Implementación de herramientas CASE que asistan en el Diseño de Data Warehouses

Verónika Peralta, Raúl Ruggia

Universidad de la República, Uruguay.
{vperalta, ruggia}@fing.edu.uy

Resumen: Un Data Warehouse (DW) es una base de datos que almacena información para la toma de decisiones. Las características de los DWs hacen que los modelos de datos y estrategias de diseño sean diferentes a los utilizados para las bases de datos operacionales, requieren de nuevas técnicas y herramientas de diseño.

Este proyecto plantea la construcción de herramientas CASE que asistan al diseñador en la construcción de un DW relacional a partir de un esquema conceptual. En concreto se quiere permitir que el diseñador complemente el esquema conceptual con estrategias de diseño. También se quiere asistir al diseñador en el relacionamiento entre el esquema conceptual y la base fuente de donde obtendrá los datos.

Las herramientas CASE serán parte de un conjunto de herramientas de diseño de DW que conforman una plataforma de investigación y experimentación para trabajos de cooperación entre la Universidad de la República y la Universidad de Versailles Saint-Quentin-enYvelines (Francia).

1. Motivación

Un Data Warehouse (DW) es una base de datos que almacena información para la toma de decisiones. Dicha información es construida a partir de bases de datos que registran las transacciones de los negocios de las organizaciones (bases operacionales). El objetivo de los DWs es consolidar información proveniente de diferentes bases operacionales y hacerla disponible para la realización de análisis de datos de tipo gerencial.

La prioridad es el acceso interactivo e inmediato a información estratégica de un área de negocios. Las operaciones preponderantes no son las transacciones, como en las bases de datos operacionales, sino consultas que involucran gran cantidad de datos y agrupaciones de los mismos.

Las características de los DWs hacen que las estrategias de diseño para las bases de datos operacionales generalmente no sean aplicables para el diseño de DW ([Kim96], [Inm96]). Los modelos de datos para representar los datos almacenados en el DW también son diferentes.

A nivel conceptual resurgen los modelos multidimensionales ([Ken96], [Car00]), que representan la información como matrices multidimensionales o cuadros de múltiples entradas denominados *cubos*.

A los ejes de la matriz se los llama *dimensiones* y representan los criterios de análisis, y a los datos almacenados en la matriz se los llama *medidas* y representan los indicadores o valores a analizar.

La Figura 1 muestra un cubo con información de ventas de autos. Tiene dos dimensiones: *modelo* y *color*. La medida es *cantidad vendida*. Cada uno de los valores se interpreta como la cantidad de autos vendida de un modelo y color dados.

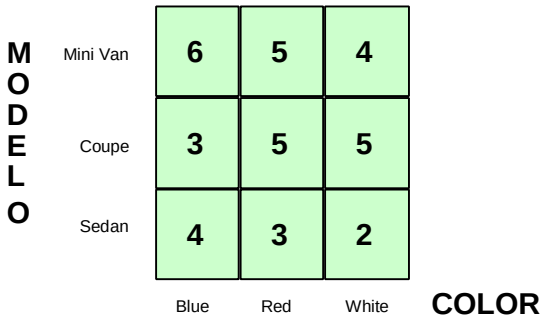

Figura 1 – Una matriz multidimensional (cubo)

Las dimensiones se estructuran en jerarquías con diferente nivel de detalle (niveles). La Figura 2a muestra la definición gráfica de la dimensión *clientes* en el modelo CMDM ([Car00]). El nombre de la dimensión está en la esquina superior izquierda del recuadro amarillo. Los cuadros de texto blanca son los niveles de la dimensión; sus nombres aparecen en negrita (*cliente, ciudad* y *departamento*). Debajo de los nombres de los niveles están los atributos que lo conforman(ítems). Las jerarquías de niveles se representan por flechas entre los niveles, del nivel con menos agregación (hijo) al nivel con más agregación (padre).

