DISEÑO DE DATOS DE UN DATA WAREHOUSE

DESIGN OF INFORMATION OF A DATA WAREHOUSE

Aguilar Mallea Freddy Wilder[1], Pilco Romero Wilder Franz[1]

[1]Departamento de Sistemas y Ciencias Exactas, Facultad de Ciencias Integradas de Bermejo
Universidad Autónoma Juan Misael Saracho.

Dirección para correspondencia: Freddy Wilder Aguilar Mallea, Departamento de Sistemas y Ciencias Exactas, Universidad Autónoma Juan Misael Saracho, Av. Bolívar Esq. Argentina, Bermejo, Tarija, Bolivia.
Correo electrónico: faguilar@uajms.edu.bo

RESUMEN

En los últimos años, el área de Data Warehouse (DW) y Aplicaciones Olap ha tenido un desarrollo importante. En este tipo de aplicaciones, se construye una base de datos con visión multidimensional de la realidad.

El presente artículo tiene como objetivo proponer un procedimiento del diseño de datos para la mejora de la implementación y construcción de un data warehouse a través de diferentes sistemas transaccionales que existen en la Universidad Juan Misael Saracho.

El diseño de datos se divide en tres etapas secuenciales: En el diseño conceptual se construye un esquema conceptual de la realidad a partir de los requerimientos y/o bases fuentes, enriquecido con requerimientos de rendimiento y almacenamiento. Se proponen estructuras de datos y un mecanismo de especificación de restricciones de integridad. En el diseño lógico, se genera un esquema lógico, que es una especificación más detallada que el esquema conceptual donde se modelan los datos según un modelo lógico de DBMS, definiéndose estructuras más cercanas al nivel de almacenamiento. Se tiene en cuenta estrategias para resolver los requerimientos de performance y almacenamiento. También se toma en cuenta la información de las bases de datos fuentes. En el diseño físico se implementa el esquema lógico en el manejador de bases de datos elegido, teniendo en cuenta técnicas de comprensión, índices, selección de vistas e índices, optimización de consultas, carga y mantenimiento, los cuales se hace necesario para acelerar los tiempos de respuesta de las consultas complejas. Finalmente se realiza un prototipo del diseño de datos de un data warehouse para un Sistema de Gestión Académica de la Universidad con sus respectivas etapas.

Palabras clave: Data warehouse, modelo multidimensional, olap, olpt, cubo.

ABSTRACT

In recent years, the area of Data Warehouse (DW) and OLAP applications has been an important development. In this type of application, building a database with multi-dimensional view of reality.

This article aims to provide a process design data to improve the implementation and construction of a data warehouse across different transactional systems that exist in Juan Misael Saracho University.

Data design has three sequential steps: In the conceptual design builds a conceptual scheme of reality from the requirements and / or foundation sources enriched performance and storage requirements. Data structures are proposed mechanism and specification of integrity constraints. In the logical design, logical schema is generated, which is a more detailed specification that the conceptual schema where data are modeled as a logical model of DBMS, defining structures closest to the storage level. You consider strategies to address performance and storage requirements. It also takes into account information from the source databases. In the physical design is implemented in the logical schema database handler chosen, taking into account compression techniques, indexes, views and indexes selection, query optimization, and maintenance burden, which is necessary to accelerate the time of complex query response. Finally, a prototype design of a data warehouse data for Academic Management System of the University with their respective stages.

Keywords: Data warehouse, multidimensional model, olap, olpt, cube.

INTRODUCCION

Debido a la gran cantidad de sistemas de información operacionales, se está dando el efecto que existe una gran cantidad de datos en las empresas, instituciones y organismos en general, datos que se multiplican diariamente. Sin embargo, contrario a lo que pudiera creerse, esta explosión de datos no está significando un aumento en el conocimiento. Las empresas (pequeñas, medianas y grandes) producen una gran cantidad de datos que no son capaces

de transformar en información. Esto nos está llevando a que cada día mientras más datos se tienen, menos información está disponible para el análisis.
Para superar estos problemas en los últimos años se han desarrollado una serie de técnicas y herramientas que facilitan el análisis de la información como son los data warehouse (almacén de datos), data marts (mercado de datos) y herramientas OLAP (Procesamiento Analítico en Línea, On Line Analytical Processing). Sin embargo, a pesar de haberse difundido bastante, estos productos no se basan en un modelo de datos bien definido ni ampliamente aceptado sino que existe un conjunto de funcionalidades y operaciones que todos proveen, y otras que son las que diferencia un producto de otro.

En los últimos años, han surgido varios modelos de diversa índole que permiten la descripción conceptual y/o lógica de los aspectos multidimensionales de una realidad determinada. Sin embargo, en estos modelos no se han propuesto en forma explícita lenguajes de restricciones que permitan la realización de especificaciones precisas. Por otro lado, algunas metodologías y modelos asumen que la implementación del Data Warehouse, se realizará en un manejador de base de datos relacional. Esta postura deja de lado a los manejadores de bases de datos multidimensionales, sin considerarlos siquiera como una alternativa posible.

Dentro del desarrollo de estos sistemas y específicamente en el modelado de datos existen varias propuestas metodológicas entre las más conocidas están:

a) Ralph Kimball, con un esquema centrado en la identificación de los procesos de la empresa, como elemento clave para la definición de la estructura de variables y dimensiones;

b) W.H. Inmon, con un esquema que parte de la construcción del modelo de datos corporativos, elaborado al más alto nivel de abstracción, para luego derivar la estructura del modelo de datos, para el diseño del almacén;

c) Golfarelli Matteo, Maio Dario, Rizzi Stefano proponen un esquema que parte de los modelos ER descriptivos de los sistemas transaccionales de la organización, para luego derivar el modelo E-R de la estructura, para el almacén de datos.

Pero en la actualidad no existe una propuesta metodológica universalmente válida y aceptada como tal, por la comunidad académica, que realice el modelado de datos de un data warehouse que abarque todas sus etapas e integre los distintos modelos utilizados en cada una de ellas de una forma coherente. Es así que en la comunidad universitaria de "Juan Misael Saracho" tampoco existe un propuesta metodológica para implementar un data warehouse partiendo de los diversos sistemas de información automatizados (Gestión Académica Pregrado – Posgrado, Recursos Humanos) los cuales son del tipo transaccional (OLTP) y no satisfacen las necesidades de información para un nivel ejecutivo o gerencial.

El objetivo general es proponer un procedimiento para el modelado de datos para mejorar la implementación y construcción de un data warehouse a través de diferentes sistemas transaccionales, tomando como caso de estudio el sistema de gestión académica de la Universidad Juan Misael Saracho. Los objetivos específicos son: analizar las distintas etapas y técnicas que existen en el modelado de datos, analizar las distintas metodologías de modelado de datos existentes en la actualidad e implementar un prototipo de modelado de datos para un caso de estudio de un sistema transaccional.

El objeto de estudio son las distintas metodologías y técnicas de el modelado de datos de un data warehouse

Se plantea la siguiente hipótesis: "La propuesta de un procedimiento para el modelado de datos de un data warehouse permitirá disponer de una herramienta para el desarrollo de sistemas de información para la toma de decisiones dentro de nuestro entorno universitario".

En este trabajo de investigación, proponemos un procedimiento de modelado de datos que integra todas las fases de diseño (conceptual, lógico y físico) de los almacenes de datos desde las fuentes de datos operacionales hasta la implementación y el propio esquema del almacén de datos así también los requerimientos de los usuarios. Con esta propuesta se podrá disponer de una herramienta para el desarrollo de sistemas de información para la toma de decisiones no solo en el entorno universitario sino también en el entorno de la sociedad en su conjunto y de sus instituciones ya que no existen sistemas de este tipo que puedan ser de ayuda en la toma de decisiones.

