# Capítulo 4 DOMINIQUE

En este capítulo se presenta DOMINIQUE un sistema de análisis del contenido de documentos de prensa electrónica. La sección 4.1 presenta su arquitectura general. Las secciones 4.2 y 4.3 presentan las fases de construcción y análisis del sistema respectivamente. Finalmente, la sección 4.4 presenta las ventajas y desventajas asociadas al diseño de DOMINIQUE.

## 4.1 Arquitectura general

La prensa electrónica es un medio masivo de información, un claro ejemplo de esto son las publicaciones de los periódicos electrónicos que constituyen un medio por el cual nos podemos mantener informados de todo lo que sucede en el mundo.

Las personas que pueden estar interesadas en la información publicada en estos periódicos son diversas. Podemos por ejemplo, imaginar a nuestro productor de huevo en la ciudad de Puebla, que se encuentra interesado en distribuir su producto en otras ciudades del estado y hasta en otros estados del país. Así dicho productor querrá conocer el consumo de huevo en los estados, ciudades o zonas del país donde tenga pensado distribuir su producto. A su vez, también estará interesado en las temperaturas promedio registradas en dichos sitios, ya que es importante saber si su producto no se descompondrá en el trayecto de transportación del mismo.

El productor de huevo accederá a la información que necesita mediante las publicaciones de los periódicos electrónicos. Sin embargo, no le bastará conocer la temperatura de un sólo

día, ni el consumo de huevo del mes en curso, necesita saber los promedios registrados durante un cierto tiempo. Requiere de un acceso a este conjunto de información de forma rápida, confiable y resumida que le proporcione un análisis suficientemente claro, el cuál le permita tomar una decisión.

En la Figura 4.1 se muestra la arquitectura de DOMINIQUE para soportar la creación y almacenamiento de consultas de datos recuperados de documentos electrónicos.

