Especificación formal de cubos de datos aplicados a la administración de las actividades docentes

Formal specification of data cubes applied to the administration of teaching activities

Pedro Fletes Gudiño
Instituto Tecnológico de Colima
pfletes@itcolima.edu.mx

Nicandro Farías Mendoza
Instituto Tecnológico de Colima
nfarias@itcolima.edu.mx

Resumen

Durante los últimos años las bases de datos multidimensionales se han convertido en algo común en el mundo de los negocios y científico. En este trabajo se propone un modelo para la especificación formal de base de datos multidimensionales (BDMD) o cubos de datos, que nos permita a través de un proceso clasificar, filtrar información del área de docencia, seccionando los registros en ejes y capas. La meta que se pretende alcanzar a través de la propuesta de este modelo es la especificación formal del cubo de datos.

Palabras clave: base de datos multidimensionales, cubos de datos, docencia y modelo.

Abstract

During the last years the multidimensional databases have become something common in the business and scientific world. In this work is proposed a model for the formal specification of Multidimensional Database (MDB) or data cubes, which allows us to through a process sort, filter information in the area of teaching, sectioning records in

axes and layers. The goal that intends to achieve through the proposal of this model is the formal specification of the data cube.

Keywords: multidimensional database, data cubes, teaching and model.

Fecha recepción: Febrero 2013 **Fecha aceptación:** Mayo 2013

Introducción

Razones que motivaron la elección del tema

Durante los últimos veinte años mi principal función ha sido como docente en el Instituto Tecnológico de Colima, siempre en el departamento de Sistemas y Computación. Una de las áreas que desde siempre como licenciado en informática me ha entusiasmado son las bases de datos y es por dicha razón este proyecto de tesis fue elegido para continuar con la temática actual de las bases de datos multidimensionales.

Contexto del problema

Actualmente a nivel nacional, no siendo la excepción nuestro instituto, se está viviendo una problemática por los altos índices de reprobación de los alumnos. Desde siempre se han buscado diversas maneras de disminuir estos índices, pero a nivel institucional no existe una herramienta computacional para llevar un control sobre estos, que además proporcione información estadística de distintos años y desde distintos parámetros. Esto nos llevó a elaborar nuestra propuesta que generará, a través de la especificación formal, un esquema computacional (cubo de datos), capaz de almacenar información relevante sobre los anteriores índices de reprobación y arrojar información

de manera rápida y eficaz sobre las diversas causas que originan la reprobación en el instituto.

Objetivo

Desarrollar un modelo formal para la especificación de Bases de Datos Multidimensionales que pueda ser representado con SQL[1] para un caso de estudio orientado a la Administración de las actividades Docentes.

Objetivos específicos

- Definir el modelo formal.
- Especificar el caso de estudio.
- Detallar el flujo de datos para el caso de estudio.

Estado del conocimiento

Marco histórico

Este proyecto tuvo sus inicios en el Departamento de Sistemas y Computación en la Jefatura de Proyectos de Docencia, donde cada semestre se lleva un control de las planeaciones de las materias ofrecidas. Los docentes al inicio del semestre entregan la planeación de cada una de las materias que impartirán así como las instrumentaciones didácticas de cada una de las unidades. Durante el transcurso del semestre, el docente entrega avances de calificaciones y al final del curso entrega las actas de calificaciones, donde aparece el índice de aprobación y reprobación del grupo.

[1] Lenguaje de Consulta estructurado

1.2 Marco contextual

En este capítulo, se presenta una breve descripción de otros trabajos en modelado conceptual y lógico de bases de datos multidimensionales. Para facilitar la comprensión de dichos trabajos y unificar terminología, previamente se presentará una introducción a las estructuras y operaciones de los modelos multidimensionales.

Base de datos multidimensionales

Son bases de datos ideadas para desarrollar aplicaciones muy concretas, como creación de Cubos OLAP (Procesamiento analítico en línea). Básicamente no se diferencian demasiado de las bases de datos relacionales (una tabla en una base de datos relacional podría serlo también en una base de datos multidimensional), la diferencia está más bien a nivel conceptual; en las bases de datos multidimensionales los campos o atributos de una tabla pueden ser de dos tipos, o bien representan dimensiones de la tabla, o bien representan métricas que se desean estudiar.

Bases de datos multidimensionales vs. cubos OLAP

Cada una de estas tablas puede asimilarse a un hipercubo o más concretamente si de herramientas OLAP se trata a un cubo OLAP, donde las dimensiones del mismo corresponden a los campos de dimensiones de la tabla (campos 'd_i...'), y el valor almacenado en cada celda del cubo equivale a la métrica o métricas (campos 'f_i...') almacenadas en la tabla.

Este tipo de base de datos se aplica sobre el sistema OLAP también llamado cubo multidimensional o hipervínculo. Se compone de hechos numéricos llamados medidas que se clasifican por dimensiones. El cubo de metadatos es creado típicamente a partir

de un esquema en estrella (ver Fig. 1) o copo de nieve (ver Fig. 2) y utilizando tablas de una base de datos relacional.