En este trabajo el Modelo Conceptual Dimensional, propone estructuras de datos y un mecanismo de especificación de restricciones de integridad. Con este mecanismo se pueden especificar conjuntos de cubos con características comunes, permitiendo la especificación de un conjunto mayor de opciones de análisis que simplemente las que se den una a una. Dado que el modelo está orientado al diseño conceptual, no se propone en forma explícita un lenguaje de consulta.

Una de las tareas más importantes en la construcción de un DW es la construcción de su esquema lógico. El esquema lógico es una especificación más detallada que el esquema conceptual donde se incorporan nociones de almacenamiento, performance y estructuración de los datos.

En el caso de diseño de DWs se toma en cuenta un componente adicional: las bases de datos fuentes. Un DW se

construye con información extraída de un cierto conjunto de bases de datos fuentes.

Durante el diseño lógico se consideran dichas bases y cómo se corresponden con el esquema conceptual. Por lo tanto es esencial poder relacionar los elementos del esquema conceptual con las tablas y atributos de las bases fuentes. Se consideran de gran importancia la existencia de técnicas para construir un esquema relacional de DW a partir de un esquema conceptual multidimensional.

De los trabajos existentes en diseño de DWs algunos proponen construir el esquema lógico a partir de requerimientos sin realizar un diseño conceptual, mientras que otros proponen la realización de un esquema conceptual y a partir de éste generar un esquema lógico basado en un tipo particular de diseño (generalmente estrella).

Estas carencias afectan la calidad del esquema resultado así como la productividad en el diseño. En la medida de que no hay una buena conexión entre la especificación conceptual y el esquema lógico diseñado pueden generarse diferencias respecto a la información que representan. El problema es mayor si no se especifica un esquema conceptual contra el cual validar el esquema lógico. También se pierde productividad en el desarrollo, ya que no se puede reutilizar el trabajo de análisis que normalmente corresponde a la etapa de diseño conceptual.

En el diseño físico se implementa el esquema lógico en el manejador de bases de datos elegido, teniendo en cuenta técnicas de optimización física, como son: índices, particiones, etc.

MATERIALES Y METODOS

Estrategia Metodológica

Con el fin de poder identificar y describir fenómenos y situaciones del objeto de estudio que es el modelado de datos de un data warehouse (DW), el tipo de investigación es el descriptivo. Este estudio descriptivo permite indicar como son y se manifiestan los fenómenos sometidos a análisis y poder determinar los rasgos de este objeto de estudio.

Entre los métodos teóricos a utilizar en este trabajo de investigación se tiene:
Análisis y síntesis, como un proceso mediante el cual se analizan y relacionan los hechos observados de manera aislada en situaciones concretas. Un análisis de las distintas metodologías existentes en nuestro entorno.

En el análisis documental se realizará una revisión bibliográfica a nivel nacional e internacional sobre el objeto de estudio; el modelado de datos de un DW.

Entre otros métodos teóricos a utilizar en este trabajo de investigación tenemos: Deductivo, porque los resultados obtenidos por otros estudios referidos al tema del modelado de datos del DW sirven para analizar situaciones particulares que se observan en nuestro contexto. Inductivo, permite obtener conclusiones generales a partir de situaciones particulares, de diferentes autores que realizaron el estudio del modelado de datos que permitirá tener un concepto global sobre esta metodología de diseño de datos.

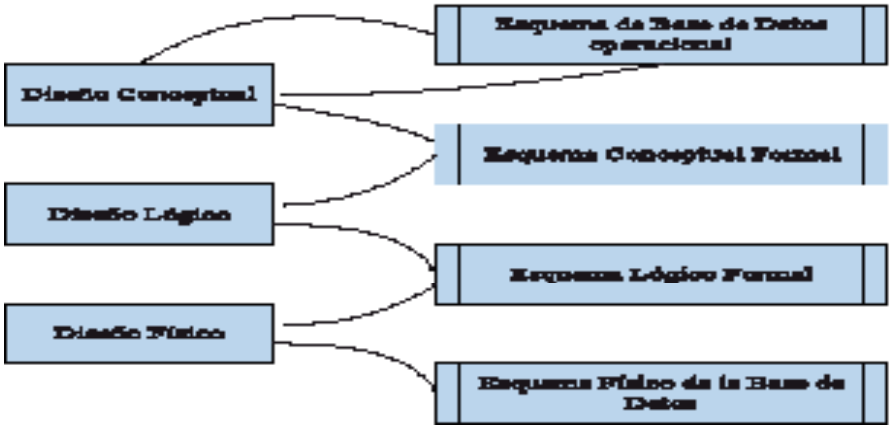
Finalmente se aplica la metodología propuesta de diseño de datos de un DW para la construcción de un prototipo

RESULTADOS

1 Diseño del Data Warehouse

Se presenta la propuesta del procedimiento de modelado de datos que incluye la descripción de tres etapas del diseño del DW, los cuales son: Diseño Conceptual, Diseño Lógico y Diseño Físico.

Figura 1. Proceso de diseño de un Data Warehouse



1.1 Diseño Conceptual

1.1.1 Estructuras en el modelo conceptual dimensional

El objetivo fundamental es permitir la especificación de una determinada realidad en términos multidimensionales. Para lograr esto, el modelo conceptual dimensional presenta tres estructuras básicas:

- Niveles
- Dimensiones
- Relaciones Dimensionales

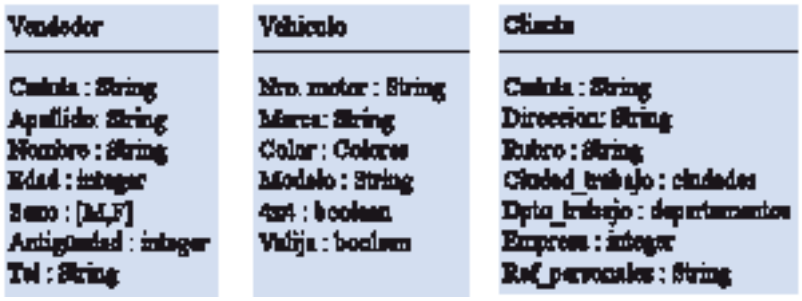
Niveles

Un nivel representa un conjunto de objetos que son de un mismo tipo. Cada nivel debe tener un nombre y un tipo. Conceptualmente, no tiene diferencia con cualquier elemento del Modelo Entidad Relación que pueda ser considerado un conjunto de Entidades.

Para representar el esquema de un nivel se utiliza un rectángulo que contiene el nombre y la estructura (o el nombre) del tipo de ese nivel.

En la figura se pueden ver tres niveles. El primero representa un conjunto de vendedores de los que se conoce un determinado número de atributos. Análogamente, el segundo representa un conjunto de vehículos y el tercero un conjunto de clientes.

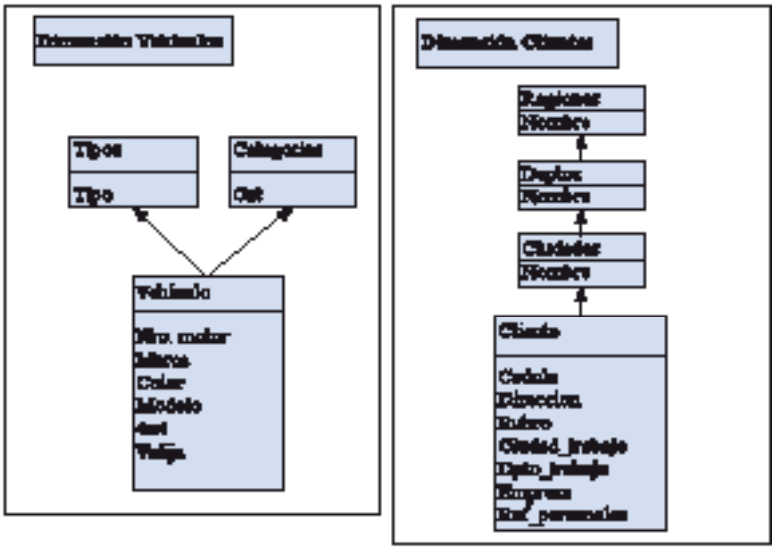
Figura 2. Tres niveles: Un nivel debe tener un nombre y un tipo



Dimensiones

Una dimensión está determinada por una jerarquía de niveles. La instancia es una jerarquía de elementos de esos niveles. De esta forma, el esquema de una dimensión está representado por un rectángulo dentro de él cual aparece un nombre para la dimensión y un grafo dirigido en donde los nodos son los niveles que participan de esa dimensión (Fig.3).

