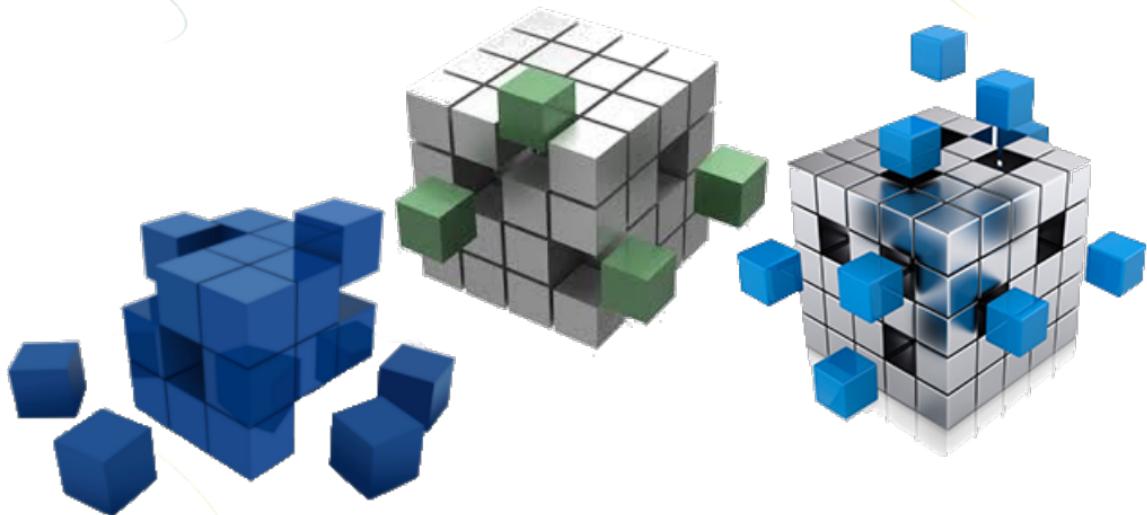


Fundamentos de Business Intelligence

Ing. Cabral, Juan B.



Universidad Nacional del Sur

SciPyCon Argentina 2014

10/2014 - Bahía Blanca - Argentina

About Me

Juan B Cabral

- Software engineer.
- Data scientist.

It's all about ME ME ME....

Unless it means being responsible for something. In which case, it's somebody else's fault, so don't look at ME.



someecards
user card

Agenda

- Historia y descripción del BI.
- Bases de datos transaccionales (OLTP) vs Analíticas (OLAP).
- DataMarts y Data Warehouse.
- Facts y Dimensiones.
- Estructura de datos para análisis multidimensional (OLAP Cubes).
- Implementaciones OLAP: ROLAP - MOLAP - HOLAP.
- Consultas MDX, DMX y XMLA.
- Modelado relacional para RDBMS (ROLAP).
- Diferentes alternativas de OLAP libres y gratuitas (Mondrian & Cubes).
- Aplicaciones BI (Pentaho - Saiku - Cubes Viewer).
- ETL (Extract, Transform and Load).
- Un repaso de la creación manual de un cubo (PostgreSQL + Mondrian + Saiku)
- Conectándose a Mondrian desde Python



Demo Time



Veamos a que apuntamos con este tutorial

The Kimball Group

The Kimball Group is a vendor-independent focused team of senior consultants specializing in the design of effective data warehouses to deliver enhanced business intelligence.

