

Herramientas y temáticas actuales de Data Warehouses y OLAP

Javier Bonet

Joel Catacora

- Base de datos avanzada • 20 de mayo del 2015



TEMAS EN INVESTIGACIÓN

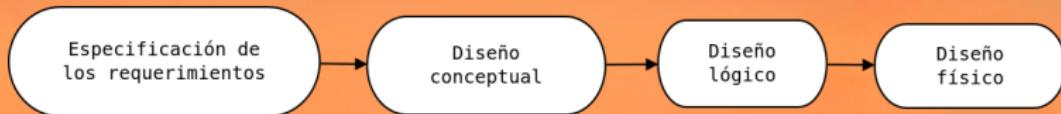
Estos son algunos temas que están siendo investigados, y que veremos rápidamente:

- El diseño de un Data Warehouse (DW).
- Ontologías para la creación de un Data Warehouse.
- Fuzzy Data Warehouse (FDW).
- Web Warehouse (WW).



DISEÑO GENERAL DE UN DW

El procedimiento, planteado en [5], se basa en la suposición de que los DWs son un tipo particular de bases de datos dedicado a fines analíticos. Por lo tanto, su diseño debe seguir las fases de diseño de bases de datos tradicionales, es decir, la especificación de los requisitos, diseño conceptual, diseño lógico y diseño físico. Sin embargo, hay diferencias significativas entre las fases del diseño de bases de datos tradicional y un DW, que se derivan de su diferentes naturalezas.



- **Especificación de los requerimientos:** determina, entre otras cosas, *qué* datos deben estar disponibles y *como* deben organizarse.
- **Diseño conceptual:** la fase anterior debe proporcionar los elementos necesarios para la construcción de un esquema conceptual inicial del Data Warehouse.
- **Diseño lógico:** primero, consiste en la transformación del esquema conceptual en un esquema lógico; y segundo, la especificación de los procesos ETL.
- **Diseño físico:** comprende la implementación del esquema lógico, y los procesos ETL. Durante la fase de diseño físico, el esquema lógico se convierte en una estructura de base de datos física.



DISEÑO CONCEPTUAL

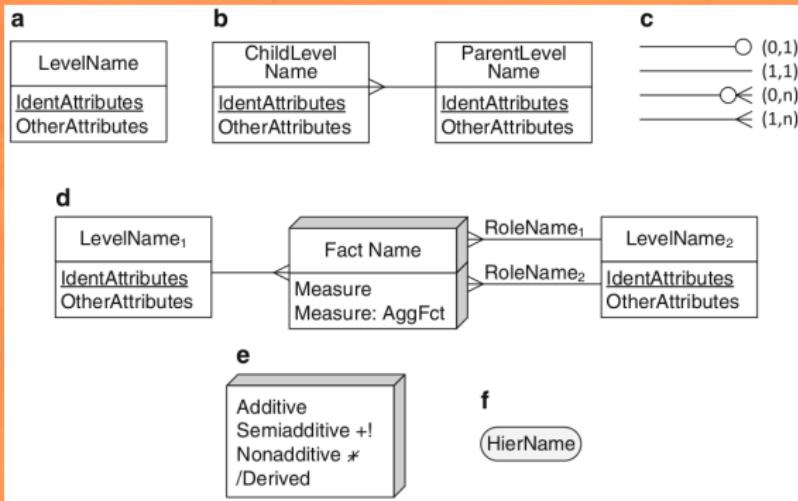
Siguiendo [5], vamos a utilizar el modelo **MultiDim** para definir los esquemas conceptuales, aunque también se pueden utilizar otros modelos conceptuales que proporcionan una representación abstracta de un esquema del DW.

Presentamos los principales componentes del modelo:

- Un **esquema** se compone de un conjunto de dimensiones y de un conjunto de hechos.
- Una **dimensión** se compone de un nivel o una o más jerarquías.
- Una **jerarquía** está a su vez compuesto por un conjunto de niveles.
- Un **nivel** es análogo a una entidad en el modelo E/R.
- Un nivel tiene un conjunto de **atributos** que describen las características de sus miembros. Además, un nivel tiene uno o varios **identificadores**.

