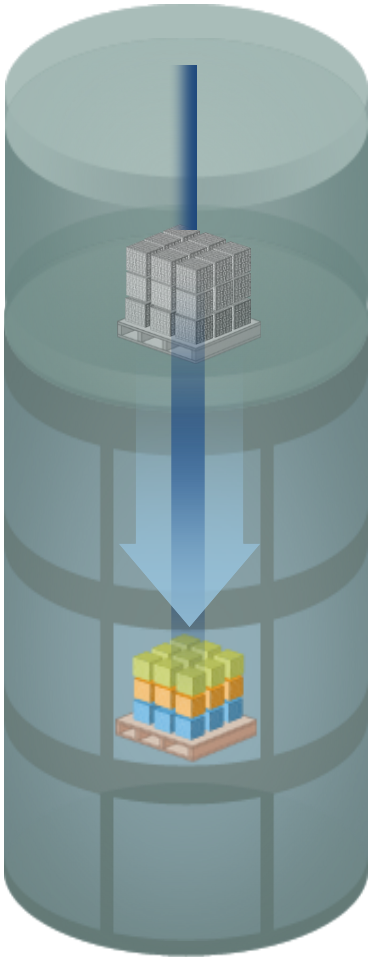
- Inspirado en Lotus Notes
- El modelo de datos Colección de colecciones Clave/Valor
- Los documentos pueden manejar versiones
- Ejemplos:
  - CouchDB
  - MongoDB

BDs orientadas a columnas

- Como las BDs relacionales pero transpuestas
- Los datos pueden ser semi estructurados
- Gran eficiencia con volúmenes grandes
- Ejemplos:
  - MonetDB
  - Infobright



Infobright



- Base de datos columnar orientada al análisis de información
- Excelente performance:
  - Análisis ad-hoc de grandes volúmenes de información
  - Tiempos de carga de 2TB/hour
  - Tasas de compresión de 10:1 a 40:1 (o aún más)
- Fácil administración e implementación:
  - Elimina la necesidad de mantener índices, tablas particionadas, tablas agregadas, etc
  - Auto-tunning: la base va “aprendiendo” de las consultas que recibe
  - Se instala en minutos, y no tiene grandes requisitos de HW
  - Basada en arquitectura MySQL (BD más utilizada en el mundo)
  - Es una base relacional, por lo que dialoga SQL
- Modelo de suscripción Low Cost

Y por qué resulta mejor para BI?

Base de datos orientada a columnas

- Diseñada para análisis de datos
- Excelente tasa de compresión

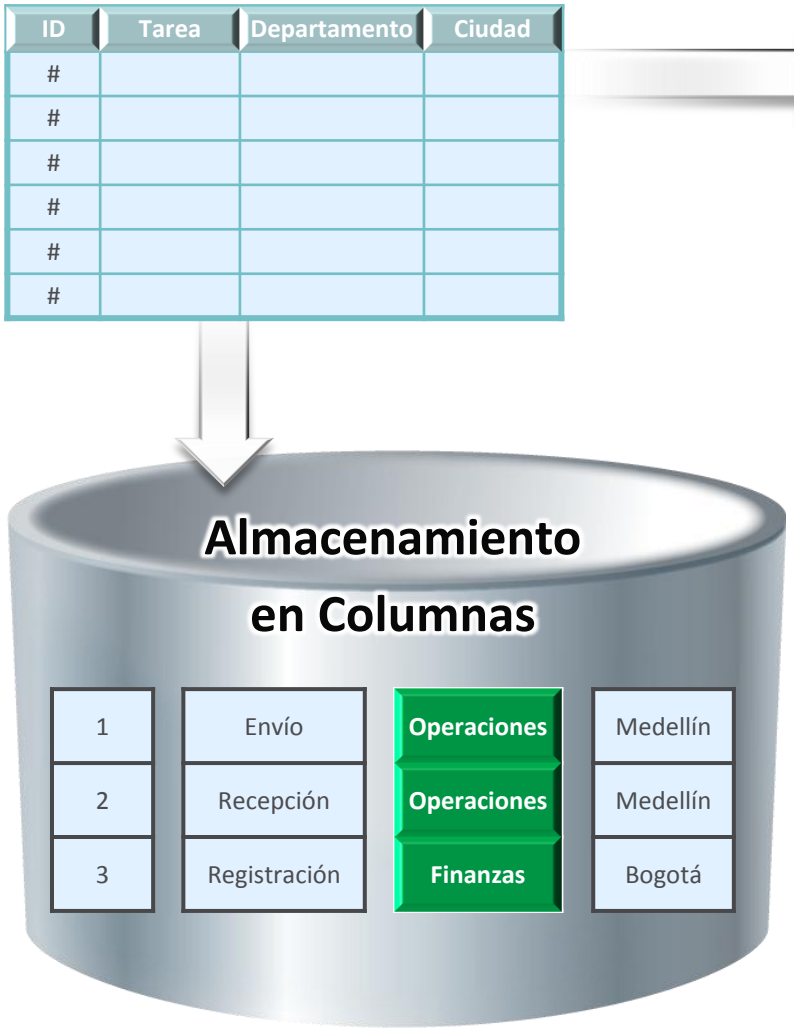
Potencia basada en inteligencia, no en Hardware