Figura 3. Dimensiones Clientes y Vehículos



Relaciones Dimensionales

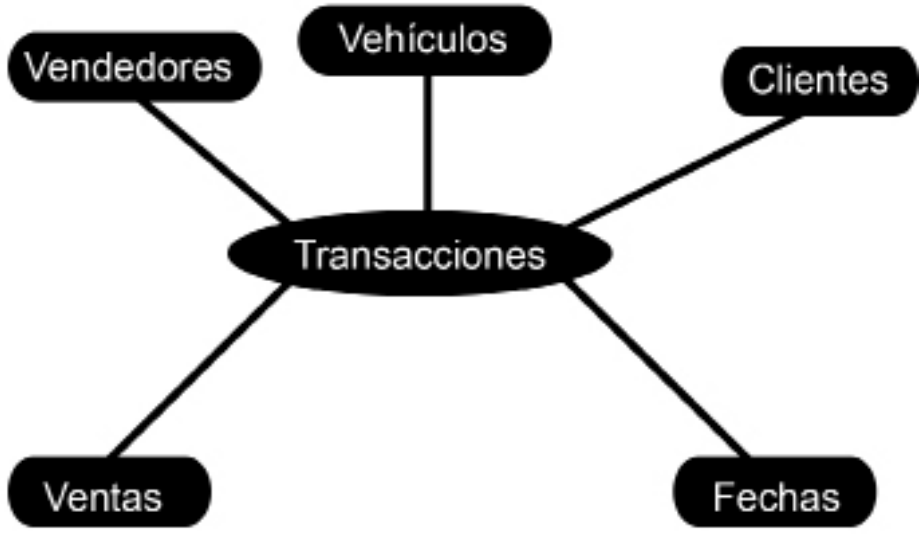
Una relación dimensional representa el conjunto de todos los cubos que se pueden construir a partir de los niveles de un conjunto dado de dimensiones. La Fig. 4 se puede interpretar como una instancia de una relación dimensional. Se asume que en cada uno de los cubos que pertenecen a la instancia de la relación dimensional, debe aparecer al menos un nivel de cada una de las dimensiones que participan en la relación.

En el modelo conceptual dimensional, un cubo es una función que va del producto cartesiano de las instancias de los niveles en los booleanos. De esta forma, cualquier nivel puede cumplir el rol de medida. Por lo tanto, el esquema de una relación dimensional está dado por un grafo en forma de estrella. El nodo central es de forma oval y tiene el nombre de la relación dimensional y los nodos "satélite" son rectangulares y tienen el nombre de cada una de las dimensiones que participan de la relación (Fig. 4).

1.1.2 Especificación del modelo

Se presenta un lenguaje de especificación para facilitar la lectura de la propia formalización del modelo y luego se presentan los esquemas de cada una de las estructuras.

Figura 4. Relación Dimensional para Representar las Transacciones (Ventas)



El lenguaje de especificación utilizado para especificar tanto esquemas como instancias pretende ser intuitivo para quien maneje los conceptos básicos sobre conjuntos y lenguajes formales. El lenguaje, hace uso intensivo de conjuntos definidos por comprensión y por extensión. Se asume como predicados la pertenencia de elementos a conjuntos, la igualdad y comparación de conjuntos (inclusión amplia y estricta) y también de elementos. De esta forma, la siguiente expresión:

{x / x Є Pares } -> { y / Ǝ(k Є Impares).y =xk }

representa el conjunto de las funciones que mapean un número par en alguna de sus potencias impares. Observar que el símbolo → se usa tanto para denotar el conjunto de las funciones en un contexto de conjuntos como la implicación en un contexto de fórmulas lógicas.

1.1.3 Restricciones de integridad en el modelo conceptual dimensional

En el Modelo Conceptual Dimensional se define un lenguaje general sobre el cual se pueden definir restricciones y también encapsularlas mediante macros. De esta forma, para extender el modelo con restricciones de integridad se necesita:

- Definir el lenguaje de restricciones.
- Modificar las estructuras de datos para que soporten las restricciones sobre ellas.

El lenguaje de restricciones que se propone es un lenguaje lógico de alto orden, en donde se permite la especificación de conjuntos por comprensión y extensión y los cuantificadores son siempre relativizados. El lenguaje es similar al lenguaje usado en la especificación del modelo, sólo que se utilizan las expresiones lógicas como lenguaje base. También se propone hacer un uso intensivo de macros para mejorar la legibilidad y la escritura de las restricciones.
El lenguaje gráfico, se podrá extender con símbolos

adecuados para la representación de ciertas restricciones que serán usadas en casi cualquier realidad.

Si el modelo que se define contiene un conjunto de restricciones inconsistente, entonces cualquier intento de cargar datos en un Data Warehouse que implemente ese modelo conceptual respetando las restricciones de integridad, será infructuoso y los programas de carga siempre desecharán todos los datos. Por ejemplo, en el nivel vendedor de la dimensión vendedores, es claro que no pueden existir diferentes vendedores con la misma cédula de identidad, la restricción se puede escribir como:

$$\forall v_1 \in Vendedores. \forall v_2 \in Vendedores. (v_1.cedula = v_2.cedula \rightarrow v_1 = v_2)$$

1.2 Diseño Lógico

El proceso de diseño lógico propuesto puede dividirse en dos grandes etapas: En la primera etapa se definen los lineamientos y se establecen los mapeos a la base fuente. Esta etapa tiene una alta participación del diseñador, incorporando información semántica a través de propiedades y vínculos entre el esquema intermedio y la fuente. La tarea del diseñador no incluye ningún tipo de procesamiento ni transformación de los esquemas, sólo la definición de propiedades que éstos deben cumplir.

En la segunda etapa se lleva a cabo la generación del esquema lógico del DW a través de transformaciones aplicadas a la fuente. Se utilizan las definiciones realizadas en la primer etapa para elegir qué transformaciones aplicar al esquema de la base fuente y en qué orden hacerlo. Este proceso es automatizable y no requiere la participación del diseñador. La figura 5 ilustra el proceso.

1.2.1 Definición de Lineamientos y Mapeos

1.2.1.1 Lineamientos de Diseño

Los lineamientos son información de diseño lógico que complementan al esquema conceptual y permiten al diseñador dar pautas sobre el esquema lógico deseado para el DW.

A través de los lineamientos, el diseñador define el estilo de diseño para el DW (snowflakes, estrella, mixto) e indica requerimientos de performance y almacenamiento (por ejemplo indicando que cubos implementar), etc.
A continuación se enuncian los lineamientos propuestos:

Materialización de Relaciones

En el Modelo Conceptual Dimensional, una relación dimensional representa un espacio de cubos resultante de cruzar niveles de las dimensiones. Dicho espacio de cubos puede restringirse mediante las restricciones de integridad del propio modelo.

Las restricciones se construyen en base a predicados con cuantificadores $(\forall, \exists, \neg)$ para indicar que "todos los cubos deben tener", o "debe existir un cubo que tenga" o "ningún cubo debe tener" dicha estructura. Por ejemplo: debe existir un cubo que cruce el nivel mes con el nivel producto.