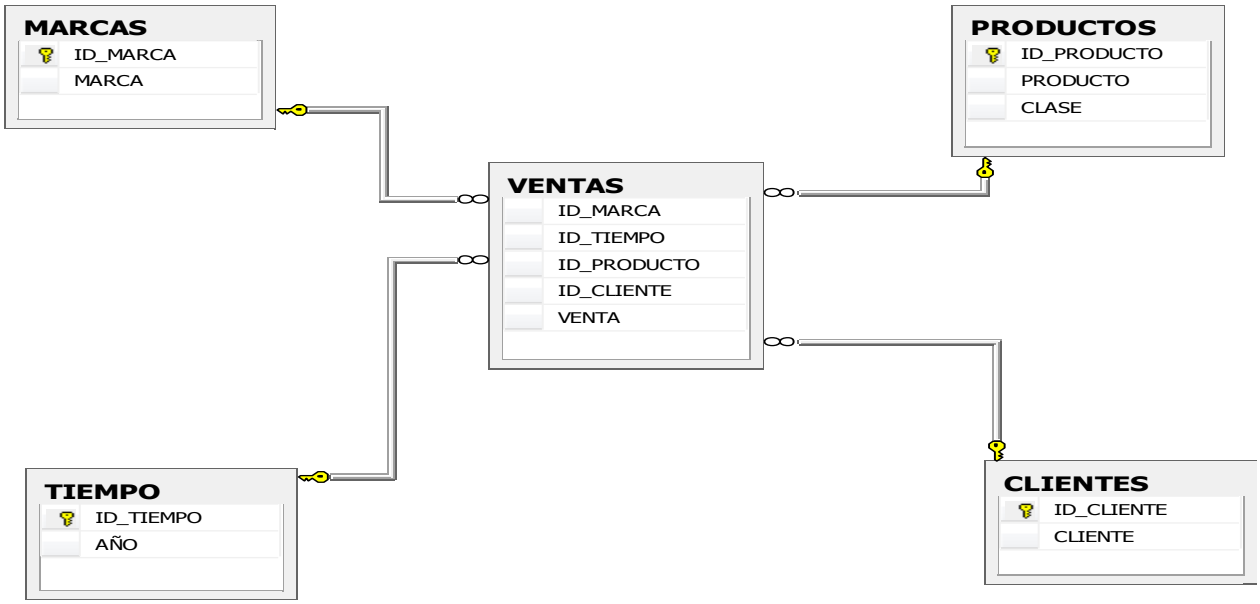

Fig.1 Esquema de estrella

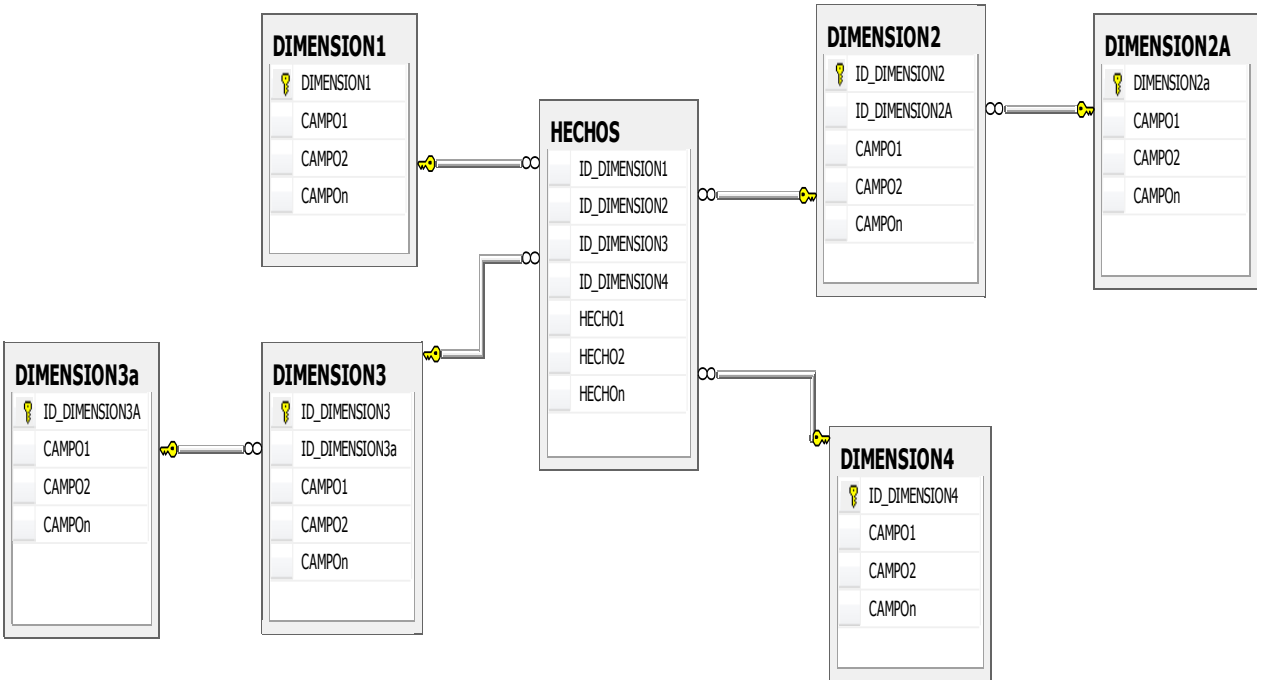

Fig.2 Esquema de copo de nieve

Los trabajos relacionados y sus aportaciones

(Seng & Habibollah, 2013) proponen utilizar KSOM para organizar los datos no estructurados para superficies cerradas. KSOM se utiliza en este trabajo mediante pruebas de su capacidad para organizar los datos de imagen médica porque KSOM se utiliza sobre todo para construir los datos de campo de la ingeniería. (Hsiao, Wo-Shun, & Petchulat, 2011) nos muestran un prototipo funcional de un sistema OLAP centrado en el cliente, que cuenta con un middleware personalizado en el lado del servidor y un cliente web que incorpora un motor ligero de datos OLAP para consultas en memoria. Ellos desarrollaron tres herramientas de visualización de datos interactivos que se ejecutan en el motor de datos en el lado del cliente. (Mansmann & Scholl, 2007) nos presentan un enfoque para explorar cubos de datos multidimensionales con técnicas de visualización jerárquicos. Un trabajo cuya principal contribución es un enfoque llamado diagrama de dispersión diferencia jerárquica (HDS). Esta propuesta permite múltiples niveles de jerarquía relativa y explícitamente visualiza las diferencias entre ellos en el contexto de la posición absoluta de los valores pivotantes; es presentado por (Piringer, Buchetics, Hauser, & Gröller, 2009). (Lafon, Bouali, Guinot, & Venturini, 2013) nos presentan varias maneras de reorganizar un cubo OLAP en función de las cuales se selecciona un conjunto de miembros de la reorganización: de la totalidad de los miembros, solo se muestran los miembros de un determinado nivel (nivel de enfoque a nivel). El trabajo que propone combinar el recorrido de las celdas de la dimensión y pruebas estadísticas paramétricas para identificar diferentes métricas significativas entre celdas del cubo, es presentado por (Ordonez, Chen, & García-García, 2011).

(Takama & Yamada, 2009) nos proponen un cubo de visualización para el modelado de la interacción en el análisis exploratorio de información de tendencias espacio-temporal. (Pitarch, Laurent, & Poncelet, 2009) proponen un modelo conceptual para modelar jerarquías personalizadas en bases de datos multidimensionales. La propuesta de un algoritmo de construcción de elipsoides basado en (ER-Tree) presentado por (Dankoand & Skopal, 2009) demuestra que estos afectan significativamente la velocidad, la indexación y el rendimiento de las consultas en bases de datos grandes como las de multimedia, medicina y geográficas. (Stolte, Tang, & Hanrahan, 2006) proponen un lenguaje de consulta visual que permita la visualización gráfica de datos basados en tablas. Este lenguaje compila tanto los comandos de las consultas como los gráficos necesarios para generar la visualización, lo que nos permite diseñar sistemas que integren estrechamente análisis y visualización. (Yaghmaie, Bertossi, & Ariyan, 2012) proponen la formalización de la trayectoria del esquema relacional, que se convierte en la base para la obtención de reparaciones dimensionales, aquí se muestra que la estrella común relacional y los esquemas de copo de nieve multidimensionales para bases de datos, no son la mejor opción para este proceso. Un novedoso sistema TEXplorer permite a los usuarios realizar búsquedas de palabras clave y la agregación de OLAP y de estilo, la exploración de los objetos de texto en un cubo construido sobre un texto multidimensional de la base de datos que nos presentan (Zhao, Lin, Ding, & Han, 2011). (Esch-Bussenmarkers & Cremers, 2004) proponen un tutorial de un sistema experimental, donde el objetivo del proyecto fue evaluar las diferentes modalidades de acceso (voz, gráficas y modo táctil) para acceder y presentar ciertos

tipos de información, y para ciertas estrategias de búsqueda al buscar y navegar en una base de datos musical multidimensional, utilizando un dispositivo móvil simulado.