<http://www.kimballgroup.com>

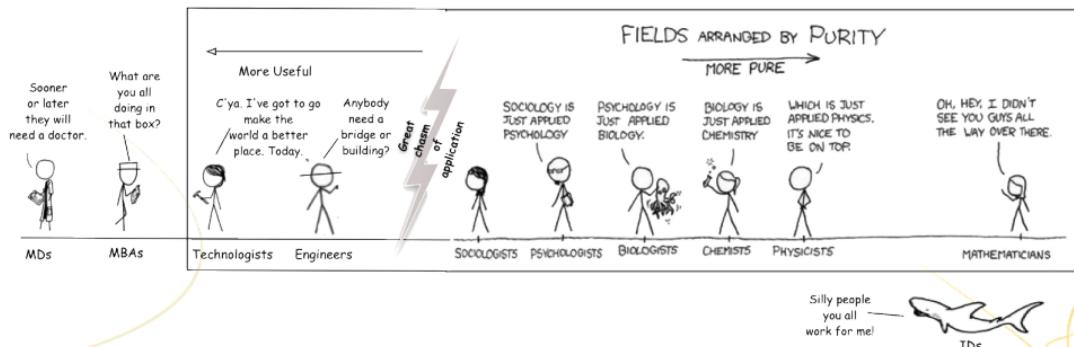


Historia y descripción del BI - Definición

El término inteligencias empresariales se refiere al uso de datos en una **empresa** para facilitar la toma de decisiones. Abarca la comprensión del funcionamiento actual de la **empresa**, bien como la anticipación de acontecimientos futuros, con el objetivo de ofrecer conocimientos para respaldar las decisiones **empresariales**.

En 1989, Howard Dresner (más tarde, un analista de Gartner Group) propuso la "inteligencia de negocios" como un término general para describir "los conceptos y métodos para mejorar la toma de decisiones **empresariales** mediante el uso de sistemas basados en hechos de apoyo"

En resumen es un nombre comercial alrededor de un conjunto de tecnologías y paradigmas para el análisis de grandes volúmenes de datos. El nombre se está abandonando a favor de **Analytics** (Y antes lo llamabas SSD)



Historia y descripción del BI - Características

- **Accesibilidad a la información.** El acceso a datos debe ser de forma independiente a su procedencia
- **Apoyo en la toma de decisiones.** La herramientas debe permitir la selección, análisis y manipulación selectiva de datos
- **Orientación al usuario final.** Se busca independencia entre los conocimientos técnicos de los usuarios y su capacidad para utilizar estas herramientas.



OLTP & OLAP - Versus otras Clasificaciones

Existen diferentes formas de clasificar bases de datos



- Segun la estructura que almacentan: **OO** (db4o), **Document-Oriented** (mongoDB, CouchDB), **RDBMS** (MySql, SQLite, PostgreSQL, Oracle, MicrosoftSQL Server, DB2), **Key-Value** (Redis, riak) o **Graph** (Neo4J)
- Segun si implementan o no SQL: **SQL** (MySql, SQLite, PostgreSQL, Oracle, MicrosoftSQL Server, DB2) o **NO-SQL** (Todas las demás)
- Segun su objetivo:**
OLAP (Mondrian, Cubes, Cognos) y **OLTP** (Todas las demás)

OLTP & OLAP - OLAP vs OLTP

OLAP es el acrónimo en inglés de procesamiento analítico en línea (On-Line Analytical Processing). Es una solución utilizada en el campo de la llamada Inteligencia empresarial (o Business Intelligence) cuyo objetivo es agilizar la consulta de grandes cantidades de datos. ... contienen datos resumidos de grandes Bases de datos o Sistemas Transaccionales (OLTP). Se usa en informes de negocios de ventas, marketing, informes de dirección, minería de datos y áreas similares.

OLTP vs. OLAP

- | | |
|---|--|
| <ul style="list-style-type: none">• provides detailed audit• supports operations• needs detailed data• find one dataset quickly• relational model | <ul style="list-style-type: none">• provides big picture• supports analysis• needs aggregate data• evaluate all datasets quickly• multidimensional model |
| Q: "WHO lives in Atown?" | Q: "HOW MANY live in Atown?" |

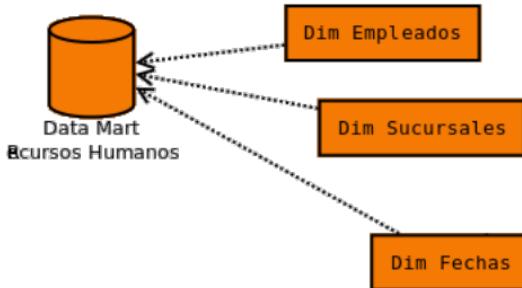
Data Marts

Un **Data mart** es una versión especial de almacén de datos. Son subconjuntos de datos con el propósito de ayudar a que un área específica dentro del negocio pueda tomar mejores decisiones.

Los Data marts son subconjuntos de datos de un almacén de datos para áreas específicas.