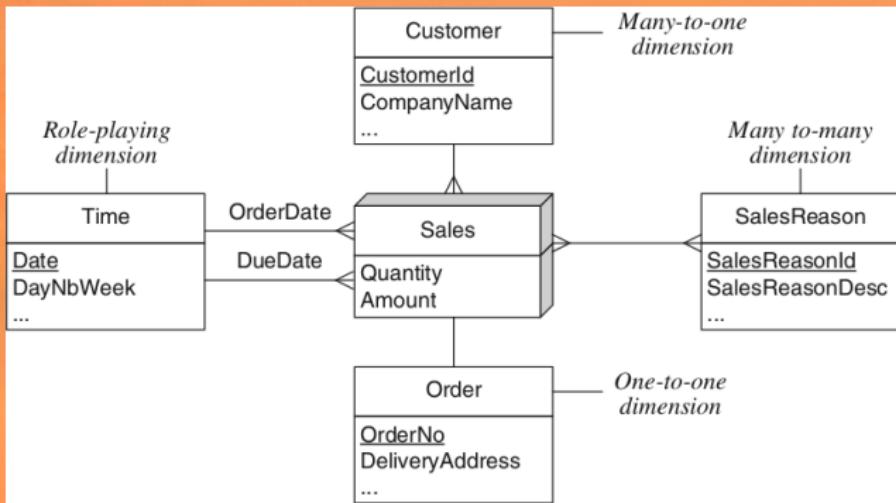
- Un **hecho** relaciona varios niveles. El mismo nivel puede participar varias veces con un hecho, mediante diferentes **roles**. Cada rol se identifica por un nombre.
- Un hecho puede contener atributos llamados **medidas**.
- Las medidas se sumarizan a largo de dimensiones cuando se realizan operaciones roll-up. La **función de summarización** asociada con una medida se puede especificar junto al nombre de la medida; SUM por defecto.
- Podemos clasificar a las métricas como **aditiva**, **semiaditiva**, o **no aditiva**. Además, las medidas y los atributos de nivel pueden **derivarse**, es decir que se calculan en base a otras medidas o atributos en el esquema. Por defecto, las medidas son aditivas.

Estas son algunas de las notaciones del modelo **MultiDim**.



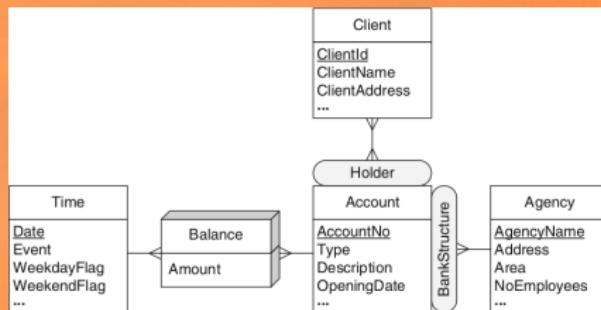
- (a) Nivel, (b) jerarquía, (c) cardinalidades, (d) hecho con las medidas y niveles asociados, (e) tipos de medidas, (f) nombre de la jerarquía.

Ejemplo 1:





Ejemplo 2



Ejemplo 3



ONTOLOGÍAS

Como se remarca en [4] la utilización y el análisis de datos no estructurados, como pueden ser los archivos de texto, la web semántica, etc., han sido incluidos entre las posibles fuentes de datos, ya que presentan una gran cantidad de información que pueden aportar para la tarea de toma de decisiones.

Breve repaso de ontologías

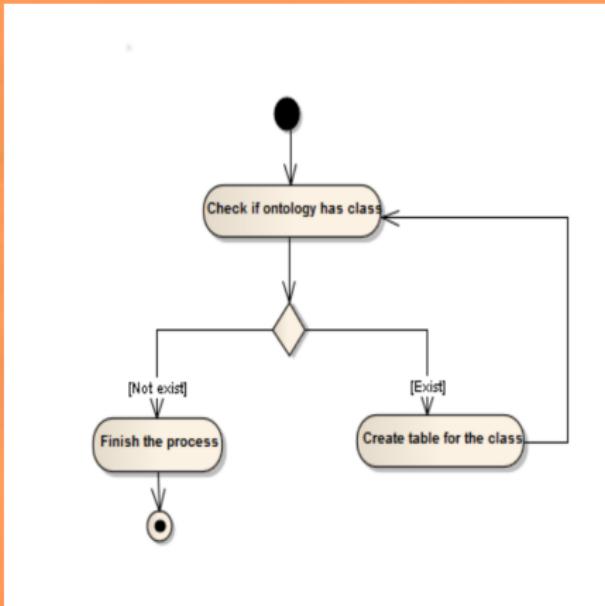
Object property: Son relaciones entre instancias de 2 clases. Por ejemplo, *ownedBy*, podría ser una object type property con la clase *Vehículo* como dominio y la clase *Persona* como rango.

Data property: Son como las object properties pero tienen como dominio una clase y rango un valor literal. Por ejemplo, si para una persona yo quisiera darle el nombre completo, entonces defino la data property *fullName* con dominio en *Person* y rango *string*.

Individuals: Representan los objetos o instancias de las clases que tenemos definidas en nuestra ontología.