- Knowledge Grid
- Motor iterativo

Administración simplificada

- No existe el tuning manual
- Tareas de administración mínimas

Almacenamiento en columnas, no filas



1	Envío	Operaciones	Medellín
2	Recepción	Operaciones	Medellín
3	Registración	Finanzas	Bogotá

El **almacenamiento en filas** sirve si…

- Todas las columnas son necesarias
  - *Por ejemplo, el detalle de una compra de supermercado*
- Ideal para un mundo transaccional donde usualmente se necesita todo el detalle de una entidad

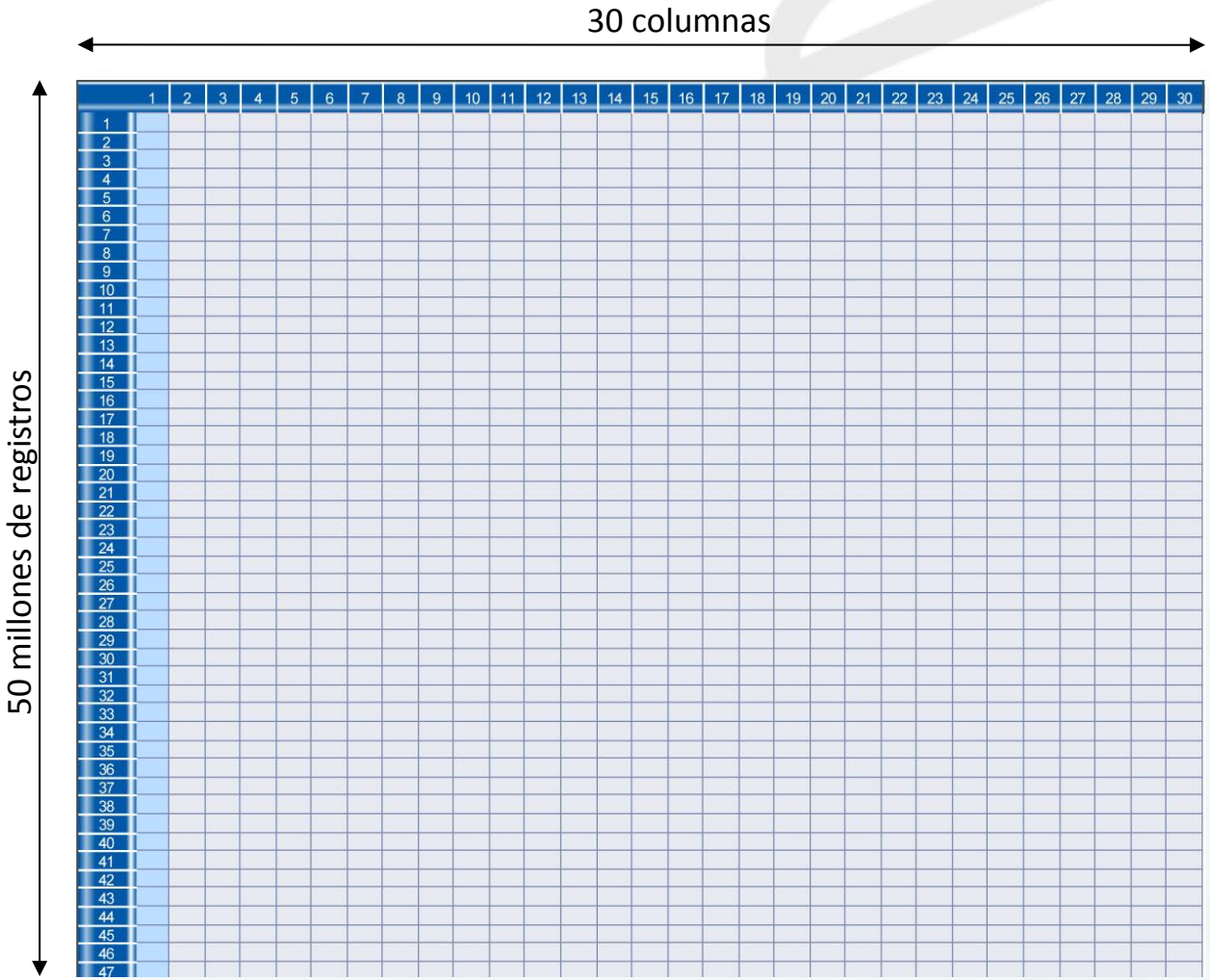
El **almacenamiento en columnas** sirve si…