Estas restricciones sugieren qué cubos sería interesante tener y cuáles no deberían existir. Sin embargo, la decisión de cuáles de esos cubos se deben materializar debe ser tomada en un momento posterior. En este contexto, materializar un cubo corresponde a precalcular los valores para los cruzamientos de las dimensiones y almacenarlos en una tabla. Luego se pueden obtener otros cruzamientos mediante operaciones efectuadas sobre éstos. (Fig.6)

Fragmentación Vertical de Dimensiones

El diseñador puede indicar el grado de normalización que quiere lograr al generar estructuras relacionales para cada dimensión. Por ejemplo, le puede interesar un esquema estrella, es decir, de normalizar todas las dimensiones y mantener fact tables. Por el contrario, le puede interesar un esquema de snowflakes, normalizando todas las dimensiones.

También le puede interesar tratar diferente cada dimensión, indicando para cada una si normaliza, denormaliza o efectúa una estrategia intermedia, indicando en este último caso, qué niveles quedan en la misma tabla.

Como lineamiento, el diseñador debe indicar para cada dimensión qué niveles desea almacenar juntos,
conformando una fragmentación de los niveles de la dimensión.

Fragmentación Horizontal de Cubos

Representar un cubo en el modelo relacional puede dar como resultado una o varias tablas (fact tables) dependiendo del grado de fragmentación que se quiera lograr.

Fragmentar horizontalmente una tabla relacional corresponde a construir varias tablas con la misma estructura, y dividir las instancias entre ellas. Con esto se logra almacenar juntas las tuplas que son consultadas juntas y tener tablas más pequeñas, lo cual resulta en un aumento de la performance en las consultas.

Como ejemplo se considera el cubo venta-1 de la Figura 6, cuyas consultas más frecuentes corresponden a las ventas posteriores al 2000. Se puede fragmentar el cubo en dos fragmentos, uno para almacenar las tuplas correspondientes a ventas posteriores a ene-2000 y otro para las tuplas correspondientes a meses anteriores. Las tuplas de cada fragmento deben cumplir, respectivamente:

Figura 5. Proceso de diseño lógico

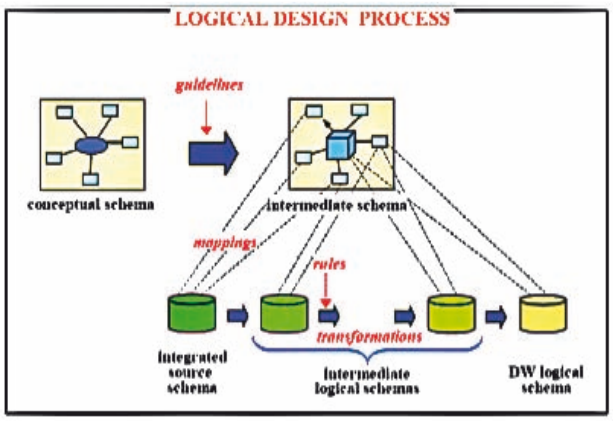


Figura 6. Materialización de relaciones (Cubos

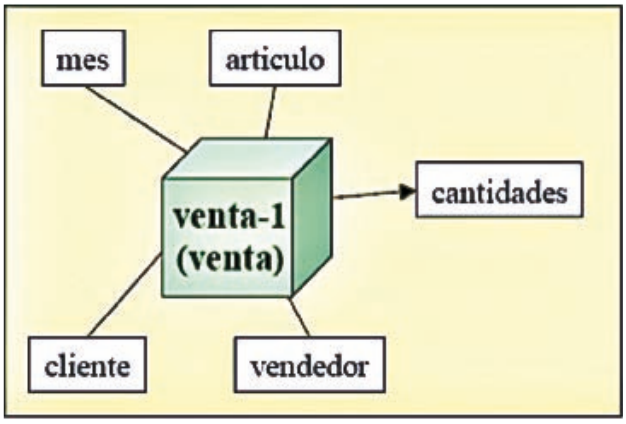


Figura 7. Fragmentación vertical de dimensiones

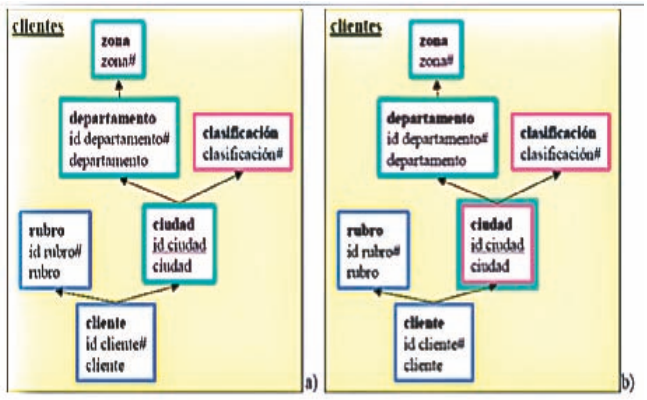
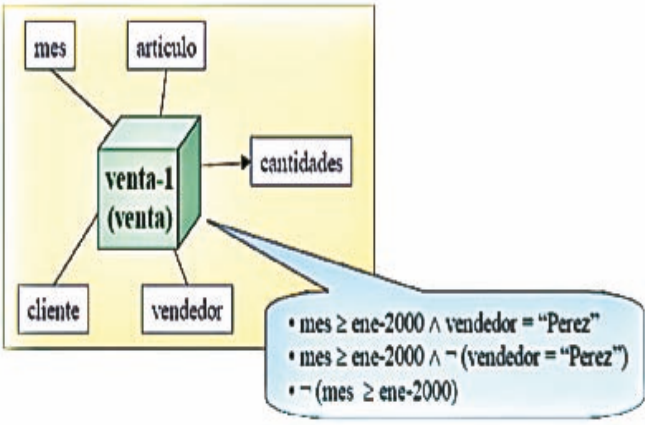


Figura 8. Fragmentación horizontal de cubos



- mes ≥ ene-2000
- mes < ene-2000

Definición de Lineamientos

Para definir los lineamientos el diseñador usualmente se basa en las restricciones de performance y almacenamiento de su sistema.

1.2.1.2 Esquema Intermedio

Al especificar los lineamientos el diseñador introduce elementos que complementan el esquema conceptual. En concreto, especifica qué cubos van a materializar las relaciones dimensionales y cómo se van a fragmentar las dimensiones y los cubos.

El esquema intermedio consiste de los ítems, niveles y dimensiones y restricciones del esquema conceptual, e incorpora los cubos, fragmentación de dimensiones y fragmentación de cubos definidos en los lineamientos.

1.2.1.3 Especificación de la Base de Datos Fuente

Se considera que la base de datos fuentes es una base relacional integrada, de la cual interesa representar las tablas, sus atributos (con sus tipos) y su clave primaria.

Dadas dos tablas, interesa reflejar cómo se vinculan, es decir, cómo se debe realizar el join entre ellas. A un vínculo de este tipo se le llama link. Cada link tiene asociado un predicado construido con los atributos de ambas tablas. Los links son relaciones simétricas.(Fig.9)

1.2.1.4 Mapeos entre el Esquema Intermedio y la Base de Datos Fuente

El esquema conceptual especifica la información que contendrá el DW, y a través de los lineamientos el diseñador indica las características que debe cumplir el esquema lógico. El esquema conceptual con el agregado de los lineamientos conforma el esquema intermedio.

