



# UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CIENCIAS

DATA WAREHOUSE APLICADO  
A LA TOMA DE DECISIONES

TESIS

QUE PARA OBTENER EL TITULO DE:

A C T U A R I O

PRESENTA:

ISRAEL IGNACIO GOMEZ LOMAS



DIRECTORA DE TESIS:  
ACT. HORTENSIA CANO GRANADOS

2004



FACULTAD DE CIENCIAS  
SECCION ESCOLAR



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**UNAM – Dirección General de Bibliotecas**  
**Tesis Digitales**  
**Restricciones de uso**

**DERECHOS RESERVADOS ©**  
**PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL**

Todo el material contenido en esta tesis esta protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.



UNIVERSIDAD NACIONAL  
AUTÓNOMA DE  
MÉXICO

**ACT. MAURICIO AGUILAR GONZÁLEZ**  
**Jefe de la División de Estudios Profesionales de la**  
**Facultad de Ciencias**  
**Presente**

Comunicamos a usted que hemos revisado el trabajo escrito: "DATA WAREHOUSE APLICADO A LA TOMA DE DECISIONES"

realizado por ISRAEL IGNACIO GOMEZ LOMAS

con número de cuenta 8926938-3 , quien cubrió los créditos de la carrera de:ACTUARIA

Dicho trabajo cuenta con nuestro voto aprobatorio.

**Atentamente**

Director de Tesis  
Propietario

ACT. HORTENSIA CANO GRANADOS

Propietario

ACT. LETICIA DANIEL ORANA

Propietario

ACT. BENIGNA CUEVAS PINZON

Suplente

ACT. CARLOS FLAVIO ESPINOSA LOPEZ

Suplente

ACT. LAURA MIRIAM QUEROL GONZALEZ

**Consejo Departamental de**

ACT. JAIME VAZQUEZ ALARILLA  
FACULTAD DE CIENCIAS

SERVICIO DE INVESTIGACIONES

MATEMÁTICAS

#### Agradecimientos:

Padre Eterno que pequeño es el sentimiento de devolución de mi hacia ti, cuando veo que tu me has dado absolutamente todo y has puesto todo absolutamente en mí. Y que enorme es el sentimiento de devolución de mi hacia ti, cuando comprendo, que YO SOY TU PRESENCIA en Acción, ¿Por qué como no poder agradecerte? si tu mismo has puesto el agradecimiento en mi corazón. GRACIAS PADRE ETERNO.

#### A mis papas:

Juan Antonio y Elvia;  
Elvia y Juan Antonio.  
Gracias papá por enseñarme la fuerza.  
Gracias mamá por enseñarme la voluntad.  
Gracias a ambos por su amor, su apoyo y su guía en todo lo que llevo de vida.  
Quiero que sepan que sin ustedes todo sería tan difícil y nunca se hubiera llevado a cabo. Los Amo.

#### A mis hermanos:

Victor, Isela y Diana  
Gracias, por que en esta obra ustedes han sido grandes e insustituibles protagonistas, que han entregado su comprensión, cariño y amor.

#### A mis Amigos:

Gracias, a todos y cada uno de los que estuvieron presentes en mi vida, pero especialmente a Mónica, Gustavo, Fernando, Eduardo y Blanca con los cuales compartí grandes batallas en esta carrera.

A mis Profesores:

Gracias, por los conocimientos transmitidos  
y por el tiempo invertido para hacer de este  
país uno mejor, especialmente a los que  
ahora tan gustosamente han accedido a ser  
mis sinodales

Laura, Flavio, Beny y Leticia.

A mi Directora de Tesis:

Act. Hortensia Cano Granados.

Gracias por su gran paciencia y entusiasmo  
en el andar de este trabajo, de igual manera  
agradezco sus tan atinados comentarios  
sobre la materia y por que sin su apoyo, este  
trabajo hubiera sido muy difícil de realizar.



---

<b><u>INDICE.</u></b>	I
<b><u>Introducción.</u></b>	1
<b><u>Capítulo I.</u></b>	
<b><u>PRESUPUESTO.</u></b>	
1.1 Antecedentes históricos de los sistemas presupuestales	2
1.2 Planeación en la administración	3
1.3 Principios del Presupuesto	4
1.3.1. Ventajas de Presupuestar	6
1.4 Presupuesto por Programas	7
1.4.1. Características del Presupuesto por Programas	8
<b><u>Capítulo II.</u></b>	
<b><u>PROGRAMACIÓN PRESUPUESTAL.</u></b>	
2.1 Conceptos de la Programación Presupuestal	10
2.2 Ejemplo de la Programación Presupuestal	13
<b><u>Capítulo III.</u></b>	
<b><u>DATA WAREHOUSE Y OLAP.</u></b>	
3.1 Antecedentes	23
3.2 Conceptos Data Warehousing y Business Intelligence	23
3.3 Datos vs. Información de Negocios	24
3.4 OLTP, Data Warehouse y OLAP	25
3.4.1. Bases de datos operacionales (OLTP On-Line Transaction Processing)	26
3.4.2. Diferencias entre un sistema OLTP y el Data Warehouse	26
3.4.3. Componentes del Data Warehouse	27
3.4.4. OLAP (On Line Analytical Process)	29
3.4.4.1. Tipos de Esquemas	34
3.4.4.2. Cubos	40
3.4.4.3. Modos de almacenamiento	42

**Capítulo IV.****Diseño e Implementación OLAP.**

4.1 Diseño de la Base de Datos	46
4.2 Conocer el formato de las fuentes de datos	48
4.3 Homogeneizar las fuentes de bases de datos a través de procedimientos	48
4.4 Creación de dimensiones	50
4.5 Creación de cubos o hipercubos	55
4.6 Manejo y Análisis de cubos	59

<b><u>Conclusiones.</u></b>	80
-----------------------------	----

**Anexo 1.****Microsoft Analysis Server.**

Registro del Servidor	82
Seguridad de la base de datos	83
Límites del Analysis Server	83
Dimensiones y Cubos Virtuales	83
Particiones	84
Proceso de dimensiones y cubos	85

**Anexo 2.****Lenguaje SQL.**

Antecedentes	88
Definición de Datos	89
Datos numéricos	89
Datos alfanuméricos	91
Datos de Tiempo	92
Datos de tiempo en Transac-SQL	92
Datos de tiempo en ANSI-SQL	93
Datos binarios	94
Datos monetarios	94
Sentencias de SQL	95
Crear Base de Datos	95
Crear Tablas	95
Crear Procedimientos almacenados	96
Eliminar Tablas	96
Eliminar Procedimientos almacenados	96



Consulta y manipulación de datos	97
SELECT	97
INSERT	100
UPDATE	102
DELETE	104
<b><u>Glosario</u></b>	106
<b><u>Bibliografía.</u></b>	109



## INTRODUCCIÓN.

Existen nuevas tecnologías de información que han desarrollado posibilidades para almacenar y manejar información en las organizaciones, por esta razón las empresas que logren aprovechar estas tecnologías y mejorar sus procesos de negocio, tendrán ventajas competitivas y estratégicas muy importantes.

Este trabajo trata de mostrar una herramienta que pueda ayudar a conseguir ciertas ventajas con respecto a sus competidores; para ello es necesario contar con información más integral o globalizada, en el sentido de no sólo tener información de distintas áreas de manera aislada, para luego conjuntarlas, sino más bien para tener información de diferentes áreas relacionadas entre ellas en un sólo instrumento; para permitir una visualización más amplia y distintas alternativas para concluir con la mejor opción en el menor tiempo posible. Cabe resaltar que esta información al ser más global sólo se enfoca a altos directivos o altos investigadores que son los encargados de tomar las decisiones finales.

Por lo tanto este trabajo trata de dar una alternativa para lograr este objetivo y para ello tomó el ejemplo del presupuesto por programas (Capítulo 2), por tener más a la mano la información necesaria, y recalcando que no es el objetivo principal el mostrar como está compuesto un presupuesto, sino más bien tomándolo como información meramente histórica desde el punto de vista de cómo está formado y jerarquizado un presupuesto de tipo gubernamental (Capítulo 1) y que nos va a servir para tener un concepto más claro a la hora de mostrar e interpretar la información mediante la alternativa de Data Warehouse.

En el capítulo 3 se mostrará la parte técnica de esta alternativa (Data Warehouse) y como interactúan los sistemas de captura, borrado, modificación así como las ventajas y desventajas en la consulta entre estos sistemas y la alternativa presentada.

En el capítulo 4 se ilustra el desarrollo técnico de esta alternativa a través de la información del presupuesto por programas de una entidad educativa (una universidad). Además se incluye un par de anexos de algunas herramientas de software utilizadas intrínsecamente en el desarrollo de esta alternativa; así como un glosario de términos técnicos para una mejor comprensión.

Y por último se presentan las consideraciones de esta alternativa.



## Capítulo I

### PRESUPUESTO.

#### 1.1 Antecedentes históricos de los sistemas presupuestales .

El Presupuesto nace en Francia, y proviene de la palabra Bougette que es un sinónimo de Control de Gastos. En nuestros días proviene de dos palabras : *Pre* que significa antes y *Supuesto* que significa hecho formado. Y genéricamente hablando se dice que los presupuestos son tan antiguos como el hombre mismo, y en la actualidad los presupuestos nacen como la necesidad de prever y controlar las actividades de una entidad; como un proceso lógico y normal del desarrollo económico en general y de las empresas en particular.

Oficial e informalmente los presupuestos se implantan en Inglaterra en el siglo XVI y dos años después obtuvieron más formalidad cuando un Canciller Ejecutivo tenía que suministrarle al Parlamento un reporte anual que incluía lo siguiente:

- a) Un estado de gastos del año anterior
- b) Una estimación de erogaciones para el período siguiente
- c) Un detalle de los impuestos con que se cubrirían los gastos sugeridos
- d) Recomendaciones para su aplicación.

En 1921 es adoptado por los países Americanos y principalmente Estados Unidos de Norte América, donde tuvo unas ligeras modificaciones en relación al sistema inglés.

El presupuesto no es un mecanismo puramente contable ó administrativo ya que es un instrumento de planeación por contener metas a cumplir en determinados medios, y es un *instrumento de administración* ya que debe realizar acciones específicas para coordinar, ejecutar y controlar los planes y programas.



## 1.2 Planeación en la administración.

La planeación constituye una de las funciones administrativas más importantes. Consiste en un proceso de anticipación mediante la selección de alternativas que combinan un conjunto de acciones.

La planeación es proyectar un futuro deseado y los medios efectivos para conseguirlo. Es un proceso de toma de decisiones, aunque la toma de decisiones no es siempre equivalente a planear o planificar. Es por ello que primero se planeará para más tarde hacer una herramienta que sirva para poder tomar decisiones con la planeación ya hecha. Pues la planeación es algo que hacemos antes de efectuar una acción, es un proceso de lo que se va a realizar para ello requerimos de los siguientes principios.

### Principios de la Planeación:

**Racionalidad:** Seleccionar alternativas de acción en forma inteligente. Se deben analizar en forma comparativa ventajas y desventajas; y reducir a través del conocimiento científico y del razonamiento sistemático, a una conducta final coherente que permita el máximo aprovechamiento de los recursos.

**Previsión:** La planeación es previsión del futuro. Los planes y programas tienen por objeto orientar la conducta de los organismos hacia delante. En función de este principio, se fijan plazos definidos para la ejecución de las acciones que se planean. Los períodos varían en el tiempo de acuerdo a las decisiones de los gobiernos, dependen también de los recursos disponibles de las posibilidades de control, así como de los objetivos perseguidos o de las diversas legislaciones.

**Universalidad:** La planeación debe abarcar las diferentes etapas o fases del proceso económico, social y administrativo y prever las consecuencias que producirá su aplicación. Desde luego, la planeación deberá aplicarse en todos los sectores y niveles de la administración a fin de dar consistencia y armonía a la gestión pública.

**Unidad:** Como consecuencia del principio anterior, se desprende la necesidad de que los planes tengan unidad, es decir, que estén debidamente integrados entre sí, que formen un todo, orgánico y compatible. Este principio está ligado al concepto de coordinación. Ello debe surgir como consecuencia de la debida integración de los planes sectoriales en el programa global, evitándose el mal gasto de los recursos en acciones duplicadas o no debidamente ajustadas.



**Continuidad:** La planeación no tiene fin en el tiempo, mientras subsistan los agentes económicos se deberán hacer las cosas, por ello pueden variar los objetivos de los planes, el énfasis de la acción o los plazos fijados. Pero la necesidad de obtener el máximo de los recursos no desaparecerá. La planeación es un proceso de acción ilimitada y que cada año se podrá revisar, evaluar y agregar uno o más estimadores.

**Inherencia:** La planeación es necesaria en cualquier organización humana, es inherente al hombre, por tanto la administración. El Estado o la empresa deben planificar la forma de alcanzar sus objetivos y no debe verse en función de metas mediatas ó inmediatas.

Habitualmente el planeamiento se distingue en **estratégico** y **táctico**. El primero se refiere al conjunto de decisiones fundamentales para el desarrollo futuro de la empresa proyectándose hacia el largo plazo, correspondiendo sus decisiones a la alta dirección. El planeamiento táctico debe conformarse dentro del marco de referencia del estratégico y abarcando el corto plazo, administrando los medios a través de los que habrán de procurarse los objetivos específicos.

Dado que el presupuesto es un instrumento de planeación estratégica y a su vez táctica como se menciona en el párrafo anterior, el siguiente punto a tratar será el de los Principios del Presupuesto.

### 1.3 Principios del Presupuesto.

Para iniciar con los principios del presupuesto es necesario tener en cuenta el concepto de presupuesto. El presupuesto es un instrumento fundamental que sirve de apoyo para la realización de los programas de trabajo de cualquier organización o institución, y que en él se expresan los recursos que se asignan para la ejecución y el cumplimiento de objetivos y metas. Asimismo, sirve como una herramienta para evaluar el grado de racionalidad en el uso de los recursos disponibles. La teoría presupuestaria ha formulado algunas reglas de aplicación general destinada a perfeccionar y explicitar la información del documento presupuestal con fines de discusión y análisis

a) En cuanto a la relación entre el presupuesto y las actividades financieras del gobierno.



- **Universalidad.-** Todos los ingresos y gastos deben encuadrarse en el mecanismo presupuestario y someterse a su proceso. La existencia de créditos y gastos fuera del presupuesto, torna menos claras las finanzas públicas y sustrae del control colectivo operaciones que pueden ser de un volumen considerable.
  - **Exclusividad.-** Esta se refiere a la exclusión del presupuesto de toda materia no financiera o programática.
- b) En cuanto al tratamiento dado por el mecanismo presupuestario a los factores comprendidos en el presupuesto.
- **Unidad.-** Esta incluye tres conceptos:
    - Unidad Formal* : Cuando un único documento presupuestario comprende todos los ingresos y gastos del gobierno.
    - Unidad de Fondo*: Cuando requiere de todos los ingresos y gastos incorporados a un solo fondo con el cual se puedan atender todos los requerimientos presupuestales. Esta regla no admite el compromiso previo de una renta para su aplicación a un destino especial.
    - Unidad Orgánica ó Sistemática*: Esta se obtiene cuando el presupuesto es presentado de manera tal que el total de los gastos y el total de los ingresos se obtiene por simples adiciones; es decir, sustrayendo un total del otro para conocer la existencia de superávit, déficit o equilibrio.
  - **Especificación.-** El problema de la especificación es una cuestión de grado. No debe ser excesiva por que se hace muy compleja la ejecución del presupuesto por parte de los organismos públicos; además dificulta la fiscalización parlamentaria. Por otro lado no debe ser muy sumaria pues disminuiría el control legislativo.
  - **Periodicidad.-** Esta consiste en no fijar límites de tiempo a los presupuestos en cuanto a las necesidades de programación, sobre todo en el caso de obras de gran envergadura, hacen indispensable el asegurar la provisión de fondos para períodos superiores a un año.
  - **Exactitud.-** Consiste en mantener la rigurosidad de los métodos en la estimativa de los ingresos y en la evaluación precisa del costo de los programas formulados.
  - **Autorización Previa.-** Esta regla exige que todos los gastos y la mayor parte de los ingresos, sean votados y autorizados por el poder legislativo antes de su ejercicio.



c) En cuanto a la presentación formal del documento presupuestario:

- **Claridad.** - El presupuesto debe ser presentado en términos que no dejen margen alguno para dudas o interpretaciones subjetivas; es fundamental que el presupuesto resulte inteligible para cualquier ciudadano.
- **Publicidad.** - Este requisito indica que el presupuesto no puede ser un documento reservado, de difícil acceso, sino por el contrario, un informe y un plan de actividades futuras del gobierno, que sea hecho para enterar a las personas en general.

Algunos elementos del documento presupuestal se verá más adelante una vez que se haya hablado del presupuesto por programas (sección 1.4.1).

### 1.3.1 Ventajas de presupuestar

Es importante hacer notar las ventajas que se tienen, al utilizar el instrumento de planeación estratégica (Presupuesto), que en este caso lo llamaremos presupuestar . A continuación se muestran las principales ventajas.

- Presiona para que la alta gerencia defina adecuadamente los objetivos básicos de la empresa.
- Propicia que se defina una estructura organizacional adecuada, determinando la responsabilidad y autoridad de cada una de las partes que integran la organización.
- Incrementa la participación de los diferentes niveles de la organización, cuando existe motivación adecuada.
- Obliga a mantener un archivo de datos históricos controlables.
- Facilita a la administración la utilización óptima de los diferentes insumos.
- Facilita la coparticipación e integración de las diferentes áreas de la compañía u órgano donde se implementa.
- Obliga a realizar auto análisis periódico.
- Facilita el control administrativo.
- Ayuda a lograr mayor efectividad y eficiencia en las operaciones.



Es de mencionar que existen varios tipos de presupuestos como:

- *Presupuesto de Ventas*
- *Presupuesto de Producción*
- *Presupuesto de Paso*
- *Presupuesto de Compras*
- *Presupuesto de Mano de Obra*

Entre otros, los cuales quedan fuera de este trabajo donde solo se tratará el Presupuesto por Programas y que se desarrollará en el siguiente punto.

#### **1.4 Presupuesto por Programas.**

Es un instrumento de gobierno, de administración, de concreción y ejecución de los planes generales de desarrollo socio-económico. Es un documento ó conjunto integrado de documentos, que en forma adecuada expresa el contenido del indicado programa.

Para crear un buen modelo presupuestal ó llevar a cabo este, es necesario tener una buena planeación y para ello existen principios básicos de planificación como son: La Racionalidad, Previsión, Universalidad, Unidad, Continuidad, e Inherencia (mencionados anteriormente); además de contar con una programación es decir; hacer planes, fijar metas cuantitativas a la actividad, destinar recursos humanos, materiales, financieros y tecnológicos necesarios, definir los métodos de trabajo a emplear, fijar la cantidad y calidad de los resultados y determinar la localización especial de las obras y actividades.

El Presupuesto por programas es una técnica presupuestaria que permite correlacionar los programas y subprogramas con los recursos financieros, teniendo como sustento los objetivos y las metas de las dependencias que conforman la Institución u Organización.



### 1.4.1 Características del Presupuesto por Programas.

Como se mencionó en la última parte de la sección 1.3.1, hay diversas clases de presupuestos los cuales tienen características comunes, es por ello que a continuación se listan las de más importancia para el presupuesto por programas.

- El presupuesto por programas es un instrumento que facilita la racionalización de las decisiones de alto nivel en lo que se refiere a la orientación de la política de acción global y por funciones.
- Además es un instrumento con posibilidades de mejorar sensiblemente la eficiencia y la productividad de los servicios.
- Permite los estudios de costos a todos los niveles de la acción programada.
- Facilita el desarrollo de los sistemas de administración de recursos humanos; de administración de compras y almacenes; y de actualización de inventarios de activos fijos y a su vez depende del desarrollo de estos sistemas para su propia consolidación.
- Permite y requiere del funcionamiento paralelo de sistemas de registros y controles de administración financiera, tanto de carácter contable y presupuestario como de tesorería así como de registros y controles de resultados de la acción y efectos de los servicios.
- Desarrolla y requiere el funcionamiento de un sistema de información y evaluación de las actividades programadas de tal forma que desde los niveles de jefatura de las unidades de base hasta el más alto nivel de autoridad se pueda seguir de modo permanente la marcha de gestión, para poder decidir oportunamente los cambios que correspondan, con el objeto de asegurar la consecución ó finalización de las metas programadas.
- El presupuesto por programas cumple con la función de instrumentar la coordinación de la gestión de cada entidad de los diferentes sectores con el resto de la actividad.
- El proceso a través del que se elabora, expresa, aprueba, ejecuta y evalúa el nivel de cumplimiento de la casi totalidad del programa de gobierno, de un organismo autónomo o descentralizado, ó de una empresa pública para cada período presupuestal.



A continuación se citan algunas diferencias en lo que vendría siendo un presupuesto tradicional y el de programas.

<i>Características</i>	<i>Tradicional</i>	<i>Por Programas</i>
Contenido	Contable	Administrativo- Económico
Información	Por Objetos	Por Actividades y Propósitos
Tipo de Medición	Detallada	Detallada y Analítica
Papel de la oficina de Presupuesto	Fiduciario	Administrativo y Político
Flujo de Decisión	Agregativo y Ascendente	Desagregativo y descendente
Tipo de Decisión	Incremental	Proyectivas
Responsabilidad		
a) Por Planeación	Dispersa	Central
b) Por Administración	Dispersa	Operativa
c) Por Control	Central	Operativa
Clasificaciones	Fiscales	Fiscales y Analíticas

Algunos elementos de un documento presupuestal pueden ser los siguientes cubriendo los aspectos que se ven en la sección 1.3 en el inciso c:

- *Exposición de motivos:* Que regularmente es dirigida a el consejo de la institución poniendo a consideración el ejercicio del presupuesto en determinado período de tiempo.
- *Documento que aprueba el presupuesto:* Este generalmente es aprobado por una comisión de presupuestos que analiza la distribución del monto a asignar.
- *Presupuesto de Ingresos:* Este puede estar integrado por distintas fuentes de subsidio y por ingresos generados por si mismo que también pueden ser de distinta índole.
- *Resumen de Ingresos:* Específica la índole y el monto del ingreso dependiendo de sus características, es decir; como se va a obtener esos ingresos (fuentes de ingreso).
- *Resumen de Egresos:* Específica la índole y el monto del egreso dependiendo sus características, es decir; como se van a distribuir esos egresos (Rubros).



## Capítulo II.

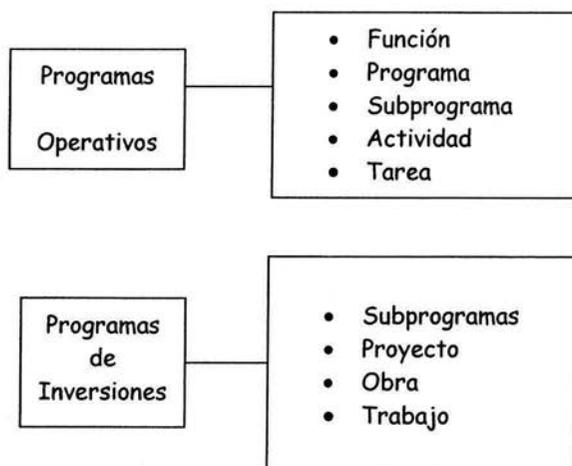
### PROGRAMACIÓN PRESUPUESTAL

#### 2.1 Conceptos de la Programación Presupuestal.

En el capítulo anterior traté sobre la historia del presupuesto y las ventajas del presupuestar además de los elementos que conforman al documento presupuestal general que es el que se presenta a las autoridades de la organización o al público en general, pero ahora las preguntas que quedan en el aire son las siguientes: ¿Cómo se implanta el presupuesto programático? y ¿Cómo se llega a ese documento?. Las respuestas parecerían no muy sencillas de responder y de hecho lo son, pues va a depender de varios factores como son los objetivos o metas a cumplir, así como del tamaño de la organización o de la institución.

De lo que se trata es de llegar a una clave o código que nos facilite el manejo de la información del presupuesto.

Y para contestar estas preguntas tomaremos en cuenta que existen niveles de programación entre los más comunes se encuentran los programas operativos y programas de inversión.





**Función:** Es un conjunto de acciones encaminadas a un mismo fin, y agrupa a los niveles restantes (descendientes). Esta agrupación se utiliza para los organismos que definen dos o más objetivos. La función es la categoría más genérica y engloba a un conjunto de programas relativamente homogéneos.

**Subprograma:** Es una división de ciertos programas complejos, para facilitar la ejecución de un campo específico y en él se fijan metas principales que se cumplirán mediante acciones concretas que realizan determinadas unidades de operación, con los recursos humanos, materiales y financieros asignados y con un costo específico. Por ejemplo: Para un caso específico como el de una Universidad, puede ser que dentro de un programa de formación profesional, se establezcan subprogramas de formación de Licenciados en Economía, Derecho, Actuarios, Biólogos, Químicos, Ingenieros en Computación etc; los cuales son ejecutados por unidades de operación o dependencias diferentes y bien definidas a las cuales se les asignará una clave para hacer una distinción entre ellas.

**Actividad:** Es una división más reducida en cada una de las acciones que se llevan a cabo para cumplir las metas de un programa o subprograma de operación y puede ser realizada por una entidad administrativa de nivel intermedio o bajo, como una sección.

**Tarea:** Es una operación específica, que forma parte de un proceso que se destina a producir un resultado, por ejemplo para la publicación de un libro, se deberán ejecutar las tareas de redacción, edición, almacenamiento y distribución.

**Proyecto:** Es un conjunto de obras requeridas para el logro de un fin, y que una vez puesto en marcha da resultados de crecimiento o expansión en cuanto a infraestructura.

**Obra:** Es cada una de las partes o etapas de la ejecución de un proyecto.

**Trabajo:** Será cada una de las acciones específicas para el logro de una obra por ejemplo: excavación, remoción de materiales .