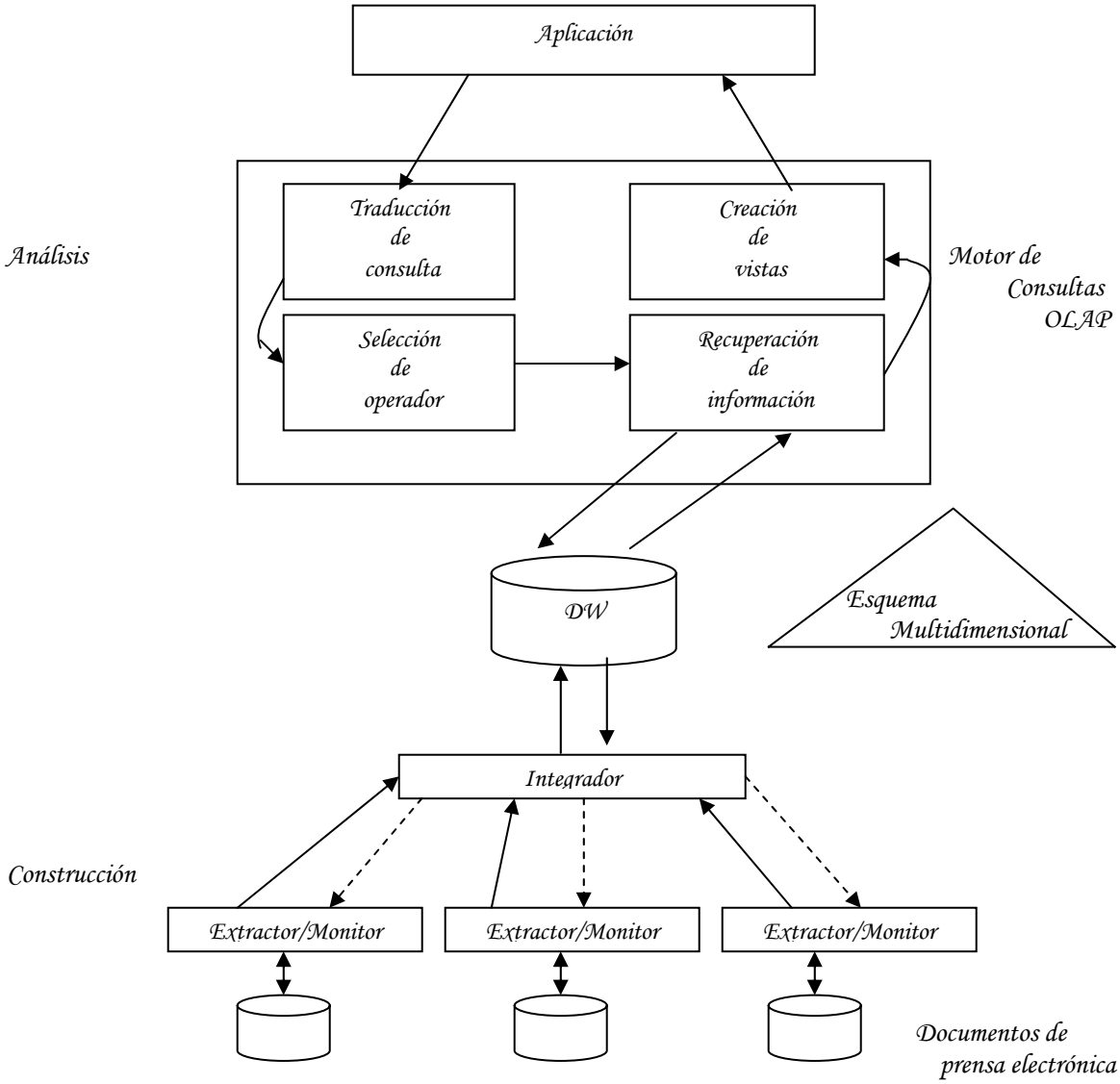



Figura 4.1 Arquitectura de DOMINIQUE

DOMINIQUE conserva la arquitectura típica de un DW. Su elaboración comprende dos fases, una de construcción y otra de análisis, es decir, primeramente llevar los datos de las fuentes al repositorio y después tomar esta información para soportar consultas. Estas dos fases son explicadas en la sección 4.2 y 4.3 respectivamente. Como se aprecia en la Figura, DOMINIQUE utiliza un esquema interno de representación, que es un esquema multidimensional.

## 4.2 Esquema de DOMINIQUE

Como ya se mencionó anteriormente hablar del diseño de un DW es referirse a la estructura en que los datos serán almacenados y manipulados dentro del repositorio. Para llevar a cabo un correcto diseño es necesario analizar los tipos de datos que se ingresarán, así como también su formato.

DOMINIQUE es un DW que almacena y manipula datos de periódicos electrónicos. El formato en que dichos periódicos (fuentes de DOMINIQUE) están construidos, es de manera semi-estructurada, ya que carece de una buena estructuración la información en las diversas fuentes. Refiriéndose a los tipos de fuentes mencionadas en el capítulo 2, las publicaciones de periódicos electrónicos se ubican en el tipo de fuentes de instancia múltiple, puesto que manejan información en un formato semi-estructurado en su página.

DOMINIQUE implementa un esquema multidimensional que es instrumentado a través de un esquema relacional con un esquema en copo de nieve. Con un modelo multidimensional, surge la necesidad de seleccionar una estrategia de representación, la cuál es el esquema

relacional. La ventaja que ofrece el modelo relacional es la facilidad de representar la información en forma de tablas, lo que facilita el acceso a través de las diversas granularidades de los datos multidimensionales.

El esquema en estrella y el copo de nieve son usados comúnmente para representar los datos multidimensionales. DOMINIQUE, emplea el esquema de copo de nieve, que no es otra cosa que la normalización del esquema en estrella. En vez de tener una sola tabla por cada dimensión, en el esquema copo de nieve se tienen sub-tablas de dimensiones, esto con la única intención de simplificar la información, evitar duplicidad en la misma y representar mejor la semántica de las dimensiones del aspecto de los negocios. Vale la pena también mencionar que ambos cubos emplean dicho modelo de representación. A continuación se describe la representación del modelo multidimensional para las dos secciones de prensa electrónica.

### 4.2.1 Análisis climatológico

Para diseñar el cubo del clima, primero se procedió a analizar la información que era proporcionada por la fuente. Con fundamento en esto se decidió que dicho cubo estaría conformado por cuatro dimensiones:

- Estación: Estación del año correspondiente a una cierta fecha.
- Cielo: Condiciones atmosféricas del cielo.
- Región: Ubicación dentro del país.
- Fecha: Un momento en el tiempo, como día, mes o año.

Dichas dimensiones están integradas por ciertos niveles de agregación (granos). Para algunas dimensiones sólo se tiene un nivel de agregación, esto dependiendo sobre todo de la información proporcionada en la fuente, quedando pues las siguientes granularidades:

- Para estación: Temporada del año.
- Para cielo: Estado del cielo.
- Para región: Ciudad, estado, zona.
- Para fecha: Día, mes, año.

Y por último teniendo como medida del cubo la temperatura expresada en grados *Celcius* (°C). La Figura 4.2 la representación del cubo de datos para la climatología.