Los cruzamientos entre dimensiones se denominan relaciones dimensionales. La Figura 2b muestra la definición de la relación dimensional *venta* en CMDM. En la relación se cruzan las dimensiones *fechas, clientes, colores, modelos* y *cantidades.*

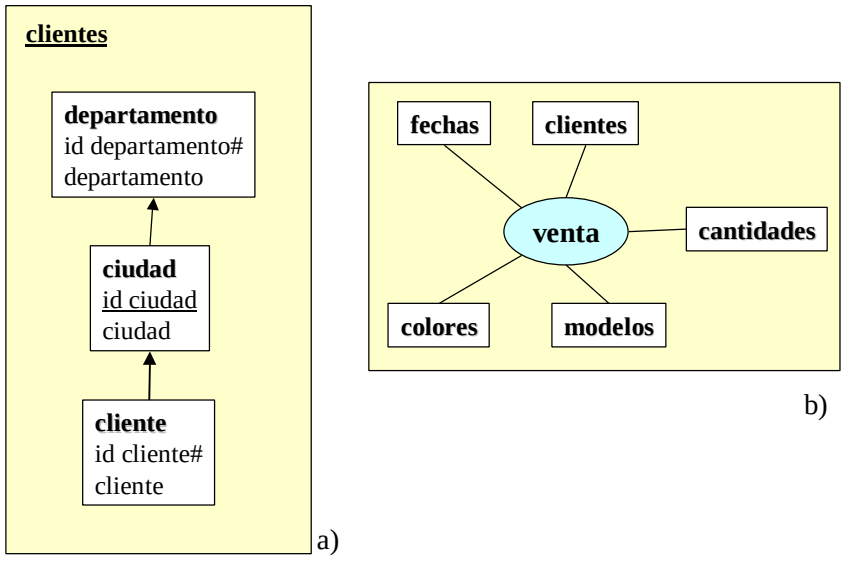

Figura 2 – Representación gráfica de: a) dimensiones y b) relaciones dimensionales

A nivel lógico surgen implementaciones de los cubos tanto para manejadores relacionales, como para manejadores multidimensionales.

Para el caso de manejadores relacionales surgen nuevas técnicas y estrategias de diseño que apuntan esencialmente a optimizar la performance en las consultas introduciendo redundancia, lo cual eventualmente sacrifica la performance en las actualizaciones. ([Kim96], [Moo00]).

El modelo más popular es el estrella ([Kim96], [Bal98], [Moo00]). Consiste de una tabla central que contiene información sobre los cruzamientos de dimensiones y las medidas asociadas (fact table), y tablas de dimensiones que tienen todos los atributos asociados a cada dimensión (dimension tables). La Figura 3 muestra un esquema estrella para el ejemplo anterior.

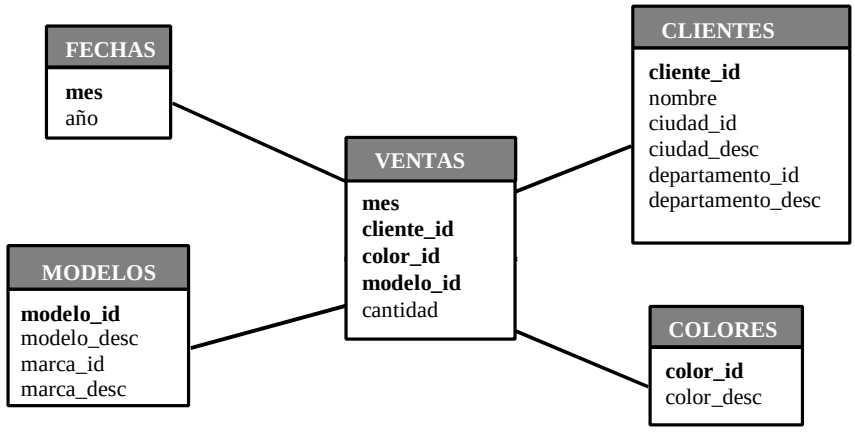

Figura 3 – Un esquema estrella

Si en lugar de almacenar juntos todos los atributos de la dimensión se almacenan en tablas separadas, se llama esquema snowflake ([Bal98], [Moo00]). La Figura 4 muestra el esquema estrella para el mismo ejemplo.