- Sólo se requieren algunas columnas para el análisis
  - Por ejemplo, el total vendido del producto X en cada una de las sucursales en los últimos 3 meses.
- Información consolidada (sumas, cantidades, promedios, …)
- Ideal para un mundo analítico, donde la información se concentra en métrica de distintas entidades

Almacenamiento en filas vs. columnas

Ejemplo de aplicación:

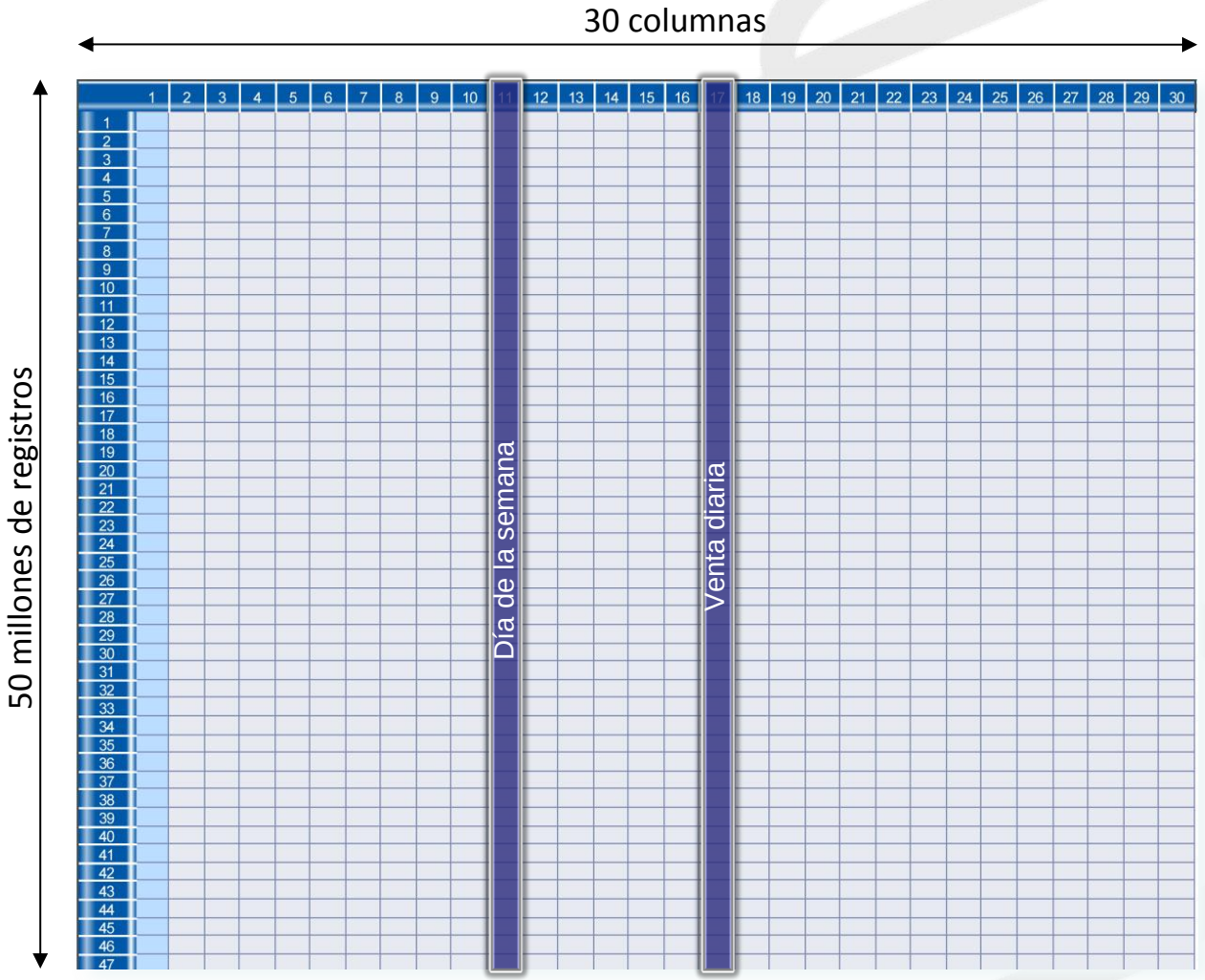
- 50 días con información; 30 columnas/variables por cada registro; un millón de registros por cada día
- El acceso a disco es un problema
- El almacenamiento en filas obliga a recuperar la información de todas las columnas
- Al incrementarse el tamaño de la tabla, se incrementan los índices
- La velocidad de carga se degrada dado que los índices deben recrearse al incorporarse nuevos datos