Marco teórico

A continuación se detallarán los conceptos incluidos en este proyecto:

Bases de datos

Un sistema de gestión de base de datos (DBMS Data Base Management System) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos (Silberschatz, Kotrh, & Sudarshan , 2002, pág. 1). Una colección compartida de datos lógicamente relacionados, junto con una descripción de estos datos, que están diseñados para satisfacer las necesidades de información de una organización (Connolly & Begg, 2005, pág. 15).

Base de datos multidimensionales

Base de datos de estructura basada en dimensiones orientada a consultas complejas y alto rendimiento. Puede utilizar un SGBDR en estrella (Base de datos Multidimensional a nivel lógico) o SGBDM (Base de datos Multidimensional a niveles lógico y físico o Base de datos Multidimensional Pura) (De la Herrán Gascón, 2004).

Cubos de datos

El Cubo OLAP, que acuña su nombre por su característica multidimensional, es una base de datos que posee diversas dimensiones.

Los cubos de datos se utilizan en los sistemas de procesamiento analítico en línea (OLAP) para apoyar la toma de decisiones. Construido a partir de las bases de datos de

un negocio, este sistema de visualización interactiva presenta una estructura cúbica 1D condicional jerárquica de árbol para representar a los cubos de datos y el uso de iconos gráficos 2D para ilustrar los elementos de datos. Los usuarios pueden entonces explorar interactivamente datos multidimensionales en los niveles jerárquicos.

Desarrollo del cubo de datos aplicado a la administración de las actividades docentes

Análisis del sistema

En todos los institutos tecnológicos a nivel nacional, existen departamentos académicos que administran las carreras que se ofrecen, y en cada uno de ellos existen jefaturas de docencia. Actualmente no existe un sistema automatizado que nos permita hacer un registro de datos y después obtener información, como la que hoy se demanda.

Por ello, no es posible obtener información de control (estadísticas de aprobación, reprobación y deserción) de cada uno de los docentes (por materia, hora de la clase, semestre), de las carreras, del departamento y del Instituto. Además, tampoco es posible obtener fácilmente información de semestres anteriores.

Modelo conceptual

El esquema multidimensional propuesto (ver Figura 1) muestra cómo a partir de la información del SIITEC[2] y de las diferentes áreas de docencia de los departamentos académicos, la información va sufriendo transformaciones y segmentándose (pasos 1 y

[2] Sistema de Información Integral del Instituto Tecnológico de Colima.

2) hasta llegar al cubo de datos (paso 3) y ya ahí se puede procesar para realizar las consultas que se requieran (paso 4).

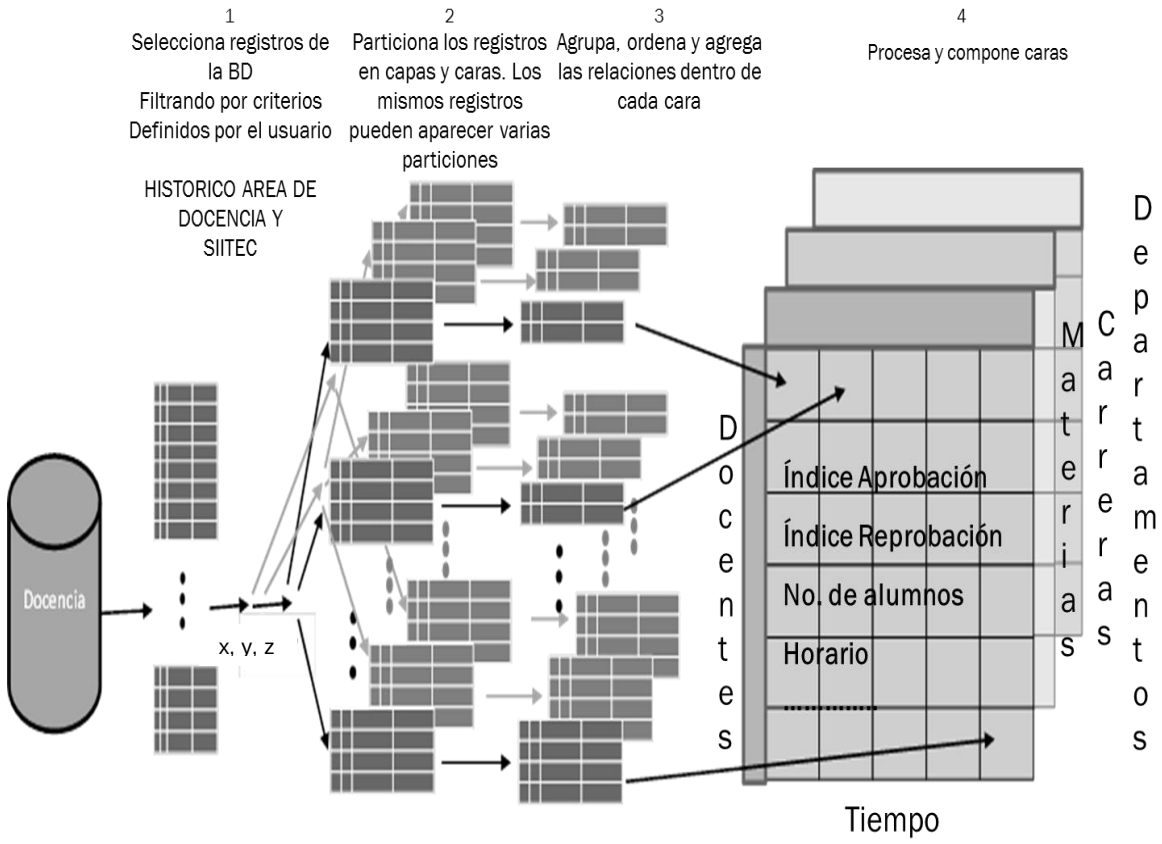

Figura 1 Big Picture del esquema multidimensional propuesto.

Requerimientos del sistema

En este punto se considera toda la información que se puede obtener del cubo de datos para evaluar los aspectos por los cuales ocurre la reprobación de alumnos en cada una de las materias que cursa, durante su estancia en el Instituto y que va desde:

- Materias con mayor índice de reprobación
- Horarios con mayor índice de reprobación

- Docentes con mayor índice de reprobación
- Aspectos que influirían en la reprobación
  - Estado civil de los alumnos.
  - Situación laboral de los alumnos (si tienen algún trabajo).
  - Género de los alumnos.
  - Lugar de residencia de los alumnos (locales o foráneos).
  - Beca (si el alumno cuenta con algún apoyo).
  - Si cuentan con alguna carrera anterior.
  - El horario de la materia.

Toda esta información puede ser obtenida desde el punto de vista de cada una de las dimensiones del cubo: docentes, tiempo, departamento, carrera y materia.