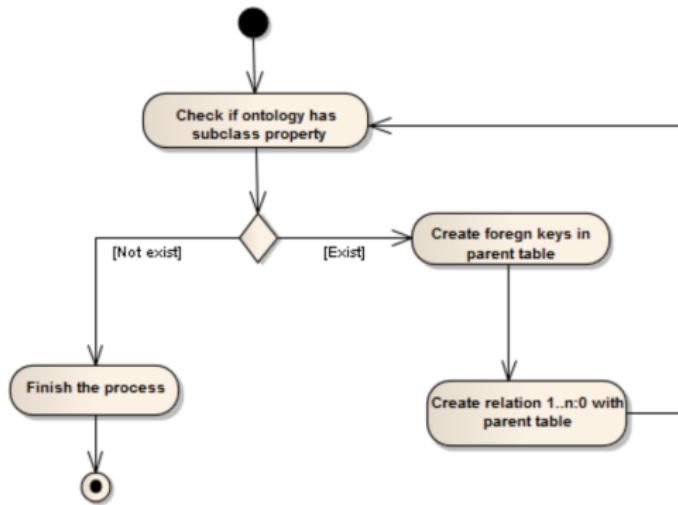
Pasos a seguir para generar el DW

- Creación de tablas a partir de las clases.



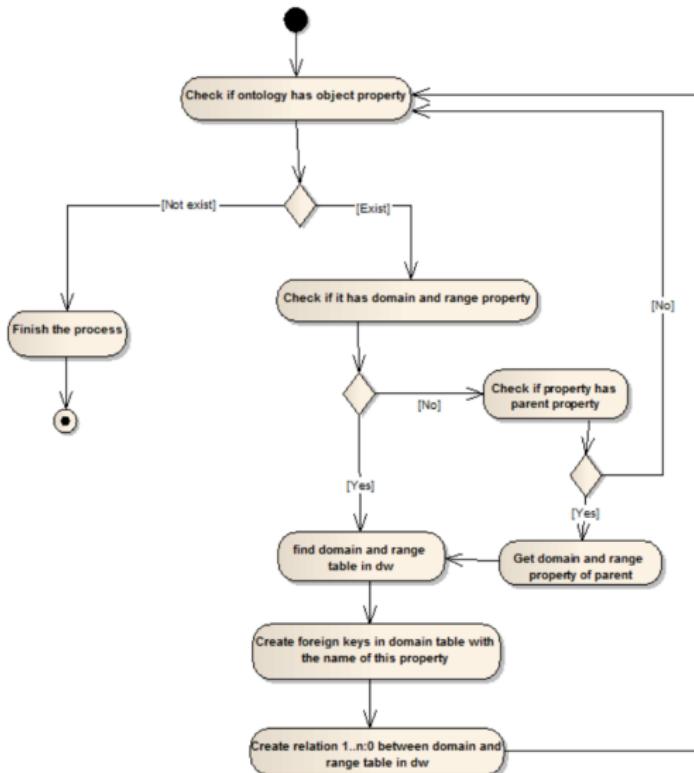
Pasos a seguir para generar el DW

- Creación de tablas a partir de las clases.
- Relaciones de subclase.



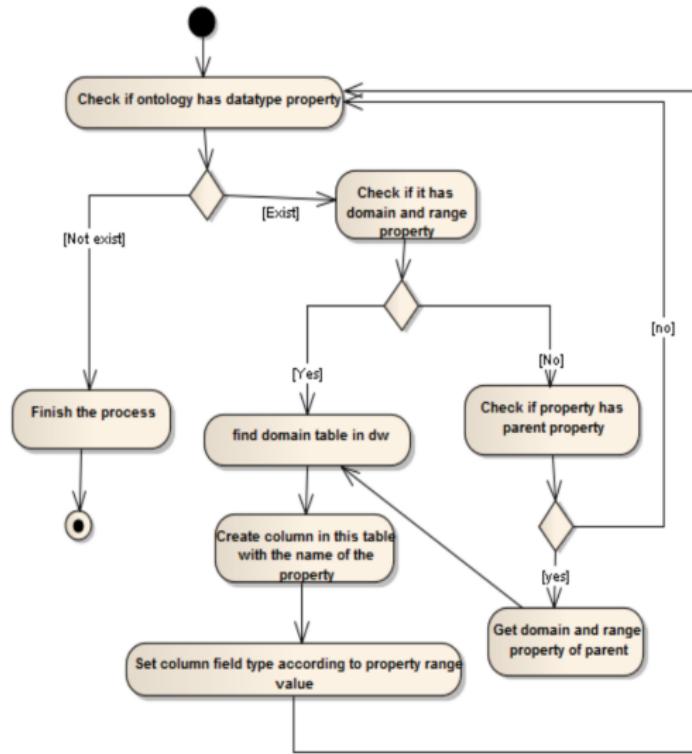
Pasos a seguir para generar el DW

- Creación de tablas a partir de las clases.
- Relaciones de subclase.
- Definiendo relaciones de propiedad de objetos