Almacenamiento en filas vs. columnas

Objetivo: obtener el total de ventas de una semana del año.

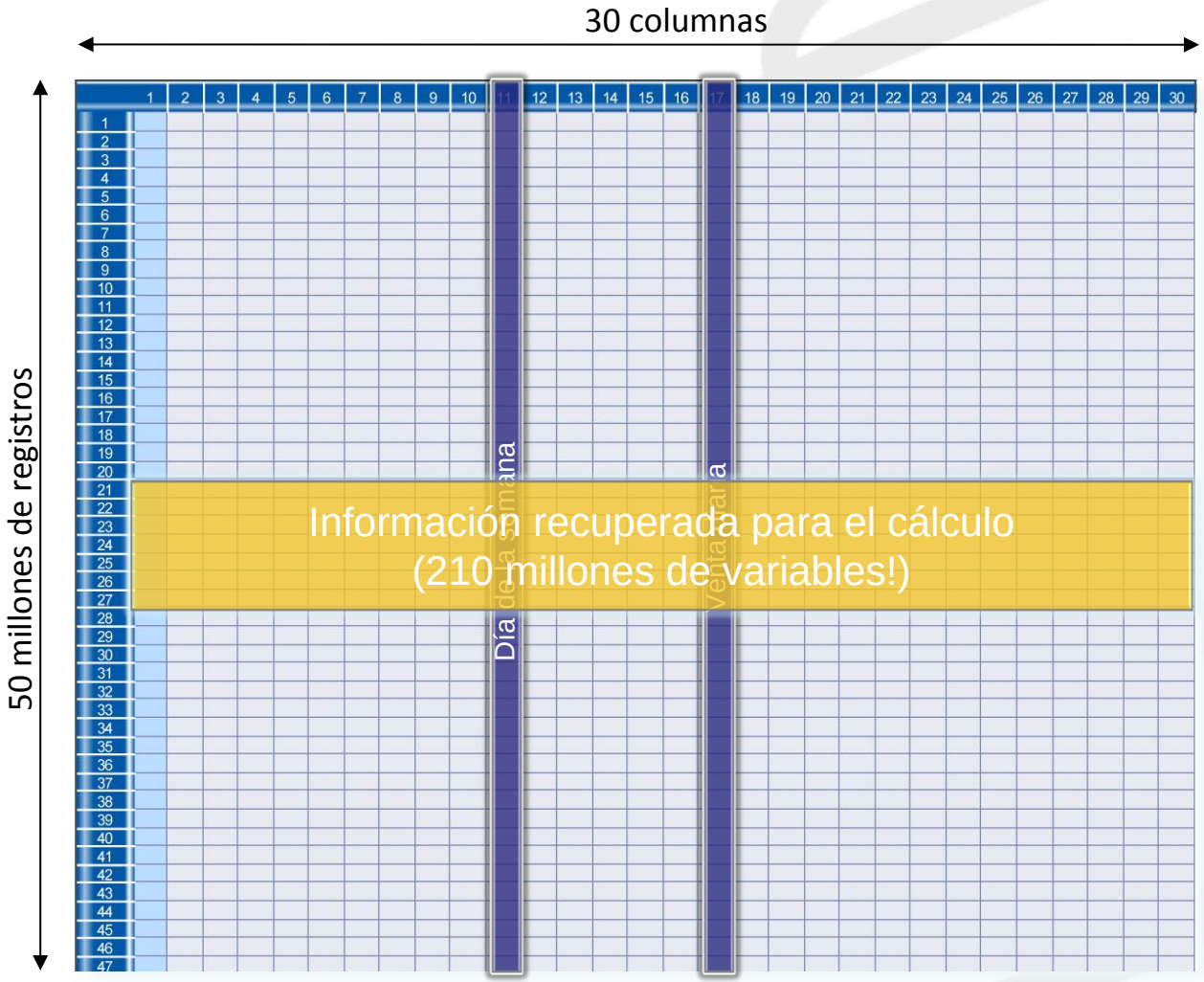
- La consulta SQL sería algo como:
  Select sum(ventas)
  from tabla
  where semana = X