Como nos podemos dar cuenta son niveles que van de lo general a lo particular (de arriba hacia abajo), además utilizando esta técnica nos va a permitir ir de lo particular a lo general (de abajo hacia arriba) entre otras opciones.

En este trabajo supondremos un caso particular para alcanzar una mayor comprensión y que mejor que el ejemplo de una Universidad donde podemos aplicar



un programa Operativo, en el cual la estructura programática será la parte central de la técnica del presupuesto por programas; en ella se conjugan sistemáticamente los programas y subprogramas con los recursos encaminados al cumplimiento de las funciones de Docencia, Investigación, Extensión Universitaria y Gestión Institucional.

Esta estructura es el marco conceptual en el cual las dependencias presentan los objetivos y metas que pretenden lograr, así como los recursos humanos, materiales y financieros que se requieran para su cumplimiento.

En otras palabras para la implantación del sistema de Presupuesto por Programas es necesario contar con lo siguiente:

- Estructura de programas
- Un catálogo de cuentas presupuestales por objeto del gasto
- Un código programático
- Formatos e instructivos para la formulación del anteproyecto de presupuesto
- Una metodología para el análisis y discusión de los anteproyectos.

Antes debemos tomar en cuenta la complejidad del organismo para dividir sus actividades, tanto por el tipo de necesidades a satisfacer, como el grado de especialización técnica que requieran para cumplir su finalidad; desde luego es muy importante considerar que para alcanzar en mayor o menor medida los objetivos dependerán de los programas que se asignen.

Para diseñar e instrumentar los elementos básicos que requiere la implantación del sistema presupuestario mencionado arriba y para elaborar un proyecto de estructura programática, se debe tomar en cuenta la estructura de organización para el establecimiento de las funciones principales en que se puede dividir la acción que desarrolla el Organismo, su base legal, los manuales de organización y procedimientos y demás documentos que permitan profundizar en el conocimiento del Organismo.

**Niveles programáticos.** Debido a la enorme y compleja actividad que una universidad presenta, se clasifica de tres maneras distintas:

1. Funciones básicas de la Institución: Por que permite evaluar los recursos asignados a cada uno de los programas asociados a cada función.

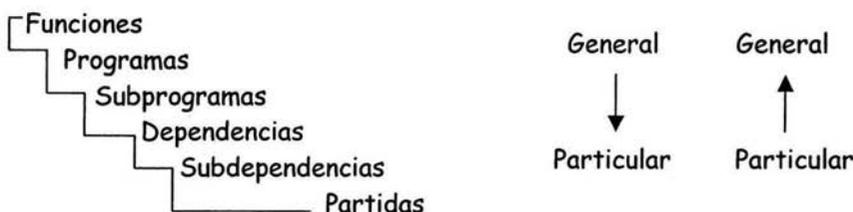


2. **Objeto de Gasto:** Que permite tener una óptica sobre la forma en que se destinan los recursos como pueden ser remuneraciones personales, gasto corriente (erogaciones que no aumentan el patrimonio de la institución por ejemplo sueldos, bienes de consumo y/o servicios) y gasto de inversión (erogaciones que repercuten directamente a el incremento del patrimonio de la institución, por ejemplo mobiliario, plantilla de autos, bienes inmuebles etc.), donde la mínima expresión será llamada **Partida**.
3. **Naturaleza de las Actividades:** Que agrupará a cada una de las dependencias u órganos, por actividades a esa agrupación se le llamará **Ramo**, donde la mínima expresión se llamará **Dependencia** que a su vez estará dividida por sus **Subdependencias**.

En la siguiente sección a través de los ejemplos se aclararán un poco más los conceptos mencionados en esta sección.

## 2.2 Ejemplo de la Programación Presupuestal.

Es de hacer notar que cuando hablemos de programa en este capítulo no es un programa de computadora y se entenderá por Programa el agrupar una serie de acciones encaminadas a un mismo fin, a efecto de coordinar al proceso de decisión, en una unidad a la que se le asignarán recursos, para que en un tiempo determinado alcance un objetivo, una acción común en función de los objetivos y una buena estructura de programas la constituye el código ó clave de identificación, pues a través de ésta se facilita la operación o manejo del sistema.



Esta es la estructura que nos permitirá encontrar fácilmente cualquier nivel de información; ya sea por función, programa, subprograma, por la naturaleza de sus actividades (dependencias que estarán agrupadas por los ramos) y/o por la clasificación de sus erogaciones (partidas que estarán agrupadas por los grupos de gasto).



Para manejar esta estructura con mayor agilidad se le asigna una clave a cada nivel, que dará como resultado un código (**Código Programático**).

Ejemplo del Código Programático:

32.02.215.01.131

Donde:

- 3 Función
- 2 Programa
- 02 Subprograma
- 215 Dependencia
- 01 Subdependencia
- 131 Partida de Gasto(1 Grupo, 31 Subgrupo)

A continuación se verá la Estructura de Programas

Estructura de programas

### ***FUNCIONES Y PROGRAMAS.***

**1) DOCENCIA NIVEL SUPERIOR** Esta será la función 1 que a su vez tiene los programas siguientes.

#### **Programas**

1.0	Educación de Licenciatura
1.1	Educación de Posgrado
1.2	Educación de Sistema de Universidad Abierta
1.3	Educación Complementaria
1.4	Educación Continua
1.5	Coordinación Académica
1.6	Formación y Actualización Académica
1.7	Programas de Desarrollo Académico
1.8	Servicios de Apoyo Académico
1.9	Servicios de Apoyo Administrativo

**2) DOCENCIA NIVEL BACHILLERATO Y TÉCNICO** Función 2**Programas**

2.1	Educación Media Superior
2.2	Educación Técnica
2.3	Educación Complementaria
2.4	Coordinación Académica
2.5	Formación y Actualización Académica
2.6	Programas de Desarrollo Académico
2.7	Servicios de Apoyo Académico
2.8	Servicios de Apoyo Administrativo

**3) INVESTIGACIÓN** Función 3**Programas**

3.1	Investigación en Ciencias y Desarrollo Tecnológico
3.2	Investigación en Humanidades y Ciencias Sociales
3.3	Coordinación Académica
3.4	Formación Académica
3.5	Programas de Desarrollo Académico
3.6	Servicios de Apoyo Académico
3.7	Servicios de Apoyo Administrativo

**4) EXTENSIÓN UNIVERSITARIA** Función 4**Programas**

4.1	Difusión de Actividades Artísticas, Científicas y Culturales
4.2	Coordinación
4.3	Extensión Educativa
4.4	Comunicación y Divulgación Universitaria
4.5	Vinculación con la Sociedad
4.6	Programas de Apoyo Técnico
4.7	Servicios de Apoyo Administrativo

**5) GESTIÓN INSTITUCIONAL** Función 5**Programas**

5.1	Dirección Institucional
5.2	Planeación y Normatividad
5.3	Programas de Apoyo Técnico
5.4	Prestaciones Contractuales
5.5	Servicios Administrativos Institucionales
5.6	Vigilancia y Fiscalización
5.7	Servicios de Apoyo a la Comunidad
5.8	Servicios de Apoyo Administrativo

**SUBPROGRAMAS**

Tomando en cuenta la definición de subprograma y dado que estos derivan de los programas, la cantidad de subprogramas crece enormemente por lo cual sólo mostraré algunos ejemplos de los mismos

12.02.412.04 Ciencias Sociales.

12.02.434.01 Enfermería Técnica.

12.02.434.02 Enfermería Profesional

Donde:

- 12 El primer carácter (1) es la función que unido al 2 forma el programa 12 (que toma el nombre de Educación de Sistema de Universidad Abierta)
- 02 es el Subprograma para la Dependencia 412 en la Subdependencia 04 (que se llama División del Sistema de Universidad Abierta), que para otra dependencia el subprograma 02 podría tener otro nombre, como se ve en los ejemplos de arriba e inclusive si fuera sobre la misma Dependencia, pero distinta Subdependencia sería otra división lo que reeditaría en otro nombre distinto.

Como se puede apreciar el número de subprogramas serán por lo menos tantos como dependencias y subdependencias existan, por lo cual solo se presentan algunos ejemplos. Para tener más clara la idea de una subdependencia y un subprograma tomaremos el caso de la dependencia 411 Facultad de Ciencias que tiene 3 subdependencias.



411.01 Posgrado, 411.02 Licenciatura, 411.03 Investigación y cada una de las subdependencias puede tener n subprogramas que obviamente, por ser de distintas subdependencias tendrán diferentes nombres.

En los niveles programáticos se encuentra la Naturaleza de las Actividades. A continuación algunos ejemplos de la misma.

La lista de los ramos es la siguiente

200	Órganos de Investigación Humanística
300	Órganos de Investigación Científica
400	Facultades, Escuelas y Unidades Multidisciplinarias
500	Programas Complementarios a la Docencia e Investigación
600	Órganos de Extensión Universitaria
700	Servicios de Planeación, Administrativos y Jurídicos
800	Obras, Conservación y Mantenimiento
900	Órgano Coordinador de Servicios Institucionales

Veremos más a detalle el ramo 400 como ejemplo.

400.- Facultades, Escuelas y Unidades Multidisciplinarias

411.- Facultad de Ciencias

411.01.- Posgrado

411.02.- Licenciatura

411.03.- Investigación

412.- Facultad de Ciencias Políticas y Sociales

412.01.- Posgrado

412.02.- Licenciatura

412.03.- Investigación

412.04.- Educación Continua

412.05.- Educación Abierta

413.- Facultad de Contaduría y Administración

413.01.- Posgrado

413.02.- Licenciatura

413.03.- Investigación

413.04.- Educación Continua

413.05.- Educación Abierta

413.06.- Unidad Externa

.....

.....



- .....
- .....
- 472.- Dirección General del Colegio de Ciencias y Humanidades
- 472.01.- Plantel "Azcapotzalco"
  - 472.02.- Plantel "Naucalpan"
  - 472.03.- Plantel "Vallejo"
  - 472.04.- Plantel "Oriente"
  - 472.05.- Plantel "Sur"

Para concluir con la conformación del Código Programático incluiré las siguiente definiciones.

**Grupos de Gasto:** Es el elemento presupuestario que agrupa los bienes y servicios que la institución adquiere conforme a la naturaleza intrínseca de los mismos.

100	Remuneraciones Personales
200	Servicios
300	Becas, Prestaciones y Estímulos
400	Artículos y Materiales de consumo
500	Mobiliario y Equipo
600	Inmuebles y Construcciones
700	Erogaciones condicionadas al Ingreso y Programas de Colaboración Académica

**Partida de Gasto:** Es el elemento presupuestario en que se dividen los grupos (subgrupos) de gasto y que clasifican las erogaciones de acuerdo con el objeto específico del gasto.

Ejemplos de Partidas de Gasto:

Estos pertenecen al grupo 100 Remuneraciones Personales y subgrupos 10 y 80 respectivamente.

- 110 Sueldos a Profesores de Asignatura.
- 180 Honorarios

Esta pertenece al grupo 200 Servicios subgrupo 13

- 213 Gastos de viaje de alumnos en Prácticas Escolares y a Deportistas



Estas pertenecen al grupo 300 Becas, Prestaciones y Estímulos, subgrupos 10 y 55 respectivamente.

310 Becas

355 Prima de Desempeño al Personal Académico

Esta pertenecen al grupo 400 Artículos y Materiales de consumo, subgrupo 13.

413 Combustible y Lubricantes

Esta pertenecen al grupo 500 Mobiliario y Equipo, subgrupo 14.

514 Equipo de Cómputo.

Una vez concluida la conformación del Código Presupuestal el siguiente paso es ¿A dónde?, ¿Para qué? y ¿Cuánto? se va asignar, para esto es necesario llenar ciertos formatos donde se dice el nombre del proyecto, a que partida afectaría, de que dependencia es, etc. Los formatos para el anteproyecto que se presenta a continuación pueden variar dependiendo del tipo o de las necesidades del organismo.

-Nombre del Organismo en este caso El nombre de la universidad  
-Área a la que va dirigida por ejemplo Dirección de Presupuesto  
-Titulo correspondiente por ejemplo Objetivo, metas y actividades por Dependencia y programa

□ □ □ □	□ □ □ □ □ □ □ □	□ □	□ □ □ □ □ □ □ □				
Clave	Nombre	Clave	Nombre				
Dependencia		Subdependencia					
□ □	□ □ □ □ □ □ □ □	Principales metas a realizar					
Clave	Nombre	Unidad de Medida	Trimestres				Total Anual
Programa			1	2	3	4	
Descripción							

Calendarización de las principales actividades					
Indicadores de Actividad	Trimestres				Total Anual
	1	2	3	4	



- Nombre del Organismo
- Anteproyecto presupuestal del año a ejercer
- Titulo correspondiente. Para este caso Solicitud de Recursos Adicionales Resumen del Costo de Proyectos
- Clave y nombre de la dependencia

Área	Nombre del Proyecto	Costo
		Total



-Nombre del Organismo  
 -Anteproyecto presupuestal del año a ejercer  
 -Titulo correspondiente. Para este caso Solicitud de Recursos Adicionales por Grupo o por Partida  
 -Nombre del Proyecto

<input type="text"/>	<input type="text"/>	<input type="text"/>	<input type="text"/>
Nuevo	En Proceso	Apoyo	Departamento
<input type="text"/> <input type="text"/> <input type="text"/>	<input type="text"/>	<input type="text"/> <input type="text"/>	<input type="text"/>
Clave	Nombre	Clave	Nombre
Dependencia		Programa	

Objetivos

<input type="text"/> <input type="text"/> <input type="text"/>	<input type="text"/> 1 <input type="text"/> 2 <input type="text"/> 3 <input type="text"/> 4 <input type="text"/> 5 <input type="text"/> 6 <input type="text"/> 7 <input type="text"/> 8
Clave de Partida	Prioridad

Acciones y Resultados esperados

Requerimientos			
Grupo	Concepto	Descripción	Costo

Justificación



Una vez que se han llenado los formatos con la información pertinente el siguiente paso es implementar una metodología para el análisis de la información (para ver cuales proceden o cuales se pueden dejar para otro ejercicio dependiendo de los objetivos del organismo o si se puede asignar en su totalidad el monto solicitado, o tal vez se puede asignar en parcialidades etc.



## Capítulo III.

### Data Warehouse y OLAP.

#### 3.1 Antecedentes

En nuestros días se ha acrecentado la necesidad de contar con información que optimice el funcionamiento de nuestra empresa u organismo, para ello es necesario contar con datos organizados, resumidos además de igualar metas de los sistemas de información, implementar una reingeniería de las actividades, mejorar las capacidades de los ejecutivos de alto nivel, establecer conexiones con clientes, proveedores o prestadores de servicios, actualizar los sistemas obsoletos y mejorar los procesos de desarrollo de sistemas y para ello nos ayuda de gran manera el tener un Data Warehouse, no obstante de ser una tecnología relativamente nueva (1995) ya existen centros de Investigación específicos para el desarrollo de la misma.

El Data Warehouse ha crecido mucho en las grandes empresas a nivel mundial y en México existen algunas empresas que lo están implantando y están alcanzando un gran éxito como con ello; algunos ejemplos son: Elecktra, Alpura, Vitro, Productos de Maíz, algunas Cementeras y Reaseguradoras entre otras.

#### 3.2 Conceptos Data Warehousing y Business Intelligence

**Data Warehousing** es el proceso de almacenar datos o reunir información histórica de una Organización en un repositorio o depósito central. Este concepto fue desarrollado por W.G. Inmon en la década de 1990.

**Business Intelligence** es un amplia categoría de herramientas de software y soluciones que se usan para recolectar, consolidar, analizar y proveer el acceso a datos, que serán utilizados por gente de altos niveles que tengan que ver con la toma de decisiones. El término de Business Intelligence fue introducido por el grupo Garnet en 1993 que textualmente dijo que era el conjunto de herramientas de tipo Software que proveen consultas, análisis y reportes de bases de datos multidimensionales.



Existen varias definiciones de Data Warehouse pero citaré algunas de las más globales e importantes.

- Un **Data Warehouse** es un conjunto de datos orientados a una materia o tópico, que varían con el tiempo y que no son transitorios, los cuales soportan el proceso de toma de decisiones de una administración.
- Otra definición es la de ser una arquitectura o depósito semánticamente consistente en datos (separados y que no interfieren con los sistemas operativos y de producción existentes) que llenan por completo los diferentes requerimientos de acceso y reporte de datos.
- Data Warehouse es un proceso continuo que mezcla los datos de varias fuentes heterogéneas, incluyendo datos históricos y adquiridos para soportar la constante necesidad de consultas estructuradas que están hechas a la medida, así como reportes analíticos y soporte de decisiones.

### 3.3 Datos vs. Información de Negocios.

La captura de datos es un proceso que en cualquier organismo o empresa se realiza diariamente; estos datos regularmente son afectaciones de cualquier tipo y que modifican el estado actual del organismo o empresa, un ejemplo claro de esto puede ser el de un Banco donde capturamos los datos de los clientes, nombre, estado civil, dirección, crédito etc. Estos datos nos sirven para tener información acerca del cliente y de cierta manera a mejorar algunas funciones, pero no sólo estos datos nos dan información sobre el cliente y es ahí donde interviene el concepto de Información de Negocios, que nos va a ayudar a examinar los datos en diferentes contextos, a determinar como esos datos se relacionan con otros datos, y a entender como los datos afectan los objetivos y al mismo negocio en sí.

Como en el ejemplo anterior que tenemos la captura de datos entre ellos su domicilio y su estado civil (si es casada hablamos de más de un cliente).

¿Qué pasa si este cliente vive en una zona donde no existe una sucursal de este banco y además no es el único cliente; pues nuestra base de datos nos dice que hay cientos en esa misma área, entonces empezamos a considerar la propuesta de poner otra sucursal ahí, pues pudiera ser que otro banco de la competencia se percatará de ese hecho y colocará una de sus sucursales en este lugar, lo que manifestaría un descenso considerable de clientela y en casos más drásticos el cierre de la



sucursal. Para evitar esto es necesario cambiar nuestra perspectiva de no solo captura de datos sino tener información de negocios.

Una vez que la empresa u organismo ha reconocido el valor de la Información de Negocios; los datos y la información se torna muy numerosa y frecuente, entonces para satisfacer las necesidades de Información se vuelve en una tarea muy compleja, pues se tiene que saber el origen de los datos, así como consolidar, analizar y distribuir los mismos a otros miembros de la empresa u organismo. Para hacer esta tarea regularmente las empresas recurren a sistemas que proveen los datos para la mejora del análisis, pero regularmente estos sistemas son malos, pues no pueden proveer toda la información necesaria para una toma de decisiones, son muy caros y tardan mucho tiempo en dar los primeros resultados y algunas veces muestran la información sin importar que tipo de personas ven la información, si tiene derecho a ver esa clase de información; es decir no cuentan con roles según el área y la mayoría de las veces no está jerarquizada la información (forma de árbol).

### 3.4 OLTP, Data Warehouse y OLAP

Una vez que la empresa empieza a recopilar datos y esta se vuelve muy numerosa para poder manejarse con tan sólo una hoja de cálculo, nace la necesidad de crear procesamientos de almacenaje y consulta de los mismos, para ello las diferentes marcas de software dedicado a crear manejadores de bases de datos crean los procesamientos OLTP y OLAP: el procesamiento de transacciones en línea (OLTP, Online Transaction Processing) y el procesamiento de análisis en línea para toma de decisiones (OLAP, Online Analytical Processing).

Estas 2 clases tienen diferentes razones de ser por lo que su funcionamiento es diferente. Se necesita una elevada concurrencia y técnicas inteligentes para acelerar la ejecución de los procesos para permitir una elevada tasa de transacciones de actualización e inserción (OLTP), por otro lado se necesitan buenos algoritmos de evaluación de consultas y optimización de consultas para el análisis de decisiones (OLAP).

La arquitectura de algunos sistemas de bases de datos se ha ajustado para el procesamiento de transacciones y otros más para el análisis de las decisiones mientras que otros fabricantes buscan el equilibrio entre estas dos tareas. Las aplicaciones tienen generalmente una mezcla de necesidades de procesamiento de transacciones y de procesamientos de análisis. Por lo tanto el mejor sistema de bases de datos para una aplicación dependerá del conjunto de respuestas que den solución a ambas necesidades. Teniendo en cuenta lo anterior Microsoft crea una herramienta para el análisis de procesamientos en línea (Analysis Manager), sin



descuidar el procesamiento de transacciones con el Enterprise Manager y Query Analyzer que aunado a otras herramientas conforman a Microsoft SQL Server 7 o 2000 y será utilizado en el siguiente capítulo para continuar el ejemplo.

### 3.4.1 Bases de datos operacionales (OLTP On-Line Transaction Processing).

OLTP son sistemas operacionales que capturan, borran, modifican o actualizan la transacciones de un organismo o empresa.

Características:

- Proceso de Transacciones en tiempo real (Inserción, Actualización y Borrado).
- Contienen estructuras de datos optimizadas para captura y edición.
- Proveen información limitada, para toma decisiones.

Ejemplo: Aplicaciones que muestran catálogos, o que modifican datos de los clientes aplicaciones que capturan datos y los insertan directamente a la Base de Datos etc.

### 3.4.2 Diferencias entre un sistema OLTP y el Data Warehouse

Un Data Warehouse es diferente de las bases de datos operacionales que soportan las aplicaciones de un Proceso de Transacción en Línea (OLTP).

- El Data Warehouse está orientado a una materia, es decir organiza los datos desde la perspectiva del usuario final (regularmente el que toma las decisiones). Mientras que otros sistemas organizan la información de manera que los datos tengan la mayor eficiencia posible a la hora de actualizar o borrar la información; que muchas veces no la devuelve de una manera clara para que en ese mismo instante se tome cierta decisión y en algunas ocasiones se ayuda de otras aplicaciones lo que conlleva a una demora de tiempo.
- El Data Warehouse administra grandes cantidades de información. Pues continuamente los datos históricos nos ayudan para hacer proyecciones, las



cuales son importantes en la toma de decisiones dentro de la misma aplicación. A diferencia de otros sistemas que no manejan la información histórica por que implicaría una gran carga, además de no ser operacional dentro de la misma aplicación.

- El Data Warehouse guarda información en distintos medios de almacenamiento debido al volumen de información que maneja
- El Data Warehouse comprende múltiples versiones de bases de datos o datos fuente (archivos planos, Excel, DBase, Sql Server, Oracle, Informix, Sybase etc.) debido a que la información ha pasado por bastante tiempo y las tecnologías de almacenamiento cambian continuamente, de ahí la característica del Data Warehouse de integrar y asociar la información en un solo sistema. Cabe hacer mención que se pueden acceder a distintos repositorios en distintas ubicaciones y que pueden ser heterogéneas las fuentes de información.
- El Data Warehouse condensa y agrega información, pues a veces el nivel de detalle es muy alto e inconsistente en otros sistemas lo cual llevará a una toma de decisiones errónea o confusa.

Dadas las diferencias existentes entre los sistemas OLTP y el Data Warehouse, las empresas en un futuro tendrán uno o más Data Warehouse, las cuales les beneficiarán con respecto a sus competidores. Las áreas a aplicar pueden ser las siguientes: Administración de bienes corporativos, Administración de las relaciones con los clientes, Análisis de Rendimiento, Administración Estratégica, Mercadotecnia, Finanzas, Logística, Embarque y Planeación entre otras.

### 3.4.3 Componentes del Data Warehouse

Fuentes de datos: Pueden ser sistemas operacionales (OLTP), que capturan las transacciones de una empresa u organismo y estos son los que regularmente suministran de datos a el Data Warehouse o a los Data Marts que más adelante definiré. Los OLTP pueden ser de tipo relacional o no-relacional y pueden prescindir de datos históricos, pero los que siempre deben de estar son los datos que reflejan el actual estado de la empresa u organismo.

Proceso de conversión o área de preparación de datos: Es una colección de procesos que limpian, transforman y combinan entre otros las fuentes de datos para ser usadas en los Data Marts o el Data Warehouse. Es decir la información que provienen de fuentes de datos heterogéneas, aquí es donde se van a consolidar y se



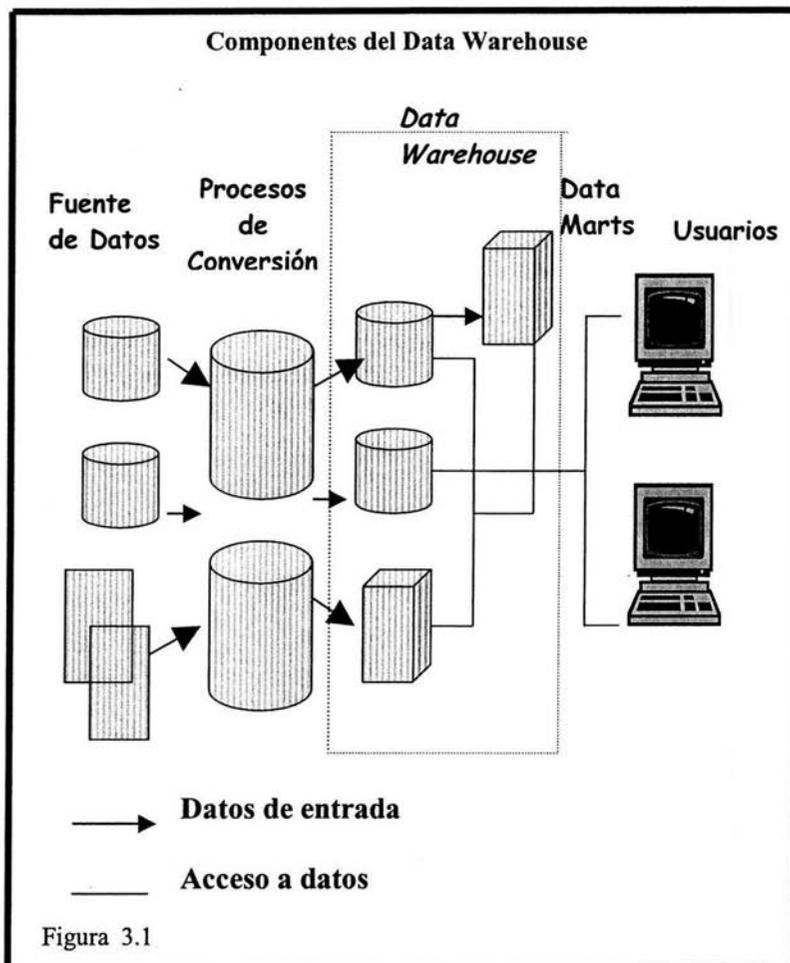
les va a dar una homogeneidad, además de checar su consistencia, integridad referencial y ser cargados a la base de datos del Data Warehouse. Sus principales características son:

- Están montados sobre una o varias computadoras.
- Pueden estar basados sobre una tecnología relacional, pero no necesariamente.
- Los usuarios no deben tener acceso a reportes en esta fase.