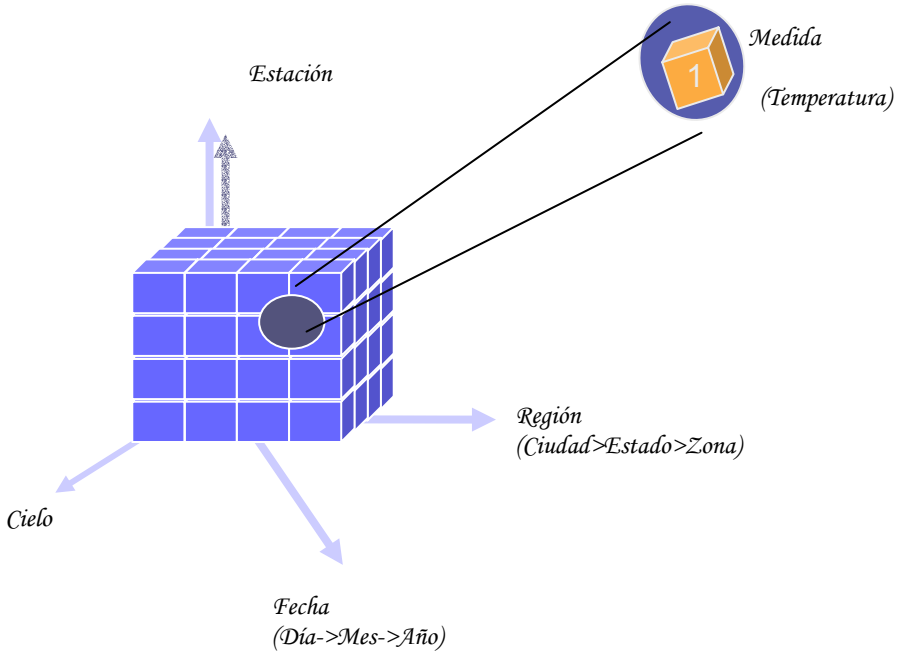



Figura 4.2 Hipercubo de Temperaturas

En la Figura 4.2 se muestran las dimensiones con sus granos correspondientes así como la medida resultante de la unión de las diferentes dimensiones.

Para diseñar el esquema relacional del cubo de datos climatológicos se hizo necesaria la creación de las siguientes tablas:

- Ciudad(CodCiudad, NomCiudad)
- Estado(IdEstado, NomEstado)
- CiudEdo(CodCiudad, IdEstado)
- Zona(NomZona, IdEstado)
- Temperatura(Mínima, Máxima, CodCiudad, Nomtemporada, Fecha, Cielo)

Dichas tablas tienen asociadas restricciones:

Restricciones de llave: Señaladas mediante el subrayado.

Restricciones referenciales: Señaladas mediante la relación de las llaves de una tabla a otra.

Como se puede observar se tiene la medida temperatura mínima y máxima. La tabla de Temperatura corresponde a la tabla de hechos la cuál tiene como llave el identificador de cada una de las tablas de dimensiones. En el caso de la dimensión región, se tiene una tabla de dimensión que es ciudad y esta se encuentra normalizada en otras dos tablas de sub-dimensiones: estado y zona, estando ambas relacionadas mediante llaves externas.

### 4.2.2 Análisis de ventas

De la misma forma, para diseñar el cubo del consumo de productos, primero se procedió a analizar la información proporcionada por la fuente. Con fundamento en esto se decidió que dicho cubo estaría conformado  por tres dimensiones:

- Producto: Refiriéndose a los productos de mayor consumo en el país
- Región: Refiriéndose a la ubicación dentro del país.
- Fecha: Como su nombre lo dice, hace referencia a un momento en el tiempo.

Dichas dimensiones están integradas por ciertos niveles de agregación, quedando pues las siguientes granularidades:

- Para producto: Producto, derivados, grupo, rama, familia.
- Para región: Ciudad, estado, zona.
- Para fecha: Día, mes, año.

Y por último teniendo como medida del cubo, el índice numérico del consumo de productos. La figura 4.6 la representación del cubo de datos para el consumo de productos.

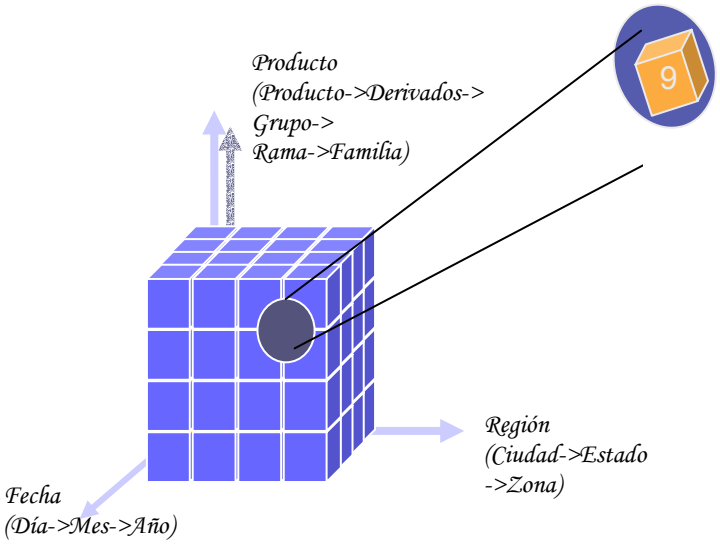



Figura 4.3 Hipercubo de Consumo de productos

Para diseñar el esquema relacional del cubo de datos de consumo de productos se hizo necesaria la creación de las siguientes tablas:

- Ciudad (NomCiudad, CodCiudad)
- Estado (NomEstado, CodCiudad)
- Zona (NomZona, NomEstado)
- Producto (NomProducto, CodProducto)
- Derivados (CodDerivado, NomDerivado)
- Grupo (CodGrupo, NomGrupo)
- Rama (CodRama, NomRama)
- Familia (NomFamilia, CodRama)
- Derivado_Producto (CodDerivado, CodProducto)
- Grupo_Derivado (CodGrupo, CodDerivado)
- Rama_Grupo (CodRama, CodGrupo)
- Ventas (Indice, CodProducto, CodCiudad, Fecha)

De la misma forma éstas tablas tienen restricciones asociadas:

Restricciones de llave: Señaladas mediante el subrayado.

Restricciones referenciales: Señaladas mediante la relación de las llaves de una tabla a otra.

Restricciones de dominio: Señaladas con el uso de un mismo nombre en las tablas.