Luego de construir el esquema intermedio, el siguiente paso es vincularlo con la base fuente. Para ello se establecen mapeos o correspondencias (mappings) que indican dónde se encuentran en el esquema lógico de la fuente los diferentes elementos del esquema intermedio. (Fig.10)

1.2.2 Generación del Esquema Lógico del DW

A partir del esquema intermedio, la base fuente y los mapeos, se construye el esquema lógico del DW mediante la aplicación de transformaciones sucesivas sobre esquemas relacionales, partiendo del esquema de la base de datos fuente. En este trabajo se propone un algoritmo y un conjunto de reglas para construir el esquema lógico en forma automática.

Figura 9. Links entre tablas fuentes

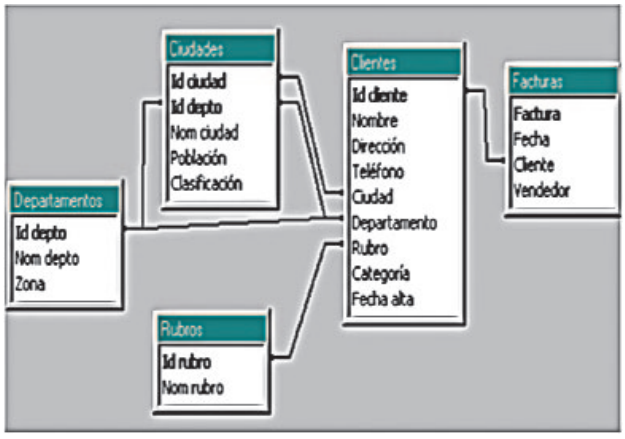
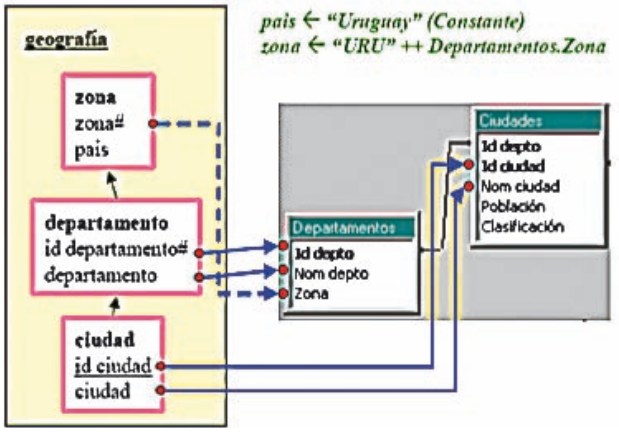


Figura 10. Representación gráfica de una función de mapeo



Reglas de Diseño

Se presenta un mecanismo para construir el esquema lógico del DW partiendo del esquema intermedio, la base fuente y las correspondencias entre ellos. El objetivo es automatizar el proceso de diseño teniendo en cuenta las estrategias de diseño especificadas mediante los lineamientos. Se propone un algoritmo para diseñar el esquema lógico del DW basado en la aplicación de reglas de diseño. Dichas reglas se aplican a objetos del esquema intermedio (fragmentos de dimensiones, cubos, etc.) que cumplen determinadas condiciones y dan como resultado la aplicación de primitivas al esquema lógico del DW.

El objetivo de las reglas es determinar cuándo puede aplicarse una determinada transformación, según si los objetos del esquema intermedio cumplen una serie de premisas o condiciones. Las reglas utilizan y modifican las tablas y las funciones de mapeo asociadas. Entre estas reglas se encuentra:

	Regla	Condiciones de aplicación	Descripción
R1	Join	Un fragmento o cubo mapea a más de una tabla. Dos de esas tablas están relacionadas (tienen definido un link).	Combina las dos tablas generando una nueva tabla.

Especificación de un Algoritmo de Generación del Esquema Relacional

Se presenta un algoritmo para construir el esquema lógico de un DW. Dicho algoritmo propone un orden de aplicación para las reglas descritas en la sección anterior.

El algoritmo se divide en dos partes: (1) construcción de las tablas de dimensión y (2) construcción de las tablas de hechos. La última parte, a su vez, se divide en tres sub-partes: (a) construcción de tablas de hechos para cubos con mapeo base, (b) construcción de tablas de hechos para cubos con mapeo recursivo, y (c) construcción de tablas de hechos para franjas de cubos.

PARTE 1: Para construir las tablas de dimensión se ejecutan los siguientes pasos:

Step 1: Construir los esqueletos.
Step 2: Renombrar atributos para ítems con mapeo directo.

Step 3: Generar atributos para ítems con mapeo calculado o externo.
Step 4: Aplicar filtros.
Step 5: Eliminar atributos no mapeados.
Step 6: Ajustar las claves.

PARTE 2:
a) Para construir las tablas de hechos para cubos con mapeo base se ejecutan los siguientes pasos:

Step 7: Construir los esqueletos.
Step 8: Renombrar atributos para ítems con mapeo directo.
Step 9: Generar atributos para ítems con mapeo calculado o externo.
Step 10: Aplicar filtros.
Step 11: Eliminar atributos no mapeados.
Step 12: Ajustar las claves.

b) Para construir las tablas de hechos para cubos con mapeo recursivo se ejecutan los siguientes pasos:

Step 13: Construir las tablas de las jerarquías.
Step 14: Aplicar los drill-ups.

c) Para construir las tablas de hechos (para las franjas de todos los cubos) se ejecuta un único paso:

Step 15: Construir las tablas de las franjas

1.3 Diseño Físico

Los sistemas data warehouse requiere la ejecución de operaciones costosas, como por ejemplo joins y aggregations. Esta situación se hace más compleja por el hecho de que las consultas OLAP deben realizarse sobre estructuras que tienen potencialmente millones de registros y porque los resultados tienen que ser entregados interactivamente al analista de negocios que opera el sistema. Dadas estas características, el énfasis en el ambiente OLAP está en el procesamiento eficiente de consultas.

Dentro del diseño físico de data warehouse se debe considerar las siguientes áreas de investigación:

- Modelo de Datos: En esta área se estudian las estructuras de datos utilizadas en almacenamiento físico, así como los mecanismos de aplicación de estas según los requerimientos generales del sistema OLAP.

- Compresión: Dado el gran volumen de información manejado y los altos grados de dispersión involucrados, debido a que no todos los posibles cruzamientos de dimensiones determinan valores para las medidas, muchas de las estructuras estudiadas no optimizan el espacio de almacenamiento. Por esta razón se han desarrollado varios métodos de compresión que permiten consultar los datos sin descomprimirlos.

  Se presenta FCompress, una técnica de compresión de información que mantiene la propiedad de consulta sin necesidad de descomprimir los datos. Esta técnica es aplicable a los datos de las tablas de medidas, sumarización y cubos. Está diseñado para integrarse al ambiente del Data Warehouse y se basa en el reemplazo de los datos originales por valores aproximados de los atributos, almacenando bitcodes compactos.

- Índices: Muchos de los índices ya existentes para sistemas OLTP son revisados y modificados para sistemas OLAP. La selección óptima de índices, se basa en el esquema lógico y en la carga de trabajo y requiere que se tomen en cuenta estructuras específicas de acceso. Los pasos para agregar índices van de técnicas simples, como índices secundarios a estructuras complejas. En general se piensa que el mejor subconjunto de índices es el que reduce al mínimo el costo de acceso. Si bien en la mayoría de los casos cualquier estructura de índice resuelve los requerimientos de performance, el problema es el costo asociado. Cuando el diseñador tiene la posibilidad de poner unos o más índices en una relación para mejorar ciertas consultas, la ventaja de materializar una vista puede ser afectada por el espacio que, invariablemente, dicho índice va a utilizar. Además del costo en espacio se agrega el tiempo de creación, administración de datos y overhead de algunas operaciones.

- Selección de vistas e índices: Una de las principales actividades en la etapa de diseño es la selección del conjunto adecuado de vistas e índices a materializar para obtener el mejor desempeño, respetando las restricciones de espacios y tiempos de mantenimiento. Este problema es conocido como el problema de selección de vistas e índices, View and Index Selection Problem (VIS Problem).

