

Universidad Peruana de Ciencias Aplicadas

OLAP

Terminología y Tecnología Multidimensional

INTRODUCCION

El presente trabajo de investigación pretende explicar los conceptos fundamentales que sustentan al procesamiento analítico en línea (OLAP). Estos conceptos son abordados de la manera mas clara posible, con ejemplos prácticos que permitirán una rápida comprensión de los mismos.

Para comprender las ventajas de la tecnología OLAP es necesario, primero, hacer una comparación con el procesamiento transaccional en línea (OLTP), de tal forma que podamos valorar el alcance de esta tecnología de información.

Entre los principales temas que trataremos se encuentran: datos multidimensionales, la consolidación, los conceptos de jerarquías , bases de datos n-dimensionales, seguridad y finalmente el lenguaje de consulta multidimensional MDSQL.

La principal fuente de información que hemos utilizado es Internet, los sitios web relacionados con OLAP y MsSQLServer.

Esperamos que el contenido del trabajo sea lo mas completo y a la vez práctico posible, para que permita una rápida comprensión.

CONTENIDO

- ¿Qué es OLAP? 1
- ¿Qué son Datos Multidimensionales?..... 3
- Consolidación: La Clave para una Respuesta Consistentemente Rápida..... 5
- Jerarquías Simples dentro de las Dimensiones..... 8
- Variables..... 11
- Aritmética del vector..... 13
- Bases de Datos n-dimensionales..... 14
- Limitaciones Prácticas en el Tamaño de la Base de Datos..... 15
- Tipo de Datos Serie de Tiempo..... 16
- Datos Esparcidos..... 20
- Todas las Dimensiones no se crean Igual..... 21
- Jerarquías Múltiples y Clases dentro de las Dimensiones..... 23
- Profundizando en los Datos Relacionales..... 24
- Seguridad y Robustez..... 25
- El Lenguaje de Consulta Multidimensional MDSQL..... 27

OLAP

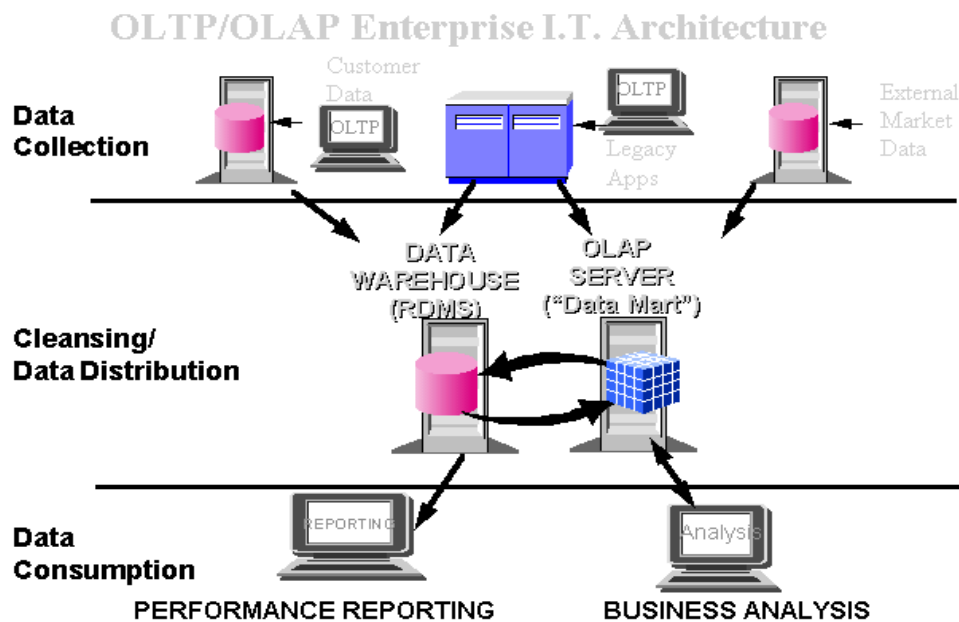
Terminología y Tecnología Multidimensional

1. ¿QUÉ ES OLAP?

OLAP simboliza On-Line Analytical Processing (Procesamiento Analítico en Línea). A diferencia del conocido OLTP (On-Line Transaction Processing – Procesamiento Transaccional en Línea), OLAP describe una clase de tecnologías diseñadas para mantener específicamente el análisis y acceso a datos. Mientras el procesamiento transaccional generalmente confía solamente en las bases de datos relacionales, OLAP viene a ser un sinónimo con vistas multidimensionales de los datos del negocio. Estas vistas multidimensionales se apoyan en la tecnología de bases de datos multidimensionales. Estas vistas multidimensionales proporcionan la base técnica para cálculos y análisis requeridos por las Aplicaciones del Negocio Inteligente.

Tener un RDBMS no significa tener el nirvana inmediato para el soporte de decisiones. Al permitir que los RDBMS estuvieran cerca de los usuarios, nunca se pensaba que los RDBMS proporcionaban funciones poderosas para síntesis, análisis y consolidación de datos (funciones colectivamente conocidas como análisis de datos multidimensional).