Data Mart: Es un subconjunto de un Data Warehouse con datos específicos para un objetivo o tema en particular, así como para una actividad específica de la empresa, tales como el área de finanzas, crédito o clientes. Pueden existir uno o más data Marts dentro del Data Warehouse; pueden ser de tipo relacional o de tipo OLAP (más adelante definiré OLAP); pueden contener detalles o resúmenes de datos los cuales pueden o no ser compartidos entre data marts.

Usuarios: Usualmente los usuarios suelen ser el personal capacitado y autorizado para tomar decisiones, dependiendo del área de consulta (Clientes, Finanzas, Recursos Humanos etc.)

Un Data Warehouse puede ser visto como una aplicación de software empresarial de misión crítica que utiliza una o más bases de datos para almacenar y recuperar información y lo que vendría siendo diferente a las demás aplicaciones operacionales de la empresa sería el modo de almacenamiento y organización y devolución de los datos; así como de el manejo voluminoso de los mismos. Cabe mencionar la diferencia de optimización de actualizaciones con las otras aplicaciones, en contra de una optimización de consulta que se da en el data Warehouse.



### 3.4.4 OLAP (On Line Analytical Process).

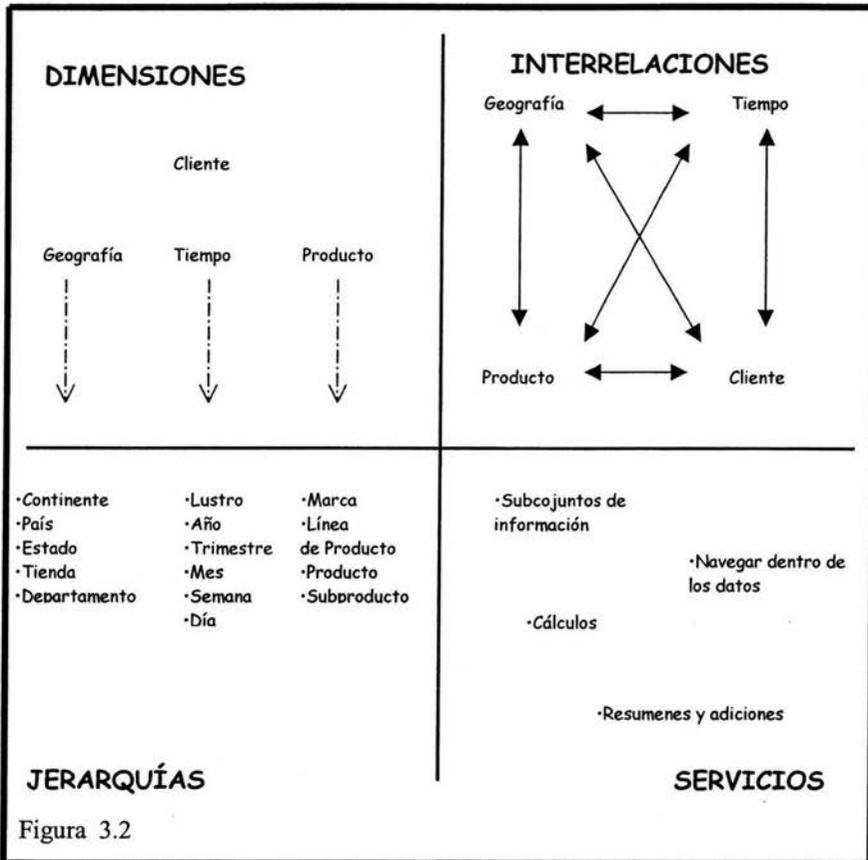
El Procesamiento Analítico nace de la necesidad que tienen los dueños o gerentes empresariales de manejar su empresa eficientemente de tal manera que pueda minimizar lo más posible el costo y de obtener la máxima utilidad posible, así como de planificar o planear su desarrollo, para esto; ellos buscan respuestas a cuestiones operacionales como por ejemplo:



- ¿Cuáles fueron los ingresos por ventas en el día de la madre, en X's artículos ? Esta pregunta la pueden hacer conforme se acerca el fin de mes.
- ¿Cuáles fueron los 5 artículos más vendidos ese día?.
- ¿Cuáles fueron los 5 menos vendidos?.
- ¿Cuáles continuaron vendiéndose más hasta el último día del mes?.
- ¿Cómo se comparan estas ventas con las de los 3 últimos años en el mismo período de tiempo?.
- ¿Y en que región, tienda y de ser posible el departamento donde se vendieron más?.
- ¿Y en cuáles menos, además de quien es el gerente de ventas de esa área?.

Los empresarios saben que deben tener respuestas donde puedan comparar lo real con lo presupuestado o planeado, de tal manera que puedan predecir el futuro comportamiento de su empresa, por eso la importancia de la funcionalidad del procesamiento analítico.

Un hecho muy importante es que las respuestas a las anteriores preguntas son de carácter multidimensionales si consideramos que la información esta relacionada además de depender entre si y por supuesto tiene una jerarquía. ¿Cómo es esto?. Bien si consideramos el hecho de llamar dimensión al parámetro *tiempo* donde su jerarquía es tal vez año, mes, día; y por que no; llamar a las *ventas* una nueva dimensión donde su escalafón puede ser producto, subproducto etc.; otra dimensión podría ser la *zona geográfica* donde sus diferentes categorías puede ser país, estado, tienda y departamento. Por lo tanto en el análisis multidimensional los datos se representan mediante dimensiones y comúnmente las dimensiones tienen distintos niveles que llamaremos jerarquías.





**OLAP (On Line Analytical Process ):** es una base de datos con tecnología diferente a la Base de Datos Relacional o bien una alternativa que ofrece mayor velocidad de consulta, vistas de datos y análisis más flexibles.

<b>Características</b>	<b>Especificaciones</b>
OLAP tienen un esquema de optimización, para agilizar las consultas	Los queries son devueltos más rápidos y son por mucho, más interactivos con el usuario que los de las bases de datos relacionales o los sistemas (RDBMS). Los cubos OLAP almacenan varios niveles de resúmenes en estructuras altamente optimizadas para la devolución de los resultados.
OLAP tiene un potente motor para los cálculos y análisis numéricos	Se usan cubos OLAP para análisis numéricos sencillos hasta algoritmos complejos de procesar. Varios de estos cálculos avanzados son optimizados en el motor de la base de datos lo que no podría hacer las bases de datos relacionales por las limitaciones en los motores RDBMS
OLAP es un concepto intuitivo de modelo de datos	Más que una tecnología de bases de datos OLAP es un concepto o modelo intuitivo que a los usuarios les es fácil de entender, sin el desarrollo de aplicaciones o reportes complejos o hechos a la medida
OLAP provee una vista multidimensional de datos	Los cubos proveen de vistas multidimensionales que son prácticamente una extensión del modelo bidimensional (tablas), estos incorporan la navegación a través de los datos por medio de expandir hacia abajo (Drill down) y contraer (Drill up) es decir pueden llegar al nivel de detalle que se haya especificado anteriormente, también lanza las funciones Pivot tables que pueden cambiar de columnas a filas y de filas a columnas en un reporte o sea el usuario puede cambiar la orientación del reporte.



### Almacenamientos de Datos en OLAP.

Hay dos maneras principales de almacenar los datos en OLAP una de ellas es el depósito de datos Multidimensional y la otra forma es la de tipo Relacional.

Características	Base de Datos Relacional	Base de Datos Multidimensional
Deposito de datos, acceso y visión	<ul style="list-style-type: none"><li>➤ Relacional</li><li>➤ Tablas campos, registros</li><li>➤ Lenguajes SQL con ampliaciones</li><li>➤ Herramientas de terceros que usan API's</li></ul>	<ul style="list-style-type: none"><li>➤ Dimensional</li><li>➤ Arreglos: Hipercubos Multicubos</li><li>➤ Tecnología de matriz inversa</li><li>➤ Hoja de cálculo como front</li></ul>
Utilización e incorporación	<ul style="list-style-type: none"><li>➤ OLTP</li><li>➤ Motores RDBMS</li><li>➤ Profundización a nivel de detalle</li></ul>	<ul style="list-style-type: none"><li>➤ OLAP</li><li>➤ Motor multidimensional</li><li>➤ Profundización a nivel resumen</li><li>➤ Desempeño de consultas rápido</li></ul>
Tamaño y actualización de bases de datos	<ul style="list-style-type: none"><li>➤ Gigabytes a Terabytes</li><li>➤ El depósito de índices aumenta el tamaño</li><li>➤ Actualización durante el uso</li></ul>	<ul style="list-style-type: none"><li>➤ Gibabytes</li><li>➤ Compresión y adición de datos dispersos</li><li>➤ Difícil de actualizar</li><li>➤ Cualquier cambio puede requerir una reorganización.</li></ul>

Cabe mencionar que todavía no hay estándar de acceso para el modelo multidimensional. Lo cual hace que los más importantes fabricantes de manejadores de bases de datos Oracle, Sybase, Microsoft entre otros utilicen el modelo relacional, por esta razón y dado que en la aplicación que se mostrará más adelante esta hecha con software de Microsoft hablaré sólo del modelo Relacional.

En este modelo (relacional) donde los datos se almacenan en forma relacional (columnas, filas) se presentan como dimensiones, también se crean metadatos



adicionales para cualquier resumen o adición y esto es para mejorar el tiempo de respuesta y son almacenados en un depósito de metadatos.

**Metadatos:** Son datos acerca de los datos aunque pareciera redundante el término. Es decir son características, información, observaciones o atributos acerca de los mismos datos, por ejemplo: Nombre del cliente que es de 35 caracteres de longitud, que se encuentra en la tabla de Clientes, que está almacenada de cierta manera y no es el nombre del cliente en si; o sea el dato.

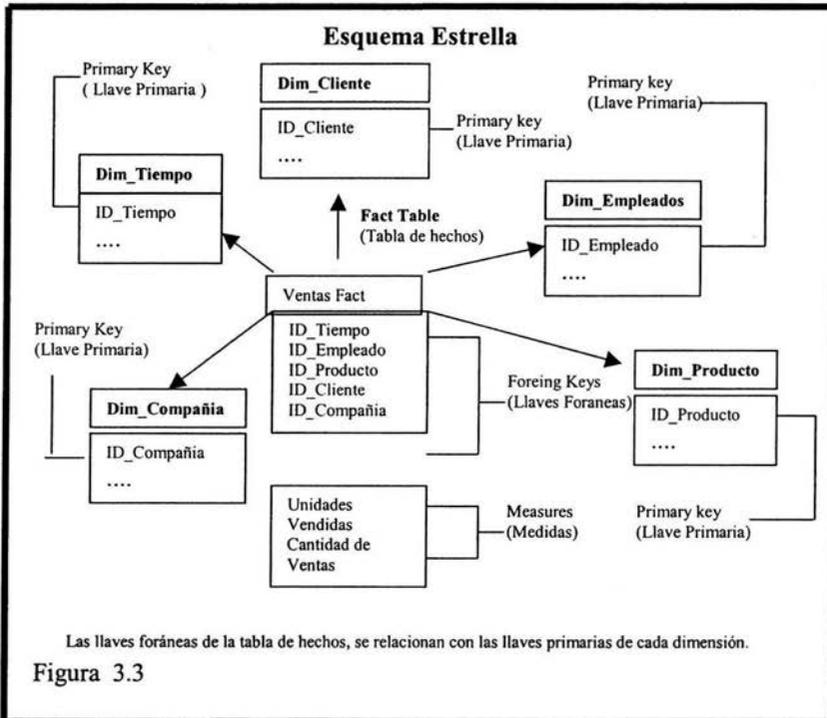
Es importante destacar que para el modelo Relacional es necesario diseñar la Base de Datos, para ello es necesario tener en cuenta los esquemas de Estrella, Copo de Nieve, Padre hijo y los modos de almacenamiento (MOLAP, ROLAP y HOLAP).

#### 3.4.4.1 Tipos de Esquemas

Para facilitar la devolución y el análisis de los datos en un Data Warehouse es necesario organizar los datos conforme a ciertas estructuras o esquemas .



El **esquema estrella** (Star Schema) es caracterizado por tener una tabla central conocida como *Fact Table* y alrededor de ella tablas desnormalizadas con datos que regularmente se conocen como *Dimensiones*. Es decir existe una tabla central con medidas del negocio que describen eventos específicos en una empresa u organización, tales como transacciones bancarias, venta de productos, ingresos, egresos etc. Figura 3.3



- **Fact Table:** Es la tabla central que presenta datos numéricos sobre una entidad del negocio.

Esta conformada por:

**Measures:** Es un valor, cantidad numérica o columna numérica en la *Fact Table*. Las measures regularmente representan los valores a analizar

**Foreign keys:** Es la llave primaria de una dimensión, que está como columna(s) en la *Fact Table*. Continuamente la combinación de las



Foreign keys forman la llave primaria de la *Fact Table*; es decir son el identificador único de cada registro en la *Fact Table*.

**Fact Table Grain:** (Granularidad) Es el nivel más bajo de detalle almacenado en cada una de las dimensiones asociadas a la *Fact Table*.

➤ Dimensiones (Tablas):

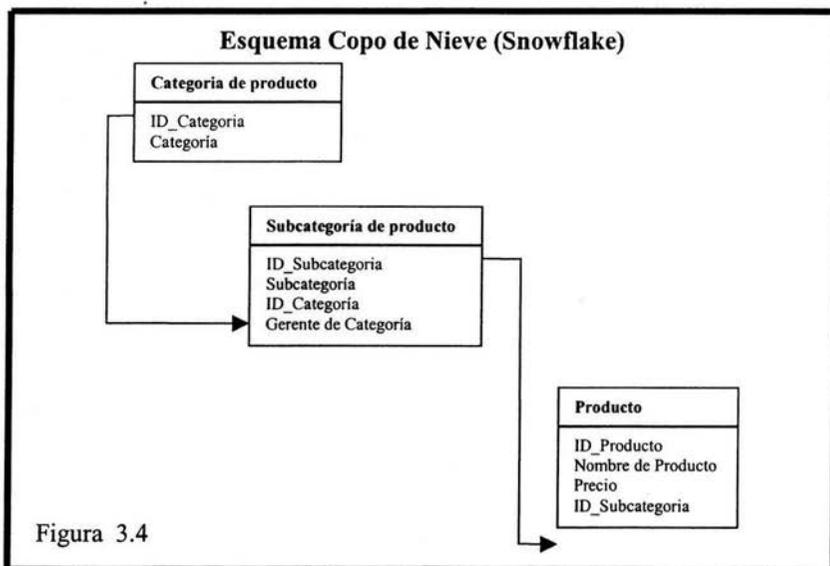
Características:

Entidades que describen el negocio

Atributos que proveen de un contexto numérico

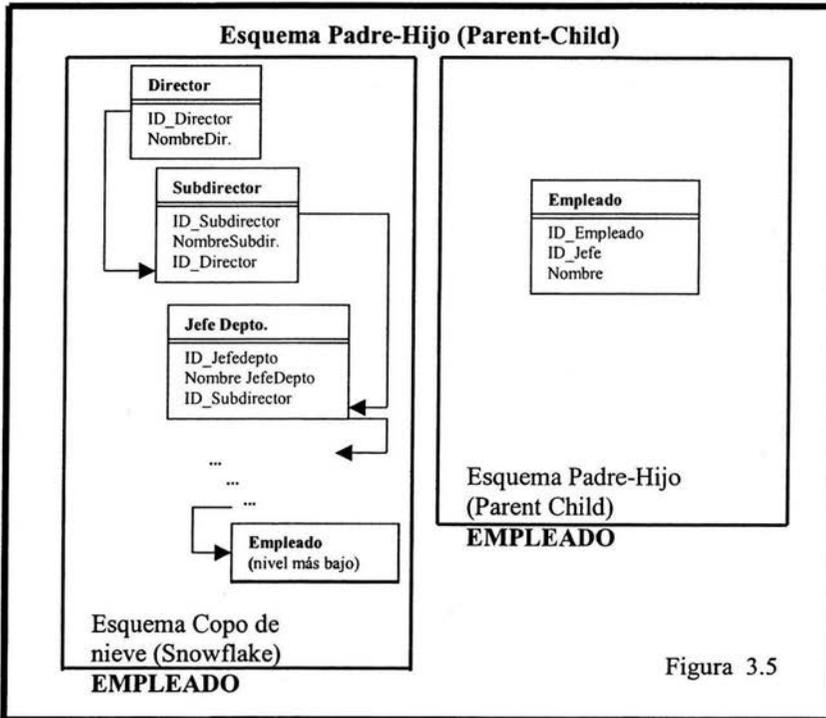
Presentan datos organizados en jerarquías

**El Esquema Copo de Nieve:** (Snowflake). Es una variante del esquema de estrella en la cual la jerarquía de las dimensiones esta almacenada en múltiples tablas o dimensiones y otra característica es que las tablas están más normalizadas que si fuera una sola tabla.





**Esquema Padre-Hijo:(Parent-Child).** Este esquema es muy recurrido cuando una dimensión tiene cierto desbalance jerárquico. El desbalance ocurre cuando algunas de las ramas de la jerarquía contiene más niveles que otras ramas. El esquema Parent-Child es muy similar al esquema Snokflake en que cada liga (join) crea un nivel nuevo. La diferencia fundamental es que en el nivel más alto en la jerarquía parent-child; así como la del nivel más bajo deriva de la misma dimensión como se puede ver en el ejemplo .





En este ejemplo donde se desea saber ¿De quien depende X empleado y que jerarquía tiene?, se podrá notar la clara diferencia entre un esquema Snowflake y el Parent-Child y por que es más conveniente utilizar el esquema Parent-Child a el Snowflake

**Director**

Id_director	NombreDir.
01	Juan Gómez
02	Elvia Lomas
03	Victor Rojas

**Subdirector**

Id_Director	Id_Subdirector	Nombre
01	011	Pedro Jiménez
01	012	Sofía Esquivel
02	021	Francisco Torres
02	022	Ricardo Ledesma
03	031	Margarita Hernández

**Jefe de Departamento.**

Id_Subdirector	Id_JefeDepto	Nombre
011	0111	Elena Rodríguez
011	0112	Roberto Mendieta
011	0113	Alfonso González
012	0121	José Díaz
012	0122	Arturo Suárez
021	0211	Humberto Álvarez
021	0212	Marco Martínez
022	0221	Alejandro Gutiérrez
022	0222	Rosa Trujillo
022	0223	Eduardo Galván

**Empleado (Parent-Child)**

Id_Empleado	Id_Jefe	Nombre	Puesto
01	01	Juan Gómez	Director
02	02	Elvia Lomas	Director
03	03	Víctor Rojas	Director
04	01	Pedro Jiménez	Subdirector
05	01	Sofía Esquivel	Subdirector
06	02	Francisco Torres	Subdirector
07	02	Ricardo Ledesma	Subdirector
08	03	Margarita Hernández	Subdirector
09	04	Elena Rodríguez	Jefe de Departamento
10	04	Roberto Mendieta	Jefe de Departamento
11	04	Alfonso González	Jefe de Departamento
12	05	José Díaz	Jefe de Departamento
13	05	Arturo Suárez	Jefe de Departamento
14	06	Humberto Álvarez	Jefe de Departamento
15	06	Marco Martínez	Jefe de Departamento
16	07	Alejandro Gutiérrez	Jefe de Departamento
17	07	Rosa Trujillo	Jefe de Departamento
18	07	Eduardo Galván	Jefe de Departamento



### 3.4.4.2 Cubos

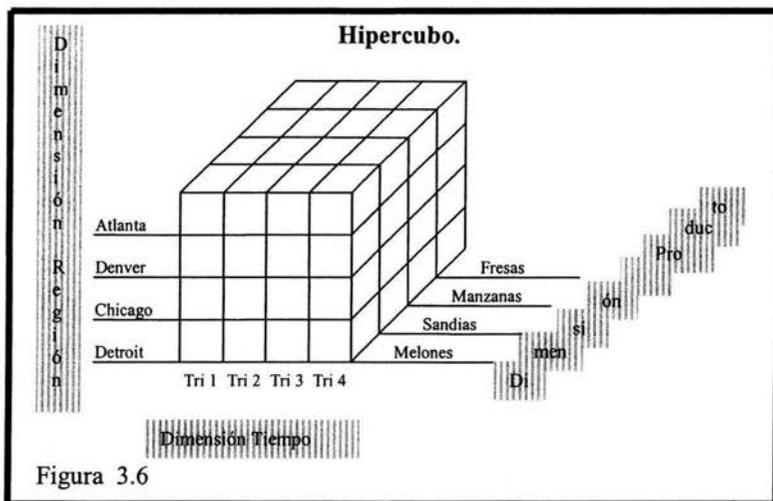
Una vez que se tiene identificado que esquema(s) se va(n) a utilizar se introduce el concepto de cubo en OLAP, que es básicamente una estructura de almacenamiento lógica que combina varias dimensiones y medidas, en modelos flexibles para que el usuario pueda manipular de una manera intuitiva y sencilla la información.

Definición de cubo: Conjunto de dimensiones (tablas) relacionadas que forman una matriz n-dimensional.

Entre sus principales características destacan que en cada celda del cubo está contenido un valor y es producto de la intersección de las dimensiones. Es importante recalcar que un cubo en OLAP no es solo de 3 dimensiones y que algunos lo conocen como hipercubo cuando este, excede a las 3 dimensiones y que para visualizarlo es más sencillo representarlo con rectas donde cada recta representa una dimensión.

En el Ejemplo siguiente veremos un hipercubo con 3 dimensiones

- Región
- Tiempo
- Producto





Con él podemos contestarnos preguntas como ¿Qué productos se vendieron en el trimestre 2 en la región de Chicago? La respuesta será muy sencilla si encontramos el punto de intersección Chicago, segundo trimestre, podemos notar que no es una sola celda la solución a la respuesta; es un plano que contiene a manzanas, melones, sandías y fresas, pero podemos elaborar preguntas más específicas que sólo nos devuelva algunos cuantos valores (subconjuntos más pequeños de información) que llamaremos Slice o rebanadas del cubo.

Por ejemplo ¿Cuántas toneladas de manzana se vendieron el trimestre 2 en Chicago? o ¿Cuántas toneladas de manzana verde se vendieron en la segunda semana de el trimestre 2 en la parte norte de Chicago?, es intuitivo el hecho de que entre más elaborada o específica sea la pregunta se va a obtener un subconjunto cada vez más pequeño, que siempre dependerá del nivel de detalle de las dimensiones; por ejemplo en nuestra dimensión tiempo podemos tener niveles de jerarquía como año, trimestres, meses, semanas, días, horas, minutos inclusive segundos.

Por otro lado si tomamos en cuenta que nuestra pregunta involucra milisegundos y no está ese nivel de jerarquía en nuestra dimensión es claro que no tendremos respuesta en nuestro cubo, pero podemos auxiliarnos de las matemáticas haciendo interpolaciones, regresiones, estadísticas o estudios probabilísticos para obtener algún resultado.

Ahora que pasa si necesitamos hacernos una pregunta como la siguiente ¿Qué tipo de clientes compraron el producto sandía en el trimestre 3 en la zona sur de Atlanta?. Es obvio que no tendremos respuesta alguna si es que no involucramos una nueva dimensión llamada Cliente, pero no podemos imaginarnos como expresarla gráficamente con un hiper cubo de 3 dimensiones, de ahí que nos ayudaremos de columnas que representaran a las dimensiones. Por ello la importancia de tener herramientas ejecutivas (no técnicas) que puedan utilizar los empresarios o directivos que les den una idea real de cuándo, cómo, dónde y por qué está el negocio como hasta ahora.

Es muy importante que estas herramientas tengan una buena velocidad de respuesta y esto va a depender en cierta medida de la forma física de almacenar la información que se verá en el siguiente tema.



### 3.4.4.3 Modos de almacenamiento.

Los modos de almacenamiento van a depender completamente del software en este caso particular usamos la tecnología Microsoft y más adelante se explicará el por que. La herramienta que nos proporciona los servicios OLAP es el Analysis Server y dentro de él existen 3 modos de almacenamiento MOLAP, ROLAP y HOLAP.

Los tres modos de almacenamiento tienen en común las siguientes características:

Son administrados de igual manera e incluyen elementos como:

- Diseño de agregaciones (Como son creados los resúmenes de información)
- Construcción de agregaciones (Creación de resúmenes de información)
- Particiones (Unidad física de almacenamiento definida para un cubo OLAP)
- Construcción de dimensiones (Entidad o atributos que proveen de un contexto numérico)

Ninguno de los modos de almacenamiento aloja agregaciones nulas. Por ejemplo si no se venden bikinis en la Antártica, no se almacenará información y por lo tanto no se desperdiciará espacio en disco.

Los tres proveen el mismo modelo multidimensional a clientes, usuarios y aplicaciones. Es decir es completamente transparente para los clientes, usuarios y aplicaciones que solo verán el cubo y no el modo en que fue hecho .



En cualquier momento del desarrollo o de vida del cubo puede cambiarse el modo de almacenamiento.

Es importante destacar que las estructuras de las dimensiones se almacenan sobre el Analysis Server, excepto cuando las dimensiones son diseñadas para usarse en el modo ROLAP.