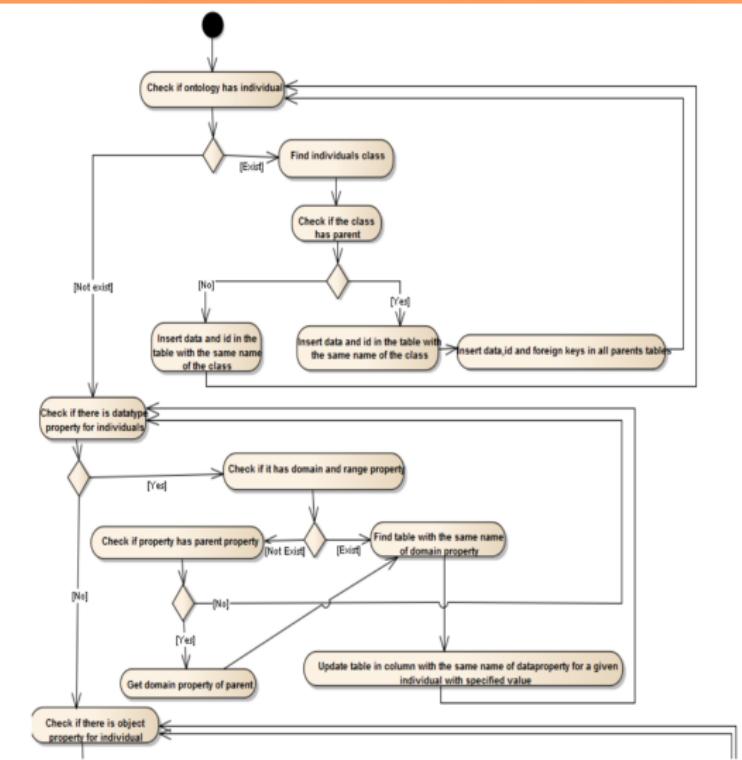
Pasos a seguir para generar el DW

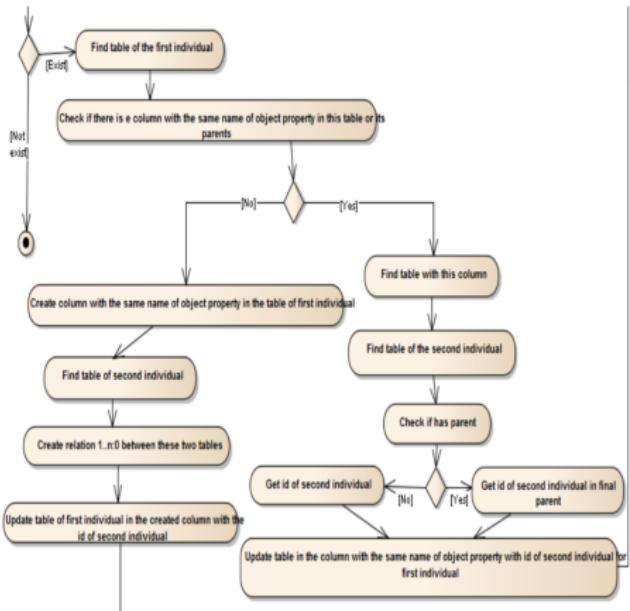
- Creación de tablas a partir de las clases.
- Relaciones de subclase.
- Definiendo relaciones de propiedad de objetos
- Definiendo relaciones de propiedad de datos



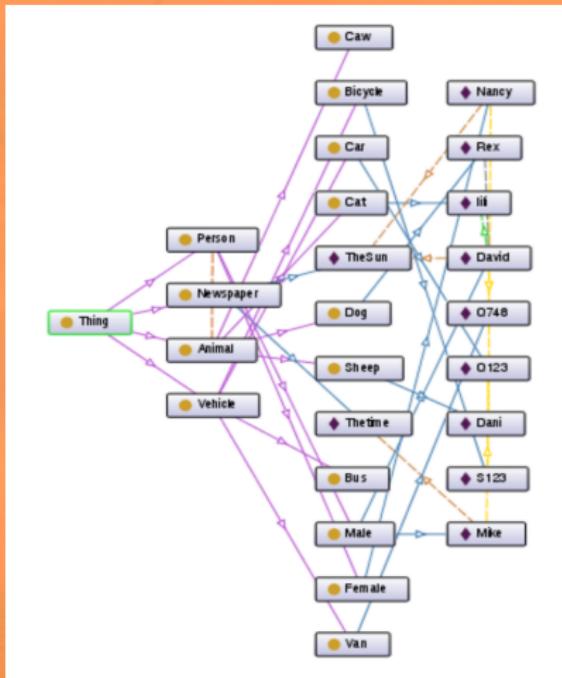
Pasos a seguir para generar el DW

- Creación de tablas a partir de las clases.
- Relaciones de subclase.
- Definiendo relaciones de propiedad de objetos
- Definiendo relaciones de propiedad de datos
- Insertando datos en las tablas