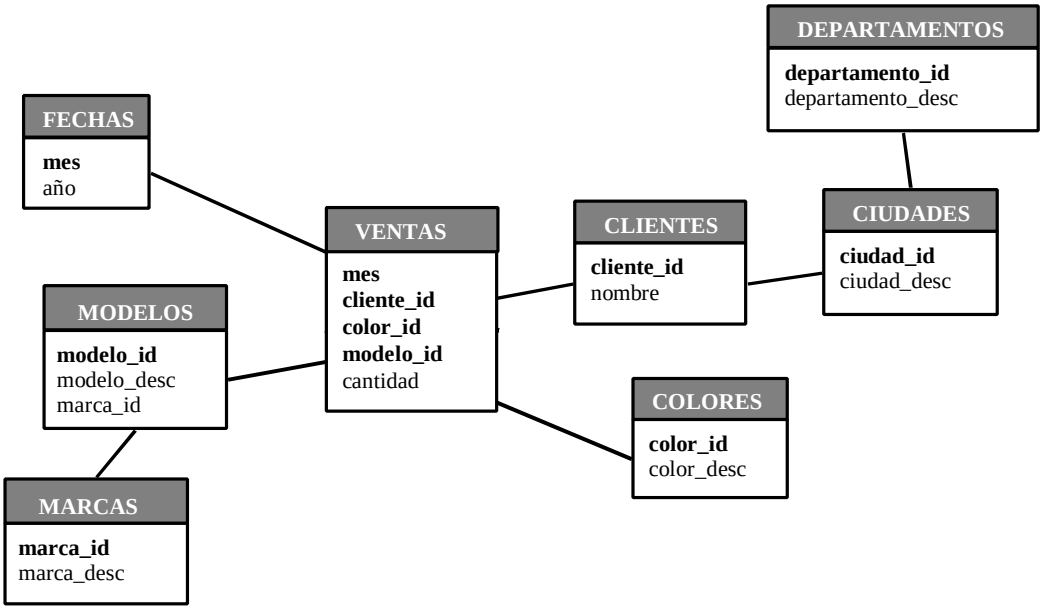

Figura 4 – Un esquema snowflake

Para resolver los requerimientos planteados, los sistemas resultantes (Sistemas de Data Warehousing) combinan diferentes tecnologías. En lo referente a bases de datos se utilizan tanto bases relacionales como multidimensionales ([Kim96], [Ken96], [Ada98]).

2. Plataforma

En el Laboratorio Concepción de Sistemas de Información (CSI) del In.Co. se está trabajando en el desarrollo de herramientas CASE que automaticen algunas tareas de diseño y asistan al diseñador a través de técnicas e interfases gráficas.

Incluye el desarrollo de un Modelo Conceptual Multidimensional: CMDM ([Car00]), [Pic00]), técnicas para diseño lógico basado en transformaciones de esquemas ([Mar00], [Gar00], [Per00]), la repercusión de la evolución de los esquemas fuentes en el DW ([Mar00], [Alc01]) y la traducción del esquema conceptual a un esquema lógico ([Per01]), y la persistencia de objetos ([Arz00]).

La Figura 5 muestra una arquitectura simplificada de la plataforma. Actualmente se cuenta con prototipos para diseño conceptual (editor de CMDM) y diseño lógico relacional y evolución (DwDesigner). Se está trabajando, en el contexto de una tesis de maestría, en el traductor de CMDM a un DW relacional. Faltan implementar prototipos para los editores de lineamientos y mapeos, que se proponen para este proyecto.

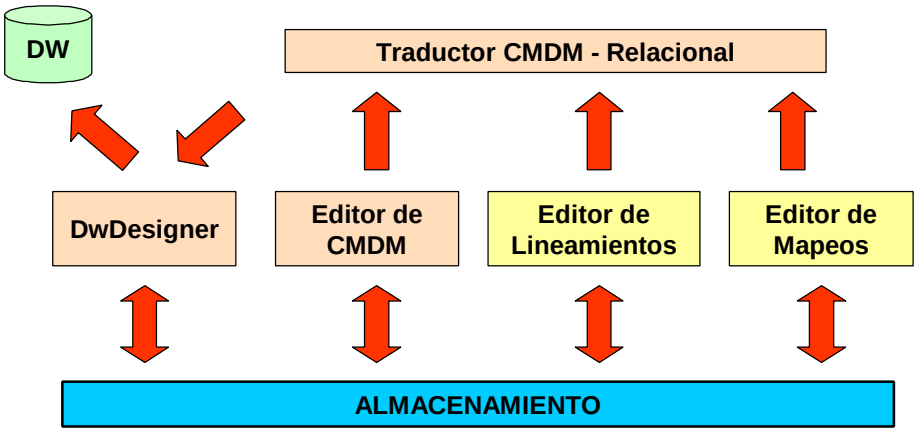

Figura 5 – Arquitectura de la Plataforma de Diseño de DW

La plataforma es parte de un marco de investigación y cooperación entre la Universidad de la República y la Universidad de Versailles Saint-Quentin-enYvelines (Francia).

