Inteligencia de Negocios en PostgreSQL

http://www.PostgreSQL.org/

Álvaro Herrera
alvherre@2ndQuadrant.com

2ndQuadrant Ltd.
http://www.2ndQuadrant.com/

PGConf.EU 2014
http://2014.pgconf.eu/
22 de octubre de 2014

2ndQuadrant
Professional PostgreSQL

Inteligencia de Negocios en PostgreSQL

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 318633

¿Qué es “inteligencia de negocios”?

“Inteligencia de negocios”: traducción del término *Business Intelligence* (BI)

”habilidades, tecnologías, aplicaciones y prácticas que se utilizan para ayudar a una organización a adquirir una mejor comprensión de su contexto comercial”

¿Qué es “inteligencia de negocios”?

“Inteligencia de negocios”: traducción del término *Business Intelligence* (BI)

”habilidades, tecnologías, aplicaciones y prácticas que se utilizan para ayudar a una organización a adquirir una mejor comprensión de su contexto comercial”

En otras palabras:

Utilización de datos sobre el pasado y el presente para tomar mejores decisiones en el futuro.

Inteligencia de negocios: Objetivos

Requisito

- Tenemos los datos
- (idealmente muchos de ellos)

Inteligencia de negocios: Objetivos

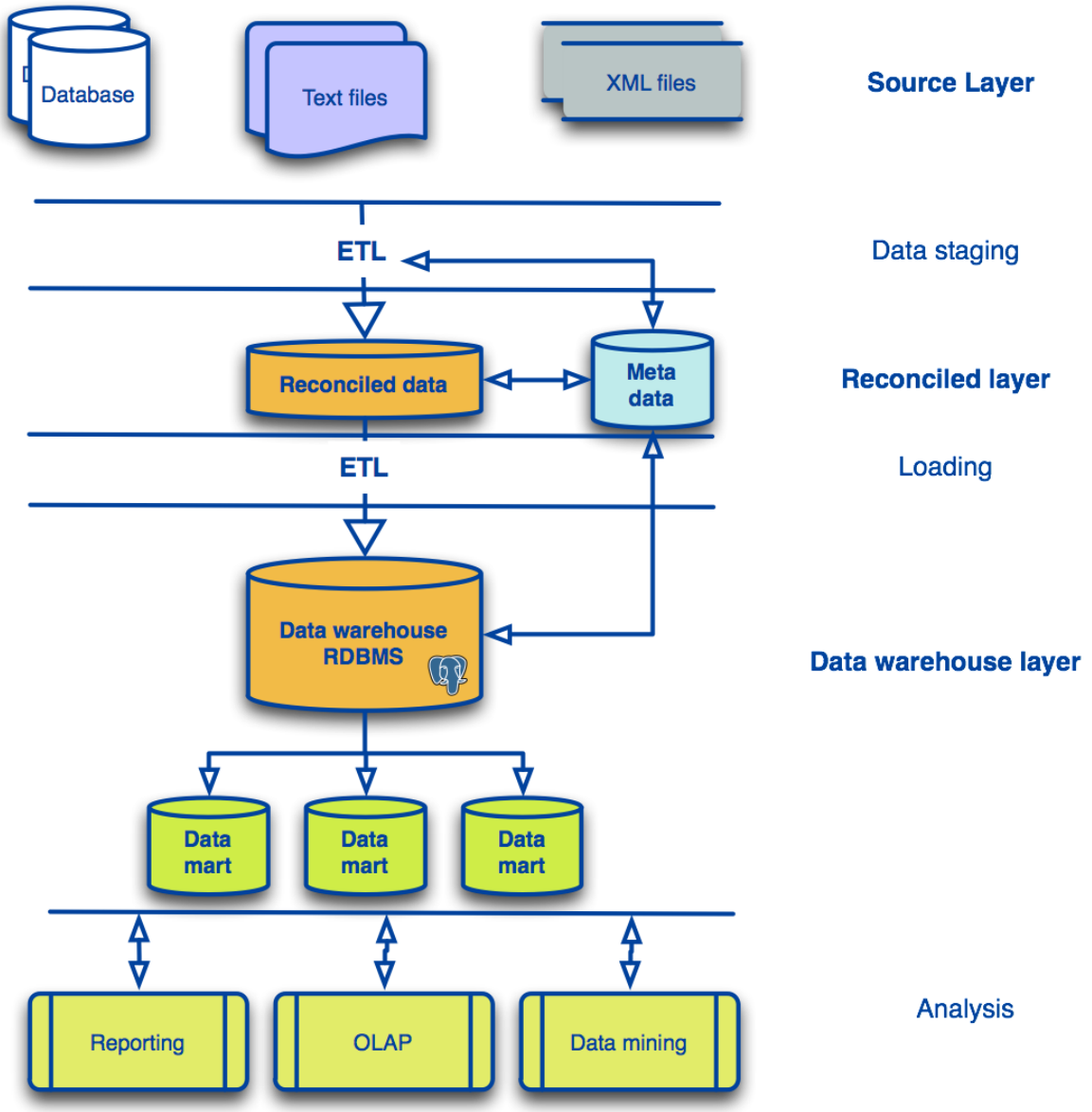
Requisito

- Tenemos los datos
- (idealmente muchos de ellos)

Objetivos:

- Queremos explorar estos datos ...
- ... y descubrir el conocimiento escondido en ellos

Inteligencia de negocios: Arquitectura de tres capas



¿En qué ayuda Postgres?

En esta arquitectura de tres capas:

1. Cuarentena y carga de datos (*stage and load*)
2. Almacenes de datos (*data warehouse*, *data marts*)
3. Ejecución de consultas (*analysis*)

Cuarentena y Carga de datos

Herramientas ETL

- *Extract – Transform – Load*
- Cargar datos externos en variedad de formatos
- Lidiar con errores e inconsistencias
- Manipular los datos antes de insertarlos
- PGLoader: `http://www.pgloader.io`

Aproximación ELT

- *Extract – Load – Transform*
- El proceso de transformación se hace dentro de la BD
- La transformación puede usar herramientas de PostgreSQL

Conectores de datos externos

- *Foreign Data Wrappers (FDW)*
- Conectar, desde Postgres, a servidores de datos remotos
- (En realidad, a cualquier cosa que provea datos)
- El mecanismo es extensible: puedes escribir el tuyo