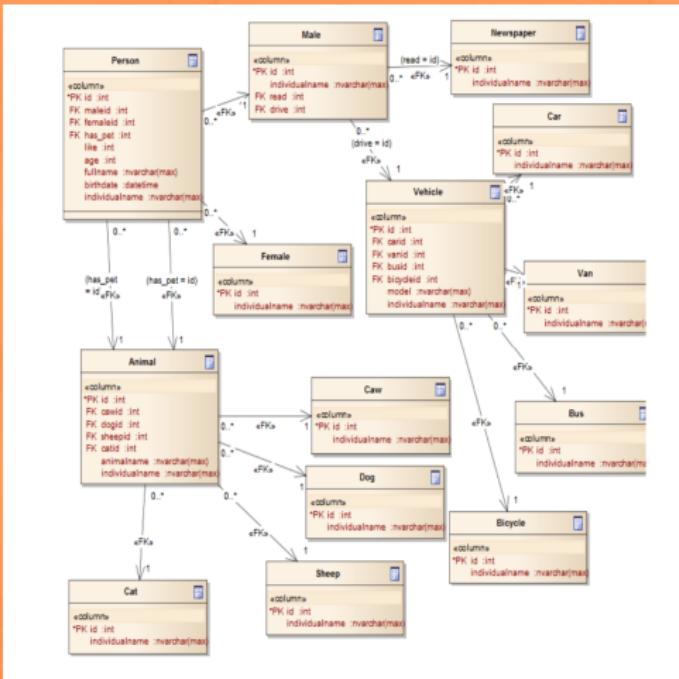




Tomando la siguiente ontología



Obtenemos





FUZZY DATA WAREHOUSE

Breve repaso de lógica difusa

Conjunto difuso. Un conjunto difuso A en M , también llamado universo del discurso, es caracterizado por una función $\mu_A(m)$, que asocia a cada elemento en M , un número en el intervalo $[0, 1]$. Los números en el intervalo $[0, 1]$ definen la pertenencia de un elemento m al conjunto difuso A , donde 1 implica plena pertenencia, y 0 implica ninguna pertenencia. Un conjunto difuso A en M , puede ser representado como un conjunto de tuplas $\{(m, \mu_A(m))\}$.

Una **variable lingüística** es una quintupla, $(X, T(X), G, M, F)$, definida como sigue:

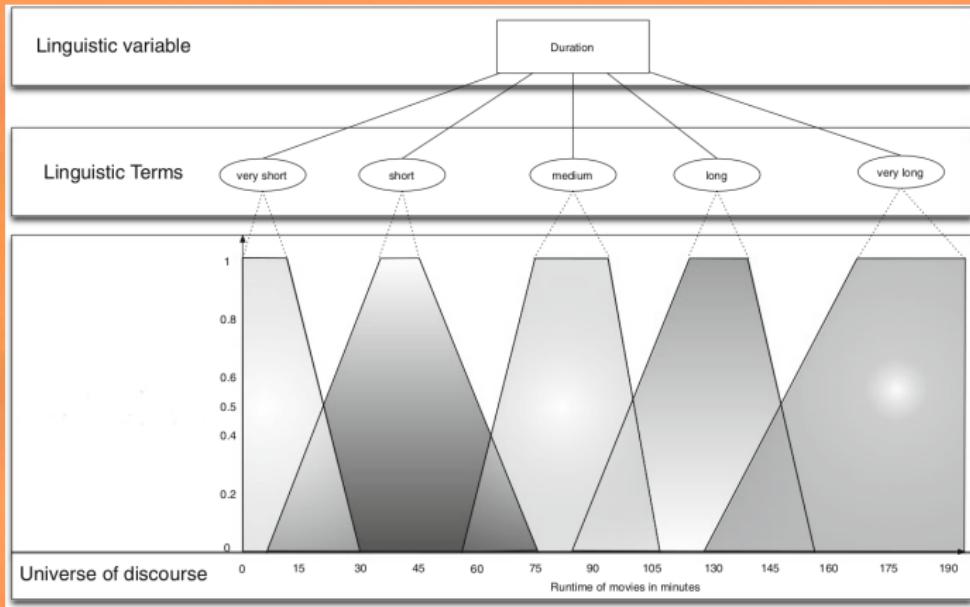
- X es el nombre de la variable lingüística.
- $T(X)$, es el conjunto de términos lingüísticos de X .
- G es regla sintáctica que genera el conjunto de términos lingüísticos.
- M es el universo de discurso.
- F es una regla semántica que define para cada término lingüístico, su significado en función de un subconjunto borroso en M .

Ejemplo

Consideramos un conjuntos de películas, donde cada película tiene una duración específica. Podemos definir una variable lingüística *duration*, y podemos dividirla en un conjunto de términos lingüísticos $\{ \text{very short}, \text{short}, \text{medium}, \text{long}, \text{very long} \}$.

Difinimos la variable ligüística *duration*, como sigue:

$(X = \text{duration}, T(X) = \{ \text{very short}, \text{short}, \text{medium}, \text{long}, \text{very long} \}, G, M = [0, 200], F)$.



Representación gráfica de la variable lingüística *duration*

¿Qué es un Fuzzy Data Warehouse (FDW)?