Este cubo es un tanto diferente al anterior ya que en este se tiene una tabla más de sub-dimensiones. También es de notar que la dimensión de productos tiene muchas más

granularidades, lo que hace más complejo e interesante el diseño construcción y explotación de dicho cubo. La metodología de referencia a la par del cubo de climatología es la misma, una tabla de sub-dimensión y una de dimensión se relacionan mediante una llave fóranea.

## 4.3 Construcción

En base a un esquema adaptado al análisis de datos y un conjunto de fuentes predefinidas, el proceso de construcción consiste en extraer los datos, integrarlos y eventualmente refrescar el contenido del DW. El esquema de construcción se muestra en la Figura 4.4.

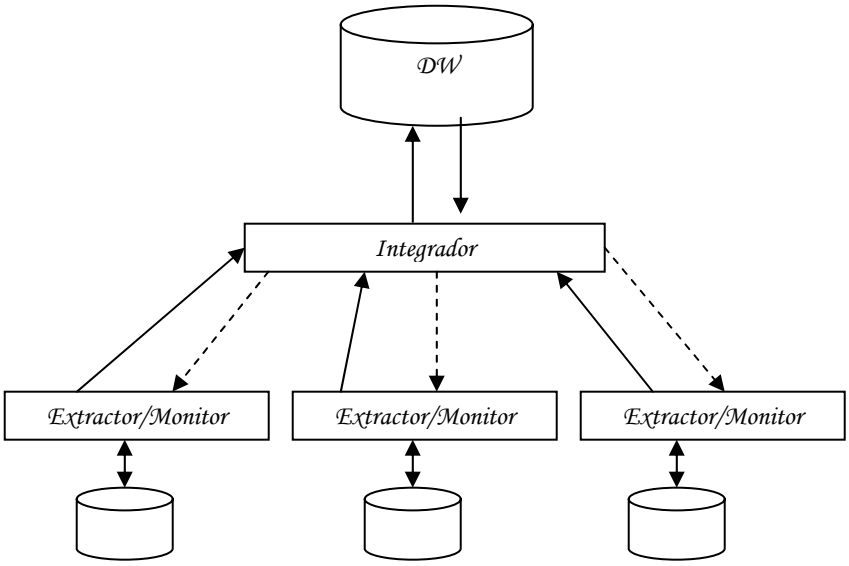



Figura 4.4. Arquitectura de construcción

Como se muestra en la Figura 4.4 primero se debe localizar la información cuyas características ya fueron analizadas, después se extrae dicha información. El proceso de extracción es particular para cada fuente, ya que cada una maneja una representación y un formato diferentes, por lo cual se hace necesario la construcción de un extractor individual

por fuente. Ya teniéndose la información concentrada ahora lo consecuente es darle un mismo formato, homogeneizarla en otras palabras, proceso que será realizado por el extractor, posteriormente es concentrada por el integrador, mismo que se encarga de poblar el repositorio. La construcción de DOMINIQUE es presentada más a detalle a continuación.

### 4.3.1 Extractor

Es el encargado de realizar la extracción de datos. Consiste en llevar los datos de las fuentes a DOMINIQUE. A pesar de que DOMINIQUE extrae información sólo de periódicos electrónicos, los formatos de cada uno de estos son cambiantes, inclusive de una sección a otra dentro de un mismo periódico pueden estar estructuradas de forma diferente. Por lo que se puede observar una heterogeneidad elevada de los datos.

Los extractores conocen perfectamente el protocolo y ubicación de las fuentes y del DW. Esto quiere decir que recibe datos en lenguaje HTML y los comienza a reformar en lenguaje SQL. Siendo este el esquema general de datos que manipulan los extractores.

DOMINIQUE se compone de tres extractores, uno que es capaz de extraer los datos de las temperaturas registradas en forma diaria, otro que extrae los índices de consumos de productos registrados en forma mensual y por último un extractor que recupera la información de las poblaciones más importantes de la República Mexicana, así como a los respectivos estados a los que éstas pertenecen.

El encargado de recuperar la información de las fuentes es un extractor, un *"gateway"* o bien la migración de datos. DOMINIQUE emplea el diseño de extractores los cuales conocen el formato de las fuentes así como el formato de representación de datos dentro del DW, conocen el protocolo de comunicación de ambos, así como también su ubicación.

**Extractor de temperaturas**

El extractor de las temperaturas también tiene un módulo de monitoreo de las fuentes. Por fuente se entiende la página web de un periódico de prensa electrónica la que sirve de información al DW. Como es de suponerse la página Internet donde aparecen las temperaturas de las principales ciudades de México es actualizada día con día, así pues se hace necesario que un monitor actualice la página cada 24 horas.

Éste extractor consta a la par de un módulo de acceso a fuentes y otro de traducción de datos, el cuál enviará la información finalmente al integrador, mismo que se discutirá mas adelante. La arquitectura del extractor de temperaturas es la Figura 4.5.