- `http://wiki.postgresql.org/wiki/FDW`
- postgres_fdw oracle_fdw mysql_fdw odbc_fdw jdbc_fdw informix_fdw firebird_fdw sqlite_fdw tds_fdw couchdb_fdw MonetDB FDW mongo_fdw redis_fdw Neo4j fdw Tycoon FDW file_fdw file_text_array_fdw file_fixed_length_record_fdw json_fdw twitter_fdw ldap_fdw PGStrom Hadoop FDW s3_fdw www_fdw cstore_fdw **multicorn**

Almacenes de Datos

Soporte extensivo JSON

- Tipos de datos `JSON` (9.2) y `JSONB` (9.4)
- Almacenamiento semi-estructurado
- Facilita el acceso a datos de proveniencia no relacional
- Indexable (índices GIN, GiST)

```
{ "id" : "142857",
  "nombre" : "Javier González",
  "nacimiento" : { "fecha" : "1976/10/14",
                   "lugar" : { "pais" : "Venezuela" }
                 },
}
```

Vistas Materializadas

- *Materialized views*
- Útil para data-mart
  - tiene otros usos también
- Permite generar “resúmenes” de datos que pueden ser consultados repetidamente
- Mejora el rendimiento en consultas repetitivas sobre los mismos resúmenes
- Simplifica refresco de vistas resúmenes según necesidades de la organización

Tables “unlogged”

- *UNLOGGED TABLES*
- Tablas que no son respaldadas en WAL
- Mejora en rendimiento de escrituras por ahorrar tráfico WAL
- Desventaja es perder datos en caso de caída

Tables “unlogged”

- *UNLOGGED TABLES*
- Tablas que no son respaldadas en WAL
- Mejora en rendimiento de escrituras por ahorrar tráfico WAL
- Desventaja es perder datos en caso de caída
- En un *DWH* o *data mart* no importa
  - los datos pueden volver a crearse
  - es un riesgo aceptable porque las caídas son infrecuentes
    - o deberían serlo

Hot Standby

- Permite tener una copia de datos en un servidor separado
- ... el cual puede ejecutar consultas de sólo lectura
- Para ejecutar reportes
  - sin sobrecargar el servidor en operación OLTP
- Puede tener réplicas “en cascada”
  - distribuir geográficamente
  - *high availability*
  - tomar respaldos *pg_dump*

En desarrollo: UDR

- *Uni-Directional Replication* (replicación uni-direccional)
- Replicación a nivel lógico, no físico
  - como Slony, Londiste
- Permite hacer cambios en la réplica
  - a diferencia de *hot standby*
  - tablas temporales
  - índices adicionales
- Proyecto en desarrollo de 2ndQuadrant
- Basado en BDR: `http://wiki.postgresql.org/wiki/BDR`