Almacenamiento en filas vs. columnas

Utilizando un esquema de **almacenamiento en filas** (esquema tradicional)

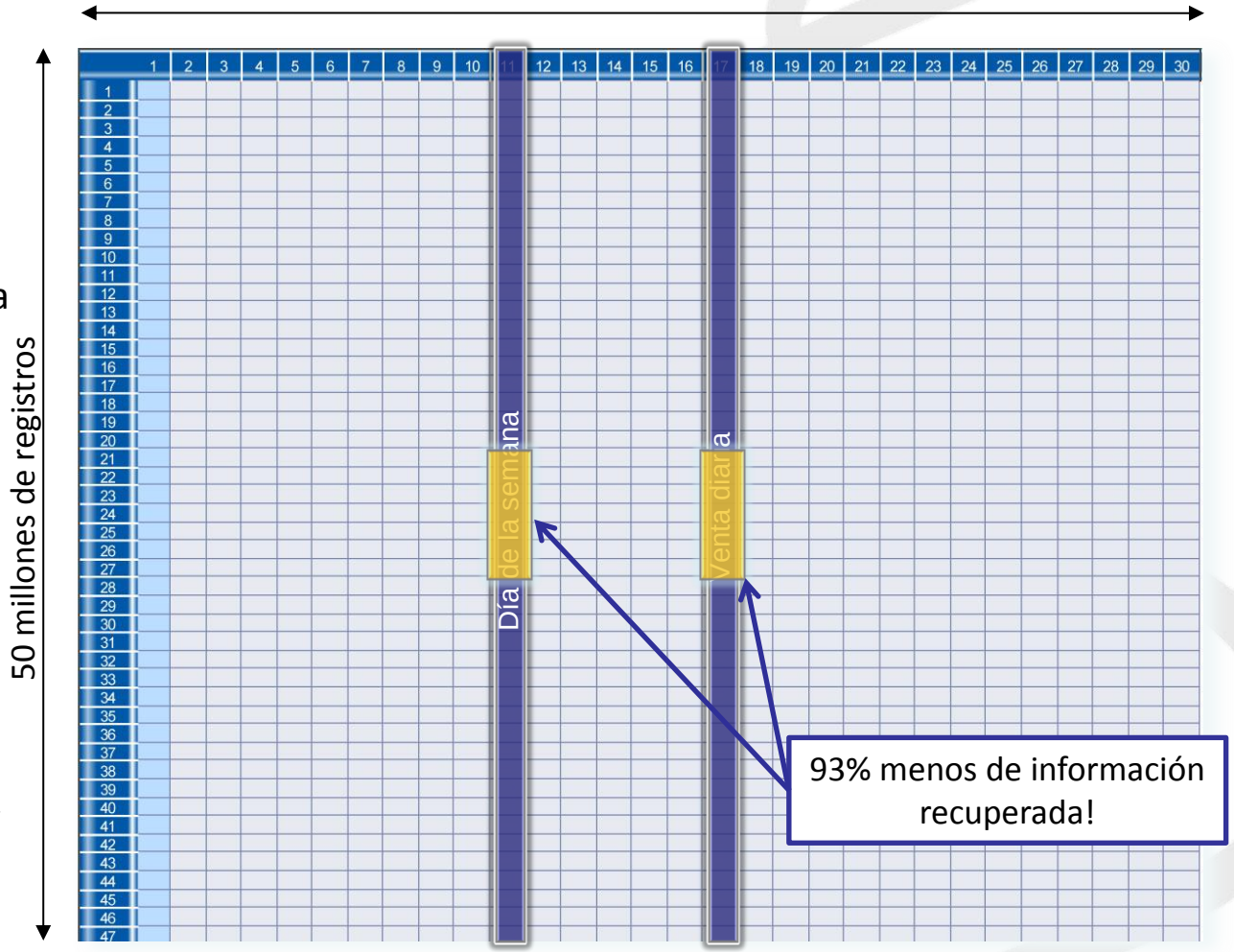
- Se recorren 7 millones de registros (un millón por cada día)
- Para acceder a un valor en particular (el valor de la venta) es necesario leer las 30 columnas, aún cuando no tengan información relevante.
- **Se procesan TODAS las columnas** (210 millones, 30 x 7), **y se DESCARTAN TODAS MENOS una.**
- Es decir **se procesó un 93% más de información.**