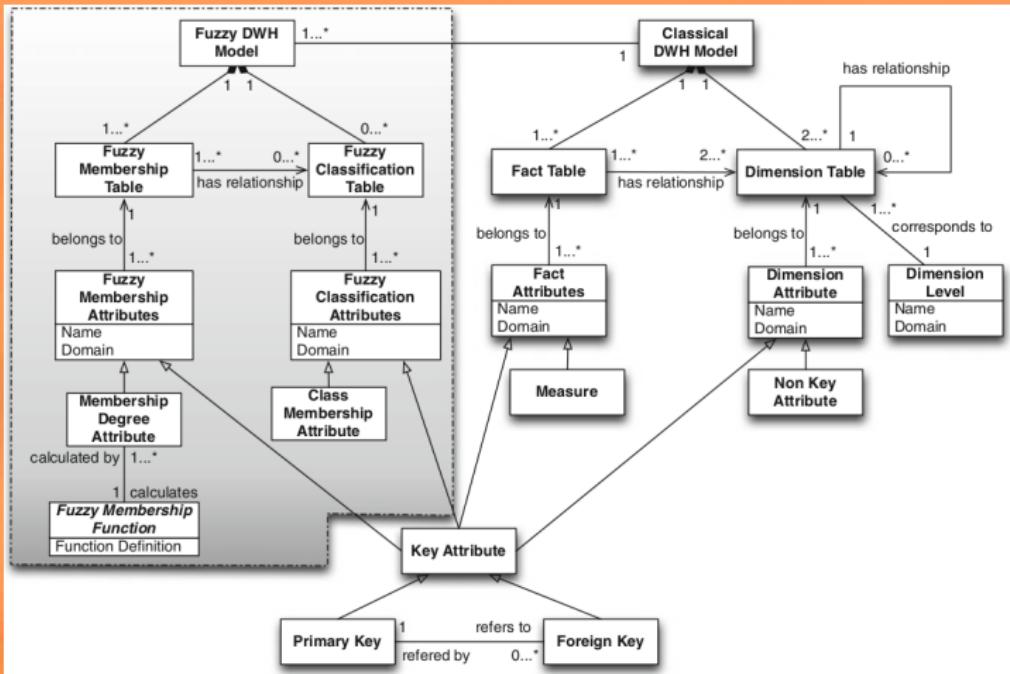
FDW es un *data warehouse*, que contiene datos difusos y permite el procesamiento difuso de los datos. Al incorporar conceptos difusos, los sistemas de *data warehouse* tienen la posibilidad de procesar datos a mayor nivel de abstracción y mejorar el análisis de los datos imprecisos.

Conceptos básicos del **Modelo Fuzzy Data Warehouse**, propuesto en [2]:

- El **atributo objetivo**, lo notamos TA , es un atributo, de una dimensión o hecho, que quiere ser clasificado como difuso.
- El **atributo de pertenencia a una clase**, para un TA , es un atributo que tiene un conjunto de términos lingüísticos T_1, T_2, \dots, T_k . Lo notamos CMA_{TA} .

- La **función de pertenencia**, para un término lingüístico t en CMA_{TA} , $\mu_t : TA \rightarrow [0, 1]$, es utilizada para calcular los grados de pertenencia de los valores en TA, al término t .
- El **atributo grado de pertenencia**, de un TA, lo notamos MDA_{TA} , tiene un conjunto de grados de pertenencia del TA, para términos lingüísticos en CMA_{TA} .

- La **tabla de clasificación fuzzy**, FCT , es un tabla con dos atributos, los términos lingüísticos y sus identificadores. Es decir, $FCT_{TA} = \{ID, CMA_{TA}\}$.
- La **tabla de pertenencia fuzzy**, FMT , es una tabla que almacena el grado, en el que los valores del TA, están relacionados con los términos lingüísticos. Es decir, $FMT_{TA} = \{ID, ID\ de\ TA, ID\ de\ CMA_{TA}, MDA_{TA}\}$.
- Un **modelo Fuzzy Data Warehouse**, es un conjunto de tablas, $FDW = \{Dim, Fact, FCT_{TA}, FMT_{TA}\}$.



El meta modelo Fuzzy Data Warehouse

Ejemplo

En [1] se propone un Fuzzy Data Warehouse para un sistema de análisis web, donde se demuestra que las medidas no siempre son fáciles de interpretar, pues no es posible contar las visitas de distintos usuarios con precisión. Entonces, el concepto difuso sobre las visitas a páginas web proporciona una clasificación más certera de los valores numéricos.

Beneficios

- Los datos pueden ser analizados en forma difusa y no difusa.
- El modelo puede ser utilizado para la creación nuevos DWs, o aplicado a uno ya existente.

Desventajas

- Se basa en un enfoque relacional.
- La estructura de las meta tablas no considera la posibilidad de tener diferentes versiones de datos difusos, en función del tiempo.



WEB WAREHOUSE

Motivación

Un **Web Warehouse** es un DW, en el cual la información es obtenida de las posibles fuentes de datos de la web y, por la naturaleza de la web, se debe tener muy en cuenta la calidad de los datos. Como se menciona en [3] la idea de Web Warehouse es algo similar a la de un DW, ya que en su arquitectura ya son bastante parecidos.