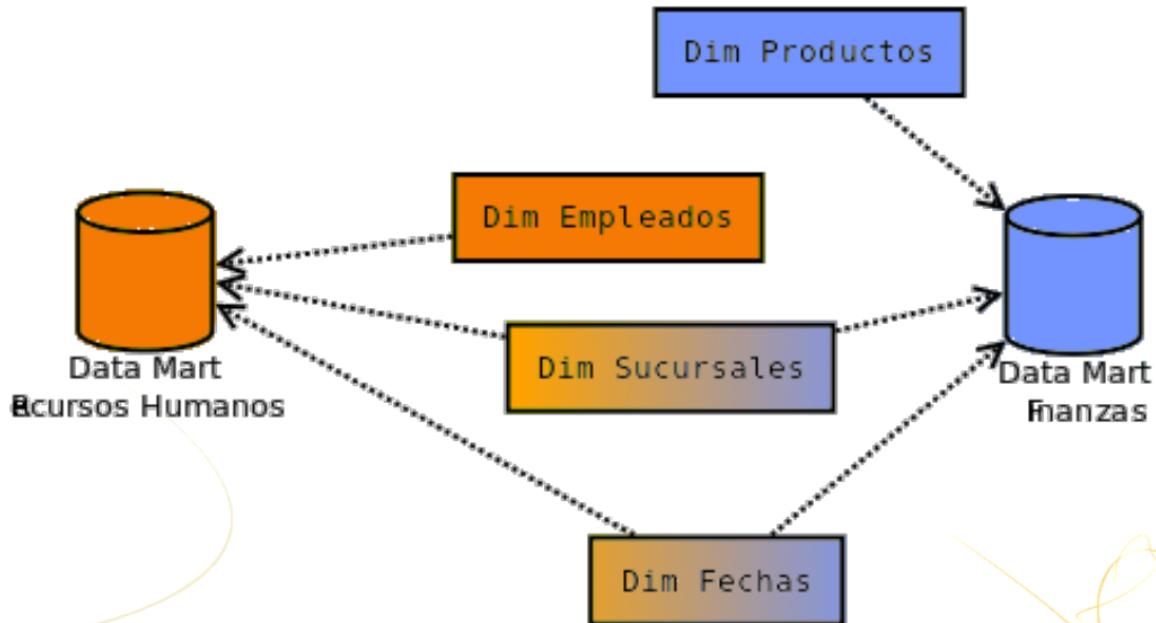
Entre las características de un data mart destacan:

- Usuarios limitados.
- Área específica.
- Tiene un propósito específico.
- Tiene una función de apoyo.



Data Warehouse

Según Ralph Kimball un almacén de datos o **Data Warehouse** es: "una copia de las transacciones de datos específicamente estructurada para la consulta y el análisis"[cita requerida]. También fue Kimball quien determinó que un data warehouse no era más que: "la unión de todos los *Data marts* de una entidad". Defiende por tanto una metodología ascendente (bottom-up) a la hora de diseñar un almacén de datos.



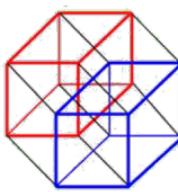
Hechos y Dimensiones - Definición

Hechos (o Facts)

Un hecho es un valor o una medida que representa un hecho (*sic) sobre una entidad o un sistema.

Es algo que efectivamente sucedió o existe y sobre los cuales queremos efectuar análisis.

Los hechos tienen valores que se llaman **Métricas** y definen una dimensión en sí misma.



Dimensiones

Es una estructura que categoriza a hechos y medidas para permitir responder preguntas del negocio.

Dividiendo Dimensiones

- Una **dimension** SIEMPRE se divide en una o mas **Jerarquias**.
- Una **Jerarquia** SIEMPRE puede dividirse en **Niveles**.
- Un **Nivel** PUEDE se dividir en **Niveles**.
- Los **Atributos** pueden estar en las **Jerarquias, Niveles**
- A los registros individuales de una dimension se los llama **Miembros**



Dividiendo Dimensiones - Ejemplos

Data Dimension Hierarchies

Week Hierarchy

Year of week

Week (Week in Year)

- Week begin date
- Week number

Day (Date)

- Day name
- Day num of week
- Weekday indicator
- Holiday indicator

Calendar Hierarchy

Year

Quarter (Year-Qtr)

- Quarter number

Month (Year-Month)

- Month name
- Month number

Fiscal month (FY-Month)

Fiscal quarter (FY-Qtr)