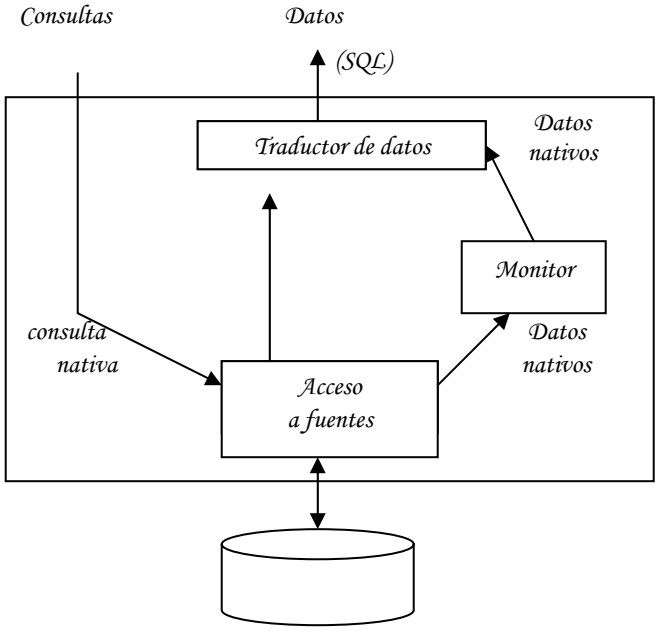



Figura 4.5 Arquitectura extractor de temperaturas

Como se muestra en la Figura 4.5, la fuente provee la información al extractor, por lo tanto el extractor conoce bien el protocolo, ubicación y formato, tanto de la fuente como del DW. Por lo tanto el extractor recibe los datos en el lenguaje HTML y los interpreta reformulándolos en el lenguaje SQL, formato en el que los envía al integrador.

**Extractor de consumo de productos y poblaciones**

Tanto el extractor de consumo de productos, como el de las ciudades y localidades de México tienen una arquitectura idéntica. A diferencia del extractor de temperaturas estos dos extractores no tienen un módulo de monitoreo. La razón de lo anterior, es que las fuentes donde se obtienen las ciudades no se actualizan, mientras que las fuentes donde se recuperan los índices de consumo de productos se actualizan una sola vez al mes.

Los módulos de traducción de datos y acceso a fuentes se conservan en ambos extractores. La arquitectura para los extractores de consumo de productos y ciudades es presentada a continuación en la Figura 4.6.

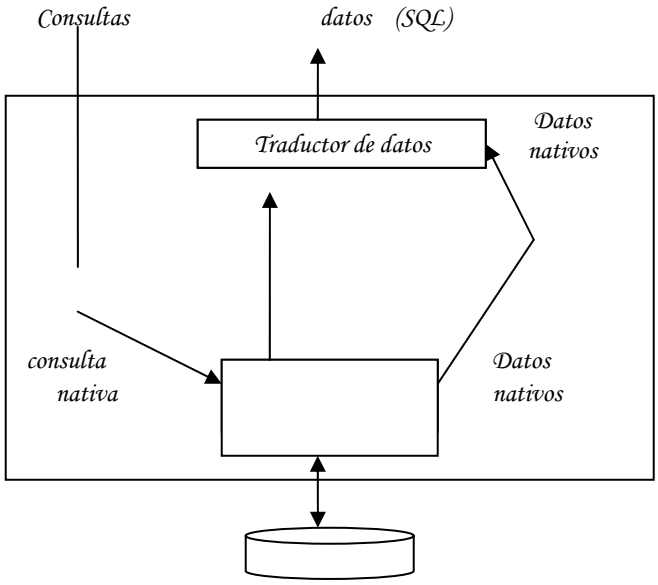



Figura 4.6 Arquitectura extractor de índices y de poblaciones.

De igual manera que el extractor para las temperaturas el extractor recupera los datos en un formato HTML y luego con el módulo de traducción de datos los transforma en SQL, lenguaje en el que los datos recuperados son enviados al integrador.

### 4.3.2 Integrador

Una vez que se ha extraído la información de las fuentes, el siguiente paso consiste en dar un mismo formato, es decir, quitar la redundancia y completar los datos, para finalmente llevar esta actualización de información al DW. Como es de suponerse, la información proporcionada por el extractor de las fuentes puede contener datos erróneos o quizá información incompleta, así que el integrador es el encargado de no actualizar dicha información, así como también de fusionar los datos que provienen de las diferentes fuentes (problema no considerado en esta tesis) y finalmente cargar los datos ya fusionados en el DW.

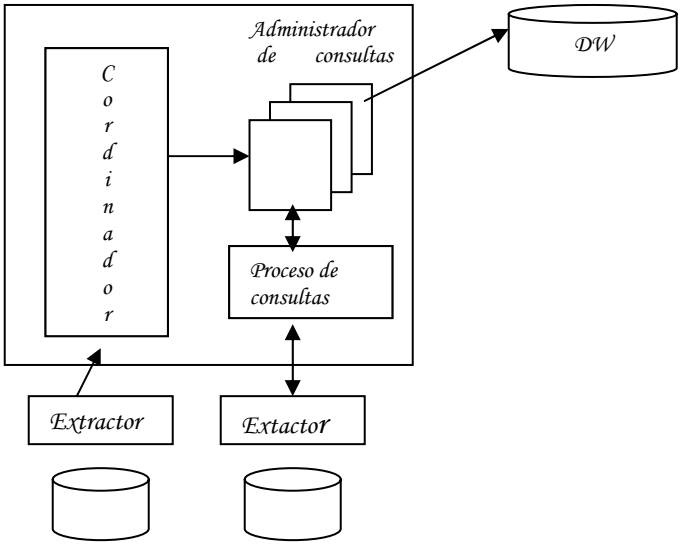



Figura 4.7 Arquitectura del integrador.