Características propias de los modos de almacenamiento:

#### *MOLAP:*

- El detalle de los datos y las agregaciones son almacenadas en Analysis Server .
- Los cubos MOLAP tienen la más alta velocidad de respuesta a una consulta dada.
- Dado que el nivel de detalle de la Fact Table es muy amplio, esto repercute en duplicidad de datos lo que implicaría mayor uso de espacio en disco, pero debido a la eficiencia de algoritmos de compresión de datos hace que MOLAP sea una buena opción en terminos de espacio en disco.
- Como las consultas (Querys) del Analysis Server no acceden a las bases de datos relacionales, no importa si está abajo el servidor de RDBMS.

#### *ROLAP:*

- Los detalles de los datos y las agregaciones son almacenadas en el RDBMS, y solo las jerarquías de las dimensiones son almacenadas en el Analysis Server; lo que implica un menor uso de espacio en disco.
- El Analysis Server provee de una capa intermedia entre los clientes y el RDBMS.
- Los resultados son devueltos por el RDBMS y se alojan en la memoria cache del cliente o servidor.
- Por varias circunstancias entre ellas las ligas (joins) el modo ROLAP tiene menos velocidad de respuesta (performance) que el modo MOLAP.

*HOLAP:*

- Los detalles de los datos son almacenados en el RDBMS y las agregaciones son almacenadas en el Analysis Server en forma multidimensional.
- HOLAP es una combinación entre MOLAP y ROLAP en términos de velocidad (performance) y de almacenaje en disco.
- Los detalles de los datos no son duplicados y las agregaciones están en el Analysis Server en forma multidimensional, lo que reduce en menor espacio en disco en comparación a el modo MOLAP y obviamente mayor espacio con respecto al modo ROLAP.
- Las consultas no son tan lentas como en el modo ROLAP, pero no tan rápidas como en el modo MOLAP.
- Los tiempos de procesamiento para el modo HOLAP es muy similar a el modo MOLAP.



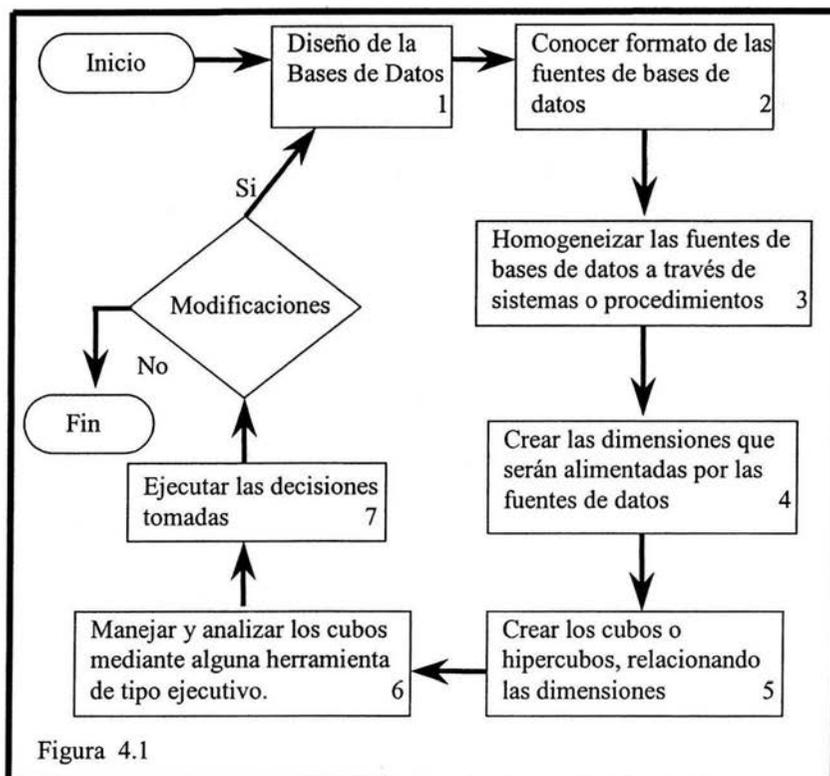
## Capítulo IV.

### Diseño e implementación OLAP.

En este capítulo se plantea la forma de construir un procedimiento ágil, veraz y oportuno que ayude a tomar decisiones correctas al personal no técnico en informática, como puede ser un directivo. Tomando en cuenta las ventajas de las herramientas OLAP que permiten jugar con la información de la cual cuenta la empresa o institución, además de reducir el tiempo de respuesta y sin que el desarrollador de software de la empresa, tenga que construir previamente esas consultas, como se realizarían en un sistema cliente-servidor, que accede a una base de datos relacional por medio de una interfaz creada en algún lenguaje de programación. Para ello es necesario considerar los siguientes aspectos:

- Un diseño de bases de datos para conformar la fuente de datos que solución al problema planteado.
- Conocer el formato de las fuentes de datos.
- Homogeneizar estas fuentes de datos a través de sistemas o procedimientos.
- Crear las dimensiones que serán alimentadas por nuestras fuentes de datos.
- Crear los cubos o hipercubos, relacionando las dimensiones.
- Manejar y analizar los cubos mediante alguna herramienta de tipo ejecutivo.
- Ejecutar las decisiones tomadas.
- Si hay modificaciones o ampliaciones se repite el ciclo de lo contrario se finaliza el proceso.

Ver siguiente figura 4.1



#### 4.1 Diseño de la Base de Datos.

Siguiendo el diagrama de flujo de la figura 4.1 se crea el diseño de la base de datos (1).

De acuerdo al ejemplo de una Institución Educativa que se ha venido manejando, y ajustándose a el código presupuestal del que se habló en el capítulo 2 y debido a que la información es requerida diariamente y con cortes a fin de mes, se considera que el siguiente diseño de la base de datos da respuesta eficiente, veraz, y rápida aunque no suficiente para tomar alguna decisión; por tal motivo y debido a la naturaleza del Data Warehouse resulta más conveniente para cubrir las necesidades de información, que cualquier programa que esté basado en una bases de datos relacional.



## Diagrama Entidad-Relación

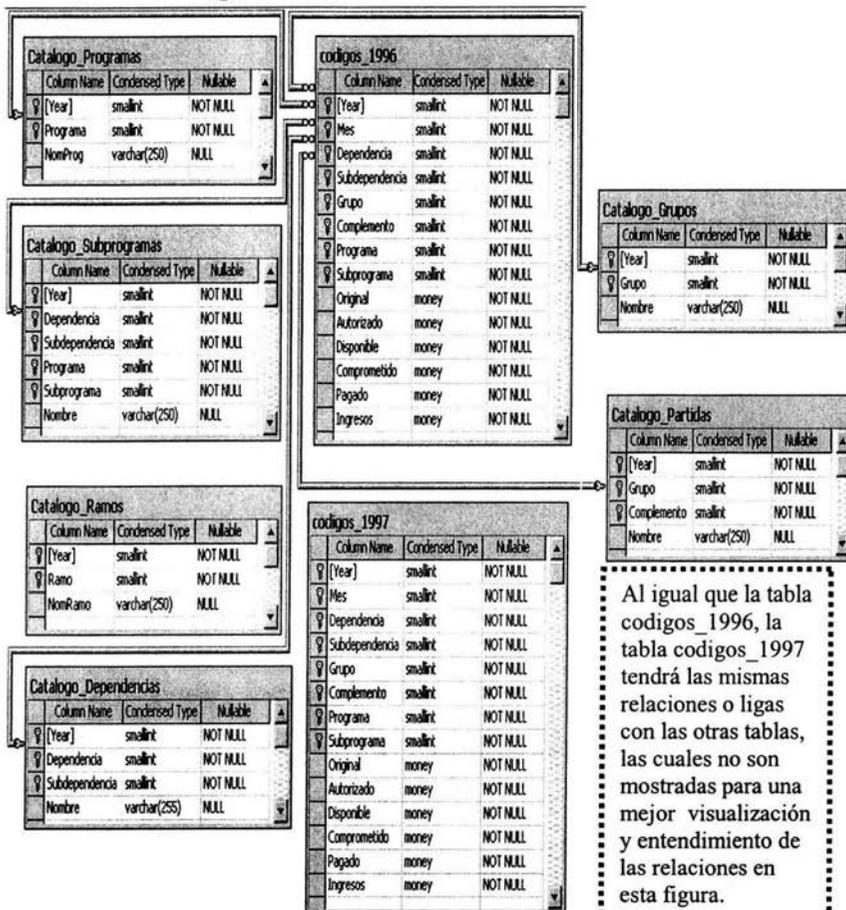


Figura 4.2

Este diseño permitiría visualizar los movimientos diarios a través de un programa que hiciera acceso a esta información, además de insertar y/o modificar registros y probablemente dar respuesta a preguntas muy específicas o cotidianas, a diferencia de las herramientas OLAP, que utilizan como soporte estos datos además de tener una mayor versatilidad, creando una extensa gama de posibles consultas y respuestas al instante, sin la necesidad de crear estas



opciones en el programa que de igual manera accederían a esta fuente de datos. Cabe mencionar que no todo se mejora, pues esto tiene un costo en cuanto a espacio y redundancia de datos como se habló en el capítulo 3, que en lo particular no considero grave, pues la velocidad de creación de un Data Warehouse es muy rápida, así como la devolución de los resultados esperados y si se toma en cuenta que hoy en día el costo de almacenaje es muy barato considero que es mayor la ganancia.

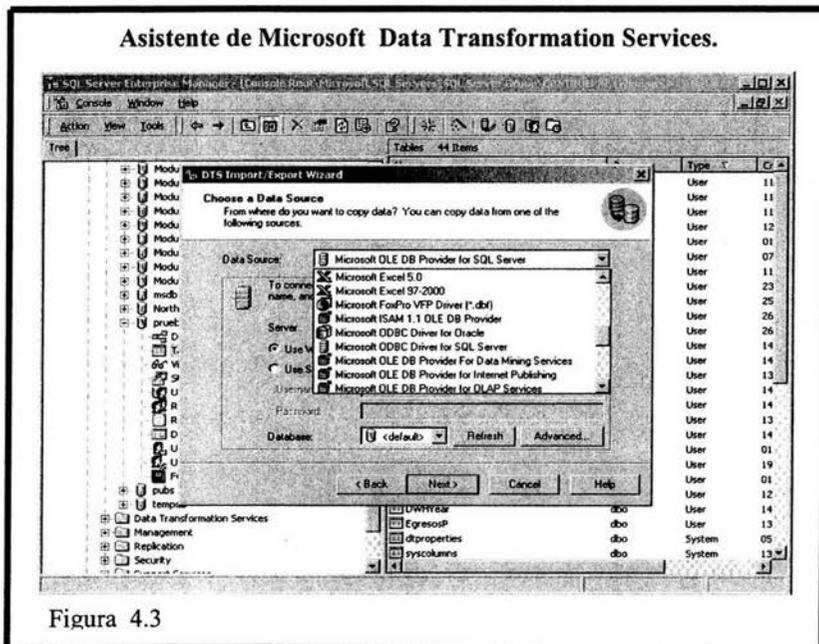
## 4.2 Conocer el formato de las fuentes de datos.

Continuando con el diagrama de la figura 4.1 conocer formato de las fuentes de bases de datos (2). Como en todo problema real la fuente de base de datos ya cuenta con datos históricos que alimentaran a nuestro Data Warehouse o Data Marts (con formatos de tipo texto, hoja de cálculo y de manejadores de datos) por lo que en la figura se toma un año en particular (1996). Es de mencionar que en la figura anterior, la tabla de Codigos\_1997 no fueron puestas las relaciones para evitar alguna confusión pero todo lo que se aplique para la tabla de Codigos\_1996 se puede aplicar exactamente igual a la de Codigos\_1997. También se puede notar que en la tabla Catálogo\_Ramos no tiene una llave foránea que haga una relación directa con la tabla de códigos\_1996, aunque realmente existe (el primer carácter de la dependencia nos dirá a que ramo pertenece), es por ello que no se unió con alguna liga, para que no se piense que el grupo es igual a la dependencia.

## 4.3 Homogeneizar las fuentes de bases de datos a través de procedimientos.

El siguiente paso de nuestro diagrama de la figura 4.1 Homogeneizar las fuentes de bases de datos (3).

Para Homogeneizar las distintas fuentes de datos Microsoft SQL Server cuenta con una herramienta de transformación de datos dentro del Enterprise Manager, llamada Data Transformation Services, la cual puede importar o exportar datos según el origen o fuente a la que se desee llevar mediante un Asistente muy fácil de usar como se ilustra en la siguiente figura 4.3.



A fin de ejemplificar pudiera ser que se tengan datos en archivos de Excel y que además, sólo algunas columnas sean de interés las cuales pasaremos a una o algunas tablas en una Base de datos que por manejador tenga a Microsoft SQL Server. Existe una gran variedad de métodos para realizar esta tarea, los cuales pueden variar en el número de pasos a seguir así como el tiempo que requiere cada uno de ellos. Aquí se muestra un método que puede ser muy eficiente: Primero se copian las columnas de interés y se pegan en una hoja nueva de Excel, de tal manera que las columnas, queden en el orden deseado, se salva el archivo y se utiliza el Data Transformation Services Asistente, en el cual se seleccionará la plataforma en la que está el archivo (Excel) y después se escoge la plataforma a la cual se desea llevar (Tabla de SQL Server 2000).

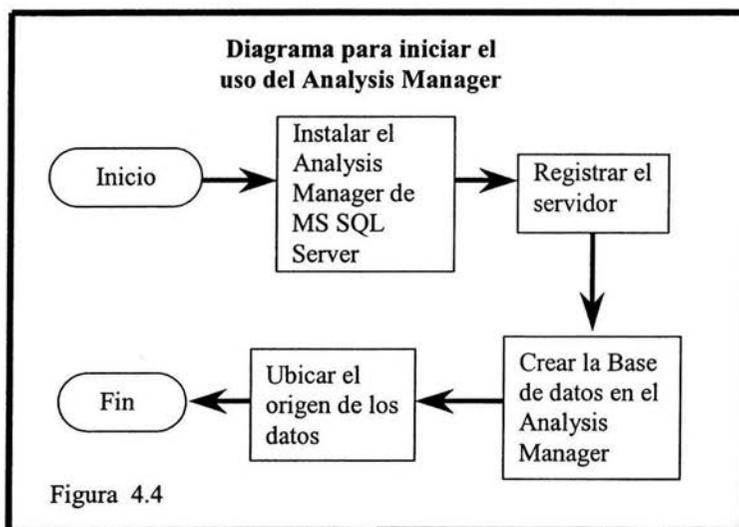
En el siguiente paso se selecciona la hoja y se asigna un nombre a la tabla así como las características de tipo y tamaño de los campos de la tabla y esto da por finalizado la importación de datos para el caso que hemos planteado. Este procedimiento se utiliza hasta finalizar el depurado y homogenización de la plataforma que contendrán a los datos (SQL Server 2000), cabe aclarar que puede ser una o más tablas.



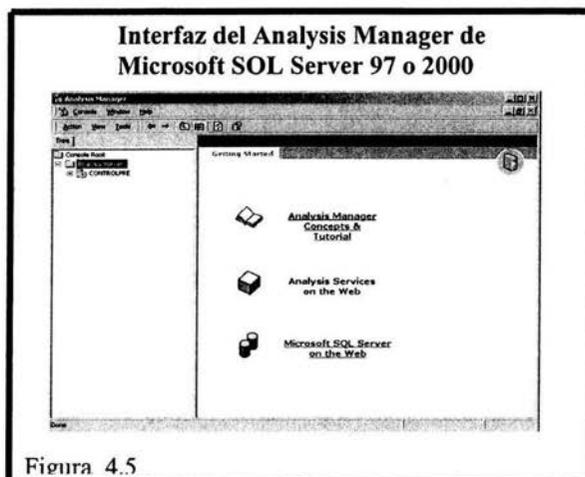
Una vez que se ha depurado y homogeneizado la información; la estructura resultante son tablas con algunos de los movimientos referentes al presupuesto y que son asentados conforme el código programático según nuestro caso.

#### 4.4 Creación de dimensiones.

Los pasos para crear las dimensiones (4) se muestran en la figura 4.4

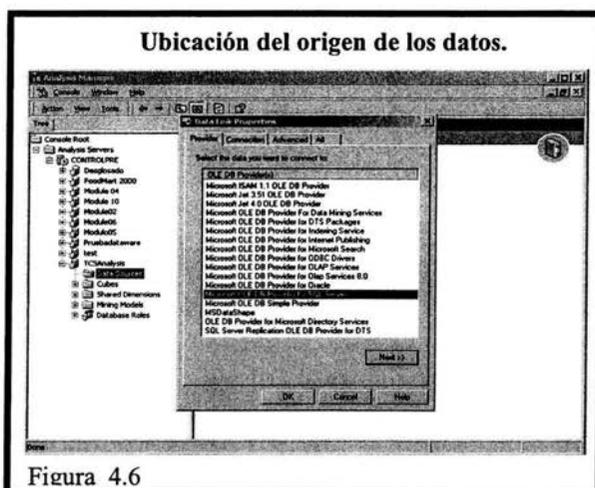


Para crear una dimensión hay varios caminos de los cuales tomaremos el más fácil para esto necesitaremos utilizar el Analysis Manager que es la interfaz para diseñar y administrar los servicios de análisis y cubos dentro de Microsoft SQL Server (figura 4.5).



Es importante mencionar que una vez que se registra el MSSQL ServerOLAPService dentro del Analysis Manager se procede a crear una nueva Base de Datos (Se llamará TCSAnalysis) que puede o no tener el mismo nombre que la del Enterprise Manager que ya hemos creado y de la cual se ha mostrado el Diagrama Entidad-Relación figura 4.2.

Otra nota importante y que es previa a crear una dimensión es ubicar el origen de los datos para TCSAnalysis del Analysis Manager (figura 4.4), como se puede apreciar en la figura 4.6.





Continuando con nuestro diagrama de flujo figura 4.1 (crear una dimensión). Esta se puede crear de varias formas, pero en este trabajo se utilizará el Asistente de Dimensiones por ser una herramienta de rápido y fácil manejo para la construcción de dimensiones, es de aclarar que los otros métodos se hacen sobre el Analysis Manager sobre el folder de Shared Dimensions, cabe recordar que en el capítulo anterior se habló de los diferentes tipos de esquemas y de las jerarquías en una dimensión; por otro lado existirán dimensiones que podrán compartirse y otras que solo podrá accederlas uno y sólo un cubo. Para nuestro ejemplo crearemos una que se podrá compartir con otros cubos en el Asistente de dimensiones siguiendo la siguiente metodología.

- 1) Seleccionar el esquema.
- 2) Escoger la tabla que nos servirá como dimensión.
- 3) Establecer los niveles de la dimensión (jerarquía).
- 4) Asignar las columnas que servirán como enlace entre las otras dimensiones.
- 5) Elegir el modo de almacenamiento o tomar el que esta por default (ROLAP<sup>1</sup>, MOLAP<sup>2</sup>).
- 6) Asignar un nombre a la dimensión.

En la figura 4.7 se muestran la mayoría de las pantallas del asistente para la creación de dimensiones la cual lleva un orden que es especificado por las flechas y la numeración en la misma figura.

---

<sup>1</sup> ROLAP Relational Online Analytical Processing

<sup>2</sup> MOLAP Multidimensional Online Analytical Processing



### Asistente de Dimensiones del Microsoft Analysis Manager.

The figure displays four sequential screenshots of the Microsoft Analysis Manager Dimension Wizard, numbered 1 through 4, illustrating the process of creating a dimension.

**Step 1: Choose how you want to create the dimension**  
This screen offers four options for creating a dimension:  
 **By Schema: A single dimension table**  
 **Stronglike Schema: Multiple, related dimension tables**  
 **Event-Child: Two related columns in a single dimension table**  
 **Global Dimensions: The member properties of another dimension**  
A description at the bottom states: "In a later step, select one or more columns from a single table. Each column contributes a level to the dimension (or a single data column can be parsed to create multiple levels. If you select multiple columns, they often contain increasingly detailed information. For example, the columns Country, Province, City, and State create a Stores dimension. This option creates a regular dimension."

**Step 2: Select the dimension table**  
This screen shows a list of available tables under the 'CONTRIBUORE - TCSAnalysis' schema. The table 'tbl\_depco\_dependencia' is selected. The 'Details' pane shows the columns: 'Year', 'Dependencia', 'Subdependencia', and 'Numero'.

**Step 3: Select the levels for your dimension**  
This screen shows the mapping of columns to levels. The 'Available columns' list contains 'Year' and 'tbl\_depco'. The 'Dimension levels' table is as follows:

Level name	Source column
* Dependencia	Dependencia
- Subdependencia	Subdependencia

**Step 4: Finish the Dimension Wizard**  
This final screen shows the 'Dimension name' set to 'Ramo' and the option 'Create a hierarchy of a dimension' checked. A preview pane displays a tree structure of the dimension levels:

```
[-] Ramo
  [0] 111
  [0] 112
  [0] 113
  [0] 114
  [0] 115
  [0] 116
  [0] 118
```

Figura 4.7



En el ejemplo que se está tratando de la entidad educativa se han creado las siguientes dimensiones utilizando el asistente de dimensiones:

- Dimensión "Programa", que contendrá las funciones y los programas según el código programático.
- Dimensión "Ramo", que estará formada por ramos, dependencias y subdependencias
- Dimensión "Objeto del Gasto", que se conformará por las partidas
- Dimensión "Subprogramas", representada por los mismos subprogramas
- Dimensión "Año", período de tiempo
- Dimensión "Mes", período de tiempo

En la siguiente figura 4.8 se muestra el editor de dimensiones donde se puede personalizar o formatear varias propiedades de las mismas. En la parte superior izquierda se puede ver la jerarquía de la dimensión; en este caso; ramo está por encima de dependencia y esta a su vez de subdependencia, algunas características importantes son el poder concatenar campos, dar diferentes formatos a los números, cambiar tipo y tamaño de letra etc.

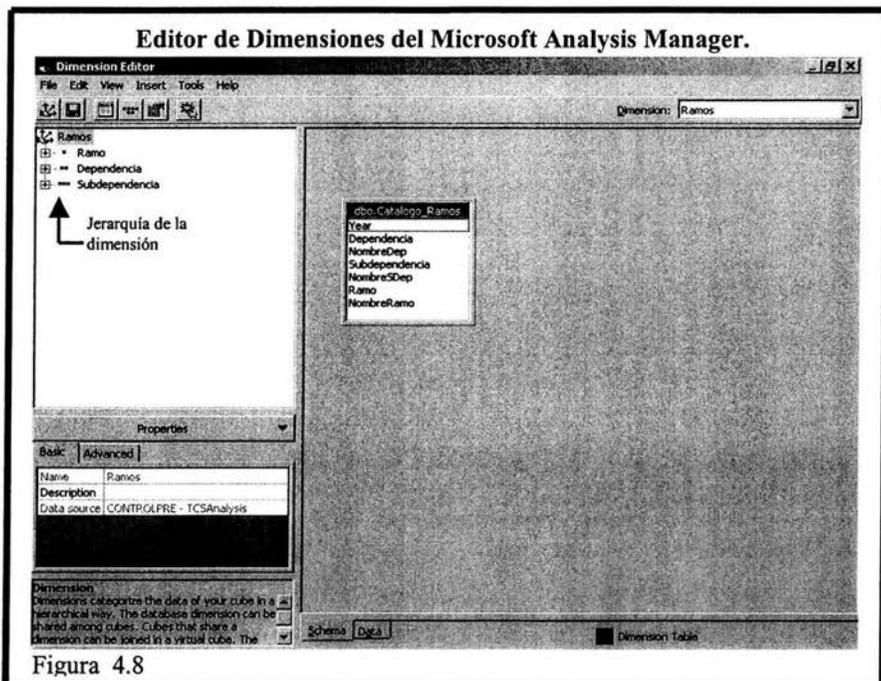


Figura 4.8



## 4.5 Creación de cubos o hipercubos.

Para seguir la estructura de la figura 4.1 paso (5), crear cubos o hipercubos y continuando con la misma dinámica; el cubo que se creará mediante el asistente de cubos que es la manera más sencilla de crear cubos o hipercubos (figura 4.9).

- Primero se selecciona la Fact Table, para nuestro ejemplo se toma la tabla de *Codigos\_1996*, pues esta será la piedra angular del cubo.
- Escoger las medidas sobre las que se realizará la inferencia de información en la que actualmente es la Fact Table (*Codigos\_1996*), se puede ver claramente que en la tabla *Codigos\_1996* las medidas son original, autorizado, disponible, comprometido, pagado e ingresos.
- Elegir las dimensiones que interactuarán entre si, como estamos manejando un data mart<sup>1</sup> de presupuesto lo lógico es utilizar todas las dimensiones que involucran a él mismo.
- Asignar un nombre al cubo.
- Personalizar o dar formato al cubo.