Fiscal year

Fiscal Hierarchy

Hechos y Dimensiones - Un ejemplo

Tito fue a comprar jabón en polvo gasto en total \$16 en la sucursal 7 el 16 de octubre del 2014

- **Hecho:** Sucedío 1 (métrica) venta que se gasto \$ 16 (métrica).
- **Dim. Cliente:** tito
- **Dim. Producto:** jabón en polvo.
- **Dim. Sucursal:** 7
- **Dim. Fecha:** 16 de octubre del 2014

Consultas multidimensionales basándonos en el ejemplo:

1. Promedio de gastos por cliente.
2. Quiero el promedio de las ventas por producto y sucursal.
3. Quiero la suma de ingresos por producto.
4. Quiero conteo de ventas por día.



Hechos y Dimensiones - Ejemplo Científico

El telescopio X encontró una estrella tipo RR-Lyrae con una magnitud aparente Y en la posición Z en la fecha W.

- **Hecho:** Sucedió 1 (métrica) descubrimiento de una estrella de magnitud aparente Y (métrica).
- **Dim. Dispositivo:** Telescopio X
- **Dim. Tipo de Fuente:** RR-Lyrae
- **Dim. Zona:** rango R tal que R contiene a Z
- **Dim. Fecha:** W

Consultas multidimensionales basándonos en el ejemplo:

1. Cantidad de descubrimientos por posición.
2. Promedio de magnitud por tipo de fuente.



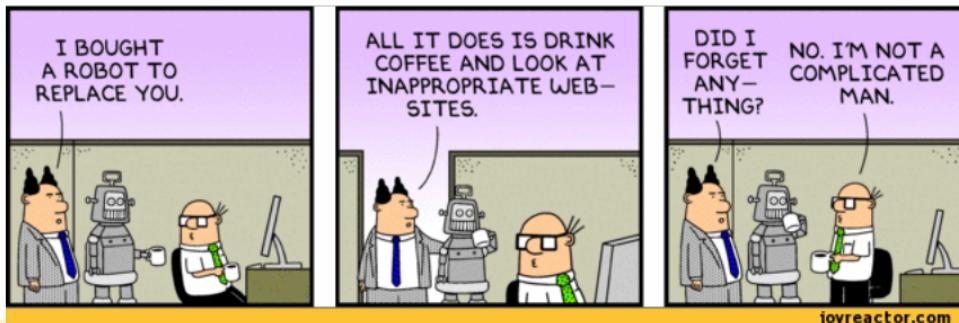
Dimensiones - Tipos

- **Regular:** cliente, articulo, tipo de fuente
- **Conformed:** Conectan mas de un datamart y tienen mismo significado semántico en todos los datamarts
- **Role Played:** Cambian de significado según el datamart
- **Junk:** Suelen tener banderas como [S|N] o Sexo
- **Dirty:** Son *role-playing* que no tienen significado en si mismos. Por ejemplo: una dimensión numero que en un datamart es un identificador de facturas y en otro es un DNI.



Dimensiones - Identificando Miembros

- Cada miembro de una dimension normalmente se extrae de una cantidad de un sistema transaccional (una tupla en una RDBMS, una fila de Excel, etc)
- En el sistema transacional es comun que esta entidad tenga un identificador unico (PK en una RDBMS, ID en una base documental, nro de orden en un Excel)
- Las claves del sistema trasaccional las llamamos **Business Key (BK)**.
- Un miembro tiene una clave calculada a partir del **BK** llamada **Surrogated Key (SK)**
- Es obligacion del analista mantener esta relación.
- Las SK pueden no ser unicas en una dimensión.