Almacenamiento en columnas, no filas

Utilizando un esquema de **almacenamiento en columnas** (esquema utilizado por Infobright)

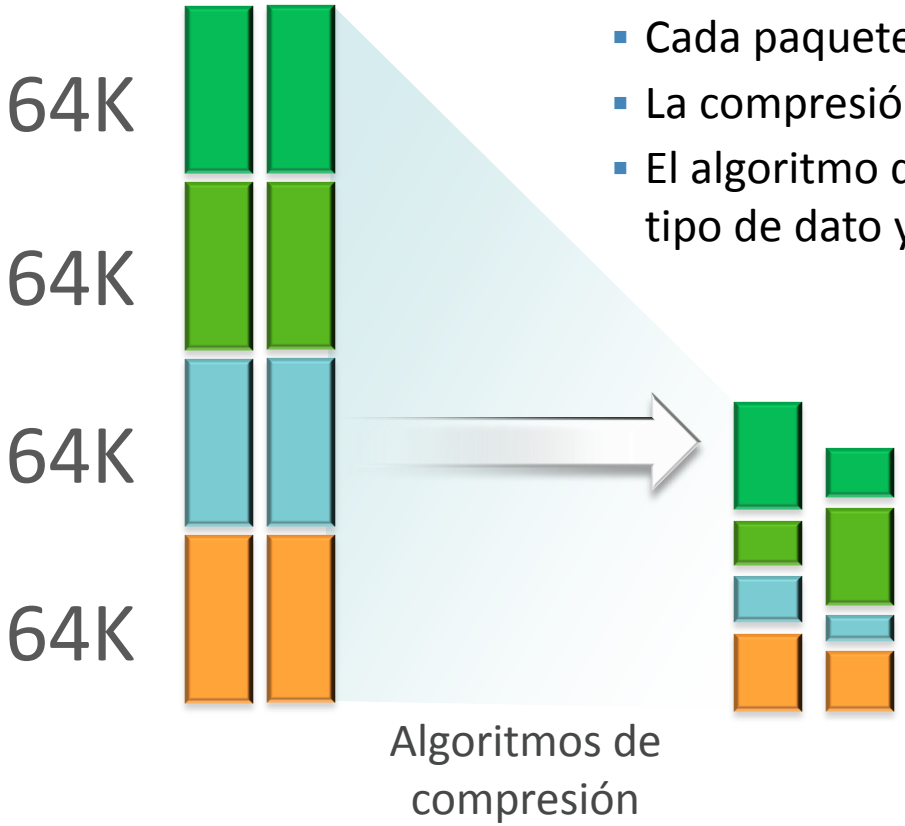
- Se recorren 7 millones de registros (un millón por cada día)
- Se descartan 28 de las 30 columnas.
- Se procesan sólo las 2 columnas necesarias: día y venta diaria.
- **Es decir, <u>sólo se procesa la información necesaria.</u>**
- La inteligencia está en la forma de almacenamiento y en el algoritmo usado para extraer los datos.



Inteligencia, no hardware

Al momento de cargar los datos, crea información (metadata) de los datos automáticamente	• Almacena esta información en la ***Knowledge Grid*** • La KG está cargada en memoria • Menos del 1% del tamaño de los datos comprimidos
Utiliza esta información al procesar consultas para eliminar/reducir acceso a datos	• Al disminuir los datos que deben accederse, aumenta la tasa de respuesta • Respuestas por debajo del segundo cuando la información está contenida en la KG
Beneficios de la arquitectura	• No existe la necesidad de particionar los datos, crear/mantener índices, hacer proyecciones o tuning para mejorar la performance.