El integrador recibe del DW consultas, las cuales son peticiones de recuperación de información para las fuentes. Estas peticiones son enviadas a los extractores los cuales recuperan la información la homogenizan, la envía al integrador, el cual se encarga de poblarla en el repositorio.

El integrador consta de un módulo de proceso de consultas, otro que será el administrador de relaciones y finalmente un coordinador. El primero, encargo de comunicarse directamente con los extractores para realizar un filtrado de los datos y homogeneizarlos; el segundo es el encargado de dar mantenimiento a los datos almacenados actualizando la información que se esté recuperando.

### 4.3.3 Mantenimiento

DOMINIQUE implementa un refrescado incremental. La razón de esto se encuentra precisamente en sus fuentes. Los periódicos electrónicos que sirven de fuente a DOMINIQUE se actualizan con una cierta frecuencia de tiempo, siendo ésta diferente para cada uno de ellos. De esta forma es posible actualizar la información la información administrada dentro del DW, cada vez que se actualizan las fuentes.

## 4.4 Análisis empleando técnicas OLAP

Para consultar a DOMINIQUE se hace uso de un motor de consultas, mismo que es mostrado en la sección 4.1. Éste motor de consultas es el encargado de recuperar la información solicitada por el usuario a través de la interfaz y recuperar los datos del

repositorio. Para realizarlo se emplean las técnicas OLAP con sus operadores *slice'n dice*, *roll-up* y *drill-down,* mismos que serán descritos en esta sección.

La estrategia que emplea el motor de consultas, es almacenar las consultas que se van ejecutando. Para realizar esta tarea existe la posibilidad de almacenar las tuplas seleccionadas como respuesta a la consulta en ejecución en forma de tablas. Sin embargo, esto sería muy costoso en términos de almacenamiento.

Como la cantidad de consultas que se realizan es realmente alta, para la implementación de los operadores OLAP se crean vistas materializadas, las cuales son "fotos fijas", que no se almacenan físicamente en la base de datos. En otros términos las vistas materializadas pueden ser consideradas como tablas virtuales que almacenan las tuplas seleccionadas por una consulta.

DOMINIQUE hace uso de vistas materializadas que guardan los resultados de las consultas. También emplea dos tablas que son encargadas de llevar un control de los cubos, granos, dimensiones y medidas empleadas en las consultas. La finalidad de estas dos tablas es evitar el rehacer una consulta y mantener datos agregados (vistas materializadas) que agilicen el tiempo de respuesta de una nueva consulta. Estas tablas son; Bitácora y Bitácora_resultados, mismas que son descritas en le siguiente capítulo.

Un ejemplo de creación de vistas materializadas se muestra en la Figura 4.8

*Create materialized view consulta1 As*

*( select codciudad*

*from ciudedo cd, estado e1*

*where e1.idestado = cd.idestado and e1.nomestado = 'PUEBLA' )*

Figura 4.8 Instrucción SQL de creación de una vista materializada

### 4.4.1 Slice'n dice

La operación *Slice'n dice* es la encargada de restringir valores dentro de los granos del cubo original, incluso una sola. El diseño que se siguió para la implementación del operación *slice'n dice* es el que se muestra en la Figura 4.9.

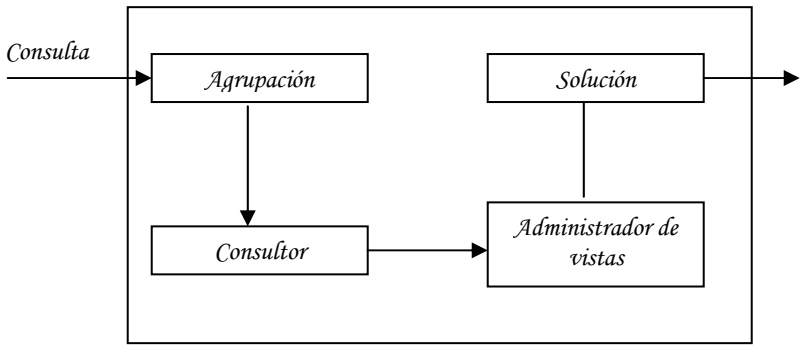



Figura 4.9 Diseño de operación *Slice'n dice*

Como se muestra en la Figura 4.9 el diseño de esta operación inicia con la recepción de una consulta. Esta consulta es recuperada en la interfaz mediante la selección de valores seleccionados por el usuario. De esta manera el usuario mediante la interfaz elige las dimensiones, granos y medidas que quiere consultar, todos estos valores van siendo agrupados (módulo de agrupación), para finalmente tener una consulta en una sola oración o sentencia (ésta ya en lenguaje SQL).