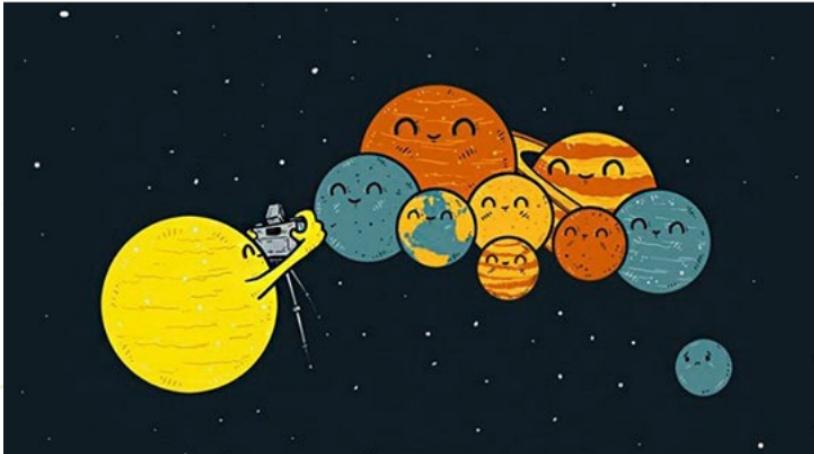
Slowly Change Dimension

- Se supone que una DW no cambia mucho en sus dimensiones.
- Si alguna cambia: **cambia lentamente**

Suponiendo que tengo alguna dimension con un miembro parecido a:

```
{sk: 1, bk: 001, nombre: "Plutón", cat: "Planeta"}
```

Ahora Plutón no es mas un planeta...



Slowly Change Dimension - Enfoques

1. **SCD Tipo 0:** No hacemos nada. No siempre un cambio en OLTP refleja un cambio en OLAP.
2. **SCD Tipo 1:** No Guardo Historia.

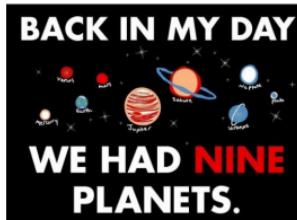
```
{sk: 1, bk: 001, nombre: "Plutón", cat: "Planeta Enano"}
```

2. **SCD Tipo 2:** Guardo Historia Versionando.

```
{sk: 1, bk: 001, nombre: "Plutón", cat: "Planeta", ver: 1}  
{sk: 1, bk: 001, nombre: "Plutón", cat: "Planeta Enano", ver: 2}
```

3. **SCD Tipo 3:** Guardo Historia Cambiando la Dimensión.

```
{sk: 1, bk: 001, nombre: "Plutón", cat0: "Planeta", cat1: "Planeta Enano"}
```

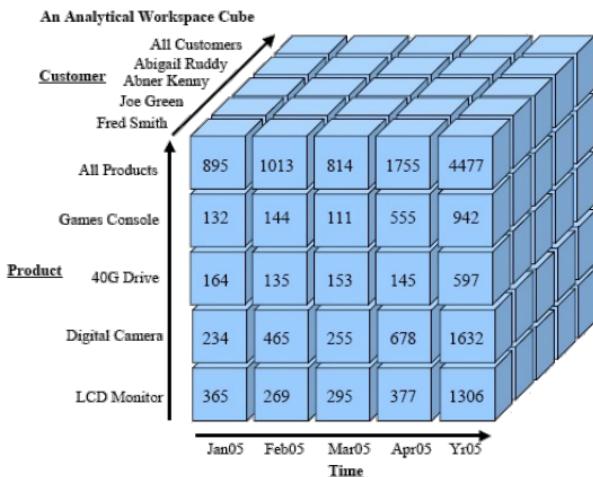


Cubos OLAP

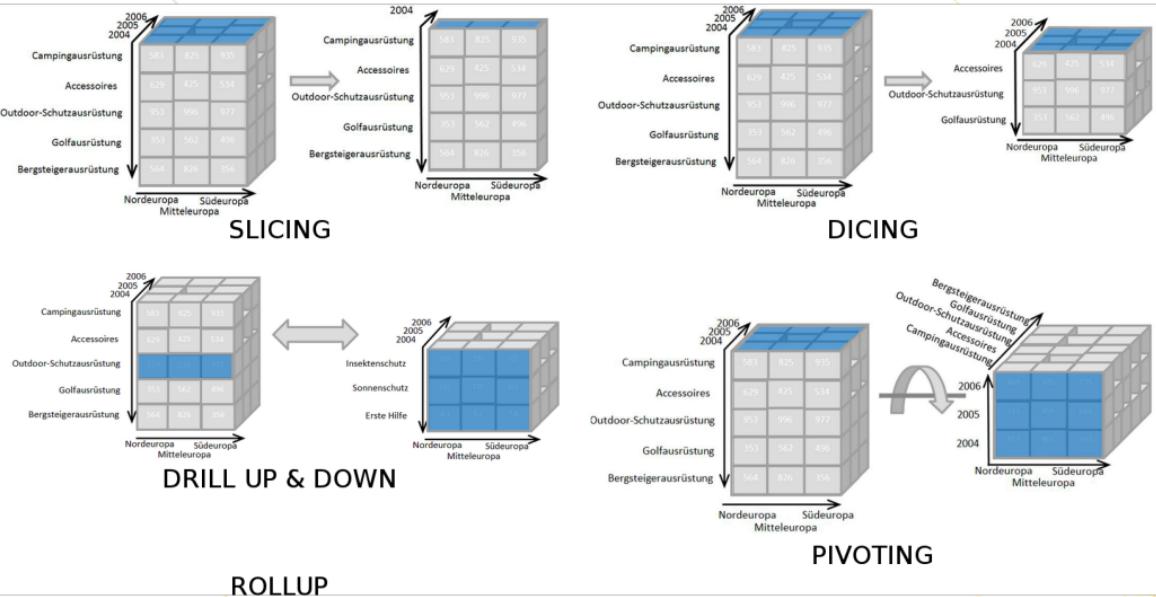
Es una base de datos multidimensional, en la cual el almacenamiento físico de los datos se realiza en un vector multidimensional.