Paquetes de Datos y Compresión



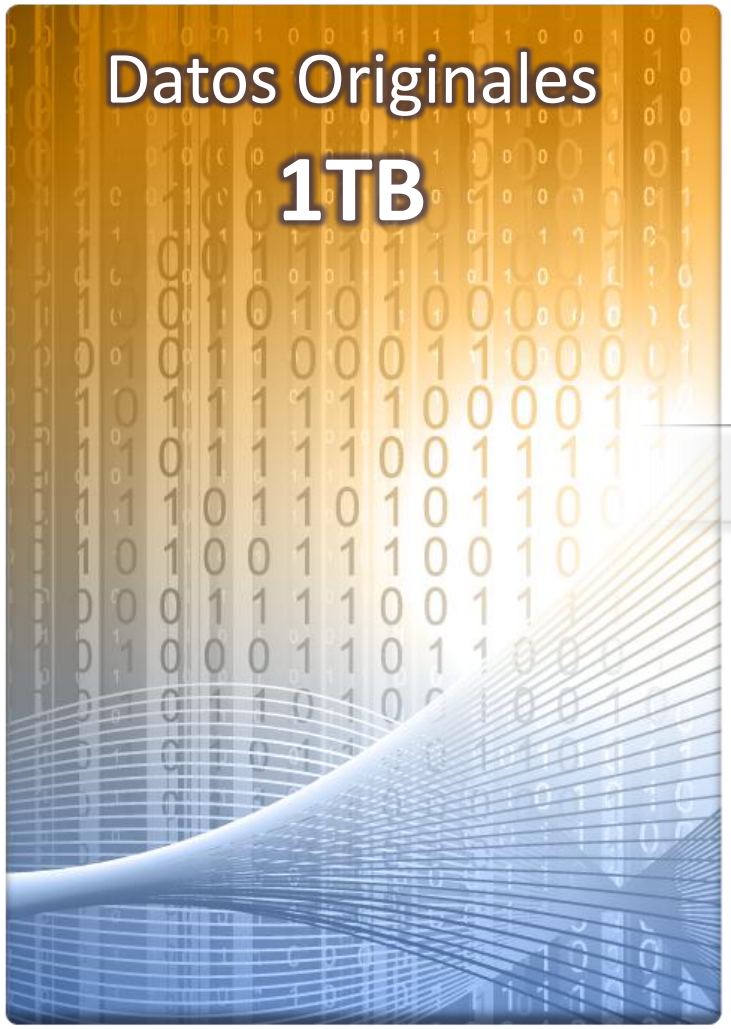
Paquetes de Datos

- Cada paquete contiene 65,536 valores de datos
- La compresión se realiza a nivel de paquete individual
- El algoritmo de compresión cambia de acuerdo con el tipo de dato y su distribución.

Compresión

- Los resultados dependen de la distribución de datos en los paquetes
- La media observada en diferentes implementaciones es **10:1**
- En algunos casos se han detectado compresiones mayores a **40:1**
- *Por ejemplo, con una compresión 10:1, 1TB de datos requeriría solamente 100GB de almacenamiento*

Almacenamiento de los datos



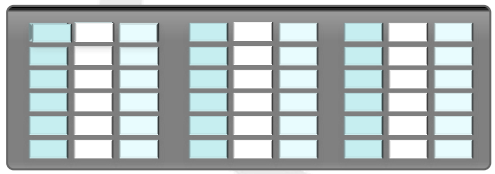
Datos comprimidos

100 GB

Promedio de compresión 10:1

Knowledge Grid

< 1 GB

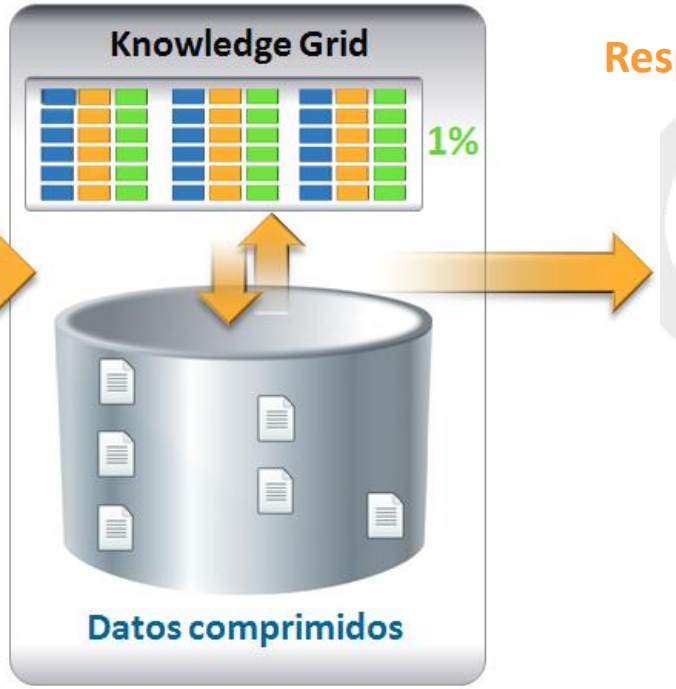


< 1% datos comprimidos

Inteligencia, no hardware

Consulta

Cual es el total de ventas de los últimos 3 meses?