Ejecución de Consultas

Funciones y Procedimientos en la base

- Muchos procesos se pueden ejecutar en la base
- Ahorra tráfico de datos
- Programas definidos por el usuario ...
  - implementar lógica de negocio
- ... en una variedad de lenguajes
- Mantener el conocimiento en la base
  - Permite reutilizar en múltiples herramientas

MADlib

- `http://madlib.net`
- *MADlib: Big Data Machine Learning in SQL for Data Scientists*
- Powerful analytics for Big Data
- También `PL/R`
  - análisis estadístico, etc
  - visualizaciones de datos (gráficos, etc)

En desarrollo: Aggregate pushdown

- Ejecuta la agregación más cerca del recorrido de la tabla
- Permite ahorrar movimiento de datos a través de nodos de ejecución que no lo necesitan

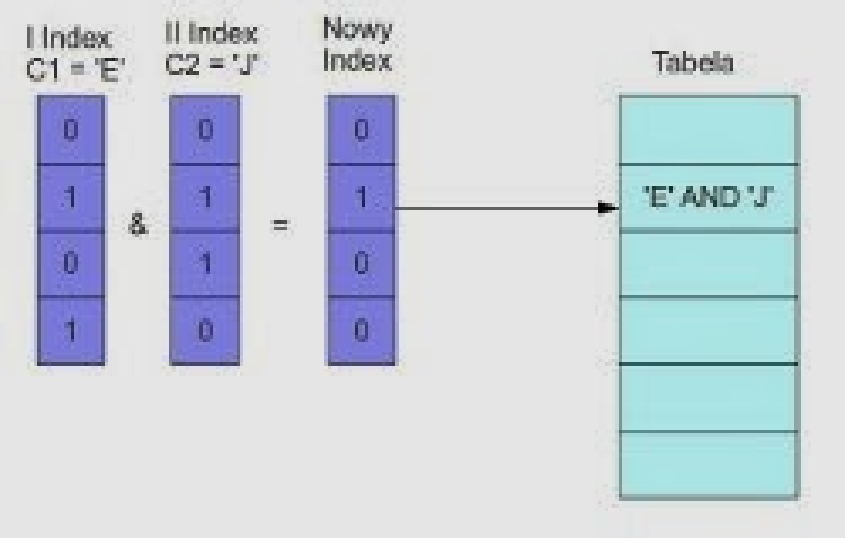
```
WITH w_ventas AS (
SELECT ventas_valor_pagado, ventas_cliente_id
FROM ventas, dim_fechas
WHERE fechas_anno = 2001
        AND fechas_fecha = ventas_fecha
        AND ventas_cliente IS NOT NULL
), w_clientes AS (
SELECT clientes_cliente_id
FROM clientes, clientes_detalles
WHERE clientes_cliente_id = detalles_cliente_id
        AND detalle_estado_credito = 'Bueno'
        AND detalle_genero = 'F'
)
SELECT clientes_cliente_id, sum(ventas_valor_pagado) as total_pagado
FROM w_ventas, w_clientes
WHERE
    ventas_cliente_id = clientes_cliente_id
GROUP BY clientes_cliente_id
HAVING total_pagado > 0
ORDER BY total_pagado DESC
LIMIT 100;
```

Bitmap index scan

- Recorrer un índice, generar un bitmap
- El bitmap permite recorrer la tabla en orden físico
- El bitmap puede ser operado con otro bitmap
  - de otro o del mismo índice

Bitmap index scan

- Recorrer un índice, generar un bitmap
- El bitmap permite recorrer la tabla en orden físico
- El bitmap puede ser operado con otro bitmap
  - de otro o del mismo índice