Modelo para la especificación formal de Base de Datos Multidimensionales (BDMD) o cubos de datos

El m**odelo** para la especificación formal de c**ubos de datos** está formado por el siguiente quíntuple:

MCD[3] = (C, OAT, OPAT, PR, OBD)

En donde:

C = {campos ordinales, campos cuantitativos, dimensiones, mediciones} son los conceptos básicos.

OAT = {A, B, C, P, Q, R} son los operandos del álgebra de tablas.

[3] Modelo de cubo de datos.

OPAT = {+, X, /} son los operadores del álgebra de tablas.

PR = {Filtro, *Fila (i), Columna (j), Capa (k)*} son los predicados.

OBD = {Agregados, dim, SUM, AVG} Operadores de la Base de Datos.

Este modelo propuesto consiste en un formalismo basado en un álgebra de tablas que captura la estructura de las tablas en modelo relacional y realiza la conversión de estas a un formato de cubos de datos. Conceptos utilizados:

Campos Ordinales: La escala ordinal es representada en forma discreta, como encabezados o clases diferentes.

Campos cuantitativos: Los campos cuantitativos son continuos y mostrados como ejes o como valores con una variación continua.

Álgebra de tablas

Definimos un álgebra como el mecanismo formal para especificar la configuración de las tablas. Una configuración completa consiste de tres expresiones separadas en el álgebra de tablas. Dos de las expresiones definen la configuración del eje ***X*** y del eje ***Y*** de la tabla seccionando la tabla en ejes y columnas. La tercera expresión define el eje de las ***Z*** de la tabla, la cual parte el desplegado en capas. Las expresiones ***X, Y*** *y* ***Z*** forman las cláusulas en el lenguaje.

Los operandos en el álgebra de tablas son nombres de campos ordinales o cuantitativos de la base de datos. Se utiliza ***A, B y C*** para representar campos ordinales y ***P, Q*** *y* ***R*** para representar campos cuantitativos. Asignamos secuencias de valores a cada símbolo de campo de la siguiente manera: a los campos ordinales asignamos los

miembros de un dominio ordenado del campo y a los campos cuantitativos se asignan conjuntos de elementos individuales conteniendo el nombre del campo.

Los campos ordinales y cuantitativos generan tablas con diferentes estructuras (1)-(2). Los campos **ordinales** segmentan la tabla en renglones y columnas utilizando encabezados, mientras que los campos **cuantitativos** generan ejes.

$$\boldsymbol{A = domain\ (A) = \{a_1,...,a_n\}} \quad (1)$$
$$\boldsymbol{P = \{P\}} \quad (2)$$

Una expresión válida en el álgebra consiste de uno o más símbolos con operadores entre cada par de operandos adyacentes y con paréntesis utilizados para alterar la precedencia de los operadores.

Operadores del álgebra de tablas

Concatenación (+) El operador **+** (3)-(5) concatena dos secuencias de la siguiente forma:

$$\boldsymbol{A} + \boldsymbol{B} = \{a_1, ..., a_n\} + \{b_1, ..., b_m\} \quad (3)$$
$$= \{a_1, ..., a_n, b_1, ..., b_m\}$$

$$\boldsymbol{A} + \boldsymbol{P} = \{a_1, ..., a_n\} + \{P\} \quad (4)$$
$$= \{a_1, ..., a_n, P\}$$

$$\boldsymbol{P} + \boldsymbol{Q} = \{P\} + \{Q\} \quad (5)$$
$$= \{P, Q\}$$

Producto: (X) El operador ***Producto*** (6)-(7) realiza el producto cartesiano de dos secuencias.

$$\boldsymbol{A} \times \boldsymbol{B} = \{a_1, ..., a_n\} \times \{b_1, ..., b_m\} \quad (6)$$
$$= \{a_1b_1, ..., a_1b_m,$$
$$a_2b_1, ..., a_2b_m, ...,$$

$a_n b_1, ..., a_n b_m\}$

$$\boldsymbol{A} \times \boldsymbol{P} = \{a_1, ..., a_n\} \times P = \{a_1 P, ..., a_n P\} \quad (7)$$

Proyección (*/*) El operador **Proyección** (8) es similar al operador producto, pero solo crea secuencias para las cuales existen registros.

$$A/B = \{a_i b_j \mid \exists r \in R \text{ st } A(r) = a_i \ \& \ B(r) = b_j\} \quad (8)$$

La interpretación intuitiva del operador **Proyección B** dentro de **A**.

Por ejemplo, dados los campos trimestre y mes, la expresión trimestre/mes podría interpretarse como aquellos meses dentro de cada trimestre, resultando en tres entradas para cada trimestre. En contraste, trimestre X mes podría resultar con 12 entradas para cada trimestre. Los cubos de datos representan jerarquías explícitamente y no es necesario calcular la relación de proyección.

Transformación de datos y generando consultas en la base de datos

Un aspecto importante del formalismo propuesto es la transformación de datos. Entonces el formalismo debe soportar el rango completo de transformaciones de datos en una consulta en un lenguaje de consultas como SQL, incluyendo los operadores relacionales comunes: selección, filtrado, agrupación, agregación y ordenamiento.

Flujo de datos total en el modelo MCD

El formalismo debe ser compatible con la gama completa de posibles transformaciones de datos en un lenguaje de consulta como SQL, incluyendo los operadores relacionales comunes: selección, filtrado, agrupación, agregación y clasificación. Se puede

demostrar que cualquier consulta definida en SQL se puede expresar como una especificación en el formalismo MCD.

En esta propuesta, los campos medida se agregan mientras que los campos de dimensión se insertan en un estatuto ***GROUP BY***, con campos de dimensiones adicionales especificados en el nivel de detalle. Cada dimensión también puede ser ordenada, y las diferentes funciones de agregación se pueden asociar a cada medida y estas opciones pueden ser elegidas. También hay un filtro que representa los elementos en la cláusula WHERE. Por último, también se exponen los cálculos generales y joins.

Tomando como referencia la Figura 1, se explica a detalle cada uno de los pasos mostrados en la figura descrita a través del modelo MCD.

Paso1. Seleccionando los registros. La primera fase del flujo de datos recupera registros de la base de datos, aplicando filtros definidos por el usuario para seleccionar subconjuntos de datos.

Para un campo ordinal A, el usuario puede especificar un subconjunto del dominio del campo como si filter(A) es un subconjunto seleccionado por el usuario. Entonces **un** predicado relacional expresando el filtro para A es:

A in *filter* (A)

Para Un campo cuantitativo P, el usuario puede definir un subconjunto del domino del campo como válido si $min(P)$ y $max(P)$ son extensiones definidas por el usuario del subconjunto. El predicado relacional que expresa el filtro para P es:

($P \geq min(P)$ and $P \leq max(P)$)

Podemos definir el predicado relacional **filtro** como una conjunción de todos filtros de los campos individuales. Entonces, la primera etapa de la red de transformación de datos es equivalente a la siguiente declaración en SQL:

SELECT *
WHERE {*filtros*}

Es posible dentro del formalismo completo definir un filtrado más sofisticado, tal como filtros del producto cruz de campos múltiples o filtros con dependencias de ordenamiento (el filtro A es calculado relativo al filtro B).