  Los sistemas OLAP están orientados principalmente a operaciones de consultas sobre grandes volúmenes de datos. Los dos principales factores que influyen en el incremento de la performance de este tipo de sistemas son, el almacenamiento de los resultados precalculados, vistas y vistas intermedias, y la utilización de estructuras apropiadas tanto de datos como de índices.

  Existen varias estrategias de diseño que apuntan a mejorar la performance de este tipo de sistemas. La trivial y con la que se obtiene el mejor rendimiento es la materialización de todas las vistas, incluyendo las intermedias, junto a varias estructuras de índices. Esto implica diseñar una vista para cada consulta más el conjunto de índices asociado. Esta estrategia requiere mucho espacio de almacenamiento, mucho tiempo de carga del sistema y muchos recursos de administración y gestión de las estructuras de datos e índices.

- Caché: Cuando no se conoce la frecuencia de ocurrencia

de las consultas no es posible seleccionar un conjunto de vistas a materializar. Por lo tanto, se requieren técnicas de materialización dinámicas basadas en las condiciones de uso del OLAP. En estos casos, el caché es la mejor respuesta a esta problemática.
La materialización de vistas mejora notoriamente el desempeño de los sistemas de DW. La dificultad está en determinar la frecuencia y el costo de las consultas. A priori, el diseñador puede suponer o estimar la frecuencia de ciertas consultas. Sin embargo estos datos pueden cambiar a lo largo de la vida del DW. Lo ideal sería realizar la estimación del costo y frecuencia a medida que las consultas se realicen y el sistema evolucione. De esta forma surgió la idea de utilizar caché de resultados en sistemas de DW.

En el caché se almacenan los resultados de las consultas para que estas sean utilizadas por otras consultas. Un ejemplo sería el de un usuario que realiza una consulta por las ventas en las distintas tiendas de una determinada ciudad y que luego de estudiar estas medidas, trate de comparar las ventas en la misma ciudad contra las ventas del resto de las ciudades en una zona.

- Optimización de Consultas: Una vez aplicados los métodos de diseño anteriores, es necesario realizar cambios a las consultas realizadas a los sistemas DW. De esta manera, se podrán aprovechar los beneficios brindados por los métodos vistos, principalmente la materialización de vistas y el caché de respuestas.

  Si bien la materialización de vistas o el uso de sistemas de caché debe ser transparente para los usuarios del Data Warehouse, es posible que deban realizarse cambios para obtener el máximo provecho de la estrategia seleccionada. Esto implica un proceso de cambio de la consulta original conocido como query rewritting. En particular, este problema se reduce a la aplicación serial de dos problemas: en primer lugar, estudiar cuales son los resultados de vistas que total o parcialmente pueden ser utilizados; en segundo lugar, minimizar la consulta es decir, eliminar todos aquellos elementos de la nueva query que son redundantes, por ejemplo atributos semánticamente equivalentes que se repiten. La combinación de ambos problemas se conoce con el nombre de query containment.

- Fragmentación y Distribución: Tal como se estudia en el diseño de bases de datos para sistemas OLTP, muchas veces es necesario distribuir los datos entre distintos sitios para mejorar la performance y la disponibilidad.

Dado el gran volumen de información, la fragmentación tanto horizontal como vertical permite obtener respuestas más veloces a las consultas que se realizan sobre el DW. Es pertinente saber en qué casos es conveniente una distribución de datos entre más de un nodo en una red, Al igual que en las bases de datos distribuidas, se puede pensar en distribuir los distintos fragmentos en varios nodos. Esto permitiría explotar el principio de localidad de datos así como la posibilidad de emplear técnicas de paralelismo

Distribución

Una vez realizadas las fragmentaciones se debe elegir la ubicación de los fragmentos. Por lo tanto se establece la distribución de los fragmentos de la tabla de medidas según las consultas que se ejecuten en los distintos lugares y que dan origen a los minterms que sirven de input al algoritmo. Sin embargo no se establece si esta distribución se debe realizar siguiendo un patrón estadístico respecto a la ocurrencia de las consultas en los distintos sitios. Las relaciones de dimensiones se replican en todos los sitios. Esto se basa en la baja frecuencia de renovación de las dimensiones y la necesidad de acceso a los datos en todos los sitios que acceden al Data Warehouse.

- Carga y Mantenimiento: Dado el gran volumen de información involucrado, la complejidad de las estructuras de datos y el extensivo uso de índices, es necesario contar con mecanismos de carga y mantenimiento eficientes.

- Teniendo las sentencias que definen las vistas a materializar, es directa la manera de realizar la carga en el DW. Se ejecutan las distintas sentencias para las relaciones de sumarización o vistas intermedias, en el orden establecido por las precedencias establecidas en el algoritmo, por ejemplo a través del lattice. Una vez ejecutado el proceso, se tiene el DW. Con los datos materializados en las distintas vistas. Este proceso suele realizarse antes de que el Data Warehouse quede disponible a las consultas realizadas por los usuarios o herramientas OLAP.

Para el mantenimiento de vistas lo más común es que el contenido del Data Warehouse sea totalmente recalculado en cada proceso de carga a partir de los datos de fuente. Todas las técnicas se basan en tomar los datos de las fuentes, sean estos la totalidad o la porción de datos nuevos, e insertarlos en las relaciones que implementan el o los cubos multidimensionales.

2. Prototipo

Se presenta un caso de estudio sobre un Sistema de Gestión Académica de la Universidad, la misma que realiza los procesos de matriculación, convalidaciones, programación de materias y llenado de notas de los estudiantes.

De los alumnos se conoce el documento de identidad, apellido paterno, apellido materno, nombres, sexo, fecha de nacimiento, país, departamento, provincia, dirección, teléfono y colegio de egreso. Se tiene información sobre los Materias que cursan los alumnos de la universidad, los docentes que dictan las cátedras, los diferentes planes de

estudios de los programas y sus facultades.

De las notas o calificaciones se conoce la materia, el alumno, el docente, la calificación, el grupo, la gestión y el periodo. De la matriculas se conoce el alumno, la gestión, el periodo, la fecha de registro, el usuario que registra la venta.

2.1 Diseño Conceptual

Se siguen ciertas líneas para construir el diseño multidimensional de la realidad descripta en la sección anterior. Esas líneas son arbitrarias, no pretenden fijar una metodología específica de diseño.

Dimensiones

A partir de la descripción de la realidad que se presentó anteriormente, hay ciertas dimensiones que aparecen claras, aunque podría no estar clara la estructura interna de cada dimensión. En principio, se construye una dimensión para cada objeto que participa en el problema.

De esta forma, las dimensiones que se obtienen inicialmente son:

Relaciones Dimensionales

Las relaciones dimensionales representan el conjunto de cubos que se pueden construir tomando al menos un nivel de las dimensiones participantes.

De acuerdo con la realidad planteada, los cubos de interés tienen como dimensiones participantes a todas las dimensiones identificadas en el problema. Por esto, se puede representar una relación dimensional.

Restricciones

Se agrega restricciones en forma gráfica y en un lenguaje de alto orden. Se presentan las restricciones que parecen más obvias en cada tipo de restricción.

En el Modelo Conceptual dimensional es posible especificar condiciones sobre los elementos de cualquier conjunto, en particular, sobre los elementos de la instancia de un nivel. Por ejemplo, que la edad de todos los alumnos debe estar entre 17 y 90 años:

```
a.(a.edad>17 ^ a.edad<90)
```

Con este mecanismo se pueden escribir cualquier condición que sólo involucre elementos de un mismo nivel.

2.2 Diseño Lógico

2.2.1 Lineamientos

A continuación se presentan los lineamientos definidos para el ejemplo.