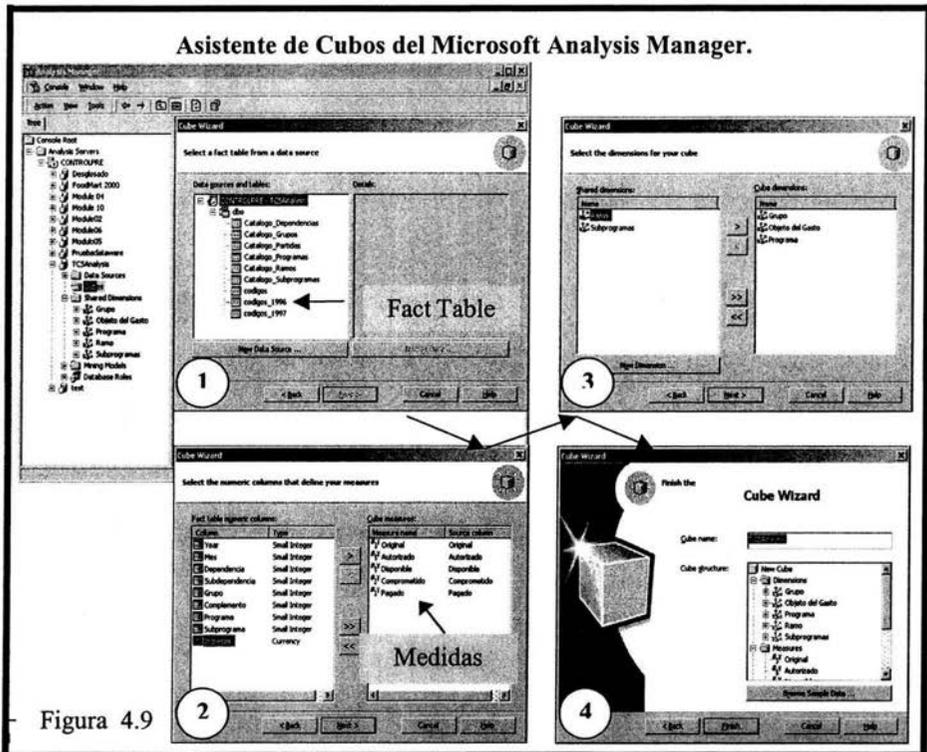


Figura 4.9



Al igual que con las dimensiones se puede modificar la apariencia de los datos a través de sus propiedades mediante el editor de cubos mostrado en la figura 4.10. Se pueden observar las dimensiones involucradas y las medidas, así como un diagrama de la relación entre la Fact table y las dimensiones; es de hacer notar dos cosas muy importantes en el editor de cubos:

La primera es que sólo se podrán modificar propiedades a la Fact table, así como a sus medidas y de ninguna forma se podrán modificar propiedades de las dimensiones involucradas.

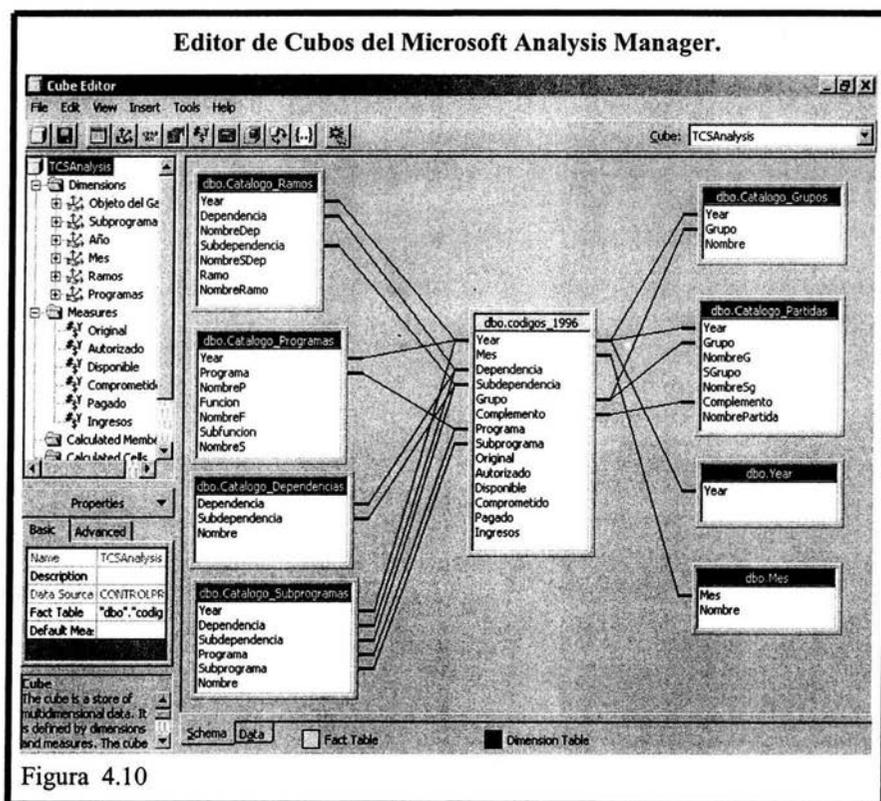


Figura 4.10

La segunda es que cuando no se cuenta con una interfaz de visualización y manipuleo de la información, este nos puede servir para tener una idea de cómo aparecerá la información que se encuentre en la fact table y las dimensiones; aunque cuando hay varias dimensiones y medidas la forma de mostrar la



información es muy pobre o definitivamente inteligible, como se puede apreciar en la figura 4.12, en esta misma figura se puede apreciar que le falta formato a la información, que como ya se ha mencionado se puede dar a través de las propiedades de las dimensiones.

Un ejemplo de información sin formato es la que aparece debajo de Grupo el 1, luego 2 más abajo 3 y así sucesivamente, estos números nos estarían diciendo la clave del grupo sin embargo no la identifica así el ejecutivo, por lo cual dándole el debido formato aparecería 100 y su nombre descriptivo (como en el catalogo de la sección 2.2 Grupos de Gasto de esta tesis), luego 200 y su nombre, más abajo 300 y su respectivo nombre y así sucesivamente de tal manera que todos los grupos, niveles y subniveles tengan su clave y su nombre respectivo que le es más familiar al tomador de decisiones y no solo al diseñador de la base de datos (técnico).

Para entender un poco más cómo opera la forma de consultar la información en el editor de cubos de la figura 4.12 es necesario ver la figura 4.11, donde podemos apreciar los objetos de los que esta compuesta la interfaz del editor de cubos en la pestaña de data. El manejo de la información es muy sencilla pues se basa en clics de botones, seleccionar niveles en los combos y arrastrar dimensiones al grid, para que se integren en el mismo y se pueda expandir o contraer niveles dentro del grid según la información y el detalle que se quiera ver dando clics a el signo de más o de menos.

Como se puede apreciar en la figura es como una especie de árbol que tiene varias ramificaciones (niveles, subniveles), que por lo regular van de lo general a lo particular o viceversa lo que permite ampliar o reducir de manera horizontal y/o vertical el grid, como una especie de acordeón que oculta subniveles cuando es requerido o expande subniveles, cuando se quiere mayor detalle de la información; esto trae como consecuencia una gran variedad de combinaciones que sirven para encontrar las fallas o aciertos e inclusive el mejoramiento del comportamiento del negocio o entidad al nivel de detalle que el directivo desee.



## Editor de Cubos "Pestaña DATA" del Analysis Manager.

Botones (Dimensiones)

Base de Datos

Combos (niveles)

Medidas

Grid

Dimensión en el grid

Niveles dentro de la Dimensión que está dentro del Grid

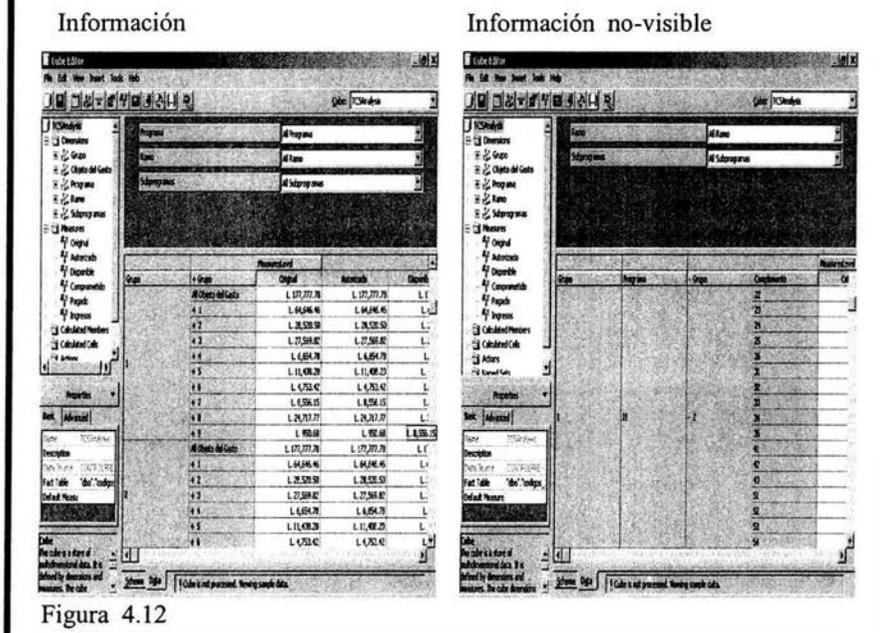
Subniveles dentro de los niveles de la Dimensión

Grupos	Original	Adaptado	Reparado
Al Grupo de Gato	L. 177,777.78	L. 177,777.78	L. 1
1.1	L. 64,646.46	L. 64,646.46	L. 1
1.2	L. 28,528.52	L. 28,528.52	L. 1
1.3	L. 27,589.82	L. 27,589.82	L. 1
1.4	L. 6,654.76	L. 6,654.76	L. 1
1.5	L. 11,488.20	L. 11,488.20	L. 1
1.6	L. 4,702.02	L. 4,702.02	L. 1
1.7	L. 6,526.65	L. 6,526.65	L. 1
1.8	L. 29,727.27	L. 29,727.27	L. 1
1.9	L. 799.80	L. 799.80	L. 1
Al Grupo de Gato	L. 177,777.78	L. 177,777.78	L. 1
1.1	L. 64,646.46	L. 64,646.46	L. 1
1.2	L. 28,528.52	L. 28,528.52	L. 1
1.3	L. 27,589.82	L. 27,589.82	L. 1
1.4	L. 6,654.76	L. 6,654.76	L. 1
1.5	L. 11,488.20	L. 11,488.20	L. 1
1.6	L. 4,702.02	L. 4,702.02	L. 1

Figura 4.11



### Editor de Cubos "Pestaña DATA" del Analysis Manager.



## 4.6 Manejo y Análisis de cubos.

Continuando con los pasos del diagrama de la figura 4.1 (Manejo y Análisis de cubos).

Para manejar y analizar los cubos son necesarias herramientas de tipo OLAP y además existen varias en el mercado, su principal diferencia estriba en la flexibilidad de los programas ya que por lo regular no es un sólo programa el que ofrecen, pues la mayoría de las empresas que se dedican al desarrollo de este tipo de herramientas suelen vender suites que son bastante caras y por ello las venden por separado, lo que en cierta medida perjudica o beneficia a el cliente, pues dependiendo de la etapa de desarrollo de su Data Warehouse y de sus necesidades, esté podrá adquirir herramientas de consolidación, transformación, manejadores, repositorios de información, de diseño e implementación de tipo OLAP, reportadores de información, visualizadores y manipuladores de datos.



Para este punto y de acuerdo con la fase en que se encuentra este Data Warehouse , necesitamos una herramienta que nos permita ver la información que se ha de generar a través de los cruces de las dimensiones.

Por otro lado, para que un ejecutivo pueda analizar la información es necesario destacar algunos roles. Por ejemplo deberá existir un grupo de personas las cuales conozcan los objetivos y el manejo de la empresa, para que puedan definir las necesidades y alcances, de los cuales podrá dar respuesta el Data Warehouse.

Otro de los roles será el que se encargue del diseño del Data Warehouse y de la base de datos.

El siguiente rol será el administrador que poseerá el conocimiento técnico de las herramientas OLAP, este se puede encargar de vigilar el buen funcionamiento del mismo, así como de la integridad de la información.

Existen otros roles que pueden variar según la herramienta que adquiera la organización, por ejemplo puede existir un equipo que se dedique a crear reportes solicitados (*ad hoc*) o al diseño del sitio Web, así como de la programación de eventos dentro del mismo e inclusive un equipo de depuración de información (según tamaño de la empresa) y hasta un equipo de seguridad para evitar intrusiones o fuga de información; para que finalmente llegue un resumen con información, datos, tendencias y gráficas que permitan al ejecutivo, dueño o socios de la empresa tomar una decisión acertada, libre de errores con tan solo una hojeadada al documento.

Hay algunos programas, que permiten interactuar con la información de una manera amigable, sin necesidad de recurrir al equipo técnico e inclusive crear sus propios reportes o exportar la información requerida a herramientas de uso cotidiano como hojas de cálculo; con tan solo contar con los cubos ya hechos.

Existe una herramienta en Excel que permite extraer información de los cubos, pero está demasiado restringida, además de no ser muy fácil de ver y usar la información, por lo que se utilizará el software de nombre Decisión perteneciente a la Compañía Comshare, sólo para visualizar y manipular la información del cubo (El objeto de este trabajo no es mostrar las ventajas o desventajas de este software en comparación con otros; así que solo se utilizara de una manera muy rudimentaria, como se podría ver con cualquier otro software par visualizar cubos).

En la figura 4.13 se observa el cubo con todas sus dimensiones y con el formato adecuado para poder ser manejado por un ejecutivo, con todas la restricciones que el editor de cubos presenta para la visualización de la información, además de ser de manejo delicado, dado que se pueden modificarse las propiedades, las



relaciones y hasta perder el cubo por completo, por esto no es adecuado para el uso de un ejecutivo que no tiene los conocimientos o experiencia para tratar el cubo a este nivel.

### Editor de Cubos “Pestaña DATA” del Analysis Manager con el Formato “ad hoc”.

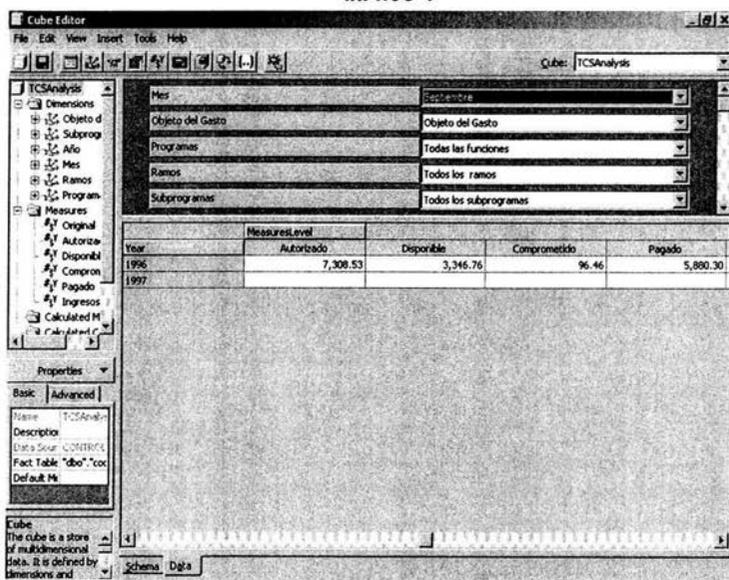


Figura 4.13

En la figura 4.14 se interpretada la información de la figura 4.13 mediante un gráfico de barras.

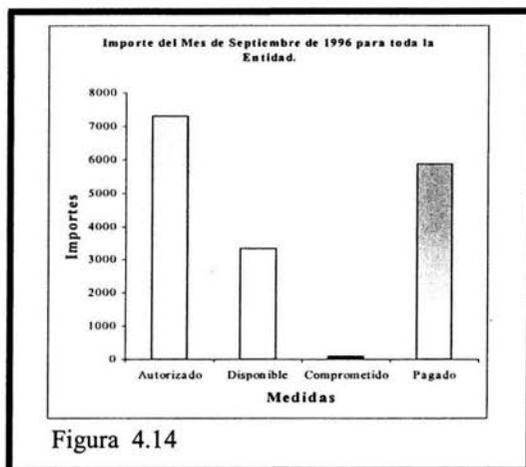


Figura 4.14

En la figura 4.15 se podrán observar las mismas dimensiones y medidas que en la figura 4.13, que como se apreció en su momento tiene varias restricciones, esto es solo para poder tener una mejor apreciación de la información aprovechando ventajas como por ejemplo: al combinarse varias dimensiones se podrán ver todas las medidas correspondientes y no se correrá el peligro de modificar o perder la estructura con la que fue creado el cubo, para ello utilizaré el Software Decisión de Comshare, sin utilizar muchas de las características que lo hacen ser un producto líder en el mercado, para este tipo de tecnología, de tal manera que solo me enfocaré a la presentación de los datos de la misma manera como se haría en la pestaña "DATA" del Analysis Manager de Microsoft SQL Server.

En las subsecuentes figuras se podrán ver algunos ejemplos del tipo de información que podría requerir algún ejecutivo que puede ir de lo general a lo particular o viceversa o tal vez requerir información que se puede derivar de la misma.

Para el primer ejemplo mostraré la figura con la información en el Decisión y luego el Query que se tendría que hacer en una base de datos relacional además de el tiempo de respuesta, sin tomar en consideración que todavía se tendría que hacer una aplicación Cliente-Servidor, con una infinidad de consultas definidas previamente, para mostrar la información, que con herramientas de tipo OLAP no llevaría tanto tiempo, además de poder hacer consultas que en ese instante sean requeridas por la parte ejecutiva, sin ayuda alguna de la parte técnica.



Por otro lado omitiré cada uno de los procedimientos que se tendrían que hacer en una base de datos relacional para obtener exactamente el mismo resultado que con el Decisión, pues en todos los casos es un trabajo difícil y laborioso que requiere de bastante tiempo utilizando lenguaje SQL (ver Anexo de Lenguaje SQL) y algún otro lenguaje de programación como Visual Basic, Delphi o Visual C por nombrar algunos, para la parte del Cliente.

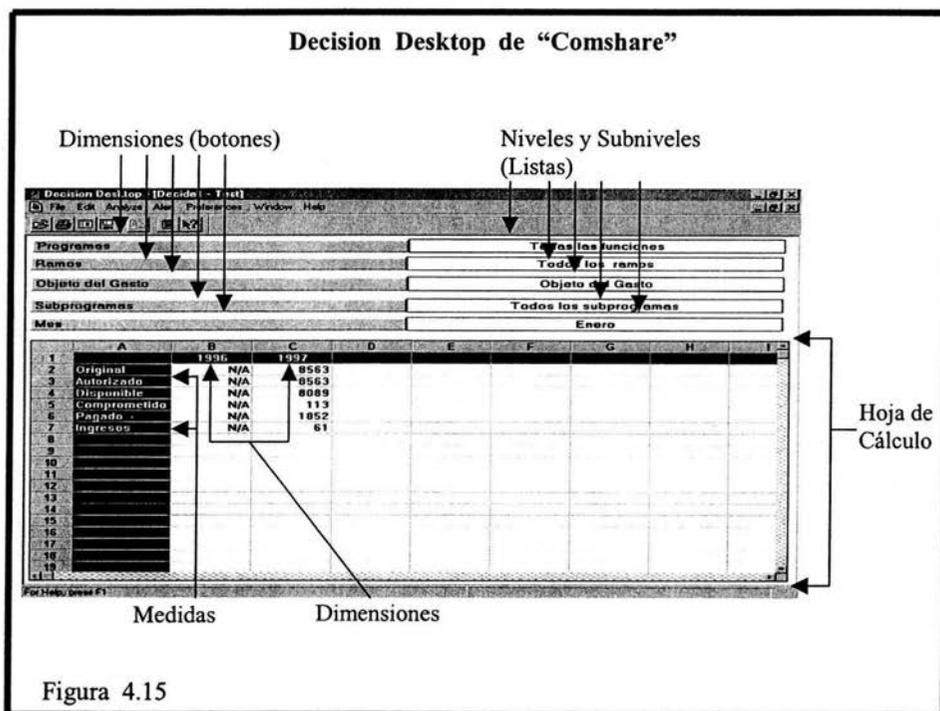


Figura 4.15

En la figura 4.15 podemos ver las medidas de lo más general(no particular); es decir sólo queremos saber cuanto tenemos de importe en las diferentes medidas, para los años 1996 y 1997 en el mes de Enero, sin importar ninguna Función, Ramo, Objeto de Gasto o Subprograma en particular.

El siguiente Query dará respuesta a esta pregunta tomando como base el diagrama entidad-relación de este mismo capítulo figura 4.2 (Las instrucciones del Lenguaje SQL están en cursivas y mayúsculas).



```
SELECT 1996 AS Año, SUM(Original) AS Original, SUM(Autorizado) AS Autorizado,  
SUM(Disponible) AS Disponible, SUM(Comprometido) AS Comprometido, SUM(Pagado)  
AS Pagado, SUM(Ingresos) AS Ingresos FROM codigos_1996 WHERE mes=1  
UNION
```

```
SELECT 1997 AS Año, SUM(Original) AS Original, SUM(Autorizado) AS Autorizado,  
SUM(Disponible) AS Disponible, SUM(Comprometido) AS Comprometido, SUM(Pagado)  
AS Pagado, SUM(Ingresos) AS Ingresos FROM codigos_1997 WHERE mes=1
```

Tiempo : 5 segundos

El Query se ejecutó en el SQL Query Analyzer en una máquina Pentium III a 833 Mhz.; misma donde se ejecutan los cubos.

En la figura 4.16 se muestra la misma información sólo cambiando de posición la dimensión de año y las medidas de horizontal a vertical con tan sólo arrastrar las dimensiones y/o las medidas visualizadas en la hoja de cálculo a la posición donde se desean tener en la misma hoja de cálculo (columna, renglón).

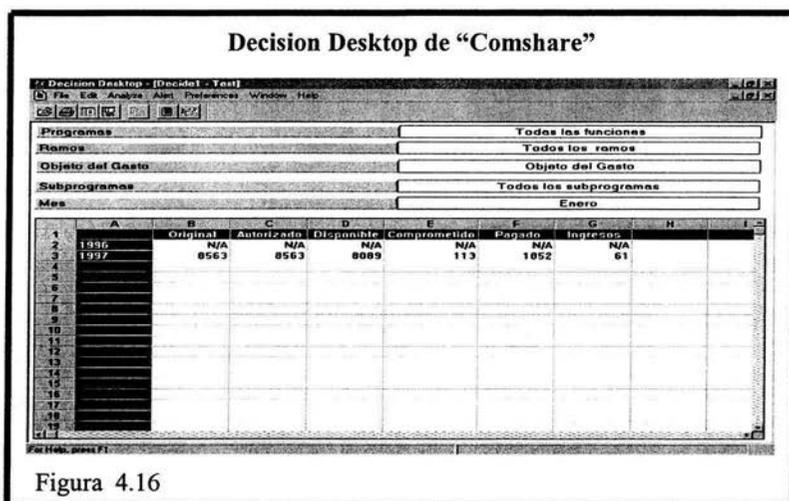


Figura 4.16

En la figura 4.17 se muestra la información correspondiente al mes de Septiembre de los años 1996 y 1997, con tan solo haber dado un clic sobre el botón de Mes y haber seleccionado el mes de septiembre en la lista que aparece del lado derecho del botón, quedando así el mes de Septiembre.

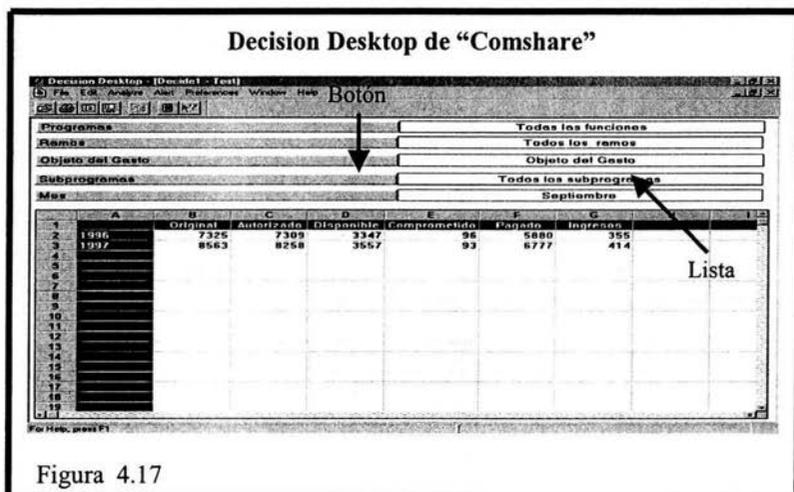
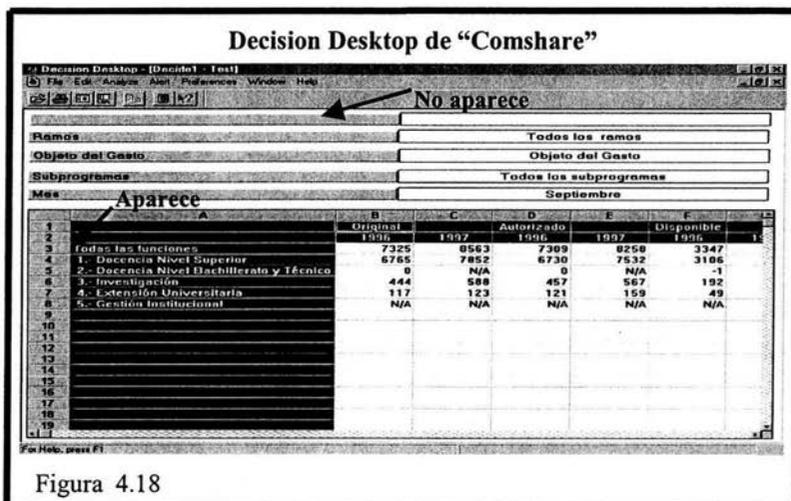


Figura 4.17

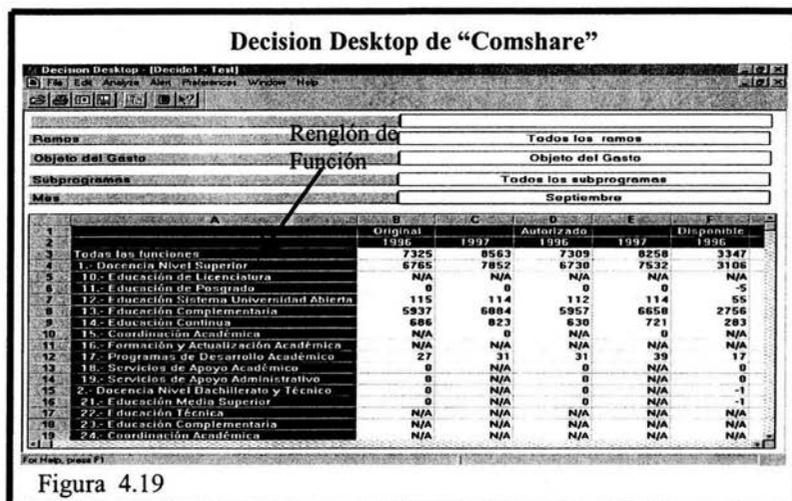
En la figura 4.18 podemos ver que la información en la hoja de cálculo, ya no es de una dimensión (Año), ahora en la hoja de cálculo se pueden apreciar dos dimensiones (Año y Funciones), la dimensión de funciones ya no aparece fuera de la hoja de cálculo, pero es importante recalcar que las seis dimensiones están interactuando al mismo tiempo, más no se ven visualizadas en la hoja de cálculo. Para tener la dimensión de Funciones dentro de la hoja de cálculo sólo es necesario arrastrarla y ponerla en el renglón y columna que se requiera.