Paso 2. Partición de los registros en paneles: La segunda fase de las particiones del flujo de datos de los registros recuperados en grupos corresponden a cada panel en la tabla. La tabla se divide en filas, columnas y capas correspondientes a las entradas en estos conjuntos.

Los valores ordinales en cada entrada establecidos definen los criterios por los cuales los registros se ordenarán en cada fila, columna y capa. Dada la Fila (i), será el predicado que representa los criterios de selección para la fila i-ésima, columna (j) será el predicado de la columna j-ésima, y Capa (k) el predicado para la k-ésima capa. Por ejemplo, si el eje **Y** de la tabla está definida por el conjunto normalizado:

$\{a_1b_1P, a_1b_2P, a_2b_1P, a_2b_2P\}$

A continuación hay cuatro filas de la tabla, cada uno definido por una entrada en este grupo, y la fila se definiría como:

Fila (1) = (A = a_1 and B = b_1)
Fila (2) = (A = a_1 and B = b_2)
Fila (3) = (A = a_2 and B = b_1)
Fila (4) = (A = a_2 and B = b_2)

Dadas estas definiciones, los registros en que se divide el panel es la intersección de la fila i-ésima y la columna j-ésima, y la capa k-ésima se pueden recuperar con la siguiente consulta:

SELECT *
WHERE {*Fila (i)* and *Columna (j)* and *Capa (k)*}

Paso 3. Transformación de registros dentro de los paneles: La última fase del flujo de datos es la transformación de los registros en cada panel. Si la especificación visual incluye la agregación, entonces cada medida en el esquema de base de datos debe ser asignada a un operador de agregación.

Por ejemplo, si la base de datos contiene los campos cuantitativos ganancias, ventas y nómina, y el usuario ha especificado de forma explícita que el promedio de ventas debe ser calculado, entonces los agregados se definen como:

Agregados = SUM(Ganancias), AVG(Ventas), SUM(Nómina)

Los Filtros de los campos agregados (por ejemplo, SUM (Ganancias > 500), no se pudo evaluar en el Paso 1 con todos los otros filtros, porque los agregados aún no se habían calculado. Por lo tanto, los filtros se deben aplicar en esta fase. Definimos los filtros predicados relacionales como en el paso 1 para los campos no agregados.

Además, se definen las siguientes listas:

G: Los nombres de campo en la plataforma de agrupación,

S: Los nombres de campo en la plataforma de clasificación, y

Dim: Las dimensiones en la base de datos.

La transformación necesaria se puede expresar por la sentencia SQL:

SELECT {dim}, {agregados}
GROUP BY {G}
HAVING {*filters*}
ORDER BY {S}

Si los campos agregados no se incluyen en la especificación, entonces la transformación restante simplemente ordena los registros en orden:

SELECT *
ORDER BY {S}

Diseño

Debido a que la propuesta es la generación del esquema de un cubo de datos para la explotación de información en el área de la docencia a través de una especificación formal, se toma como entrada un esquema de base de datos relacional. En la Figura 2 se muestra el esquema multidimensional donde se especifica cada una de las dimensiones propuestas, así como las medidas de la tabla de hechos (GRUPO).

Diseño arquitectónico

Se propone que la tabla de hechos GRUPO contenga además de los atributos ya mencionados (Índice de aprobación, índice de reprobación, número de alumnos y horario de la materia), otros datos adicionales que complementen la información y nos ofrezcan la posibilidad de extraer información desde diferentes puntos de vista. En la Figura 2 se muestra el modelo conceptual del esquema propuesto. En el siguiente apartado se especifican todos los atributos de cada una de las dimensiones y de la tabla de hechos.

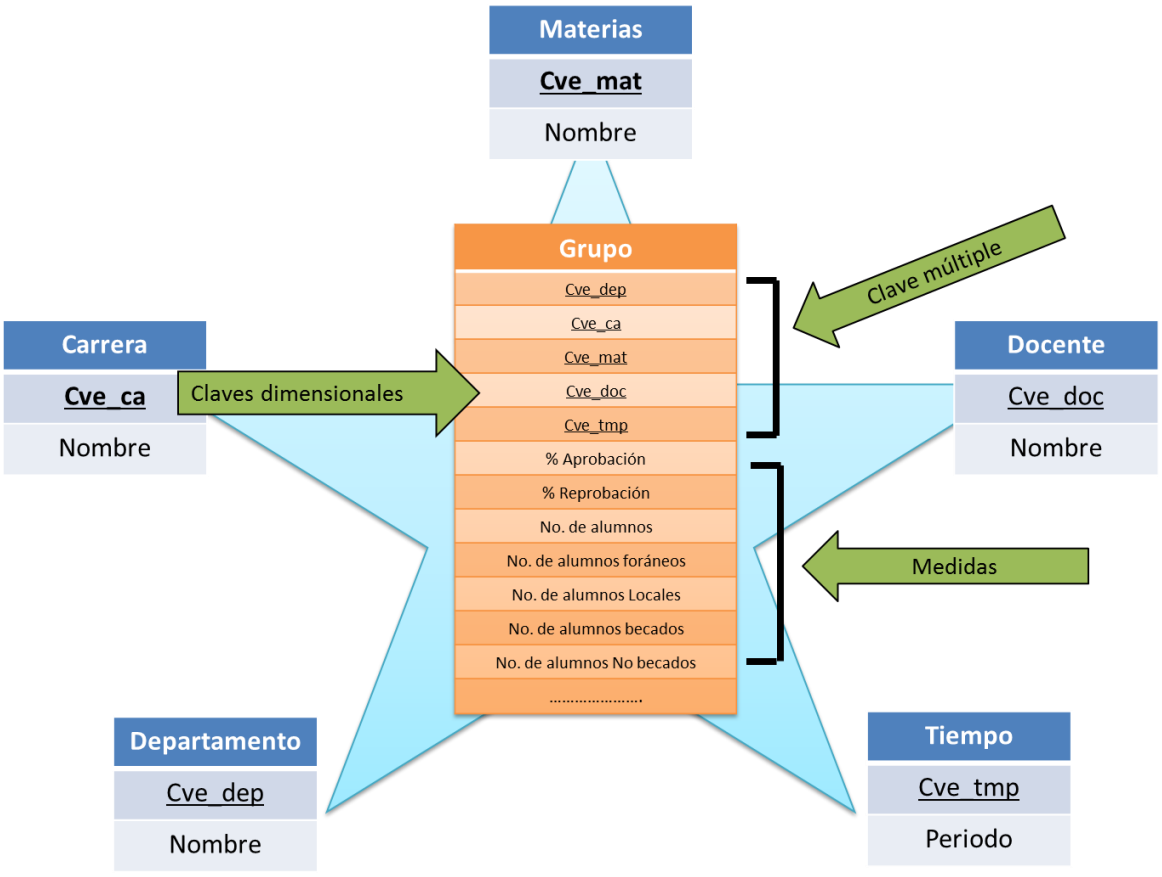

Figura 2 Modelo Conceptual.