ARQUITECTURA

Módulos de la arquitectura

- **Módulo DSi:**

- Proveer mecanismos para extraer datos de los diferentes WDS
- Monitorear los DS y controlar periódicamente la calidad de los datos
- Proveer mecanismos para reaccionar ante la degradación de los DS

Módulos de la arquitectura

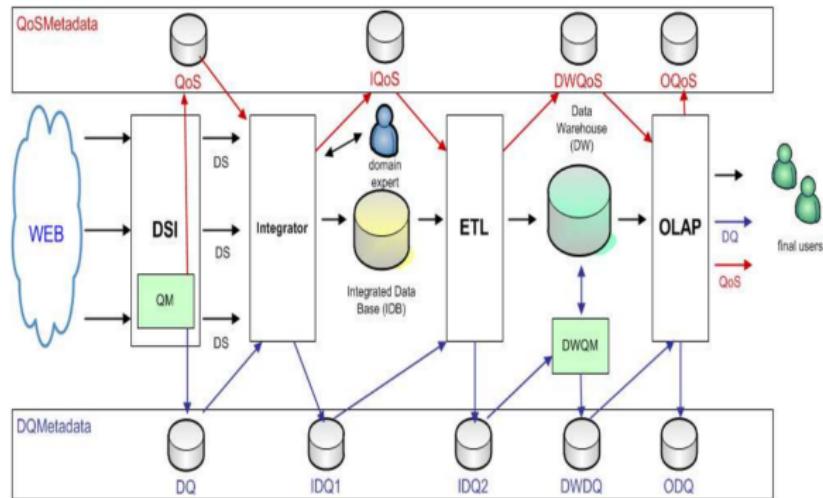
- **Módulo integrador:** Lleva a cabo la integración de los datos y provee los mismos en formato normalizado, construyendo con ellos la IDB

Módulos de la arquitectura

- **Módulos ETL y OLAP:** Realizan las mismas funciones que las que conocemos y, además, tienen la tarea de propagar los metadatos