Por ejemplo la información que nos presenta la hoja de cálculo es la siguiente: Importe con el que se cuenta en cada una de las medidas, por función en los años 1996 y 1997 en el mes de septiembre (sin importar el ramo, el objeto del gasto o algún subprograma en particular).

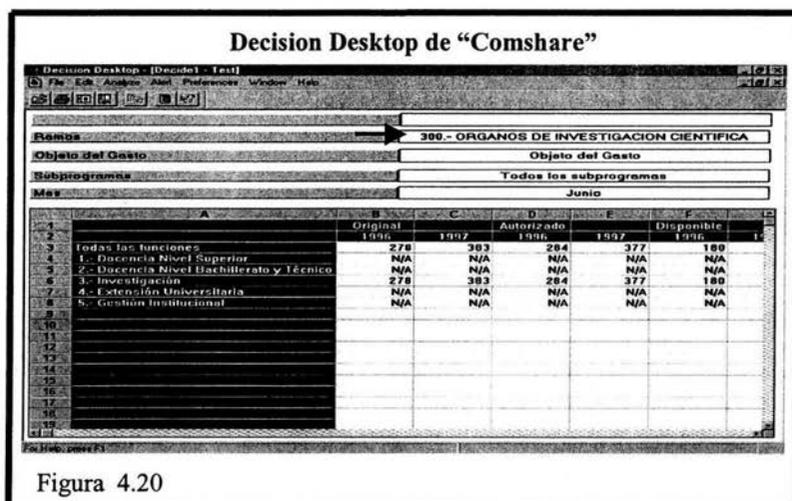


En la figura 4.19 veremos más desglosada la información anterior es decir cómo está distribuida en cada una de las funciones (recordaremos que se llaman "programas"), ahora queremos saber el importe en cada una de las medidas por programa para los años 1996 y 1997 en el mes de septiembre.

Como podemos observar la función se expande a un nivel más bajo con tan solo expandir la rama de la función correspondiente (dar doble clic sobre el renglón de la(s) función(es) que se quieran ver a más detalle es decir, a nivel programa), por otro lado es importante mencionar que en una base de datos relacional para obtener un mayor nivel de detalle se torna cada vez más compleja la consulta (Query), además de ser más lenta para dar la misma información y con estas herramientas es casi el mismo tiempo de respuesta para una consulta sencilla que una muy compleja.



En la figura 4.20 observaremos una consulta más específica. Queremos saber con cuanto se cuenta en cada una de las medidas en el mes de Junio para los años 1996 y 1997 en cada una de las funciones; pero únicamente en el ramo 300.- Órganos de Investigación Científica, para todo objeto de gasto en todos los subprogramas.





Como podemos ver en la figura 4.21 la información que se encontró nos permitió saber cuanto hay en cada una de las medidas en el mes de Junio, para los años 1996 y 1997 en cada función, pero únicamente en el ramo "300.- Órganos de Investigación Científica", teniendo como objeto de gasto "Remuneraciones Personales" sin hacer referencia a algún subprograma en especial (todos los subprogramas).

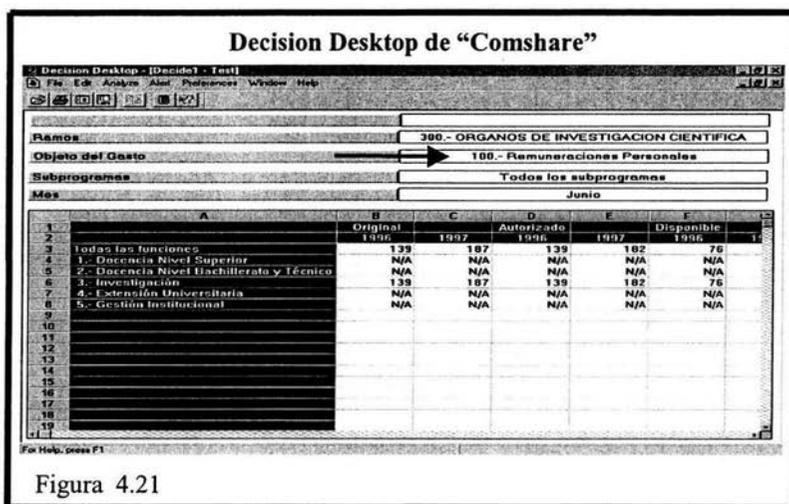


Figura 4.21

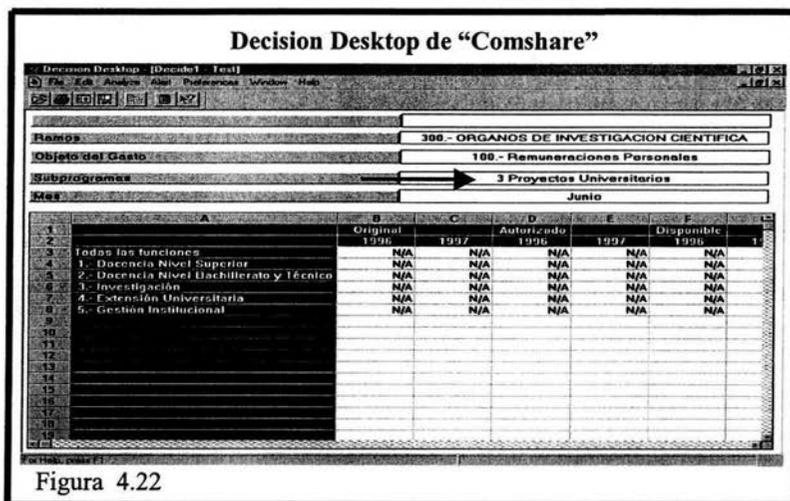


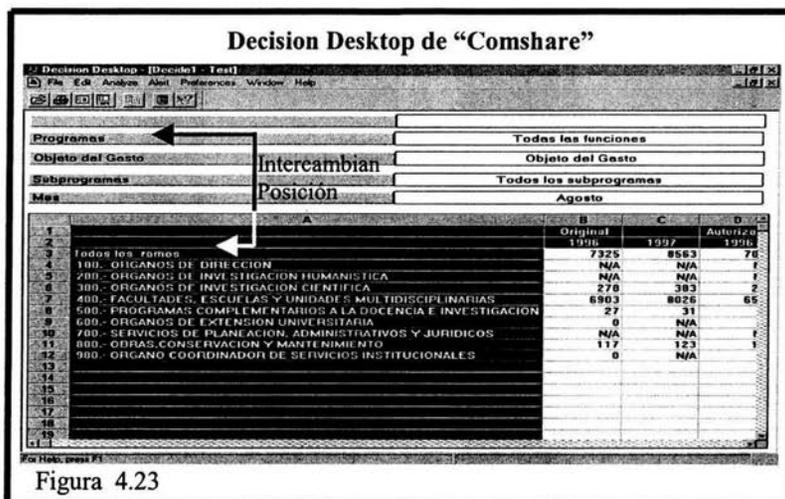
Figura 4.22



En la figura 4.22 podemos notar que la información ahora es más específica, pues ahora solicita la información para un subprograma en especial, cabe recordar que los subprogramas pueden tener la misma clave pero dependiendo del ramo y más aún de la dependencia y su subdependencia será el nombre del subprograma.

Ejemplo puede ser que la Facultad de Química tenga un subprograma de apoyo a laboratorios con clave 01.03 y en la Facultad de Derecho tienen el mismo número de clave (01.03) en X subprograma, pero como sería lógico ninguno hace referencia a laboratorios.

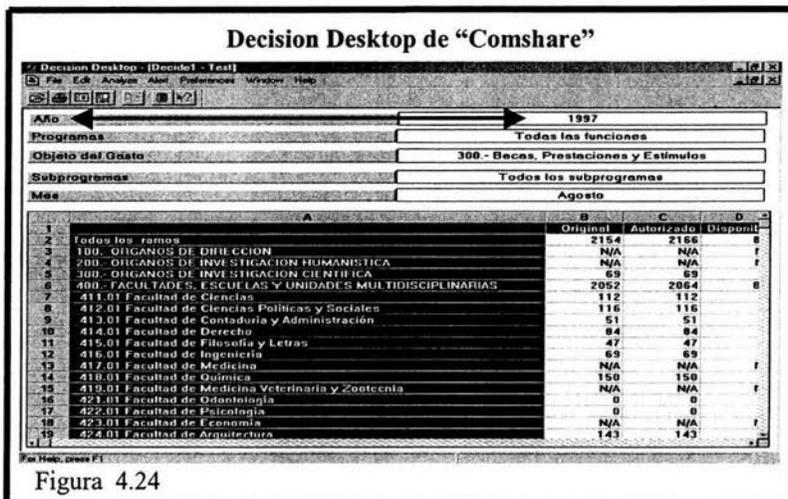
En la figura 4.23 se cambia la perspectiva de información, dado que en la figura 4.22 la perspectiva principal fue ver la información por función; ahora en la figura 4.23 el objetivo principal es ver la información por ramos, para ello todas las herramientas pueden hacer intercambio de dimensiones con tan solo arrastrar y soltar la dimensión de interés sobre la que ya no queremos que aparezca y automáticamente se intercambian de lugar, más adelante veremos como se pueden intercalar otras dimensiones sin hacer un intercambio, es decir como varias dimensiones pueden estar al mismo tiempo en la hoja de cálculo.



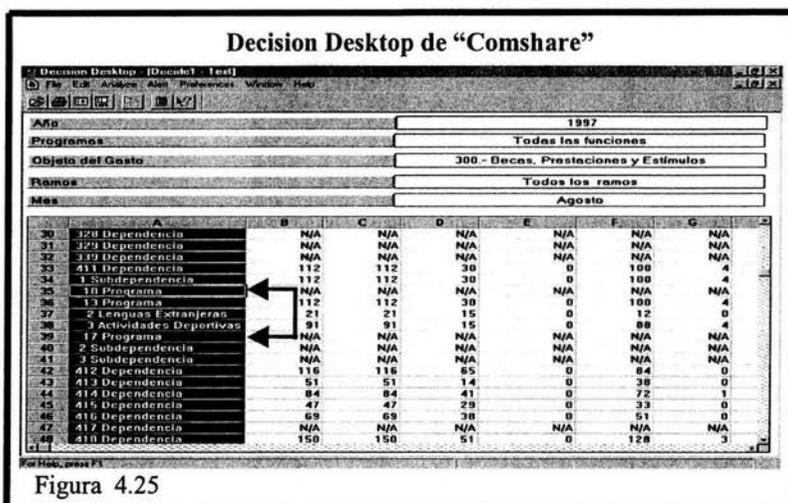
En la siguiente figura (4.24) y siguiendo con la misma perspectiva (Ramos), pero ahora expandiendo el ramo "400.- Facultades, Escuelas y Unidades Multidisciplinarias" y solo consultando el mes de Agosto del año 1997 para el objeto de gasto "300.- Becas, Prestaciones y Estímulos", en cualquier función y



ningún subprograma en particular. Es de hacer notar que ahora la dimensión año ya no está en la hoja de cálculo, más sin embargo se está seleccionando el año 1997 únicamente, para ello lo único que se hizo fue arrastrar la dimensión año de la hoja de cálculo y depositarla en el botón que está en blanco arriba de la hoja y después seleccionar de la lista el año a considerar (figura 4.23).



En la figura subsiguiente (4.25) cambiaremos una vez más de perspectiva, ahora la perspectiva principal será ver la información por subprogramas, en esta figura se señalarán los subprogramas de la Dependencia "411.-Facultad de Ciencias", Subdependencia "01.-Posgrado", para el mes de Agosto de 1997, para todas las funciones y ramos, en el objeto de gasto "300.- Becas, Prestaciones y Estímulos".



Otros ejemplos en los cuales interactuarán 2 o más dimensiones; así como sus respectivas medidas en la hoja de cálculo se muestran en las figuras siguientes. Para comenzar podemos ver cuanto tienen las medidas en cada una de las funciones por cada uno de los ramos que las comprenden, para el mes de Agosto de 1997, sin agregar, más bien seleccionar y/o dar más clics para tener más niveles de detalle. Ahora explicaré más a fondo la información que nos presenta la hoja de cálculo.

Antes de proceder a explicar y debido a que en una imagen carecemos de movimiento solo se nota la medida de Original, pero sucede lo mismo con las demás medidas; esto es con respecto a lo vertical y con respecto a lo horizontal se alcanza a apreciar "Todas las funciones" y la función "1.- Docencia Nivel Superior", pero es análogo para las funciones restantes.

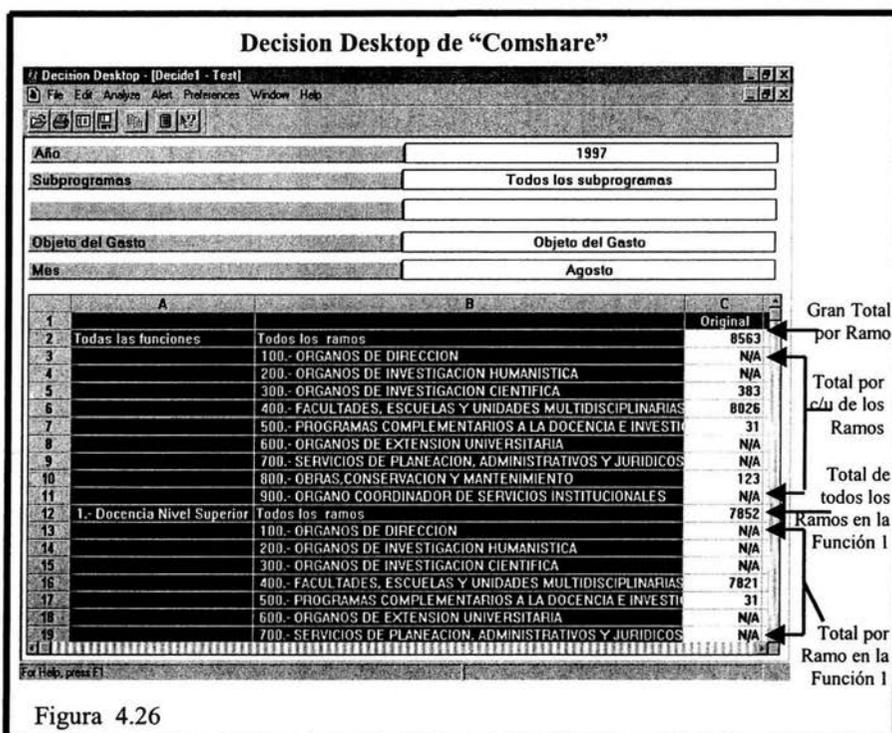
El Renglón 2 perteneciente a "Todas las funciones; Todos los ramos" en la columna de original tenemos la suma de todos los ramos sin importar función; es decir tenemos el gran total por ramo, cabe recordar que cada ramo esta constituido por dependencias y subdependencias que se pueden mostrar abriendo o extendiendo cada uno de los ramos (dando doble clic sobre el renglón de interes), pero que no se hizo en esta imagen pues no se podría apreciar ninguna medida, por que se agregarían más columnas a la hoja de cálculo.

En el renglón 3 en la columna de todos los ramos, tenemos el total de el ramo 100.- Órganos de Dirección y así sucesivamente para cada uno de los ramos hasta el renglón 11.



Para el renglón 12 " 1.- Docencia Nivel Superior; Todos los ramos " se presenta la suma de lo que tienen todos los ramos en la función 1

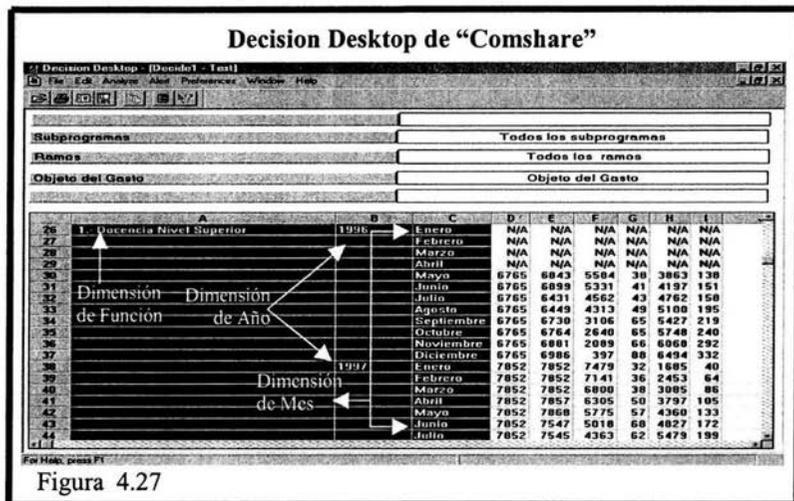
En el renglón 13 en la columna "100.- Órganos de Dirección" se presenta la suma de todas las dependencias y subdependencias que conforman el ramo 100 en la función 1, es de mencionar que debido al objetivo de la funciones y de los ramos y a su vez de las dependencias y subdependencias, puede ser que algunos ramos no tengan que ver nada con la función en cuestión.



Como cada vez es más difícil comprender la información cuando interactúan más de 2 dimensiones, además las medidas se van más a la derecha de la hoja de cálculo y muchas veces ya no se pueden apreciar junto con sus respectivos nombres y se tiene que desplazar a la derecha, por ello cuando tenemos más de 2 dimensiones nos convendrá combinar las dimensiones en la hoja de cálculo de forma vertical para ello sólo se arrastra la dimensión que está en forma horizontal y se lleva a la forma vertical posicionándola en donde se requiera sobre la misma hoja de cálculo, existe otro caso para combinar las dimensiones



pero es más sencillo pues se tienen que arrastrar de los botones a la posición que se requiera en la hoja de cálculo figura 4.27

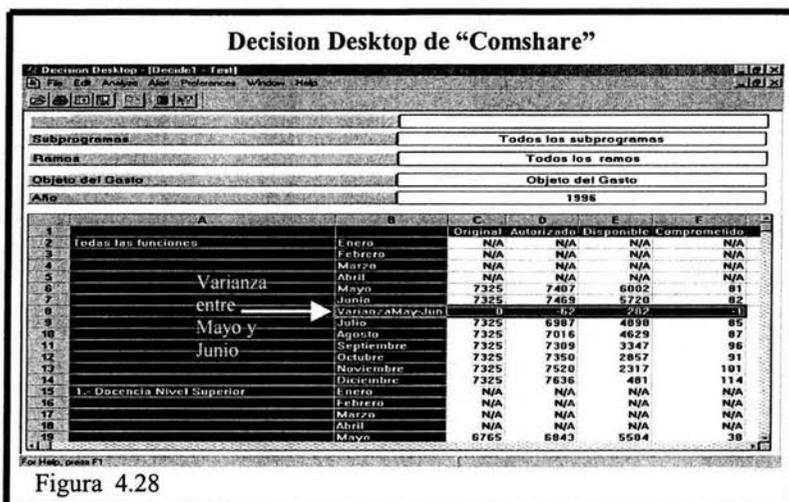


La forma que podríamos combinar y tal vez tener un mejor entendimiento de la información que se muestra en la figura 4.27 sería como se expresa en la siguiente tabla, se pueden ver las dimensiones en negritas y las medidas en cursivas y subrayadas, además de estar separadas por una línea negra horizontal. Por otro lado podemos notar que en vez de crecer de forma horizontal está crece de forma vertical y que para poder ver más a detalle la información solo bastaría dar un doble clic sobre el renglón de interes.

	<b>Enero</b>		<b>Febrero</b>		...	<b>Diciembre</b>		<b>Enero</b>		...
	<i>1996</i>	<i>1997</i>	<i>1996</i>	<i>1997</i>	...	<i>1996</i>	<i>1997</i>	<i>1996</i>	<i>1997</i>	...
	<i>Original</i>				...			<i>Asignado</i>		...
<b>Todas las Funciones +</b>	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	...	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	...
<b>Docencia Nivel Superior +</b>	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	...	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	...
<b>Docencia Nivel Bachillerato+</b>	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	...	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	<i>\$\$\$</i>	...
⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮	⋮

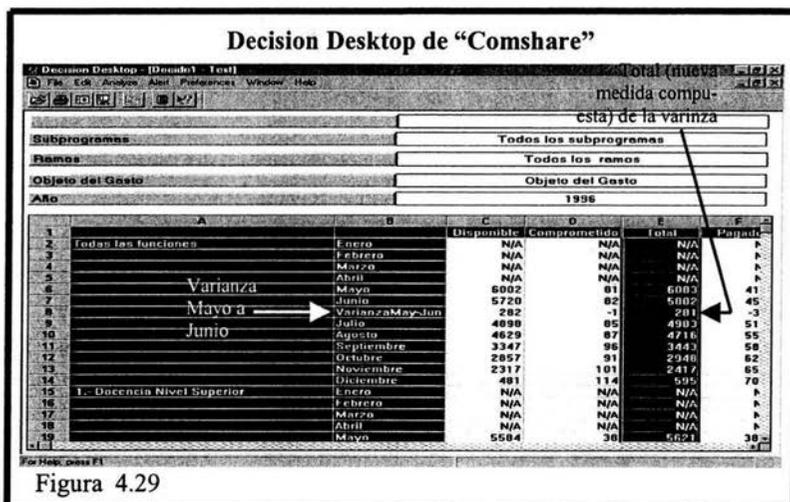


Una parte muy importante acerca del manejo de cubos y dimensiones es que a partir de sus medidas se pueden derivar otras medidas que llamaremos **medidas compuestas**, las medidas compuestas por lo regular son parámetros de relevancia para el personal cuya función es tomar alguna decisión. En la Figura 4.28 veremos un ejemplo de una medida compuesta. En este caso la medida compuesta es una simple varianza entre los meses de Mayo y Junio del año 1996 por función (se puede hacer a cualquier nivel de detalle).



Una característica muy importante y de gran utilidad es que las medidas compuestas no solamente pueden interactuar de manera vertical; es decir que podemos tener también medidas compuestas de forma horizontal y si hay alguna medida compuesta vertical y creamos una medida horizontal, esta también interactuará sobre esta medida vertical. En la siguiente figura 4.29 veremos un ejemplo que nos ayudará a clarificar este concepto.

La medida vertical es la varianza de las medidas simples, de los meses de Mayo y Junio de 1996 por concepto de función, pero además de ello nos interesaría saber la suma entre Disponible y Comprometido para cada mes e inclusive la suma de la varianza de los meses de Mayo y Junio por función, es obvio que podríamos añadir más restricciones.



Como se pudo notar este tipo de tecnología sirve para tener un mejor control de la Empresa, en este caso se enfocó a el presupuesto de una entidad educativa pero puede extenderse tanto como se quiera tener control.

Por ejemplo este caso puede extender su control si añadimos datos como plantilla docente o añadimos matricula de alumnos o materias o investigaciones o infraestructura o espacio u horas de trabajo u horas de estudio o le que se nos pudiera ocurrir que pueda tener relación con recursos económicos.

Un ejemplo más conciso sería:

En esta entidad educativa se sabe que se cuenta con \$300 en la partida "117.- Sueldo a profesores de asignatura ordinarios no definitivos" para el año 1997.

Estando en el mes de Julio de 1997 se sabe que hasta el momento se ha gastado en esa partida \$220 y todavía faltan 5 meses por cubrir. La pregunta es que hacer para poder cubrir esa obligación para el resto del año, pues ya se ha gastado el 73% del importe y todavía falta el 41% del tiempo.

Primero que nada debemos contar con información veraz para poder ofrecer una respuesta óptima.

Tal vez la primer pregunta que pasa por la cabeza, es por que se presupuestó \$300 para esa partida. y la respuesta sería muy fácil de contestar con este tipo de tecnología pues se puede ver automáticamente que se tiene un registro del año 1996, en el cual no se gastó lo presupuestado, entonces para este año se hizo una estimación errónea lo cual puede suceder, pues pudieron haber varias



causas por ejemplo el incremento la plantilla de profesores no definitivos, por el incremento de la matrícula o el alto índice de reprobación etc.

Entonces la política a seguir dependiendo de la causa puede ser que se despida el exceso de profesores, pero esto traería una repercusión, pues probablemente algunos grupos se quedarían sin profesor. Eso nos lleva a cuestionarnos. ¿A cuantos profesores asciende la plantilla? y ¿Dónde están ubicados?, ¿Se puede prescindir de algunos?, o ¿Por qué se tienen más alumnos? ¿Que es lo que causa el alto índice de reprobación, que hace que las generaciones se rezaguen y las nuevas generaciones los alcancen? ¿Se tendrá que reducir el tiempo para finalizar los estudios? ¿Qué tipos de restricciones se deben poner, a los alumnos para que no se exija un mayor número de profesores? o ¿Por qué no mejor buscar las partidas a las cuales se les ha presupuestado más de lo que en realidad gastarán y repartirlo en la partida en cuestión?.