Pueden considerar como una ampliación de las dos dimensiones de una hoja de cálculo.

Las respuestas de los cubos olap son cubos de menor dimensión (normalmente tablas de doble entrada) y los datos se le llaman celdas.



Cubos OLAP - Operaciones



Cubos OLAP - Implementaciones

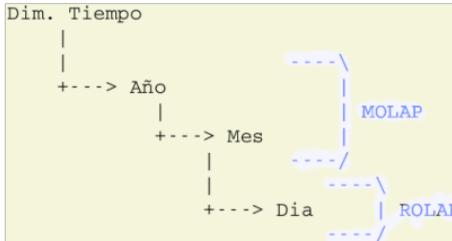
- **MOLAP** La base de datos es multidimensional hasta su nivel mas bajo. Cada miembro de cada hecho esta almacenado en una celda diferente



- **ROLAP** La base de datos es una vista lógica (schema) sobre una relacional. Existen diferentes estrategias para crear la base de datos segun necesidades. (es lo que vamos a continuar viendo en este tutorial)



- **HOLAP** Las dimensiones pueden dividir ciertos niveles en MOLAP y ciertos niveles en ROLAP



MDX - Multi Dimensional eXpressions

MDX

- Es un lenguaje de consulta para bases de datos multidimensionales sobre cubos OLAP.
- Es declarativo a diferencia de las operaciones que son imperativas.
- Es muy similar a una consulta SQL, nos devuelve un conjunto de celdas.
- Para manejar jerarquias y niveles MDX tiene funciones como Children (hijos en inglés), cousin (primos) y parents (padres).

Una consulta tiene la forma

SELECT

```
<especificación de eje> ON COLUMNS,  
<especificación de eje> ON ROWS  
FROM <especificación de cubo>  
WHERE <especificación Slicer (rebanador)>
```

MDX - Multi Dimensional eXpressions - Ejemplo

```
SELECT
{
    [Measures].[Sales Amount],
    [Measures].[Tax Amount]
} ON COLUMNS,
{
    [Date].[Fiscal].[Fiscal Year].&[2002],
    [Date].[Fiscal].[Fiscal Year].&[2003]
} ON ROWS
FROM [Adventure Works]
WHERE ( [Sales Territory].[Southwest] )
```

- The SELECT clause sets the query axes as the Sales Amount and Tax Amount members of the Measures dimension, and the 2002 and 2003 members of the Date dimension.
- The FROM clause indicates that the data source is the Adventure Works cube.
- The WHERE clause defines the slicer axis as the Southwest member of the Sales Territory dimension.

XMLA - XML for Analysis

XMLA consists of only two SOAP methods.[2] It was designed in such a way to preserve simplicity.

- **Execute** method has two parameters:

Command: Command to be executed. It can be MDX, MDXML, DMX or SQL.

Properties: XML list of command properties such as Timeout, Catalog name, etc.

The result of Execute command could be Multidimensional Dataset or Tabular Rowset.

- **Discover**

Discover method was designed to model all the discovery methods possible in OLEDB including various schema rowset, properties, keywords, etc. Discover method allows users to specify both what needs to be discovered and the possible restrictions or properties. The result of Discover method is a rowset.