Materialización de Relaciones

Se elige materializar tres cubos para la relación dimensional Notas:

1- Con detalle de alumnos, materias, docentes y meses.
2- Con detalle de alumnos, programas, docentes y meses.
3- Con detalle de alumnos y meses.

La Fig.11 muestra la representación gráfica de los cubos.

Fragmentación de Cubos

Se decide fragmentar los cubos de la siguiente manera:

1- Una franja para las notas del año actual, y otra con el resto de la historia.

La Fig.12 muestra la representación gráfica de las franjas definidas.

Fragmentación de Dimensiones

Se decide seguir las siguientes estrategias de diseño para las dimensiones:

- Alumnos: 2 fragmentos, uno con alumno y persona, y el otro con los restantes.
- Materias: 2 fragmentos, uno con materia y programa y otro con facultad.
- Docentes: denormalizada.
- Fechas: denormalizada.

Figura 11. Cubos definidos

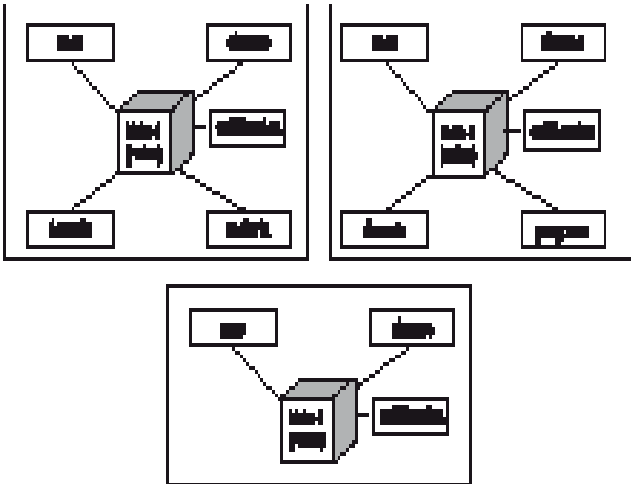
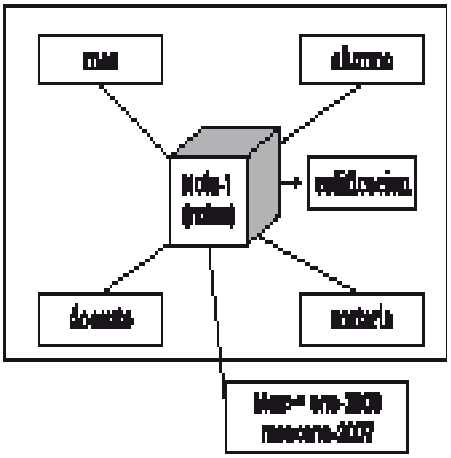


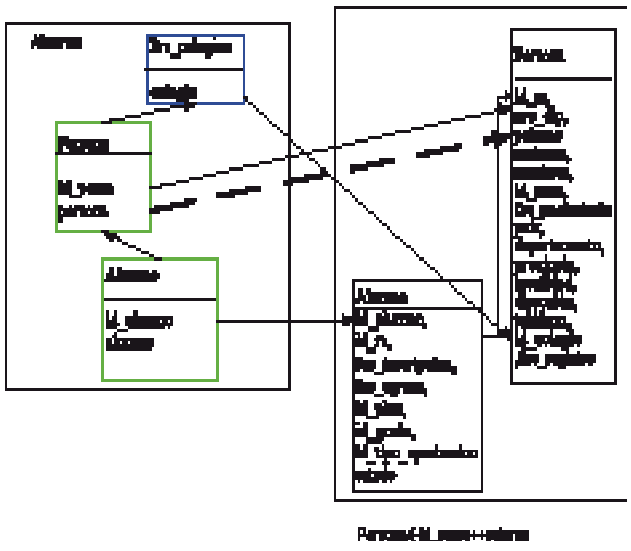
Figura 12. Franjas definidas



2.2.2 Mapeos

A continuación se presentan los mapeos definidos para los fragmentos de las dimensiones y para los cubos. En algunos casos se definen condiciones de mapeo.

Figura 13. Mapeo del fragmento verde de la dimensión alumnos



2.2.3 Aplicación Del Algoritmo

A continuación se muestra en el ejemplo la aplicación de los pasos del algoritmo, los parámetros utilizados y las tablas que se generan como resultado. Por cuestiones de espacio se omiten las funciones de mapeo.

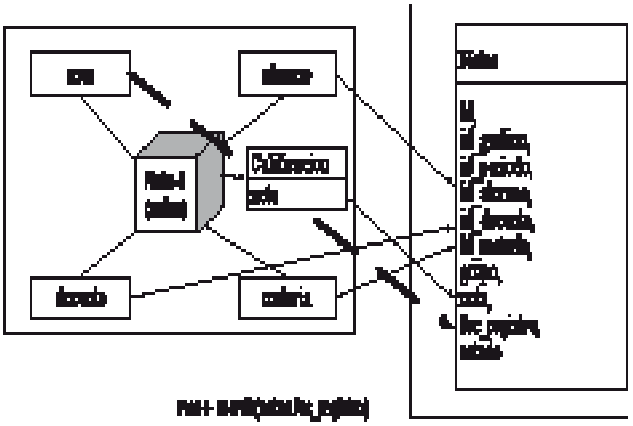
Construcción de Tablas de Dimensión

Para la construcción de las tablas de dimensión se aplican los pasos 1 a 6 para cada fragmento de dimensión. A continuación mostraremos el Paso 1:

Mapeos de Fragmentos. La Fig.13 muestra los mapeos de los fragmentos de la dimensión alumnos.

Mapeos de Cubos. La Fig.14 muestra el mapeo del cubo nota-1.

Figura 14. Mapeo del cubo nota-1



Step 1 - Construir El Esqueleto

El fragmento verde de la dimensión alumnos(se le llamará DwAlumnos) mapea a dos tablas fuentes: alumno y persona. Se itera aplicando la regla R1 (Join) de a dos tablas. Sea M = SchFMapping(DwAlumnos).Map, la función de mapeo del fragmento.

- Ejecutar Join (AlumnoPersona, M, Alumno, Persona).

Resultado DwAlumnos01 (Id_alumno, id_ra1, fec_inscripcion, fec_egreso, id_plan, id_grado, id_tipo_aprobacion,estado,

id_ra2, nro_dip, paterno, materno, nombres,id_sexo,fec_nacimiento,país,departamento,provincia,localidad,dirección,teléfono,id_colegio,fec_registro)

Construcción de Tablas de Hechos para Cubos con Mapeo Base
Para la construcción de las tablas de hechos se aplican los pasos 7 a 12 para cada cubo con mapeo base. En el ejemplo el único cubo con mapeo base es nota-1. A continuación mostraremos el Paso 7:

Step7 - Construir El Esqueleto

El cubo nota-1 mapea a una tabla fuente: Notas. Se aplica la regla R1 (Join). Sea M = SchCMapping(nota-1).
Map, la función de mapeo del cubo.

- Ejecutar Join (nota-1, M, Notas).

Resultado DwNota101 (id,id_gestion, id_periodo, id_alumno, id_docente, id_materia, grupo, nota1,fec_registro,estado,nota)

Construcción De Tablas De Hechos Para Franjas De Cubos

Para la construcción de las tablas de hechos se aplican los pasos 13 y 14 para cada cubo con mapeo recursivo. En el ejemplo el único cubo con bandas definidas es nota-1.
A continuación mostraremos el Paso 15:

Step 15 - Armar La Tabla De Cada Franja

El cubo nota-1 tiene definidas dos bandas: mes >= ene-2008 (a la que se llamará banda1) y mes <ene-2007 (a la que se llamará banda2). Se aplica la regla R8 (Filter) para cada banda. Sea M =

SchCMapping(nota-1). Map, la función de mapeo del cubo.