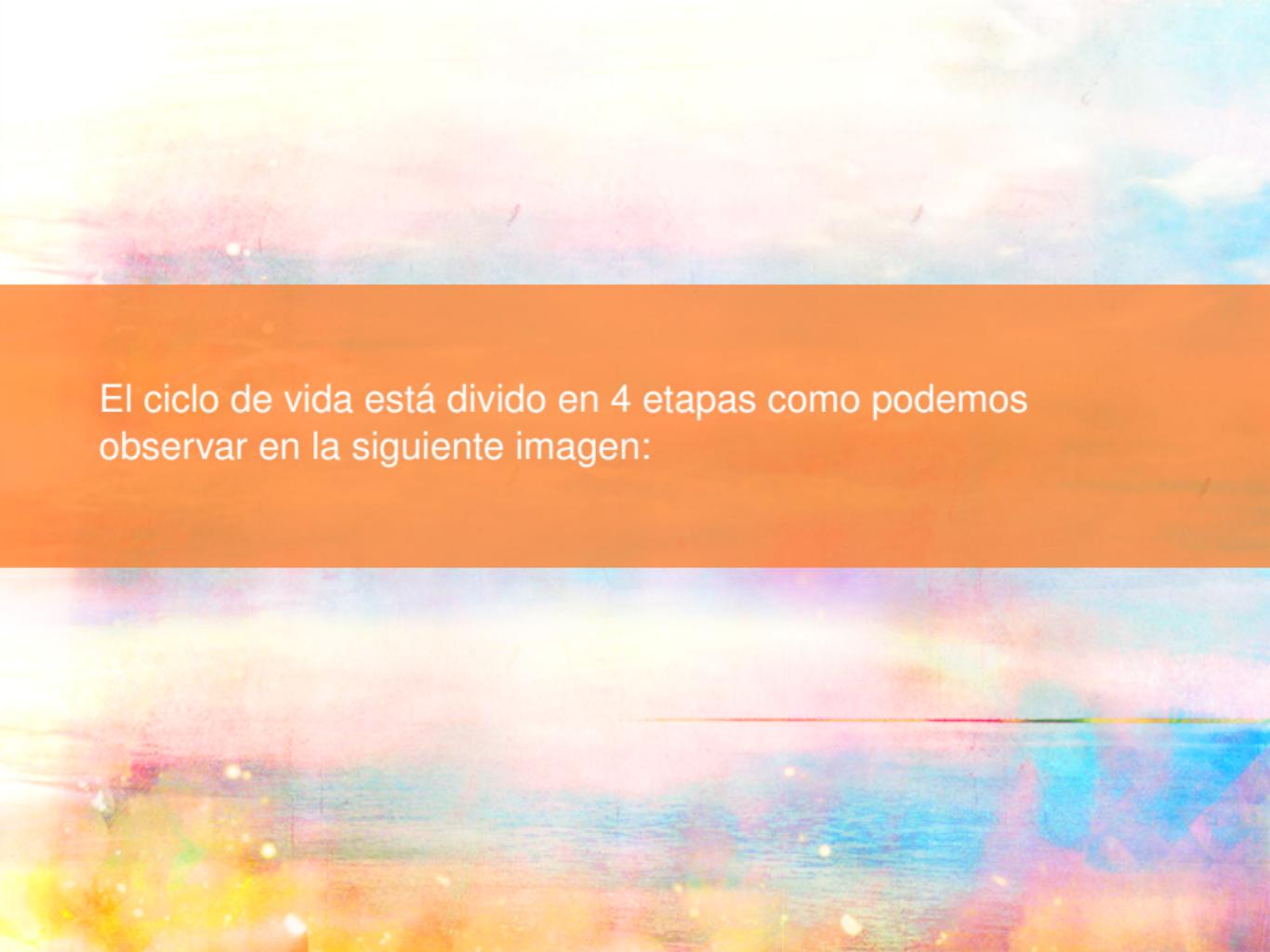
Módulos de la arquitectura

- **Módulo DWQM:** Debe controlar la calidad de los datos del DW

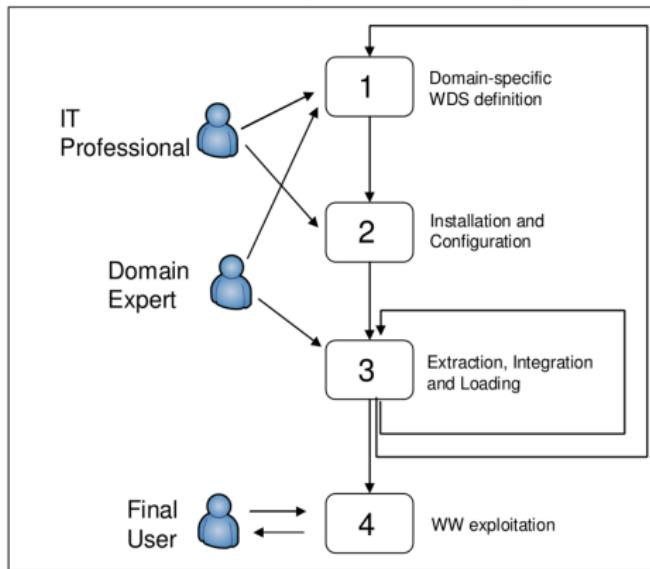




CICLO DE VIDA DE UN WW



El ciclo de vida está dividido en 4 etapas como podemos observar en la siguiente imagen:





CALIDAD DE DATOS

Como venimos mencionando, la calidad de los datos es uno de los puntos más importantes en la arquitectura de un WW. Para esto planteamos 6 factores que intentan englobar todos los aspectos referentes a la calidad de los datos

- Precisión

- Precisión
- Completitud

- Precisión
- Completitud
- Frescura

- Precisión
- Completitud
- Frescura
- Consistencia

- Precisión
- Completitud
- Frescura
- Consistencia
- Unidad

- Precisión
- Completitud
- Frescura
- Consistencia
- Unicidad
- Confiabilidad



MONDRIAN OLAP SERVER

Pentaho es un conjunto de programas libres para generar inteligencia de negocios (Business Intelligence).

Incluye herramientas para generar informes, minería de datos, ETL, OLAP, etc, ...

Uno de sus productos es **Pentaho Analysis Services**, con nombre clave **Mondrian**, es un servidor de procesamiento analítico relacional en línea (ROLAP), open-source; escrito en java. Soporta MDX, como lenguaje de consulta.

MDX (MultiDimensional eXpressions) es un lenguaje para definir y consultar las bases de datos OLAP; funciona sobre cubos de datos, dimensiones, jerarquías y miembros. MDX ha sido adoptado por una amplia mayoría de proveedores de herramientas OLAP, y se ha convertido en el estándar para los sistemas OLAP.

A continuación, mostraremos la herramienta OLAP, **Saiku Community**, la cual nos provee de una interfaz gráfica, para analizar los datos, de manera sencilla e intuitiva.

Otros servidores OLAP (algunos):

- Microsoft Analysis Services
- MicroStrategy Intelligence Server
- SAS OLAP Server

Bibliografía:

- [1] D. Fasel and D. Zumstein. "A fuzzy data warehouse approach for web analytics". In: pp. 276–285.
- [2] Daniel Fasel. *Fuzzy Data Warehousing for Performance Measurement: Concept and Implementation (Fuzzy Management Methods)*. Heidelberg: Springer, 2014.
- [3] Adriana Marotta, Laura González, and Raúl Ruggia. "A quality aware service-oriented web warehouse platform". In: *Proceedings of the 2012 Joint EDBT/ICDT Workshops*. 2012, 29–32.
- [4] Shiva Talebzadeh, Mir Ali Seyyedi, and Afshin Salajegheh. "Automated Creating a Data Warehouse from Unstructured Semantic Data". In: 2014.
- [5] Alejandro Vaisman and Esteban Zimányi. *Data Warehouse Systems: Design and Implementation*. Heidelberg: Springer, 2014.