Como podemos notar para poder tomar una decisión, que tenga la menor repercusión o que tal vez sea la óptima, es necesario tener todo el control de la entidad educativa, al mismo tiempo es decir jugar con las dimensiones de un cubo. Por que primero necesitamos haber puesto una alerta donde se excedió cierto porcentaje para saber cuanto ha sido gastado hasta el mes de Julio de 1997. Después hacer una comparación con el año anterior, después buscar las causas probablemente con otra dimensión que podría ser plantilla de profesores o matrícula de alumnos. Más tarde saber en que dependencias o subdependencias están ubicados es decir manipular la dimensión de Ramos, analizar la información devuelta y si no es suficiente buscar de nuevo manipulando alguna otra dimensión, en este caso buscar en la dimensión de Objeto de Gasto para saber que partidas no han sido ejercidas en X porcentaje para poder utilizar su asignación mediante un traspaso según la dependencia o subdependencia, pues habrá algunas de ellas que no sea suficiente el traspaso y para otras será suficiente, por que en algunas dependencias se pudo exceder mucho más que otras en la contratación de profesores no definitivos.

En fin la respuesta variará según la cantidad y la calidad de información que se tenga.

Por que podemos imaginar ¿Cuánto tiempo llevaría tomar una decisión como esta, sin tener un sistema bajo esta tecnología? y ¿Cuánta gente y de que áreas se verían involucradas?. Yo me imagino que se tendría que hacer una serie de llamadas a las distintas áreas, para poder tener algunos resultados y después de llenarnos de hojas y hojas con muchos datos, poder conjuntar algunos resultados y tener cierta información que nos sirva para buscar otras opciones, entonces eso implicaría volver a pedir otros datos, a las mismas áreas o a otras, y volver a sintetizarlos y más tarde tener información más acotada, y después



de un arduo trabajo por fin llegar a la misma solución y ejecutar la decisión después de un tiempo bastante considerable.

A manera de comprobación de este trabajo y como se puede ver claramente en el capítulo 3 y 4; están las bases para aplicar esta tecnología a distintas empresas u organizaciones, como ejemplo de ello se puede aplicar a una compañía de productos lácteos, donde se tendrán algunos productos como Quesos, Cremas, Leches, Yogurts, Mantequillas, etc. figura 4.30, pero a mi me fue más interesante presentarlo sobre una entidad educativa y con relación a un ingreso, pues es básico saber con cuánto se cuenta, para saber que se puede realizar con ello, siendo así este; un documento que puede auxiliar a nuestra noble y gran Universidad Nacional Autónoma de México; no tan sólo en el control del presupuesto, sino en distintas áreas e inclusive en distintas investigaciones, donde se requiera tener información perfectamente clasificada, jerarquizada y relacionada con algunas otras ramas de interés.



<u>Productos Lacteos.</u>	
- Productos	
- Quesos	
- Blanco	
	100 grs.
	250 grs.
	1,000 grs.
	10,000 grs.
- Oaxaca	
	100 grs.
	250 grs.
	1,000 grs.
	10,000 grs.
:	
:	
- Yogurts	
- Licuado	
Fresa	
	250 ml.
	500 ml.
	1,000 ml.
Piña	
	250 ml.
	500 ml.
	1,000 ml.
:	
:	
- Lechés	
Light	
	250 ml.
	500 ml.
	1,000 ml.
Semidescremada	
	250 ml.
	500 ml.
	1,000 ml.
:	
:	

Figura 4.30

Para continuar con nuestra comprobación y después de haber hecho una clasificación, sería fácil pensar en unidades de venta-compra, ventas, regiones, tiempo, vendedores, clientes, acreedores, etc. que son importantes para alguien que quiere tener el control total de su negocio, un caso particular de este modelo puede aplicarse, para saber cuantas unidades de yogurt, sabor fresa de 500 ml., se están vendiendo en la zona sur de la república mexicana, en el



cuadrante noroeste de la zona sur, con el cliente Walmart, en la sucursal 2, en la segunda semana de Febrero de 1999; además de saber quien es el distribuidor e inclusive si hubo una oferta especial, para esta fecha por parte del vendedor final, la cual podría hacer que se tuviera una demanda mayor para este producto. La siguiente pregunta que se podría uno hacer es la siguiente ¿En que beneficiará esta información al manejador del negocio?. Las respuestas pueden ser varias según el enfoque que se le este dando; una podría ser si se distribuyen más o menos unidades a esa zona, otra es saber si es por culpa del repartidor o de las promociones que se están manejando o inclusive si es por la temporada de la fruta que se solicitan más o menos unidades, como se puede ver también será necesaria más información como proveedores, zonas, clientes etc. y toda esta información debe clasificarse según la jerarquía que se tenga y entre más niveles o subniveles se tenga de la información con más detalle se devolverá la misma.



## Conclusiones.

Como se puede observar a través de los capítulos anteriores el concepto general de Data Warehouse en nuestros días es muy importante, por la gran necesidad de contar con información organizada, veraz, ágil y resumida; que más que encaminarnos a tomar algunas decisiones, nos lleven a decidir correctamente con un mínimo de error.

Debido a los distintos enfoques de la carrera de Actuaría; el egresado de la misma, bien puede desenvolverse en los distintos procesos que son requeridos para la conformación y desarrollo de un proyecto de Data Warehouse, en el cual, será parte fundamental para el crecimiento o decaimiento de la empresa u organización. Algunos de las funciones que puede desempeñar son:

- Ejecutivo o funcionario encargado de tomar la decisión final. (Teoría de decisiones, Planeación, Investigación de operaciones, Planeación Estratégica, Planeación Financiera, etc.)
- Analista de la información presentada (Datos estadísticos, económicos, demográficos, numéricos, porcentuales, financieros, probabilísticos, etc.)
- Técnico en el desarrollo de este tipo de sistemas informáticos. (Ingeniería del Software, Bases de datos, Programación, Desarrollo de sitios para Internet, Soporte, Administración, etc.)

Cabe mencionar que algunas de estas funciones no forman parte directa de las raíces de construcción de un sistema informático, pero que sin duda son fundamentales para el cumplimiento del objetivo, para el cual ha sido creado, por ejemplo; puede ser que el sistema esté bien diseñado y que de excelente respuesta a las demandas de información por parte del organismo, y que por otro lado el analista o el ejecutivo lleguen a cometer errores de percepción, llevando al fracaso todos los esfuerzos de las distintas áreas que lo han conformado hasta entonces.

Para lograr el éxito esperado por la parte informática, es necesario contar con herramientas de desarrollo y de front-end, que estén respaldadas por un equipo de soporte técnico, además de contar con bases de datos centralizadas que garanticen la integridad de los datos. Por el lado de la interfaz final debe de cumplir con buenos reportadores y graficadores, funciones de alertas en el caso de que algunos parámetros se salgan o entren de los criterios especificados en las políticas de la organización. Otras son propiedades para datos excepcionales o poco comunes, en algunos de ellos deberán existir interfaces de simulación e impacto en los cambios y todos tendrán que cumplir con opciones de MDX



(Multidimensional Expressions), que son útiles para realizar operaciones o evaluar funciones entre medidas y/o dimensiones. Por último deben cumplir un periodo de evaluación y soporte gratuito para determinado tiempo.

Básicamente esta tesis sirve como una introducción al concepto de Data Warehouse, para personas con conocimiento en bases de datos y que tenga el deseo de monitorear los movimientos relacionados a su negocio, mediante una de las más recientes tecnologías en cuanto al manejo de datos, cuyos cimientos están soportados principalmente por manejadores de bases de datos relacionales, que hasta la fecha han demostrado ser una buena herramienta informática para el manejo de datos; y que aunado a las herramientas de tipo OLAP, brindan una optimización para el análisis de los mismos, que más tarde se convertirán en información más entendible o manejable para el personal que se encarga de analizar y/o tomar decisiones que repercutan en los funcionamientos o estrategias del negocio según el enfoque que se le de; sin solicitar esta información a el equipo técnico o a los departamentos que intervienen en dicha información.

Se trató de ejemplificar el concepto de Data Warehouse a través de un modelo presupuestal, pero es bien claro que la utilización de este concepto se puede extender a un sin numero de proyectos, donde la única característica a cumplir es, tener datos que se puedan clasificar en niveles y subniveles y así sucesivamente.



## Anexo de Microsoft Analysis Server .

Analysis Server es un componente de Microsoft SQL Server 7 y 2000, está específicamente diseñado para crear y mantener estructuras multidimensionales de datos además de dar respuesta a consultas (queries) hechas por un cliente. La herramienta principal para administrar y manejar objetos del Analysis Server es conocido como Analysis Manager.

Cada Analysis Server tiene un repositorio llamado Analysis Services Repository, estos repositorios almacenan los metadatos de los objetos que están definidos en el Analysis Server.

El Analysis Services tiene por repositorio de default el archivo Msmdrep.mdb(Access), que de igual manera puede ser migrado a SQL Server Database . Cada Analysis Server tiene asociado un subdirectorio de datos, el cual almacena la estructura multidimensional para cada objeto definido en el Analysis Server, estas estructuras están contenidas en archivos que son creados cuando los objetos son procesados. Estas estructuras son consultadas para dar respuesta a algunas consultas pedidas por el Analysis Server, pues algunas de estas estructuras contienen agregaciones. Obviamente estos archivos tienen mecanismos de seguridad para evitar el acceso no autorizado de alguna persona.

El Servicio asociado al Analysis Server es MSSQLServerOLAPService, que por default arranca el servicio automáticamente cuando el usuario entra a Windows 2000 o NT 4.

### Registro del Servidor

En la computadora en la cual se ha instalado SQL Server 2000 , el Analysis Services es registrado automáticamente como un servidor, y aparecerá inmediatamente en la estructura de árbol del Analysis Manager. Si se quiere registrar otro servidor, se tendrá que ir a la estructura de árbol del Analysis Manager y ubicarse en el fólder de Analysis Servers, oprimir el botón derecho del mouse y seleccionar la opción de registrar servidor; aparecerá inmediatamente después una ventana solicitando el nombre del servidor a registrar, el Analysis Manager automáticamente empezará a buscar el servidor con el nombre especificado y lo registrará siempre y cuando se tengan los permisos para ello. Figura 4.5 en la sección donde aparece Analysis Servers.



## Seguridad de la base de datos

La seguridad de la Base de Datos es controlada a través de roles, al rol le es asignada cierta función o funciones las cuales podrán ser ejecutadas por los usuarios que hayan sido dados de alta en el rol estos usuarios podrán tener acceso a ciertas dimensiones o cubos (Figura 4.6 en la parte de Database Roles).

## Limites del Analysis Server

Como en todo software existen límites, los cuales deben ser considerados antes de crear base de datos, cubos y dimensiones en la siguiente tabla se especifican las limitaciones que tiene el Analysis Server

Bases de Datos por Servidor	Sin limite
Cubos por Bases de Datos	Sin limite
Cubo Virtual por medio de cubos	64
Dimensiones por cubo	128
Medidas por cubo	1,024
Miembros calculados por Cubos	65,535
Niveles por dimensiones	64
Particiones por cubo	Sin limite
Niveles por cubo	256
Tamaño de la base de datos por cubo	64 kylobytes

## Dimensiones y Cubos Virtuales

Una dimensión Virtual es una dimensión lógica que se basa en algún nivel de dimensión. Sus principales características son:

- Son mostradas a los usuarios como una dimensional normal.
- Pueden contener múltiples niveles
- No crean agregaciones
- No incrementan el tamaño de almacenamiento del cubo
- No incrementa el tiempo de procesamiento del cubo
- Incrementan el tiempo de respuesta cuando las consultas son referidas a estas dimensiones



- Se agregan a un cubo como una dimensión normal
- La fuente de la cual fue creada la dimensión, debe existir dentro del mismo cubo que utilizará la dimensión virtual; es decir el nivel de la dimensión de la cual fue creada la dimensión virtual debe estar dentro del mismo cubo.

Cubos Virtuales son procesos analíticos en línea (procesos OLAP), que combinan datos que han sido extraídos de uno o más cubos sin consumir espacio en disco.

Sus principales características son:

- Son usados para limitar el acceso de información para algunos usuarios.
- Muestra todas las dimensiones, de las cuales está conformado (No limita el acceso a ciertas dimensiones del mismo cubo).
- Pueden ser creados a través de cubos que hayan sido almacenados de manera MOLAP, ROLAP o HOLAP.
- Un cubo virtual puede ser creado de la combinación de hasta 64 cubos.
- Tiene las mismas propiedades de un cubo normal en cuanto a seguridad.
- No hay limite en cuanto a creación de cubos virtuales en una base de datos.
- Los cubos virtuales no pueden presentar sólo una parte de una dimensión, ello presenta la dimensión completa.
- No se puede escoger que niveles de la dimensión son presentados.
- Las medidas de las dimensiones, pueden ser seleccionadas según convenga.
- Los cubos virtuales toman las agregaciones de los cubos normales de los cuales fueron creados (fuentes).
- Se pueden hacer particiones de un cubo virtual, al igual que de los cubos normales.

## Particiones

Las particiones son medios físicos de almacenamiento de datos para los cubos. Un cubo puede tener una o más particiones y cada partición puede tener uno o más modos de almacenamiento (MOLAP, ROLAP, HOLAP) con diferente diseño de agregaciones. Además cada partición puede radicar en diferentes servidores. Todos los cubos tienen por default una partición.

La idea de crear particiones es para mejorar el procesamiento de los cubos y dar una respuesta más rápida a las consultas. Las particiones son procesadas una por una, pero como parte completa del proceso de un cubo; o pueden



procesarse cada una de manera independiente lo que reeditaré en una ganancia de tiempo a la hora de procesar la información. Las particiones son transparentes para el usuario y para la aplicación cliente o interfaz del usuario final. Prácticamente el cubo refleja la combinación de datos contenidos en las particiones.

Para crear particiones es necesario contar con SQL Server 2000 y con Windows NT 4.0 con service pack 5 o Windows 2000 Advanced Server o Windows 2000 Data Center Server (Figura 4.6 al expandir el fólдер de Cubes).

## Proceso de dimensiones y cubos

El proceso analítico en línea multidimensional (OLAP), incluye algunos esquemas y datos que por lo regular son modificados, por lo que es necesario actualizar la base de datos del manejador de base de datos. En OLAP la palabra procesar se entiende como una función, donde se crean Consultas(Querys), para extraer la información necesaria de el almacén de datos, para las dimensiones y cubos, también el Analysis Server crea algunas agregaciones en el cubo que previamente fueron diseñadas.

Es importante tener en cuenta la diferencia entre OLAP esquema y OLAP Datos.

OLAP Esquema consiste en jerarquías dimensionales y miembros que regularmente son jerarquizados (Figura 4.8).

OLAP Datos consiste en la información numérica como son las medidas y estas a su vez provienen de las tablas del manejador de base de datos (Figura 4.9 sección 2).

Los cubos son creados con una o más dimensiones, estas a su vez pueden combinar una o más medidas, por lo tanto cuando es procesado un cubo primero se procesan las dimensiones involucradas y una vez que las dimensiones han sido procesadas el Analysis Server carga datos de la Fact Table al cubo y este ahora es capaz de crear las agregaciones. Se debe procesar una dimensión cuando es creada por primera vez, cuando la estructura de la dimensión ha cambiado o cuando los datos de la tabla de la cual procede han sido modificados o actualizados. Por ello es necesario procesar la dimensiones cada vez que se haya efectuado algún cambio en los datos o los esquemas.

Existe dos métodos de procesamiento de dimensiones.



El primero es completo en el cual se reconstruye completamente el esquema y se actualizan los datos.

El segundo es parcial pues sólo actualiza los datos. De aquí podemos inferir que se realizará el proceso completo cuando:

- Se agregue o borre un nivel
- Un miembro sea renombrado
- Cuando un hijo sea promovido o rebajado a otro nivel.

Y se realizará el segundo (Parcial) cuando:

- Los datos de la tabla sean actualizados, borrados, agregados
- Algunas de las propiedades de los miembros hayan sido modificadas.

Al utilizar estos métodos traen consigo ciertas implicaciones.

Para el Completo.

- El Analysis Server borra y reconstruye la estructura de las dimensiones
- Los cubos que dependen de las dimensiones no pueden ser vistos por los clientes
- El tiempo de espera para la ejecución del procesado puede durar varios minutos dependiendo de su tamaño.

Para el parcial.

- Los cubos se mantienen disponibles para los usuarios
- Nuevos miembros aparecen y muchos quedan intactos.

Al igual que en el procesado de las dimensiones se tienen diferentes métodos de procesado en los cubos, los cuales dependerán de las modificaciones que se realicen tanto en la información como en los esquemas.

El proceso completo del cubo se realizará cuando:

- Se cree un nuevo cubo.
- Se agreguen o borren dimensiones.
- Se reconstruyan o compartan dimensiones entre cubos.
- Agreguen, borren o modifiquen medidas.



- Se modifique el tipo de almacenamiento (MOLAP, ROLAP, HOLAP).

Implicaciones cuando se realiza este proceso:

- El cubo es borrado y reconstruido.
- Las agregaciones son borradas y reconstruidas.
- Algunos cubos virtuales no estarán disponibles hasta después de que ellos mismos se autoprocesen.
- El tiempo del procesado es mucho mayor que en los demás métodos.

El proceso de Refrescar el cubo se efectuará cuando:

- La fuente de datos haya cambiado, pero la estructura sigue siendo la misma.
- Los datos hayan sido duplicados por algún error.

Implicaciones:

- Los datos del cubo son borrados y recargados más tarde.
- Todas las agregaciones son borradas y recalculadas.
- Las estructuras de las dimensiones compartidas se mantienen.
- El cubo original se continúa viendo y después se ven los nuevos datos al finalizar el proceso.
- Si un error ocurre el proceso se queda como antes.
- Los cubos virtuales siguen estando disponibles para los usuarios.

El proceso Parcial del cubo se llevará a cabo cuando:

- Nuevos datos son agregados a la Fact Table.
- La Estructura del cubo no ha cambiado.
- La fuente de datos no se ha modificado.

Implicaciones:

- Los nuevos datos son importados al modo de almacenamiento MOLAP.
- Todas las agregaciones son actualizadas.
- Los cubos se mantienen disponibles.
- Diferentes Fact Tables pueden ser usadas.
- Algunos filtros pueden definirse.



## Anexo de SQL.

### Lenguaje SQL.

SQL (Structured Query Language) es un lenguaje que permite expresar operaciones diversas sobre datos almacenados en bases de datos relacionales en los que la información se almacena en tablas con los datos dispuestos en filas y columnas. Una de las características más importantes de SQL frente a los lenguajes sobre bases de datos no relacionales estriba en que sus sentencias permiten manejar conjuntos de registros, en lugar de un solo registro cada vez. La mayoría de gestores tanto basados en una arquitectura Cliente-Servidor como los de sobremesa, así como los entornos de programación más utilizados hoy en día, tienen el lenguaje SQL como su medio para acceder a los datos.

El lenguaje SQL posee una estructura muy simple con un número muy reducido de sentencias que hacen que la sintaxis sea muy rápida de aprender. La versión original de SQL fue originada por IBM y se denominó SEQUEL, éste poseía muy pocas palabras reservadas y utilizaba una sintaxis sencilla, y como parte natural del desarrollo cada producto a incrementado el número de palabras reservadas, constituyendo una serie de dialectos del mismo lenguaje.

La estandarización posterior ha ido incorporando esos nuevos términos como parte del SQL normativo, sin embargo es habitual encontrar palabras reservadas en las versiones comerciales de SQL que no están incluidas en ANSI-SQL. Es de hacer notar que SQL Server de Microsoft utiliza el dialecto Transact-SQL posee la mayoría de palabras reservadas en el lenguaje estándar.

**QUERY:** Esta palabra inglesa de traducción consulta en español hace referencia a cualquier expresión en cualquiera de los dialectos SQL que definan una operación a realizar con los datos de nuestro sistema de gestión de bases de datos. Y esta constituida por los siguientes elementos:

- Una acción o verbo que determina la operación a realizar (SELECT, INSERT, etc.).
- Un objeto que usualmente será una combinación de campos de las tablas de la base de datos.
- Una cláusula que determina sobre qué objetos actúa el verbo (FROM nombre de la tabla o tablas).

Las peticiones de datos se expresan en SQL mediante sentencias. Estas sentencias podrían ser escritas por usuarios finales directamente en la pantalla en un terminal interactivo o incluidas en programas escritos con otros lenguajes



de programación. En este segundo caso, el programador deberá conocer, además de la sintaxis SQL, la propia del lenguaje de programación: C, Visual Basic, Delphi, COBOL, etc.

SQL pertenece a la cuarta generación de lenguajes y suelen llamarse no procedurales, por que se especifican los resultados que se quieren obtener por parte de la computadora, antes que los métodos para hacerlo (al gestor de bases de datos se le piden los resultados que se quieren obtener por parte del computador, antes que los métodos para hacerlo).

Una característica importante es que es un lenguaje con débil presencia de tipos. En las sentencias SQL normalmente no se especifican tipos de datos, pues se encuentran intrínsecos en la definición de las columnas (campos). Otra característica es que es un lenguaje orientados a conjuntos, pues cuando se realiza una consulta de datos se espera como resultado un cierto subconjunto homogéneo de datos, sin que conozcamos la estructura y volumen fijo de elementos, esto hace que los resultados sean con frecuencia difícilmente previsibles, cosa que no sucede con los lenguajes de tercera generación, lo que nos obligará a implementar mecanismos de programación adicional para solventar esta disconformidad.

## 1 DEFINICIÓN DE DATOS.

Permite definir nuevos objetos o destruir objetos existentes mediante las sentencias *CREATE* y *DROP* así como establecer restricciones para los campos de las tablas (*NOT NULL*, *CHECK* y *CONSTRAINT*), y establecer relaciones entre tablas (*PRIMARY KEY*, *FOREIGN KEY*). Una tabla debe ser definida antes de poder ser utilizada, especificando el nombre de la(s) columna(s) (campos) y el tipo de dato al cual pertenecerán (numéricos, alfanúmericos, tiempo), así como su respectiva longitud.

### Tipos de datos

#### *Datos numéricos*

Representan cantidades numéricas, susceptibles de participar en cálculos aritméticos. Entre ellos se encuentran los siguientes grupos:

- **Enteros**  
Representan valores enteros. Son adecuados para columnas que contenen-



gan contadores, códigos identificativos o cualquier rango de valores de los que se conoce de antemano cuáles serán los valores máximo y mínimo que podrán darse. Su ventaja es que ocupan poco espacio y que ciertas funciones se realizan sobre ellos de manera muy eficiente, sin embargo, poseen la desventaja de ser poco flexibles por el limitado rango (longitud) que soportan.

Se distinguen los siguientes enteros:

bit: Puede considerarse un dato entero, que vale 1 o 0.

tinyint: Un entero sin signo entre 0 y 255.

smallint: Representan números en el rango -32767, 32767 y tienen un tamaño de 16 bits. Corresponde con el tipo smallint de ANSI-SQL

int (integer): Representan números en el rango -2147483647, 2147483647 y tienen un tamaño de 32 bits. Corresponde con el tipo integer de ANSI-SQL.

#### ➤ Decimales

Son valores no enteros, es decir, representaciones de números racionales, en los que el número de cifras decimales es conocido o, al menos, limitado. La forma en que estos valores se almacenan y codifican internamente depende del Sistema Gestor de Bases de Datos (SGBD) y no es necesario conocerla para utilizar datos atinentes a este tipo. Se definen utilizando la siguiente sintaxis:

Decimal (p, s)

Numeric (p, s)

Ambos tipos son sinónimos y admiten datos en el rango  $-10^{38} - 1$  a  $10^{38} - 1$ . El primer entero, p, se denomina precisión y es el número total de dígitos que puede tener un valor. El segundo entero, s, se denomina escala y es el número total de dígitos de la parte fraccionaria. La escala puede omitirse, en cuyo caso diremos que se trata de un número decimal de precisión ajustable.

Las principales ventajas de este tipo de datos estriban en que pueden adaptarse a la necesidad de precisión de nuestra base de datos, y asimismo el tamaño de almacenamiento. Por otra parte, los procesos de ordenación y las operaciones aritméticas son más lentas con estos tipos de datos que con los reales, tipo que veremos posteriormente. Por otra parte, no suelen existir tipos de datos homologables a ellos en la mayoría de los lenguajes de programación. El tamaño de almacenamiento de estos



valores depende de la precisión.

➤ Reales

En aplicaciones científicas o de ingeniería es común encontrarnos con cantidades que no son conocidas exactamente sino con un cierto grado de precisión. Para este tipo de usos y situaciones son apropiados los números reales, esto es, los números aproximados en coma flotante. Su principal restricción es que poseen una precisión limitada, pues sólo conservan los dígitos más significativos de un cierto valor.

Existen diversos tamaños de tipos de datos en coma flotante.

Float(n): Es de precisión variable. *n* acepta valores entre 1 y 53 y determina el tamaño de almacenamiento. Su rango es  $1.79E - 308$  y  $1.79E + 308$ . El tipo homólogo en *ANSI-SQL*, es *double precision* y *float (n)*.

Real: Es de precisión simple y ocupa 4 bytes. Su rango es  $3.40E - 38$  y  $3.40E + 38$ .

### *Datos alfanuméricos*

Representan combinaciones de caracteres. Se utilizan para información textual o descriptiva. Podemos distinguir entre caracteres *ANSI* y *Unicode*. Los primeros se codifican con 8 bits y aceptan 256 caracteres, mientras que los segundos se codifican con 16 bits, por lo que pueden admitir 64K caracteres diferentes, con lo que los problemas de la codificación de los diversos juegos de caracteres para los diversos idiomas desaparece.

➤ De longitud fija

Existen dos versiones:

1) Cadenas de longitud fija *ANSI*: Se definen con la sintaxis *char (n)*.

*n* es un entero sin signo que representa la longitud de la cadena y que puede ser desde 1 hasta 8.000. El equivalente *SQL-92* es *character*.

2) Cadenas de longitud fija *Unicode*: Se definen con la sintaxis *nchar (n)*. *n* es un entero sin signo que representa la longitud de la cadena y que



puede ser desde 1 hasta 4000. El equivalente *SQL-92* es *national character*.

➤ De longitud variable

1) Cadena de caracteres de longitud variable *ANSI varchar (n)*, donde *n* es un entero sin signo que representa la longitud máxima de la cadena y que puede valer desde 1 hasta 8,000. Puede contener cadenas vacías. Los equivalentes *ANSI-SQL-92* son *char varying* o *character varying*.