- Ejecutar Filter (nota-1, "DwNota104.mes >= ene-2008", M, DwNota104).

Resultado DwNota1Banda01 (id_alumno, id_docente, id_materia, nota, mes)

- Ejecutar Filter (nota-1, "DwNota104.mes < ene-2007", M, DwNota104).

Resultado DwNota1Banda02 (id_alumno, id_docente, id_materia, nota, mes)

Se tienen como resultados finales las tablas:

- DwNota1Banda01 (id_alumno, id_docente, id_materia, nota, mes)
- DwNota1Banda02 (id_alumno, id_docente, id_materia, nota, mes)

Se obtiene como resultado el esquema lógico que se muestra en la Figura 15.

DISCUSION

El presente trabajo de investigación que propone un procedimiento para el modelado de datos para mejorar la implementación y construcción de un data warehouse, podemos indicar que, se ha logrado realizar una fundamentación teórica sobre este objeto de estudio que es el modelado de datos de un DW, destacando los conceptos de diseño conceptual, lógico y físico extraídos de los diferentes casos de estudio analizados.

En base al análisis de las distintas etapas que existe en el modelado de datos de un DW, se logro definir un modelo de datos conceptual adecuado para construir especificaciones de bases de datos multidimensionales.

El modelo definido, presenta buenas características desde el punto de vista de los requerimientos para los modelos conceptuales:

Se presenta un proceso de diseño para construir el esquema lógico de un DW relacional tomando como entrada el esquema conceptual y una base de datos fuente integrada.

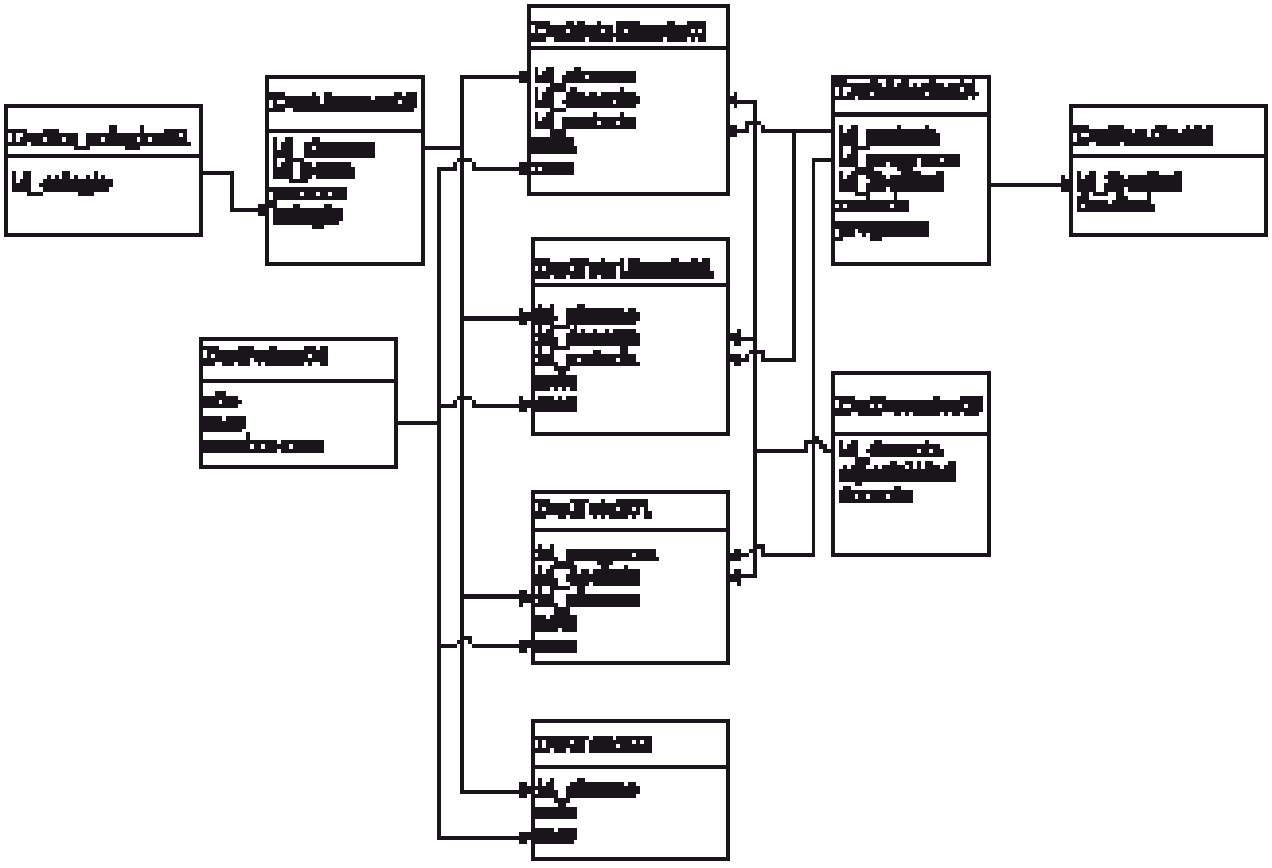
El proceso se divide en dos grandes etapas, la primera de definición de propiedades y correspondencias, es realizada por el diseñador, y la segunda de transformación del esquema fuente se realiza automáticamente.

En el diseño físico del data warehouse, el mismo debe ser tratado como un proceso global en donde los distintos componentes, que hacen al diseño físico, se conjuguen de manera que se puedan cumplir con los objetivos para los cuales el data warehouse está siendo creado, dentro de las áreas en que centra el trabajo de investigación esta: la resolución de los "VIS problem", estructuras eficientes de índices y el de mantenimiento del Data warehouse.

Se realizó la implementación de un prototipo de modelado de datos para un Sistema de Gestión Académica de la Universidad, la misma que realiza los procesos de matriculación, convalidaciones, programación de materias y llenado de notas de los estudiantes.

Finalmente se puede realizar la siguiente recomendación que consiste en realizar otros trabajos de investigación como ser el desarrollo de un sistema datawarehousing basado en la metodología de diseño de datos para un DW propuesta en este trabajo de investigación.

Figura 15. Esquema lógico del DW



BIBLIOGRAFIA

Agrawal, R., Gupta, A. y Sarawagi, S. (1997). Modelling Multidimensional Databases. UK.

Ballard, C., Herreman, D., Schau, D., Bell, R., Kim, E. y Valncic, A. (1998). Data Modeling Techniques for Data Warehousing. IBM Red Book. ISBN number 0738402451

Cabibbo, L. y Torlone, R. (1998). A Logical Approach to Multidimensional Databases. EDBT.

Codd, E.F, Codd, S.B. y Salley, C.T. (1993). Providing OLAP to user-analysts. An IT mandate .- Technical Report. E.F. Codd and Associates.

Golfarelli, M. y Rizzi, S. (1998). Methodological Framework for Data Warehouse Design. USA.

Gray, J., Bosworth, A., Layman, A. y Pirahesh, H. (1996) International Conference on Data Engineering. Data Cube: A relational aggregation operator generalizing group-by, cross-tab, and sub-totals.

Harinarayan, V., Rajaraman, A. y Ullman, J. (1996). International Conference on management data Implementation data cubes efficiently.

Inmon, William H. (1992). Building the Data Warehouse. Wiley-QED John Wiley & Sons Inc.

Kimball, R. (1996). The Datawarehouse Toolkit. John Wiley & Son Inc.

Moody, D. y Kortnik, M. (2000). From Enterprise Models to Dimensionals Models: A Methodology for Data Warehouse and Data Mart Design. Sweden.

Pressman Roger. (1993). Ingeniería del Software:Un Enfoque Práctico. McGraw-Hill Tercera Edición.