Una vez que se tiene elaborada la consulta en el lenguaje interno de DOMINIQUE, lo siguiente es verificar si la consulta ha sido realizada previamente. Esta tarea emplea la tabla de Bitácora; en caso de encontrarse que la consulta existe se recupera el resultado y en caso contrario se ejecuta la consulta (módulo consultor). Finalmente el resultado de la consulta es direccionado a la interfaz del usuario mostrándole los valores de la consulta ejecutada (módulo solución). Ya con el resultado obtenido y en caso de ser una consulta nueva, se crea una nueva vista materializada que guarda las tuplas empleadas para el cálculo de la consulta y se actualizan las tablas de Bitácora y Bitácora_resultados (módulo almacenador).

Mediante este operador podemos responder a consultas como: ¿Cuál es la temperatura máxima registrada en el estado de Puebla?, ¿Cuál es el índice de consumo de cereales en el mes de mayo del 2003? Un ejemplo de cómo la operación convierte la consulta al lenguaje SQL se muestra enseguida:

```
select max(v.maxima)
from ventas v
where v.codciudad in
        ( select codciudad from ciudedo cd, estado e1 where e1.idestado =
       cd.idestado and e1.nomestado = 'PUEBLA' and cd.codciudad in
              (select t.codciudad from temperatura t, ciudedo ce, estado e2 where
              e2.idestado = ce.idestado and t.codciudad = ce.codciudad and
              e2.nomestado = 'PUEBLA'))
```

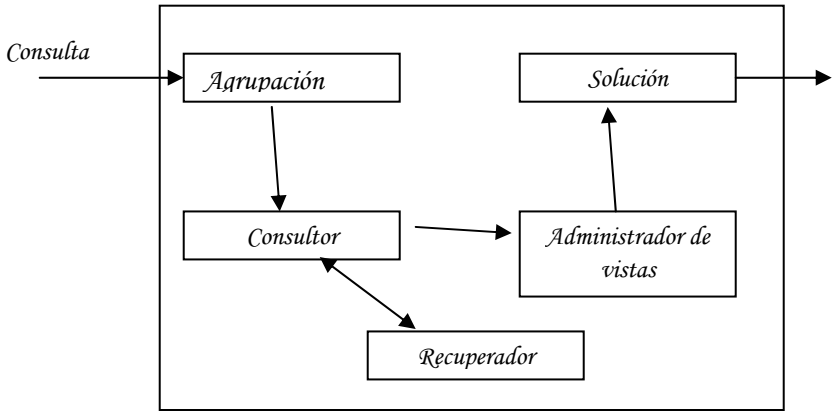
En el ejemplo se puede ver como se restringe la consulta solamente a una dimensión del cubo de temperaturas.

Como podemos observar, con dicho operador podemos aislar las dimensiones del cubo original, es decir, se hace un cubo mas pequeño, sólo con las dimensiones involucradas, restringiendo los valores dentro de los granos, hecho que agilizará el tiempo de respuesta de la consulta.

### 4.4.2 Roll – up

Al inicio de este apartado se mencionaron algunos ejemplos donde se aplicaban los resultados de consultas anteriores para responder a nuevas consultas navegando por los diferentes niveles de agregación. Esta es la función de los operadores *roll-up* y *drill-down*.

La operación *roll-up*, es la encargada de subir en nivel de agregación, ya sea partiendo del nivel más bajo o de uno medio. El diseño para la implementación de este operador esta en la Figura 4.10.




4.10 Diseño del operador *roll-up*

Es importante notar que los módulos son iguales a los del operador *slice'n dice*, sólo que el *roll-up* tiene un módulo más que es el encargado de revisar si existen datos de una

granularidad menor a la que en ese momento se consulta y en caso de ser cierto esto, recuperar dicha información para facilitar el nuevo cálculo de valores para responder la consulta actual.

La metodología de diseño para ésta operación es muy similar a la operación *slice'n dice.* Una vez que el usuario ha seleccionado los granos, dimensiones y medida dentro de la interfaz se reestructura la consulta en lenguaje SQL. Teniéndose los valores de la consulta se verifica si se tienen resultados de esa consulta (módulo consultor).

En caso de que se tenga la consulta se pasa directamente el resultado y los valores involucrados con la consulta al módulo almacenador el que actualiza las tablas Bitácora y Bitácora_resultados. En caso de que no se tenga almacenada la consulta se busca en la tabla Bitácora si se tiene almacenada una consulta de un grano superior, de ser esto cierto entonces se hace una sub-consulta que recupera el valor preguntado en la consulta original (módulo recuperador).Y por último se crea una nueva vista y se almacena la nueva consulta con su resultado en las tablas correspondientes y se presenta el resultado en la interfaz.

Por ejemplo en la dimensión de región se refiere a subir del grano de ciudad al de estado, o bien del estado al de zona. Para ejemplificar esto podemos imaginar que ya se hizo la consulta de la temperatura máxima del estado de Puebla y esta vez se quiere la temperatura máxima de la ciudad del mismo nombre. Mediante el operador *roll-up*, se puede subir de nivel de agregación empleando la consulta anterior para facilitar el proceso de ésta. En lenguaje SQL, la consulta se vería así:

*select max(v.maxima)*
*from consulta1 v*
*where v.codciudad in (select c.codciudad from ciudad c where c.nomciudad = 'PUEBLA')*

donde consulta1 es:

*select codciudad*
*from ciudedo cd, estado e1*
*where e1.idestado = cd.idestado and e1.nomestado = 'PUEBLA' and cd.codciudad in (select t.codciudad from temperatura t, ciudedo ce, estado e2 where e2.idestado = ce.idestado and t.codciudad = ce.codciudad and e2.nomestado = 'PUEBLA')*

Para aclarar lo anterior podemos suponer que el productor de huevo primero consulta el índice de consumo de huevo en la ciudad de Cholula durante el año 2003. Su consulta será realizada y almacenada por DOMINIQUE, pero ahora al mismo productor le puede surgir la necesidad de preguntar por el índice de consumo de huevo durante el año 2003 en todo el estado de Puebla; entonces de la vista materializada donde se encuentran dichos datos selecciona únicamente las tuplas que corresponden al valor de dicha ciudad, y crea una nueva vista materializada con estos valores.