Diseño del cubo de datos

Tomando como referencia el modelo conceptual de la Figura 2, el diseño propuesto se muestra en el siguiente diagrama E-R (ver Figura 3).

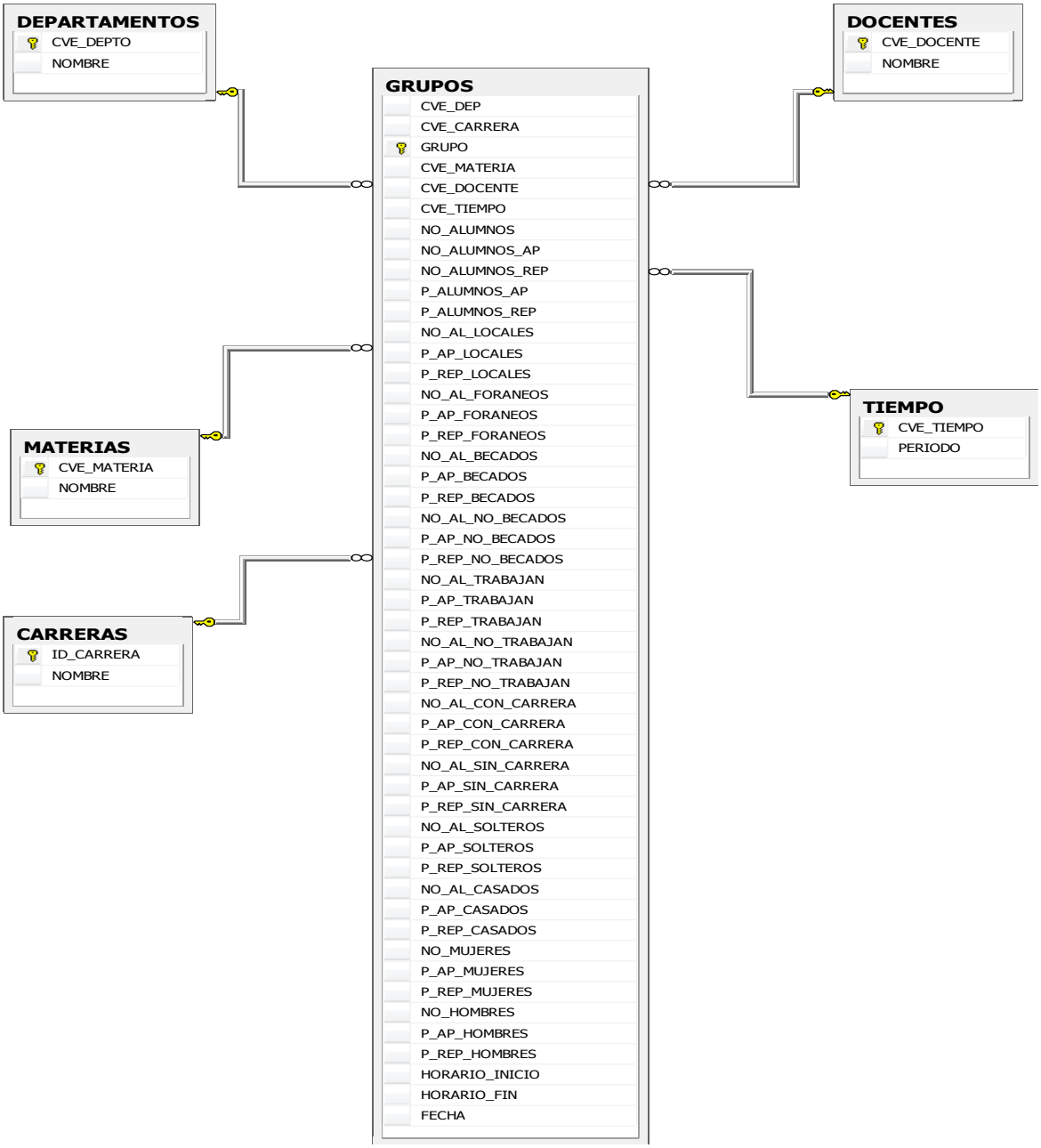

Figura 3 Diagrama de E-R del cubo de datos.

No = Número, P = Porcentaje, AP = Aprobación, REP = Reprobación.

Caso de estudio

El siguiente caso muestra cómo a partir del esquema relacional de la Figura 4 y utilizando el Modelo MCD se llega al cubo de datos de la Figura 3.