3. Problema planteado

Así como para las bases de datos operacionales existen varias propuestas para traducir un esquema E/R en un esquema relacional, se está trabajando en la traducción de un esquema conceptual multidimensional a un esquema relacional. ([Per01]).

El esquema lógico es una especificación más detallada que el esquema conceptual, donde se incorporan nociones de almacenamiento y estructuración de los datos. Durante la etapa de diseño lógico se construye el esquema lógico teniendo en cuenta no sólo el esquema conceptual, sino también estrategias para resolver los requerimientos de performance y almacenamiento.

Hay un componente adicional a tener en cuenta: las bases fuentes. Un DW no es una base de datos para construir desde cero, sino que debe construirse con información extraída de un cierto conjunto de bases fuentes. Durante el diseño lógico deben considerarse las fuentes y cómo se corresponden con el esquema conceptual.

Por lo tanto es de vital importancia poder relacionar los elementos del modelo conceptual con las tablas y atributos de las bases fuentes.

La Figura 6 muestra los pasos en el proceso de diseño lógico de un DW.

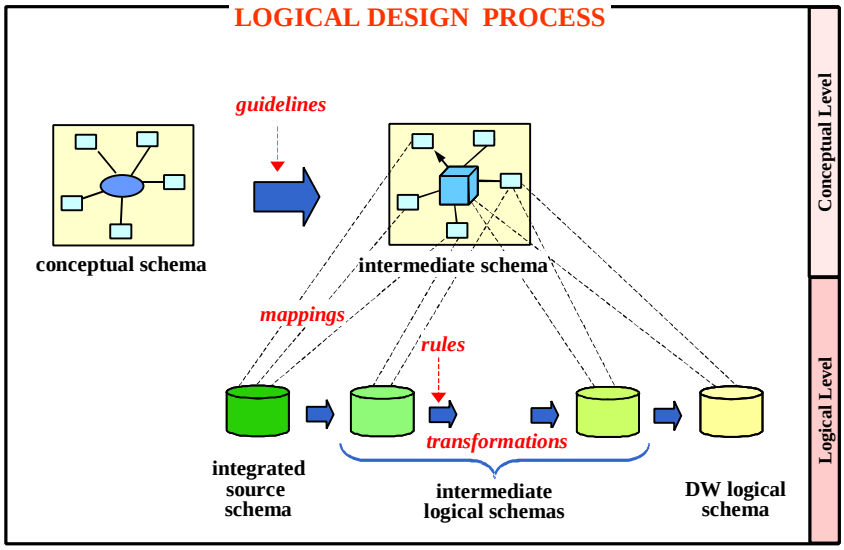

Figura 6 – Proceso de diseño lógico de un DW

El proceso de diseño lógico tiene dos entradas: el esquema conceptual (conceptual schema), y el esquema lógico de la base fuente integrada (integrated source schema).

En un primer paso el diseñador indica algunos lineamientos que complementan al esquema conceptual con estrategias de diseño lógico (guidelines), por ejemplo: como particionar datos históricos o que datos almacenar juntos. El esquema conceptual más los lineamientos conforman el esquema intermedio (intermediate schema).

Luego el diseñador establece mapeos o correspondencias (mappings) entre el esquema intermedio y el esquema lógico de la fuente. Dichos mapeos indican dónde se encuentran en la fuente los diferentes elementos del esquema intermedio.

A partir de allí, se va transformando (transformations) el esquema lógico de la fuente, refinándolo sucesivamente (transformed logical schema), hasta obtener el esquema lógico deseado para el DW (DW logical schema). Se propone un algoritmo y un conjunto de reglas de transformación (rules) para llevar a cabo dicho refinamiento.

Este proyecto se centra en la definición de lineamientos y mapeos.

3.1. Lineamientos

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.

Los lineamientos permiten elegir el estilo de diseño para el DW (snowflake, estrella, mixto), indicar requerimientos de performance y almacenamiento (indicando que cubos implementar), y elegir una estrategia para almacenar datos históricos (fragmentando los cubos).

A continuación se describen brevemente los lineamientos:

Materialización de Relaciones

En el esquema conceptual una relación dimensional indica que se quieren cruzar determinadas dimensiones. Como las dimensiones pueden tener varios niveles, una relación dimensional en realidad representa a muchos cubos.

Algunos de esos cubos se querrán implementar en tablas relacionales (fact tables), otros se calcularán en el momento de las consultas.