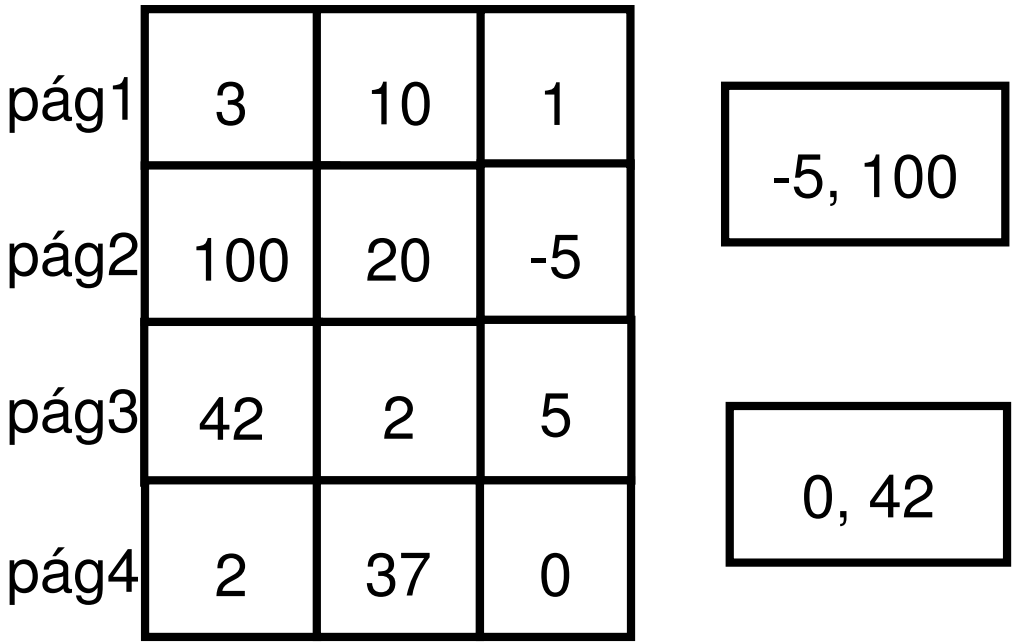
Funciones de ventana deslizante

- Permiten escribir consultas complejas
- Análisis de conjuntos de datos

```
SELECT departamento, empleado_id, salario,
       avg(salario) OVER (PARTITION BY departamento) AS a,
       rank() OVER (PARTITION BY departamento
                    ORDER BY salario DESC) AS r
FROM salario_empleados
ORDER BY r
```

En desarrollo: BRIN indexes

- (Anteriormente llamados *Minmax indexes*)
- Índices para acelerar recorridos secuenciales
- Guardar valores min() y max() de la columna, para grupos de páginas


pág1
3
10
1
pág2
100
20
-5
pág3
42
2
5
pág4
2
37
0
-5, 100
0, 42

En desarrollo: BRIN indexes (2)

- Excluir grupos de páginas del recorrido si el `WHERE` no coincide con valores min/max
- Puede mejorar 10x – 100x tiempo de consulta
- Proyecto de 2ndQuadrant para AXLE

En desarrollo: BRIN indexes (2)

- Excluir grupos de páginas del recorrido si el `WHERE` no coincide con valores min/max
- Puede mejorar 10x – 100x tiempo de consulta
- Proyecto de 2ndQuadrant para AXLE
- Podría usarse para geometry también: guardaría el bounding box
  - unas 200 líneas de código extra
- Otros: “bitmap indexes”, bloom filters, ...

En desarrollo: Almacenamiento “columnar”

- Forma distinta de almacenar datos a nivel físico
- Permite recorrer una misma columna para muchos registros más rápido
- Optimización útil en DWH
  - agregaciones de muchos datos en menos tiempo
- Puede mejorar 4x – 20x tiempo de consultas
- Proyecto de 2ndQuadrant para AXLE

En desarrollo: Freeze Map

- `VACUUM` debe recorrer toda la base cada cierto tiempo
- *freeze* de registros
- Una vez es necesario, de ahí en adelante es pérdida de tiempo
- *Freeze map* es infrastructura para evitar *freezing* inútil
- Ahorra costo de lectura/escritura
- Proyecto de 2ndQuadrant para AXLE

Otros proyectos en desarrollo en 9.5

- Procesamiento de consultas en GPU
  - CustomPlan
- GROUPING SETS

```
  SELECT brand, size, GROUPING(brand, size),
         sum(sales)
    FROM items_sold
GROUP BY rollup(brand, size) ;
 brand | size | grouping | sum
-------+------+----------+-----
 Bar   | L    |        0 |   5
 Bar   | M    |        0 |  15
 Bar   |      |        1 |  20
 Foo   | L    |        0 |  10
 Foo   | M    |        0 |  20
 Foo   |      |        1 |  30
       |      |        3 |  50
```

Addendum: Revisión de parches

- Por favor contribuir
- `pgsql-hackers@postgresql.org`
- `http://www.postgresql.org/list/pgsql-hackers/`
- `http://commitfest.postgresql.org/`

Preguntas

¿Preguntas?

El proyecto AXLE

Advanced Analytics for eXtremely Large European Databases

`http://www.axleproject.eu/`

The research leading to these results has received funding from the European Union's Seventh Framework Programme (FP7/2007-2013) under grant agreement n° 318633