### 4.4.3 Drill – down

El esquema de diseño para el operador *drill-down*, es muy similar al que emplea el operador *roll-up*, como se muestra en la Figura 4.11.

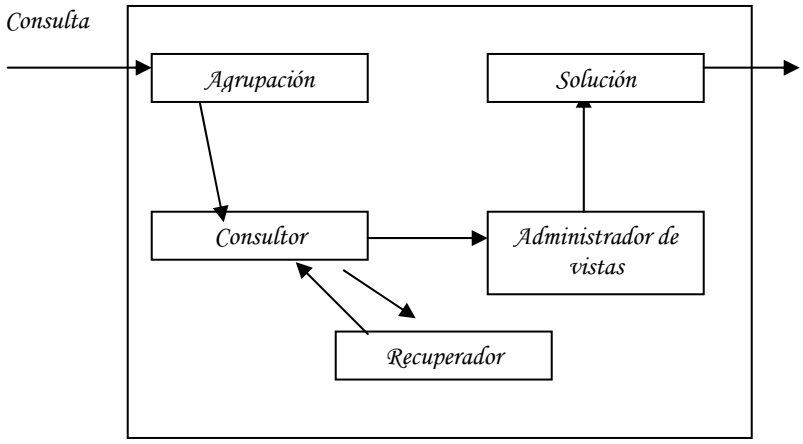


Figura 4.11 Diseño de operador *drill-down*

El operador *drill-down*, es la operación inversa a *roll-up*. Es decir, aprovecha consultas anteriores de granularidades más altas para la obtención de los resultados de una consulta de granularidad más baja, por ejemplo, si anteriormente se pregunto por el grupo de los derivados lácteos y huevo, en esta ocasión será mucho más rápido responder a una consulta sobre el índice de consumo de derivados de huevo, simplemente bajando de nivel de granularidad.

La diferencia entre ambos operadores reside en el módulo de recuperación. En *drill-down*, se busca en un grano inferior y se recuperan los valores correspondientes al grano superior, y sólo se hace la consulta con los granos que hagan falta, para clarificarlo se presenta a continuación un ejemplo.

Imaginemos el ejemplo inverso mencionado en el operador *roll –up.* Esta vez podemos suponer que el productor de huevo realiza una primera consulta preguntando por el índice de consumo de huevo en el estado de Puebla durante el año 2003. Inmediatamente después se solicita al sistema la misma consulta sólo que esta vez únicamente le interesa la ciudad

de Cholula. Lo que DOMINIQUE haría para responder a esta última consulta, es buscar si existe una consulta sobre el estado al que pertenece a la ciudad de Cholula y, como esto es cierto, esta vez el sistema no calculará todo, sino que tomará el resultado de obtenido por la primera consulta y sólo calculará el índice de las ciudades restantes correspondientes al estado.

Como ejemplo también se puede imaginar que se desea saber la temperatura máxima de la zona centro del país, entonces lo que se hace es calcular primero la temperatura de cada uno de los estados correspondientes a esta zona y en base a esos resultados se calcula la máxima de la zona. La consulta en lenguaje SQL se muestre a continuación:

```
select max(t.maxima)
 from (select c.codciudad from ciudedo c
where c.idestado in
        (  select z.idestado from zona z where z.idestado in  (select z1.idestado from
        zona z1 where z1.nomzona = 'CENTRO'))  and c.codciudad in
                (select c1.codciudad from ciudedo c1, temperatura t where
                t.codciudad = c1.codciudad and c1.codciudad in
                        (  select z.idestado from zona z where z.idestado in
                                (select z1.idestado from zona z1 where z1.nomzona =
                                'CENTRO'))group by z.idestado))
```

## 4.4 Discusión

DOMINIQUE es un DW que maneja grandes cantidades de información de la prensa electrónica, la cuál debe de encontrarse agregada para lograr su finalidad de ser un sistema de apoyo a la toma de decisiones.

Sin lugar a duda, una de las ventajas principales que presenta DOMINIQUE es su motor de consulta con técnicas OLAP que facilitan un rápido acceso a las consultas y una administración de las mismas. A su vez también existe un inconveniente en el diseño de los operadores OLAP, y es que se debe definir las dimensiones sobre las que se es más usual bajar o subir de granularidad, esto refiriéndonos a las consultas que emplean dos operadores a la vez. Por último, con el diseño de construcción y análisis mostrado en éste capítulo la implementación de DOMINIQUE se aborda en el siguiente.