Este lineamiento permite indicar que cubos se quieren implementar.

Un cubo está formado por un nombre, una relación a la cual materializa, un conjunto de niveles que conforman su nivel de detalle, y opcionalmente un nivel que es elegido como medida. El diseñador puede elegir varios niveles de la misma dimensión, pero debiera advertírsele.

Gráficamente los representamos con un cubo unido a varios niveles (rectángulos blancos) que conforman el nivel de detalle. Las medidas son los niveles marcados por una flecha. Dentro del cubo está su nombre y la relación que materializa entre paréntesis. La Figura 7 muestra el cubo *venta-1* de la relación dimensional *venta* presentada en la Figura 2b. Tiene por niveles: *mes, ciudad, color, modelo* y *cantidad,* y por medida: *cantidad*.

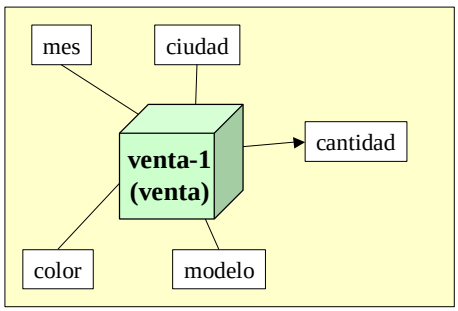

Figura 7 – Materialización de relaciones (Cubos)

Fragmentación de Dimensiones

El diseñador puede indicar el grado de normalización que quiere para implementar las dimensiones. Por ejemplo, puede querer un esquema estrella, es decir, denormalizar todas las dimensiones en una sola tabla. Por el contrario, puede querer un esquema snowflake, normalizando todas las dimensiones.

Puede querer tratar diferente cada dimensión, indicando para cada una si normaliza, denormaliza o efectúa una estrategia intermedia, indicando en este último caso, qué niveles se guardarán juntos.

En este lineamiento, el diseñador indica para cada dimensión, que niveles desea almacenar juntos, conformando una fragmentación de los niveles de la dimensión.

Un fragmento es un subconjunto de los niveles de la dimensión que se desean almacenar juntos. El diseñador define por extensión una función que a cada dimensión le haga corresponder un conjunto de fragmentos.

No todas las fragmentaciones tienen sentido, interesan las que agrupan niveles contiguos en las jerarquías. Además, no puede haber ningún nivel sin fragmento asociado.

Gráficamente representamos una fragmentación como una coloración de los niveles. Los niveles de un mismo fragmento se recuadran con el mismo color. Un nivel tendrá varios colores si está en más de un fragmento. La fragmentación es completa si todos los niveles tienen color. En la Figura 8 se muestra una fragmentación de la dimensión *clientes*, presentada en la Figura 2a. La fragmentación tiene 2 fragmentos: *departamento y ciudad* (rosa), y *ciudad y cliente* (celeste). El nivel *ciudad* pertenece a ambos fragmentos.

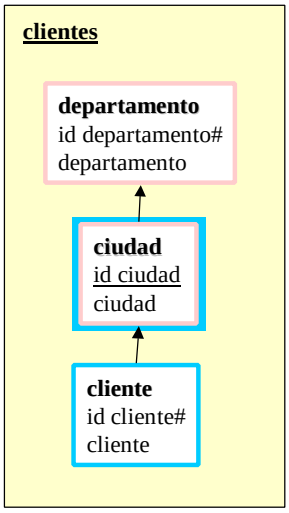

Figura 8 – Fragmentación de dimensiones

Fragmentación de Cubos

Una fact table puede contener gran cantidad de información histórica, lo que la vuelve ineficiente para las consultas. Esa tabla se puede fragmentar horizontalmente, guardando un subconjunto de las tuplas en cada fragmento (bandas).

Como ejemplo consideremos el cubo venta-1 de la Figura 7, y supongamos que las consultas más frecuentes son para los dos últimos años. Se pueden almacenar en una banda las tuplas correspondientes a ventas posteriores a ene-2000, y en otra las que corresponden a meses anteriores.

Para definir una fragmentación el diseñador debe las condiciones que distinguen una banda de otra, expresadas en términos de los ítems de los niveles del cubo.

En este lineamiento el diseñador indica que bandas quiere almacenar.

3.2. Mapeos

El esquema conceptual especifica la información que contendrá el DW, y a través de los lineamientos, el diseñador explicitó las características que debe cumplir el esquema lógico; y con eso se construyó el esquema intermedio. El siguiente paso es vincular el esquema intermedio 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.