DMX - Data Mining eXtensions

Query language for Data Mining Models supported by Microsoft's SQL Server Analysis Services product. Whereas SQL statements operate on relational tables, DMX statements operate on data mining models

- **DDL** Creates mining model (CREATE MINING STRUCTURE, CREATE MINING MODEL)
- **DML** Train mining models: INSERT INTO.
- **DML** Browse data in mining models SELECT FROM.
- **DML** Make predictions using mining model: SELECT ... FROM PREDICTION JOIN.

```
SELECT [Loan Seeker], PredictProbability([Loan Seeker])
FROM
    [Decision Tree]
NATURAL PREDICTION JOIN
(SELECT
    35 AS [Age],
    'Y' AS [House Owner], 'M' AS [Marital Status],
    'F' AS [Gender], 2 AS [Number Cars Owned],
    2 AS [Total Children], 18 AS [Total Years of Education]
)
```

XMLA - XML for Analysis - Ejemplo

```
<soap:Envelope>
<soap:Body>
<Execute xmlns="urn:schemas-microsoft-com:xml-analysis">
<Command>
<Statement>SELECT Measures.MEMBERS ON COLUMNS FROM Sales</Statement>
</Command>
<Properties>
<PropertyList>
<DataSourceInfo/>
<Catalog>FoodMart</Catalog>
<Format>Multidimensional</Format>
<AxisFormat>TupleFormat</AxisFormat>
</PropertyList>
</Properties>
</Execute>
</soap:Body>
</soap:Envelope>
```

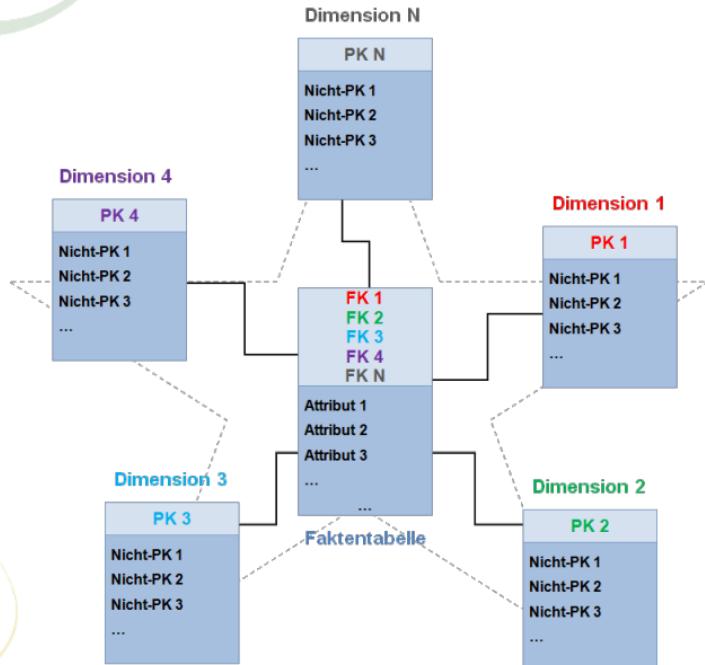


OLAP - Modelado relacional (ROLAP)