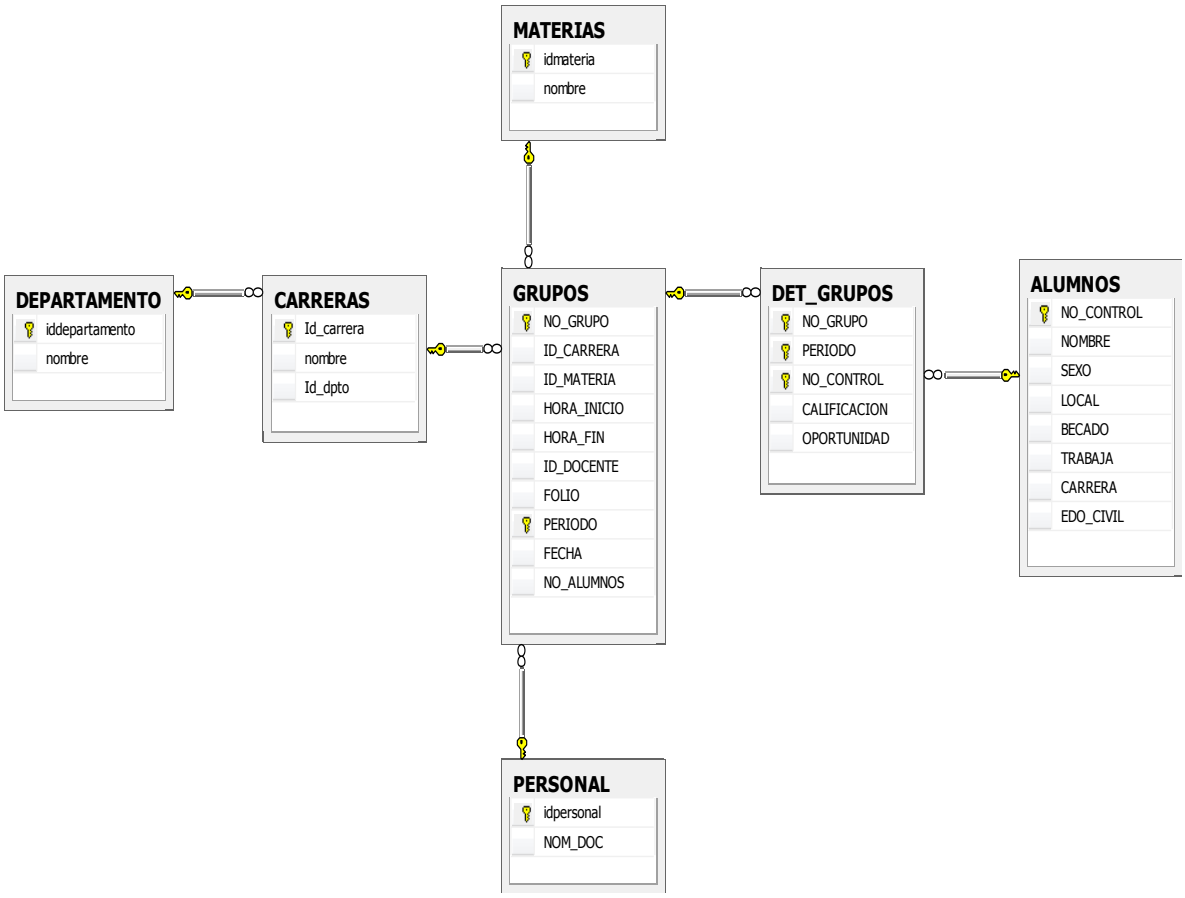

Figura 4. Modelo relacional de docencia.

En el caso mostrado a continuación se visualiza la generación de las dimensiones del cubo y la tabla de hechos (aquí se obtiene solamente la clave del grupo, las claves de las dimensiones –carrera, materia, docente, tiempo y departamento–, la hora de inicio, hora de fin de la clase, folio, el número y el porcentaje de alumnos aprobados).

Utilizando instrucciones de SQL el código sería el siguiente:

```
Select  L.No_Grupo, L.Id_Carrera, L.Id_Materia, L.Hora_inicio, L.Hora_fin,
        L.Id_Docente,  L.Folio, L.Fecha, L.No_Alumnos, P.Iddepartamento,
        Count (*), Count (*)/L.No_Alumnos
From Grupos L join  Det_Grupos M     on L.No_Grupo    = M.No_Grupo
                join  Alumnos  N     on M.No_control  = N.No_Control
                join Carreras O      on L.Id_Carrera  = O.Id_Carrera
                join Departamento P  on O.Id_Dpto     = P.Iddepartamento
Group By L.No_Grupo
Having M.Calificacion >= 70
```

Utilizando el método MCD propuesto, la especificación sería de la manera siguiente:

Select A_1, A_2, A_3, B_1, B_2, A_4, B_3, B_4, P_1, A_5, Q_1, Q_2
From Grupos L / Det_Grupos M / Alumnos N / Carreras O / Departamento P
Group By A_1
Having $Filtro_1$

Donde:

A_1 = L.No_Grupo	A_4 = L.Id_Docente	Q_1 = COUNT (*)
A_2 = L.Id_Carrera	B_3 = L.Folio	Q2 = COUNT (*) / P1
A_3 = L.Id_Materia	B_4 = L.Fecha	$Filtro_1$ = {M.Calificacion >= 70}
B_1 = L.Hora_inicio	P_1 = L.No_Alumnos	
B_2 = L.Hora_fin	A_5 = P.Idepartamento	

y L, M, N, O y P son variables tupla o alias de las tablas Grupos, Det_Grupos, Alumnos, Carreras y Departamento, respectivamente. Se visualiza en esta consulta que se genera:

- La dimensión Departamentos con el atributo P.Iddepartamento,
- La dimensión Carreras con el atributo L.Id_Carrera,
- La dimensión Materias con el atributo L.Id_Materia,
- La dimensión Docentes con el atributo L.Id_Docente y
- La dimensión Tiempo con el atributo L.Fecha.

Para obtener los atributos No_alumnos_Rep y P_Alumnos_Rep, el filtro se cambiaría a:

$Filtro_1$ = {M.Calificacion < 70}

Para obtener los atributos No_Al_Locales, P_Ap_Locales y P_Rep_Locales, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Local = 'L'} y
Q_2 = COUNT (*)/P_1
Q_3 = 100 – Q_2 y se añadiría al Select

Para obtener los atributos No_Al_Foráneos, P_Ap_Foráneos y P_Rep_Foráneos, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Local = 'F'} y
Q_2 = COUNT(*)/P_1
Q_3 = 100 – Q_2 y se añadiría al Select

Para obtener los atributos No_Al_Becados, P_Ap_Becados y P_Rep_Becados, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Becado = 'S'} y
Q_2 = COUNT(*)/P_1
Q_3 = 100 – Q_2 y se añadiría al Select

Para obtener los atributos No_Al_No_Becados, P_Ap_No_Becados y P_Rep_No_Becados, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Becado = 'N'} y
Q_2 = COUNT(*)/P_1
Q_3 = 100 – Q_2 y se añadiría al Select

Para obtener los atributos No_Al_Trabajan, P_Ap_Trabajan y P_Rep_Trabajan, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Trabaja = 'S'} y
$Q_2 = COUNT(*)/P_1$
$Q_3 = 100 – Q_2$ y se añadiría al Select

Para obtener los atributos No_Al_No_Trabajan, P_Ap_No_Trabajan y P_Rep_No_Trabajan, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Trabaja = 'N'} y
$Q_2 = COUNT(*)/P_1$
$Q_3 = 100 – Q_2$ y se añadiría al Select

Para obtener los atributos No_Al_Con_Carrera, P_Ap_Con_Carrera y P_Rep_Con_Carrera, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Carrera = 'S'} y
$Q_2 = COUNT(*)/P_1$
$Q_3 = 100 – Q_2$ y se añadiría al Select

Para obtener los atributos No_Al_Sin_Carrera, P_Ap_Sin_Carrera y P_Rep_Sin_Carrera, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Carrera = 'N'} y
$Q_2 = COUNT(*)/P_1$
$Q_3 = 100 – Q_2$ y se añadiría al Select

Para obtener los atributos No_Al_Solteros, P_Ap_Solteros y P_Rep_Solteros, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Edo_Civil = 'S'} y
$Q_2 = COUNT(*)/P_1$
$Q_3 = 100 – Q_2$ y se añadiría al Select

Para obtener los atributos No_Al_Casados, P_Ap_Casados y P_Rep_Casados, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Edo_Civil = 'C'} y
$Q_2 = COUNT(*)/P_1$
$Q_3 = 100 – Q_2$ y se añadiría al Select

Para obtener los atributos No_Mujeres, P_Ap_Mujeres y P_Rep_Mujeres, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Sexo = 'F'} y
$Q_2 = COUNT(*)/P_1$
$Q_3 = 100 – Q_2$ y se añadiría al Select

Para obtener los atributos No_Hombres, P_Ap_Hombres y P_Rep_Hombres, el filtro sería:

$Filtro_1$ = {M.Calificación >= 70 and N.Sexo = 'M'} y
$Q_2 = COUNT(*)/P_1$
$Q_3 = 100 – Q_2$ y se añadiría al Select.

Con lo anterior expuesto queda especificado de manera formal el cubo de datos, utilizando el Modelo MCD.

Flujo de datos total de la Big Picture utilizando el caso de estudio

Para explicar cada uno de los pasos de la Big Picture, Figura 1, utilizaremos el ejemplo para la obtención de los datos del modelo relacional con el cubo de datos del apartado anterior.