Los mapeos son funciones que asocian a cada ítem de un objeto del esquema intermedio una expresión de mapeo, construida en base a las tablas y atributos de la base fuente. Estas funciones son definidas por el diseñador en forma explícita. Se tendrá una función de mapeo para cada fragmento de dimensión, y una función de mapeo para cada cubo.

Una expresión de mapeo puede ser un atributo de una tabla fuente (directo), o un cálculo que involucra varios atributos de una tupla (cálculo simple), o una totalización que involucra varios atributos de varias tuplas (calculo agregado) o algo externo a las fuentes como una constante, una estampa de tiempo o dígitos de versión (externo).

Representamos gráficamente una función de mapeo, como vínculos (líneas) entre los ítems del modelo conceptual y los atributos de las tablas fuentes.

Cuando el mapeo es directo se representa con una línea corrida, cuando es un cálculo se representa con una línea cortada a cada atributo que interviene en el cálculo (y se adjunta la definición del cálculo), y cuando es externo no se utilizan líneas, pero se adjunta la expresión a la que mapea.

En la Figura 9 se muestra una función de mapeo para la dimensión *geografía* (con un único fragmento). El ítem *país* tiene un mapeo externo, el ítem *zona* tiene un mapeo calculado en base al atributo *zona* de la tabla *Departamentos*. Los demás ítems tienen mapeos directos.

Las tablas *Ciudades* y *Departamentos* joinean por el atributo *Id-depto*. Para que la función de mapeo sea correcta todas las tablas involucradas deben joinear entre si.

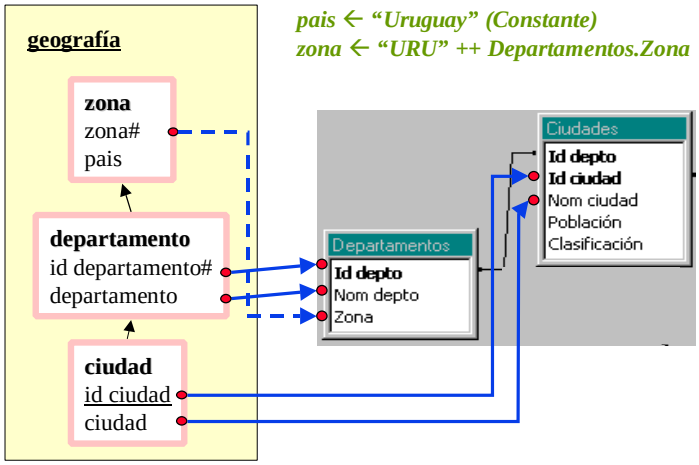

Figura 9 – Representación gráfica de una función de mapeo

En la vinculación de un fragmento o un cubo a las fuentes pueden existir condiciones, por ejemplo que un atributo se encuentre en determinado rango. Estas condiciones pueden deberse a restricciones en el esquema conceptual o a restricciones que deseen aplicarse a las fuentes.

En el ejemplo de la Figura 9, el nivel *zona* podría tener una restricción indicando que sólo interesan las zonas menores a 10, que podrían ser por ejemplo, las zonas uruguayas. Esta es una restricción del esquema conceptual. También puede ocurrir que en la tabla *Ciudades* se almacenen ciudades necesarias en varios sistemas o secciones, y que para el DW sólo interesen las que tienen *Clasificación R*. Esta es una restricción respecto a las fuentes.

Ambas restricciones deben tenerse en cuenta al establecer los mapeos, e incorporar una condición tipo:

$$Departamentos.zona < 10 \text{ Ù } Ciudades.Clasificación = \text{“}R\text{”}$$

El diseñador debe poder expresar condiciones, que son en realidad predicados sobre atributos de las tablas fuentes.

4. Objetivos

El objetivo de este proyecto es construir dos prototipos de herramientas CASE que asistan al diseñador en el diseño de un DW relacional. Una de las herramientas debe asistir en la definición de los lineamientos, la otra en la definición de los mapeos.

La herramienta de asistencia en la definición de lineamientos debe proveer las siguientes funcionalidades:

- Leer un esquema conceptual y almacenarlo en memoria.
- Sugerir lineamientos para todo el esquema. Hay dos variantes: lineamientos que conducen a un esquema estrella y a un esquema snowflake.
- Sugerir lineamientos para una dimensión o para una relación dimensional.
- Permitir especificar manualmente los lineamientos.
- Alertar al diseñador cuando sus especificaciones son incorrectas o incompletas.
- Permitir modificar los lineamientos diseñados / sugeridos.
- Visualizar gráficamente los lineamientos ya diseñados / sugeridos.
- Guardar los lineamientos.