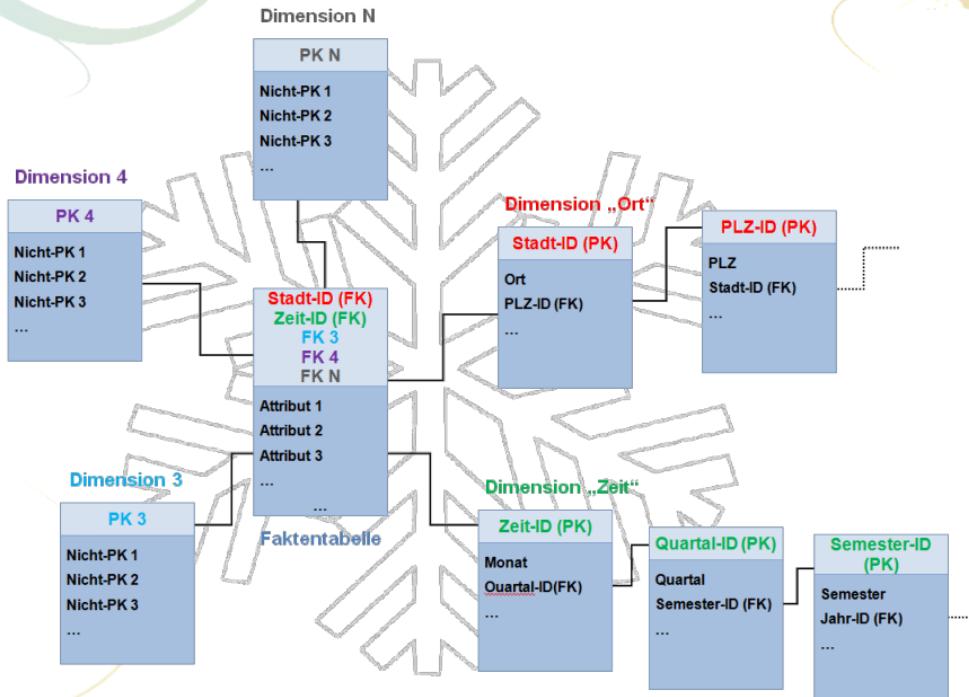
- Para facilitar en análisis de abandona la 3FN.
- Hay 3 formas de estructurar una RDBMS para ROLAP.
- Aumentan la redundancia de datos.
- Disminuyen los Join considerablemente.
- **Nota:** Recuerden esto es para facilitar el análisis sacrificando TODO lo demás de ser necesario.



OLAP - Modelado relacional (ROLAP) - Star Schema



OLAP - Modelado relacional (ROLAP) - Snow Schema



OLAP - Alternativas: Cubes



<http://cubes.databrewery.org/>

- Implementado en Python con aproximadamente ~2 años de desarrollo.
- Liviano
- Configurable con JSON (bastante feos los json)
- Usa sqlalchemy como backend de DB
- Tiene implementados dos visores cubes-views y cubes-viewer.
- Como método de análisis utiliza las primitivas de los cubos.
- Para llamadas remotas tiene una interfaz rest llamada slicer.

OLAP - Alternativas: Mondrian



<http://mondrian.pentaho.com/>

- Implementado en Java.
- Liviano como una vaca gorda corriendo con una armadura de bronce.
- Configurable con XML (increíblemente bonitos)
- Soporta MDX.
- Soporta multiples backends (Casi cualquier cosa conocida anda)
- Soporta cargas de datos muy grandes-
- Tiene cientos de visores implementados (Saiku - Pentaho - OpenI)
- Estandar de Facto del mercado.
- Soporta XMLA

BI - End To End

- Communmente se le llama BI a una serie de herramientas integradas para el análisis.
- Son muchas: Pentaho (Sobre Mondrian), Cubes Viewer (Sobre Cubes), Saiku (Sobre Mondrian), Cognos, MS-AS, OpenI (Sobre Mondrian), YellowFin...
- Es lo que vimos como ejemplo al comienzo permite la ejecución y resumen de datos de manera *Drag and Drop*

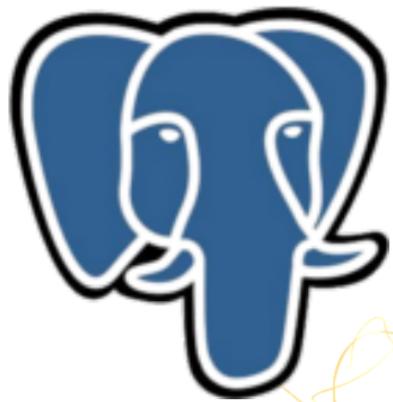
Yellowfin **COGNOS®**

SAIKU  **pentaho®**

CUTTING EDGE OPEN SOURCE ANALYTICS

Parte Práctica

- Vamos a ver un mini problema en un OLTP.
- Vamos a llevar los datos a una forma estrella OLAP en PostgreSQL.
- Vamos a Diseñar el Schema lógico para mapear la estrella.
- Vamos a configurar Saiku para que tome el cubo.
- Vamos a tirar unas consultas MDX desde Python (pip install python-xmla).



¿Preguntas?

- Charla: <http://goo.gl/3rb9QE>
- **Contactos:**
 - jbcabral.com
 - Juan B Cabral <jbc.develop@gmail.com>

I hate it when
someone starts asking
me questions just
after waking up.



AR...EHM...ARR U..... KIDDIN...MEEE?