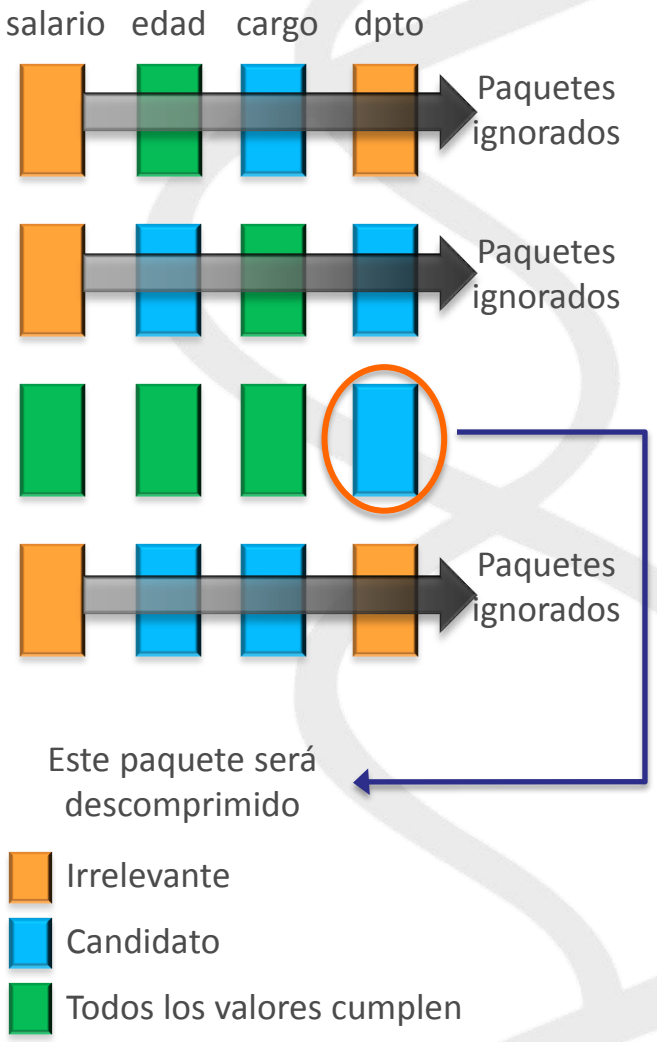
Respuesta

1. Se recibe la consulta.
2. El motor itera sobre el Knowledge Grid
3. Cada iteración elimina Data Packs (necesidad de acceso a datos en disco)
4. Sólo se accede y descomprimen aquellos que son estrictamente necesarios

Aparte de almacenar la información en columnas, se mejora la performance utilizando un acceso inteligente a los datos

Consultas con Knowledge Grid

```
SELECT count(*)
FROM   Empleados
 WHERE Salario> 100000
   AND edad    < 35
   AND cargo   = 'DBA'
   AND dpto    = 'ANT';
```



Benchmarks realizados

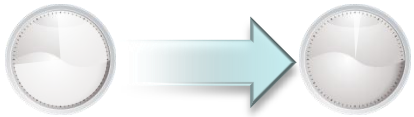
Consulta Analítica

Alternativa — INFOBRIGHT

+2 horas con MySQL — **<10 segundos**

Mobile Data
(15MM eventos)

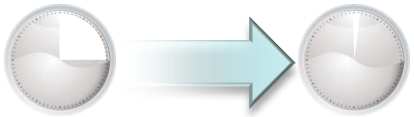
Alternativa — INFOBRIGHT



43 minutos con SQL Server — **23 segundos**

Set de Consultas

Alternativa — INFOBRIGHT



10 seg. – 15 min. con Oracle — **0.43 – 22 segundos**

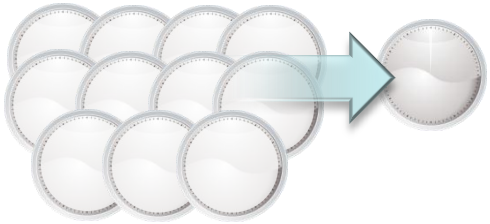
Reporte de BI

Alternativa — INFOBRIGHT

7 hrs en Informix — **17 segundos**

Carga de datos

Alternativa — INFOBRIGHT



11 horas con MySQL ISAM — **11 minutos**

Muchas Gracias!

blog.datalytics.com

@DatalyticsBI

DatalyticsBI

Datalytics

Pentaho en Español

Datalytics