La herramienta de asistencia en la definición de mapeos debe proveer las siguientes funcionalidades:

- Leer un esquema conceptual y almacenarlo en memoria.
- Leer lineamientos y almacenarlos en memoria.
- Leer la estructura de la base fuente y almacenarla en memoria.
- Sugerir mapeos por defecto siguiendo diferentes estrategias, por ejemplo macheo por nombres.
- Permitir especificar funciones de mapeo para fragmentos y cubos. Se debe poder mapear a los distintos tipos de expresiones.
- Alertar al diseñador cuando sus especificaciones son incorrectas o incompletas.
- Permitir modificar los mapeos ya realizados.
- Permitir imponer condiciones a los mapeos.
- Visualizar gráficamente los mapeos ya diseñados.
- Guardar los mapeos.

Ambas herramientas tienen como base el mismo diseño de clases; las cuales están bastante definidas y algunas programadas. El alcance del proyecto es obtener prototipos de las herramientas.

5. Bibliografía

[Ada98] Adamson, C. Venerable, M.: "Data Warehouse Design Solutions". J. Wiley & Sons, Inc.1998.

[Alc01] Alcarraz, A. Ayala, M. Gatto, P.: "Diseño e implementación de una herramienta para evolución de un Data Warehouse Relacional". Undergraduate project. InCo, Universidad de la República, Uruguay, 2001.

[Arz00] G. Arzúa, G. Gil, S. Sharoian. "Manejador de Repositorio para Ambiente CASE". Under-graduate Project. InCo, Universidad de la República, Uruguay, 2000.

[Bal98] Ballard, C. Herreman, D. Schau, D. Bell, R. Kim, E. Valncic, A.: "Data Modeling Techniques for Data Warehousing". SG24-2238-00. IBM Red Book. ISBN number 0738402451. 1998.

[Car00] Carpani, F.: "CMDM: A conceptual multidimensional model for Data Warehouse". Master Thesis. InCo - Pedeciba, Universidad de la República, Uruguay, 2000.

[Gar00] Garbusi, P. Piedrabuena, F. Vázquez, G.: "Diseño e Implementación de una Herramienta de ayuda en el Diseño de un Data Warehouse Relacional". Undergraduate project. InCo, Universidad de la República, Uruguay, 2000.

[Inm96] Inmon, W.: "Building the Data Warehouse". John Wiley & Sons, Inc. 1996.

[Inm96a] Inmon, W.: "Building the Operational Data Store". John Wiley & Sons, Inc. 1996.

[Ken96] Kenan Technologies:"An Introduction to Multidimensional Databases". White Paper, Kenan Technologies, 1996.

[Kim96] Kimball, R.:"The Datawarehouse Toolkit". John Wiley & Son, Inc., 1996.

[Mar00] Marotta, A.: "Data Warehouse Design and Maintenance through Schema Transformations". Master Thesis. InCo - Pedeciba, Universidad de la República, Uruguay, 2000.

[Moo00] Moody, D. Kortnik, M.: "From Enterprise Models to Dimensionals Models: A Methodology for Data Warehouse and Data Mart Design". DMDW'00, Sweden, 2000.

[Per99] Peralta, V. Marotta, A. Ruggia, R.: "Designing Data Warehouses through schema transformation primitives". ER'99 Posters and Demonstrations, France, 1999.

[Per00] Peralta, V. Garbusi, P. Ruggia, R.: "DWD: Una herramienta para diseño de Data Warehouses basada en transformaciones sobre esquemas". Technical Report. InCo, Universidad de la República, Uruguay, 2000.

[Per00a] Peralta, V.: "Sobre el pasaje del esquema conceptual al esquema lógico de DW". JIIO'00, Uruguay, 2000.

[Per01] Peralta, V.: "Diseño lógico de un Data Warehouses a partir de un Esquema Conceptual Multidimensional". Master Thesis. InCo - Pedeciba, Universidad de la República, Uruguay. On going work.

[Pic00] Picerno, A. Fontan, M.: "Un editor para CMDM". Undergraduate Project. InCo, Universidad de la República, Uruguay. 2000.