Si tenemos la siguiente formalización:

Select A_1, A_2, A_3, B_1, B_2, A_4, B_3, B_4, P_1, A_5, Q_1, Q_2
From Grupos L / Det_Grupos M / Alumnos N / Carreras O / Departamento P
Group By A_1
Having $Filtro_1$

Donde:

A_1 = L.No_Grupo	A_4 = L.Id_Docente	Q_1 = COUNT (*)
A_2 = L.Id_Carrera	B_3 = L.Folio	Q2 = COUNT (*) / P1
A_3 = L.Id_Materia	B_4 = L.Fecha	$Filtro_1$ = {M.Calificacion >= 70}
B_1 = L.Hora_inicio	P_1 = L.No_Alumnos	
B_2 = L.Hora_fin	A_5 = P.Idepartamento	

y L, M, N,O y P son variables tupla o alias de las tablas Grupos, Det_Grupos, Alumnos, Carreras y Departamento respectivamente.

Paso 1. Selecciona registros de la BD filtrando por criterios definidos por el usuario.

Este paso queda definido a través del $Filtro_1$ = {M.Calificación >= 70}, en este caso con el filtro se obtienen el número y porcentaje de alumnos aprobados.

Paso 2. Segmenta los registros en capas y caras. Los mismos registros pueden aparecer en varias particiones.

En este paso los datos se agrupan en caras a través de la cláusula:

Select A_1, A_2, A_3, B_1, B_2, A_4, B_3, B_4, P_1, A_5, Q_1, Q_2

From Grupos G / Det_Grupos D / Alumnos A / Carreras C / Departamento P

En el Select las variables A5, A3, A2, A4 y B4, determinan las dimensiones de Departamento, Materia, Carrera, Docente y Tiempo respectivamente. Relacionando los datos a través de la operación Proyección en la cláusula From.

Paso 3. Agrupa, ordena y agrega las relaciones dentro de cada cara.

En este paso agrupan, ordenan para agregarse en cada cara a través de la instrucción:

Select A_1, A_2, A_3, B_1, B_2, A_4, B_3, B_4, P_1, A_5, Q_1, Q_2

Donde cada atributo está definido anteriormente.

Paso 4. Procesa y compone caras.

Ya con la información almacenada en el Cubo de Datos a través de instrucciones de SQL se pueden obtener los indicadores que se requieran.

Análisis de los resultados obtenidos

El alcance de este proyecto se circunscribe a la especificación formal del cubo de datos a través del modelo MCD, el cual a través del caso práctico quedó demostrado.

Otro resultado es que cualquier diseño de un cubo de datos se puede representar y visualizar a través de esta especificación.

Además, sin necesidad de implementar el cubo cualquier diseñador puede plasmar y visualizar el esquema final del cubo de datos.

Conclusiones y recomendaciones

Observando los resultados podemos afirmar que cualquier expresión representada por el lenguaje SQL puede ser especificada formalmente utilizando el Modelo de Cubo de Datos desarrollado en esta investigación. Por lo expresado anteriormente, se observa que se cumplen los objetivos planteados en un inicio y en consecuencia queda plenamente aprobada la hipótesis del trabajo para esta investigación.

La importancia de este proyecto radica en la demostración del diseño de un cubo de datos a través de una especificación formal aplicando el modelo MCD, en este caso como propuesta de solución para poder tener indicadores de los índices de reprobación que permitan conocer las causas de estos índices en las diferentes carreras que ofrece el Instituto Tecnológico de Colima y abatirlos.

Además, es importante señalar que el esquema multidimensional obtenido es analítico y capaz de almacenar tanto información actual como histórica, dando la oportunidad de una administración eficiente de la información derivada de las actividades docentes del Instituto.

Para continuar en el futuro con esta investigación se sugiere:

- Utilizar el otro esquema de representación de un cubo de datos como es el copo de nieve para visualizar las ventajas o desventajas de este enfoque.
- Incrementar las dimensiones de los cubos, lo cual nos proporciona la posibilidad de administrar la información de otros tecnológicos agrupados en regiones geográficas del país.

Referencias

Connolly, T. & Begg, C. (2005). *Sistemas de Bases de datos.* Madrid: Addison Wesley.

Dankoand, T. & Skopal, T. (2009). *Elliptic Indexing of Multidimensional Databases.* Wellington, New Zealand: Australasian Databases Conference.

De la Herrán Gascón, M. (2004). *Red Científica.* Recuperado el 28 de Noviembre de 2013, de http://www.redcientifica.com/oracle/c0001p0005.html

Esch-Bussenmarkers, V. & Cremers, A. (2004). *User Walkthrough of Multimodal Access to Multidimensional Databases.* Pennsylvania, USA: ICMI.

Hsiao, T.; Wo-Shun, L.; & Petchulat, S. (2011). *Data Visualization on Web-based OLAP.* Glasgow,Scotland, UK: DOLAP'2011.

Lafon, S.; Bouali, F.; Guinot, C. & Venturini, G. (2013). *Hierarchical Reorganization of Dimensions in OLAP Visualizations.* Francia: IEEECS.

Mansmann, S. & Scholl, M. H. (2007). *Exploring OLAP Aggregates with Hierarchical Visualization Techniques.* Seoul, Korea: SAC'07.

Ordonez, C.; Chen, Z. & García-García, J. (2011). *Interactive Exploration and Visualization of OLAP.* Glasgow, Scotland, UK: DOLAP'11.

Piringer, H.; Buchetics, M.; Hauser, H. & Gröller, E. (2009). *Hierarchical Diffrence Acatterplots.* París, Francia: VAKD'09.

Pitarch, Y; Laurent, A. & Poncelet, P. (2009). *A Conceptual Model For Hamdling Personalized Hierarchies in Multidimensional Databases.* Lyon, France: MEDES.

Seng, P. & Habibollah, H. (2013). *Cube Kohonen Self-Organizing.* Malaysia: IEEE Transactions onNeural Networks and Learning Systems Vol.24.

Silberschatz, A.; Kotrh, H. & Sudarshan , S. (2002). *Fundamentos de Bases de datos.* España: McGraw-Hill.

Stolte, C.; Tang, D. & Hanrahan, P. (2006). *Polaris: A System for Query, Analysis and Visualization of Multidimensional Databases.* ACM.

Takama, Y. & Yamada, T. (2009). *Visualization Cube: Modeling Interaction for Exploratory Data Analysis of Spatiotemporal Trend Information.* Tokyo, Japan: IEEE/WIC/ACM.

Yaghmaie, M.; Bertossi, L.; & Ariyan, S. (2012). *Repair-Oriented Relational Schemas for Lultidimensional Databases.* Berlín, Alemania: EDBT.

Zhao, B.; Lin, x.; Ding, B. & Han, J. (2011). *TEXeplorer: Keyword-based Object Search and Exploration in Multidimensional Text Databases.* Glasgow, Scotland, UK: CIKM.