2) Cadena de caracteres de longitud variable *Unicode: nvarchar (n)*, donde *n* es un entero sin signo que representa la longitud máxima de la cadena y que puede valer desde 1 hasta 4,000. Puede contener cadenas vacías. Los equivalentes *ANSI-SQL-92* son *national char varying* o *national character varying*.

➤ De gran tamaño

Almacena un bloque de texto. Se almacenan fuera de las columnas en las que se incluyen en páginas de disco separadas. Forman parte de un tipo de objetos denominados *BLOBS* o *Binary Large Objects*.

1) Cadena de caracteres de gran tamaño *ANSI: text*. El número máximo de caracteres es  $2^{31} - 1$  (2,147,483,647).

2) Cadena de caracteres de gran tamaño *Unicode: ntext*. El número máximo de caracteres es  $2^{30} - 1$  (1,073,741,823).

### *Datos de tiempo*

Se utilizan para almacenar fechas y horas. *Transact-SQL* fusiona ambos conceptos en dos tipos de datos, *datetime*. y *smalldatetime*. *ANSI-SQL-92* distingue, sin embargo, entre fechas y horas, como veremos más adelante.

### Datos de tiempo en *Transact-SQL*

- *Datetime*: Representa un instante en el tiempo, codificado como un valor numérico, con una precisión de 3 milésimas de segundo. El rango de fechas admisibles es desde el 1 de enero de 1753 hasta el 31 de



diciembre de 9999. El rango viene determinado por la codificación numérica del valor. En este caso `datetime` se codifica como dos enteros de 32 bits. El primero de ellos contiene los días transcurridos desde el 1 de enero de 1900, pudiendo tomar valores negativos para fechas anteriores. El segundo entero representa los milisegundos transcurridos desde la medianoche.

- `Smalldatetime`: Representa un instante en el tiempo, codificado como un valor numérico, con una precisión de minuto. El rango de fechas admisibles es desde el 1 de enero de 1900 hasta el 6 de junio de 2079. La codificación es similar a la de `datetime` salvo que los enteros son de 16 bits y que se almacenan los minutos y no los milisegundos transcurridos desde la medianoche.

#### Datos de tiempo en ANSI-SQL

##### Sobre fechas

Representan un día determinado, año, mes y día, todos ellos números *enteros* sin signo. Se definen con la palabra `DATE` y ocupan 10 bytes.

##### Sobre horas

Representan una hora determinada en horas, minutos y segundos, todos ellos enteros sin signo de dos dígitos. Se definen con la palabra `TIME` y se almacenan en 8 bytes.

##### Sobre instantes

Sus valores constan de siete partes, todas ellas enteros sin signo, que representan, respectivamente: año, mes, día, hora, minuto, segundo y microsegundo. Este último tiene una precisión de seis dígitos. Se definen con la palabra `TIMESTAMP` y se almacenan en 19 bytes.

##### Sobre duración

Con los valores de tiempo solamente se permiten operaciones de suma y resta. Vamos a presentar el concepto de duración. Duración de calendario: Es un número, positivo o negativo, de tipo `DECIMAL (8, 0)` cuyo valor se interpreta como un intervalo válido de tiempo en el calendario. Por ejemplo, el resultado de la expresión `DATE ('18.4.20')` `DATE ('31.12.1999')` es una duración de calendario. El formato de duración de calendario es el siguiente:

*aaaammdd*

donde *aaaa* representan años, *mm* meses, *dd* días

Duración de reloj: Es un número, positivo o negativo, de tipo `DECIMAL (6, 0)` cuyo valor se interpreta como un intervalo válido de tiempo en un



reloj de 24 horas con precisión de segundos. Por ejemplo, el resultado de la expresión `TIME('18.50.55') - TIME('15.30.45')` es una duración de reloj.

*hhmmss*

donde *hh* representan horas, *mm* minutos, *ss* segundos

#### Sobre duración explícita

Es un intervalo expresado como un valor numérico, entero o no, seguido de una palabra predefinida que indica la unidad de tiempo en que se mide. Esta palabra puede ser una de las siguientes: `YEARS`, `MONTHS`, `DAYS`, `HOURS`, `MINUTES` o `SECONDS`.

#### *Datos binarios*

Se utilizan básicamente para patrones de bits.

##### ➤ Binarios de tamaño fijo

**Binary (n):** Una cadena de bits de longitud fija *n*. Puede almacenar hasta 8,000 bytes. El tamaño por defecto es 1. El equivalente ANSI-SQL-92 es `bit (n)`

**Timestamp:** Es un valor binario de 8 bytes que es único para cada registro dentro de una base de datos. Sólo podemos tener un valor timestamp por tabla.

##### ➤ Binarios de tamaño variable

La implementación en este tipo de datos varía enormemente de un gestor a otro. El estándar especifica el tipo de datos `BIT VARYING`. En muchas ocasiones se utilizan para almacenar objetos binarios variados como, por ejemplo, imágenes o ficheros binarios en general.

**varbinary (n):** Una cadena de bits de longitud variable, pero con un máximo de *n*. Puede almacenar hasta 8,000 bytes. El tamaño por defecto es 1, y puede contener datos vacíos, es decir, con longitud 0 bytes. El equivalente ANSI-SQL-92 es `bit varying`.

#### *Datos monetarios*

Estos datos no son soportados por ANSI-SQL, pero sí por Transact-SQL.



Sirven, cono su propio nombre indica, para almacenar valores monetarios. Preceden al valor con cualquiera de los símbolos de moneda más frecuentemente utilizados (\$, por ejemplo) y no se escriben entre comillas, ya que *SQL Server* los trata como un valor *DECIMAL* con escala igual a 4.

- *Money*: Almacena valores monetarios desde -922337203685, 477.5808 a +922337203685477.5807. El tamaño de almacenamiento es 8 bytes.
- *Smallmoney*: Almacena valores monetarios desde -214,748.3648 a +214,748.3647. El tamaño de almacenamiento es 4 bytes.

## 2 SENTENCIAS DE SQL

<sup>1</sup>*CREATE* Crea objetos como tablas, procedimientos almacenados (*stored procedures*) y Bases de Datos.

Ejemplos:

En los siguientes ejemplos la sintaxis tendrá tipo de letra courier new 11, mientras la explicación courier new 8 y solo bastaría borrar la explicación para ejecutar la instrucción en el Query Analyzer de *SQL Server 2000* (se utilizan los parámetros más comunes a estas instrucciones si desea ver todos y cada uno de ellos en la ayuda de *SQL Server* podrá encontrarlos).

*Crear Base de Datos.*

```
CREATE DATABASE Sales Base de datos con nombre Sales
ON Empieza la parte fisica
( NAME = Sales_dat, Nombre del archive de datos
```

*Crear Tablas.*

```
CREATE TABLE jobs Crea una tabla con nombre jobs
(
  job_id smallint Columna de nombre job_id(campo) y tipo smallint
    IDENTITY(1,1) Contador con tasa de crecimiento 1
    PRIMARY KEY CLUSTERED, Llave primaria como indice
  job_desc varchar(50) NOT NULL Columna job_desc de tipo varchar con
longitude de 50 y siempre debe haber dato en esta columna
  min_lvl tinyint NOT NULL Columna min_lvl de tipo tinyint y siempre debe
haber dato en esta columna
```

<sup>1</sup> Sintaxis y ejemplos de la Ayuda de *SQL Server 2000*



```
        CHECK (min_lvl >= 10), Checa que el dato para min_lvl sea mayor
igual a 10
        max_lvl tinyint NOT NULL Columna max_lvl de tipo tinyint y siempre debe
haber dato en esta columna
        CHECK (max_lvl <= 250) Checa que el dato para max_lvl sea menor
igual a 250
    )
```

### *Crear Procedimientos almacenados.*

```
CREATE PROCEDURE InfoAutor Crea un procedimiento con nombre InfoAutor
@Apellido varchar(40) Parámetro Apellido de tipo varchar con longitud 40
@Nombre varchar(20) Parámetro Nombre de tipo varchar con longitud 20
```

```
AS Especifica la acción a realizar
Select Au_Nombre, Titulo Selecciona las columnas Au_Nombre y Titulo
From Autores De la tabla Autores
Where Au_nombre=@Nombre and Au_Apellido=@Apellido Donde los
parametros a pasar coincidan con los datos de las columnas Au_nombre y Au_Apellido
Group by Au_nombre Agrupando el resultado por Au_Nombre
Order by Au_Apellido asc Ordenando por Au_Apellido ascendentemente
```

### *Eliminar Bases de Datos.*

```
Drop Database Sales Borra base de datos de nombre Sales
```

### *Eliminar Tablas.*

```
DROP Table jobs Borra la tabla de nombre jobs
```

### *Eliminar Procedimientos almacenados.*

```
Drop Procedure InfoAutor Borra procedimiento almacenado de nombre InfoAutor
```

Cabe mencionar que al utilizar las instrucciones de DROP se debe tener sumo cuidado, pues no existe ninguna instrucción de deshacer, aunque si se creó un respaldo de la base de datos se puede restaurarse mediante un RESTORE, hasta el instante que se creó el respaldo lo que querrá decir que si no se tiene un respaldo fresco tendrán que ejecutarse todos los procedimientos, hasta la fecha de borrado, para asegurar la integridad de la información.



### 3 CONSULTA Y MANIPULACIÓN DE DATOS.

Las consultas permiten obtener datos de las tablas y especificar la forma en la que deseamos que se presenten, la manipulación permite modificar, añadir y borrar filas en las tablas de las bases de datos.

- *SELECT*. Esta sentencia permite extraer datos de una o varias tablas y se utiliza para realizar consultas.
- *INSERT*. Permite añadir una o varias filas a una tabla.
- *UPDATE*. Permite modificar valores de una o más filas de una tabla.
- *DELETE*. Permite borrar una o más filas de una tabla.

#### <sup>2</sup>SELECT

Obtiene filas de la base de datos y permite realizar la selección de una o varias filas o columnas de una o varias tablas. La sintaxis completa de la instrucción *SELECT* es compleja, aunque las cláusulas principales se pueden resumir del modo siguiente:

```
SELECT
[ INTO ]
FROM
[ WHERE ]
[ GROUP BY ]
[ ORDER BY ]
```

Las cláusulas entre corchetes son opcionales y algunas cláusulas tiene sus propios argumentos como se verá en la descripción siguiente.

- *SELECT* . - Especifica las columnas que va a devolver la consulta.  
Sus argumentos son:

*All, Distinct, TOP n Percent, \*, select\_list.*

#### *All*

Especifica que pueden aparecer filas duplicadas en el conjunto de resultados. *All* es el valor predeterminado.

#### *Distinct*

Especifica que sólo pueden aparecer filas exclusivas en el conjunto de resultados.

<sup>2</sup> Sintaxis y ejemplos de la Ayuda de SQL Server 2000



### *TOP n [Percent]*

Especifica que sólo se van a utilizar las primeras *n* filas del conjunto de resultados de la consulta; *n* es un valor entero entre 0 y 4294967295. Si se agrega *Percent*, sólo se va a utilizar el primer *n* por ciento de filas del conjunto de resultados. Cuando se especifica con *Percent*, *n* debe ser un entero entre 0 y 100.

Si la consulta contiene una cláusula *ORDER BY*, se utilizarán las primeras *n* filas (o el porcentaje *n* de filas) ordenadas por la cláusula *ORDER BY*. Cuando la consulta no tiene una cláusula *ORDER BY*, el número de filas es arbitrario.

### *select\_list*

Son las columnas que se van a seleccionar para el conjunto de resultados. La lista de selección es una serie de expresiones separadas por comas, como nombres de columnas (*column name*), expresiones (*expression*) o alias (*column alias*).

#### *column\_name*

Es el nombre de una columna que se va a devolver. Califique *column\_name* para evitar realizar una referencia ambigua, como ocurre cuando dos tablas de la cláusula *FROM* tienen columnas de idéntico nombre. Si se combinan las dos tablas en una consulta, se puede especificar *Tabla.Columna* (ejemplo *Clientes.Paterno* ó *Vendedores.Paterno*).

#### *expression*

Es un nombre de columna, constante, función, o una combinación de nombres de columnas, constantes y funciones conectados mediante operadores, o una subconsulta.

#### *column\_alias*

Es un nombre alternativo para reemplazar el nombre de la columna en el conjunto de resultados de la consulta. Por ejemplo, se puede especificar el alias "Quantity", "Quantity to Date" o "Qty" para una columna llamada *quantity*.

### *\* (Asterisco)*

Especifica que se deben devolver todas las columnas de todas las tablas y vistas de la cláusula *FROM*. Las columnas se devuelven por tabla o vista, tal como se especifique en la cláusula *FROM*, en el orden en que se encuentran en la tabla o vista.



- INTO . - Crea una nueva tabla e inserta en ella las filas resultantes de la consulta.

Sus Argumentos son:

*New\_Table.*

*New Table.* Especifica el nombre de una nueva tabla que se va a crear, basada en las columnas de la lista de selección y en las filas seleccionadas por la cláusula WHERE.

- FROM . - Ruta de la cual proceden los datos de las columnas a seleccionar, regularmente solo se especifica la tabla, pero puede especificarse incluso, desde que base de datos se extraerá la información.
- WHERE . - Se especifica una condición a cumplir por los datos que serán devueltos.
- GROUP BY . - Agrupa el resultado de la petición conforme a una expresión, por ejemplo edades, años etc.
- ORDER BY . - Ordena de acuerdo a una expresión de forma ascendente o descendente.

Se puede utilizar el operador UNION entre consultas para combinar sus resultados en un solo conjunto de resultados.

```
SELECT
[ INTO ]
FROM
[ WHERE ]
[ GROUP BY ]
[ ORDER BY ]

[ UNION ]
```

```
SELECT
[ INTO ]
FROM
[ WHERE ]
[ GROUP BY ]
[ ORDER BY ]
```



## Ejemplos de SELECT

### A. Utilizar SELECT para obtener filas y columnas

En este ejemplo se muestran tres fragmentos de código. En el primer ejemplo de código se devuelven todas las filas y todas las columnas (con \*) de la tabla authors de la base de datos pubs.

```
USE pubs
```

```
SELECT *  
FROM authors  
ORDER BY au_lname ASC, au_fname ASC
```

En el siguiente ejemplo se devuelven sólo las filas de los autores que viven en California y su apellido no es McBadden.

```
SELECT au_fname, au_lname, phone AS Telephone  
FROM authors  
WHERE state = 'CA' and au_lname <> 'McBadden'  
ORDER BY au_lname ASC, au_fname ASC
```

### B. Utilizar DISTINCT con SELECT

En este ejemplo se utiliza DISTINCT para evitar la obtención de números duplicados de Id. de autor.

```
SELECT DISTINCT au_id  
FROM authors  
ORDER BY au_id  
D. Crear tablas con SELECT INTO  
SELECT * INTO #coffeetabletitles  
FROM titles  
WHERE price < $20
```

## <sup>3</sup>INSERT

Agrega una o más filas nuevas a una tabla.

```
INSERT [INTO]  
ó  
INSERT [VALUES]
```

- INSERT INTO. - Inserta una o más filas según el número de registros, que cumplan con la instrucción hecha.

<sup>3</sup> Sintaxis y ejemplos de la Ayuda de SQL Server 2000



Sus argumentos son:

*Table\_name, Instrucción*

*Table\_name*

Es el nombre de la tabla que va a recibir los datos.

- **INSERT VALUES.** - Inserta una fila de acuerdo a los campos suministrados.  
Sus argumentos son:

*VALUES*

*Values*

Presenta la lista de valores de datos que se van a insertar. Debe haber un valor de datos para cada columna definida. La lista de valores debe estar entre paréntesis.

Si los valores de la lista VALUES no están en el mismo orden que las columnas de la tabla o no tienen un valor para cada columna de la tabla, debe especificarse explícitamente la columna que almacena cada valor de entrada.

Si no hay un valor predeterminado para la columna y ésta permite el valor NULL, se inserta NULL.

## Ejemplos

### A. Utiliza una instrucción INSERT sencilla

Se inserta una fila en la tabla T1.

Después de crear la tabla T1

```
CREATE TABLE T1 ( column_1 int, column_2 varchar(30))  
INSERT T1 VALUES (1, 'Row #1')
```

### B. Cargar datos con la opción DEFAULT VALUES

En la instrucción CREATE TABLE de este ejemplo se define cada columna con un valor que se puede utilizar cuando no se especifique un valor explícito para las columnas en la instrucción INSERT. Se utiliza la opción DEFAULT VALUES de la instrucción INSERT para agregar filas sin proporcionar valores explícitos.



```
CREATE TABLE T1
(column_1 int identity,
column_2 varchar(30)
CONSTRAINT default_name DEFAULT ('column default'),
column_3 timestamp,
column_4 varchar(30),
column_5 int NULL)

INSERT INTO T1 DEFAULT VALUES
SELECT *
FROM T1
```

#### <sup>4</sup>UPDATE

Cambiar datos o actualizarlos (edita datos).

La instrucción UPDATE puede cambiar los valores de filas individuales, grupos de filas o todas las filas de una tabla. La instrucción UPDATE tiene estas cláusulas principales:

```
UPDATE
SET
[FROM]
[WHERE]
```

- UPDATE.- Edita valores en campos determinados por la instrucción. Su argumento es:

*Table\_Name*

*Table\_Name*

Especifica el nombre de la tabla donde se encuentran los campos a editar.

El nombre se puede calificar con el servidor vinculado, la base de datos y el nombre del propietario si la tabla no se encuentra en el servidor o en la base de datos actuales, o si no pertenece al propietario actual.

- SET.- Contiene una lista separada por comas de las columnas que deben actualizarse y el nuevo valor de cada columna, con el formato *column\_name = expression*. El valor suministrado por las expresiones incluye elementos tales como constantes, valores seleccionados de

<sup>4</sup> Sintaxis y ejemplos de la Ayuda de SQL Server 2000



una columna de otra tabla, o valores calculados por una expresión compleja (query).

La *column\_name* se debe encontrar en la tabla o en la vista especificadas en la cláusula UPDATE. No se pueden actualizar las columnas de identidad.

#### La expresión

Es una variable, un valor literal, una expresión o una instrucción subSELECT entre paréntesis que devuelve un solo valor. El valor que devuelve *expresión* sustituye al valor existente en *column\_name* o *@variable*.

- FROM. - Identifica las tablas que suministran los valores de las expresiones de la cláusula SET, y las condiciones de combinación opcional entre las tablas de origen.
- WHERE. - Especifica la condición de búsqueda que define las filas de las tablas de origen que están calificadas para proporcionar valores para las expresiones de la cláusula SET.

#### Ejemplos de UPDATE

##### A. Utilizar una instrucción UPDATE simple

En este ejemplo se muestra cómo se puede actualizar la tabla **publishers** si todas las editoriales de la tabla trasladan su sede central a Atlanta, Georgia.

```
UPDATE publishers
SET city = 'Atlanta', state = 'GA'
```

##### B. En este ejemplo se cambian los nombres de todas las editoriales a NULL.

```
UPDATE publishers
SET pub_name = NULL
```

##### C. También se pueden utilizar valores calculados en una actualización.

En este ejemplo se duplican todos los precios de la tabla **titles**.

```
UPDATE titles
SET price = price * 2
```



#### D. Utilizar la instrucción UPDATE con una cláusula WHERE

La cláusula WHERE especifica las filas que se van a actualizar. Por ejemplo, imaginemos que el norte de California cambiara su nombre por el de Pacifica (abreviado como PC) y los ciudadanos de Oakland decidieran cambiar el nombre de la localidad por el de Bay City. En este ejemplo se muestra cómo actualizar la tabla authors para todos los residentes de la antigua Oakland, cuyas direcciones han quedado desfasadas.

```
UPDATE authors
SET state = 'PC', city = 'Bay City'
WHERE state = 'CA' AND city = 'Oakland'
```

Deberá escribir otra instrucción para cambiar el nombre del estado para los residentes de otras ciudades del norte de California.

#### <sup>5</sup>DELETE

Elimina filas de una tabla, según la condición.

```
DELETE
FROM
[WHERE]
```

Los argumentos debido a que son los mismos que se han venido utilizando en el SELECT ,UPDATE e INSERT, por lo cual no se abundará en esta sección.

- DELETE . - *Table\_name*  
Es el nombre de la tabla que va a recibir los datos.
- FROM . - Identifica las tablas que suministran los valores a borrar.
- WHERE . - Especifica la condición de búsqueda que define las filas de las tablas de origen que están calificadas para desaparecer los valores.

<sup>5</sup> Sintaxis y ejemplos de la Ayuda de SQL Server 2000



## Ejemplos

### A. Utilizar DELETE sin parámetros

Este ejemplo elimina todas las filas de la tabla authors.

```
USE pubs
DELETE authors
```

### B. Utilizar DELETE en un conjunto de filas

Debido a que au\_lname puede no ser único, este ejemplo elimina todas las filas en que au\_lname es McBadden.

```
USE pubs
DELETE FROM authors
WHERE au_lname = 'McBadden'
```



## GLOSARIO.

***Access:***

Es un sistema miembro del paquete Office de Microsoft , que permite crear, administrar, y utilizar Bases de Datos.

***Algoritmo:***

Secuencia de pasos que conducen a la realización de una tarea. es un conjunto ordenado finito y bien definido de etapas que conducen a la obtención de un resultado.

***Analysis Manager:***

Herramienta para manejar procesamientos de análisis en línea para toma de decisiones, esta herramienta forma parte del manejador de bases de datos de Microsoft SQL Server.

***Base de Datos:***

Es una colección de datos interrelacionados, almacenados en conjunto en cualquier soporte de manera que sean accesibles y manipulables por un sistema informático.

***Comshare:***

Empresa dedicada a la implantación y desarrollo de soluciones estratégicas en el ámbito de negocios inteligentes.

***Dato:***

Hecho o detalle mínimo, que sirve de sostén al ser relacionado con un razonamiento o una investigación.

***Decisión Desktop:***

Software desarrollado por la empresa Comshare, para el manejo de información y análisis en línea de una empresa o negocio.

***Enterprise Manager:***

Es la herramienta primaria de SQL Server de Microsoft, para definir grupos de servidores que están corriendo bajo ambiente SQL Server, configura y



administra a los usuarios del sistema además de invocar a asistentes en la ayuda de tareas relacionadas con el manejo de bases de datos.

***Excel:***

Es una hoja de calculo miembro de Office de Microsoft.

***Información:***

En el sentido informático, se refiere a datos relacionados con un tema específico del cual se pueden deducir o concluir algo de interes para el lector en cuestión.

***Manejador de Base de Datos:***

Herramienta de tipo software que se utiliza para administrar y manejar bases de datos.

***Query:***

Es el mecanismo de gestión de consultas.

***Registrar:***

En este caso es pasar una serie de parámetros requeridos para dar de alta un servidor bajo el mismo ambiente tecnológico.

***Service Pack:***

Es una serie de correcciones a los sistemas o programas que ya han sido puestos en marcha.

***Servidor:***

Es normalmente una computadora muy poderosa, donde residen los programas de administración que son accedidos por otros computadores (clientes).

***Sistema Cliente-Servidor:***

Es una aplicación de gestión compartida entre un computador y otro.

***Sistema Operativo:***

Es el software encargado de ejercer el control y coordinar y administrar el uso del hardware entre diferentes programas de aplicación y los diferentes usuarios.



***Software:***

Es un conjunto de programas realizados bajo un lenguaje de programación para un fin específico.

***SQL Server 2000:***

Es un software manejador de bases de datos.

***Tabla:***

Es un conjunto de datos o entidades con atributos en común que son llamados campos.

***Transformation Services:***

Es una aplicación de transformación de datos (importador y exportador de datos) que viene dentro del Enterprise Manager.

***Windows 2000 Server:***

Es un sistema operativo creado por Microsoft, con más capacidades de administración de usuarios y programas además de mayor seguridad.



## BIBLIOGRAFÍA.

- **MOCCIARO A. Osvaldo.** "Presupuesto Integrado", Edit. Macchi, Buenos Aires Argentina, 1993.
- **C.P. COETO Mota Rodolfo.** "El Presupuesto Planeación y Elaboración", Edit. UNAM, México D.F., 1988.
- **PANIAGUA Bravo Victor M.** "Sistema de Control Presupuestario", Edit. Inst. Mexicano de Contadores Públicos A.C., México D.F., 1993.
- **DELGADO Garrón Alberto.** "Microsoft SQL Server 7", Edit. Prentice Hall, Barcelona, 1999.
- **Microsoft.** "Designing and Implementing OLAP Solutions with Microsoft SQL Server 2000, Delibery Guide", Edit. Microsoft Corporation, Colombia, 2001.
- **Microsoft.** "Ayuda de SQL Server 2000 (Transact-SQL)", Ayuda del programa.
- **Microsoft.** "OLAP Manager Tutorial de Analysis Manager 2000".
- **UNAM.** "El Presupuesto de Ingresos, el Presupuesto de Egresos y Conformación del Presupuesto", México, 1992.
- **UNAM.** "El Presupuesto de Egresos por Objeto del Gasto, el Presupuesto Operativo", México, 1992.
- **UNAM.** "El Presupuesto por Programas, Conceptos básicos, estructura y características", México, 1992.
- **BERSON, Smith.** "Data Warehousing, Datamining y OLAP". Edit. Mc Graw Hill. E.U.A. 2001.
- **SAKHR, Youness.** "Professional Data Warehousing with SQL Server 7 and OLAP". Edit. Wrox. E.U.A. 2000.
- **THOMSEN, Erik.** "OLAP Solutions: Building Multidimensional Information System". Edit Wiley Computer Publishing. USA 2001.

### Páginas de Internet

- <http://altaplana.com/olap/olap.collections.html>
- <http://www.moulton.com/olap/olap.glossary.html>
- <http://www.olapreport.com/>
- <http://www.dwinfocenter.org/>