– E. F. Codd, Computerworld



Las aplicaciones OLTP se caracterizan por la creación de muchos usuarios, actualizaciones o recuperación de registros individuales. Por consiguiente, las bases de datos OLTP se perfeccionan para actualización de transacciones. Las aplicaciones OLAP son usadas por analistas y gerentes que frecuentemente quieren una vista de datos de nivel superior, como las ventas totales por línea de producto, por región, etc. Las bases de datos OLAP normalmente se actualizan en lote, a menudo de múltiples fuentes, y proporcionan un back-end analítico poderoso a las aplicaciones de múltiples usuarios. Por tanto, las bases de datos OLAP se perfeccionan para el análisis.

Mientras las bases de datos relacionales son buenas al recuperar un número pequeño de archivos rápidamente, ellas no son buenas al recuperar un número grande de archivos y resumirlos on-the-fly. Un tiempo de respuesta lento y el uso excesivo de recursos del sistema son las características comunes de las aplicaciones de soporte de decisión construidas exclusivamente sobre la tecnología de bases de datos relacionales. Debido a la facilidad con la cuál se puede emitir un ejecutar una consulta SQL externa, muchos distribuidores IS (Information Systems) no brindan acceso directo a los usuarios a sus bases de datos relacionales.

Muchos de los problemas que las personas intentan resolver con la tecnología relacional son realmente

multidimensionales en naturaleza. Por ejemplo, una consulta SQL para crear resúmenes de ventas del producto por la región, las ventas de la región por producto, y así sucesivamente, podrían involucrar la revisión de la mayoría, si no todos, de los registros en una base de datos de mercadeo y podría tomar horas de proceso. Un servidor OLAP podría ocuparse de estas preguntas en unos segundos.

OLTP (Relational)	OLAP(Multidimensional)
Atomized	Summarized
Present	Historical
Record-at-a-time	Many records at a time
Process oriented	Subject oriented

Las aplicaciones OLTP tienden a tratar con datos atomizados registro a un tiempo, considerando que las aplicaciones de OLAP normalmente se tratan de los datos resumidos. Mientras las aplicaciones OLTP generalmente no requieren de datos históricos, casi cada aplicación de OLAP se preocupa por ver las tendencias y por consiguiente requiere de datos históricos. Como consecuencia, las bases de datos OLAP necesitan la capacidad de ocuparse de datos series de. Mientras las aplicaciones OLTP y bases de datos tienden a ser organizados alrededor de procesos específicos (como ordenes de entrada), las aplicaciones OLAP tienden a ser orientadas al tema, respondiendo a preguntas como ¿Qué productos están vendiendo bien? o ¿Dónde están mis oficinas de ventas más débiles?.

2. ¿QUÉ SON DATOS MULTIDIMENSIONALES?

Las bases de datos relacionales son organizadas alrededor de una lista de registros. Cada registro contiene información relacionada que es organizada en campos. Un ejemplo típico sería una lista de clientes con los campos para la dirección, número del teléfono, y así sucesivamente, como en el ejemplo siguiente:

Customer Name	Customer #	Telephone	Address
Jack's Hardware	10456	350-7229	40 Main St.
Value Stores	10114	266-7023	18 Elm St.
Housewares Inc.	11104	267-4040	17 Main St.
Walter Lock	11230	423-7700	6 Charles St.

Mientras esta tabla tiene varias columnas de información, cada parte de la información se relaciona con un sólo nombre de cliente. En esencia, esta tabla tiene sólo una dimensión. Si intenta crear una matriz bidimensional con el nombre de cliente abajo y cualquier otro campo (como el Teléfono) cruzado, verá rápidamente que sólo hay una correspondencia uno a uno:

Customer		Telephone			
Dimension		Number	Dimension		
	Jack's Hardware	350-7229	266-7023	267-4040	423-7700
	Value Stores				
	Housewares Inc.				

	Walter Lock				
--	-------------	--	--	--	--

Podría intentar colocar cualquier campo abajo y cualquier otro campo cruzado y todavía conseguiría sólo una correspondencia uno a uno. Esta tabla le dice que estos datos no son multidimensionales y no se prestarían a guardarse en una base de datos multidimensional.

Ahora veamos un ejemplo de una tabla relacional dónde hay más de una correspondencia uno a uno entre los campos. En el ejemplo siguiente, tenemos los datos de las ventas para cada producto en cada región. Suponga que la compañía tiene cuatro productos (tuercas, tornillos, pernos y lavaderos) qué se vende en tres territorios (Este, Oeste y Centro). Aquí está cómo se cargarían esos datos en una tabla relacional:

Producto	Región	Ventas
Tuercas	Este	50
Tuercas	Oeste	60
Tuercas	Centro	100
Tornillos	Este	40
Tornillos	Oeste	70
Tornillos	Centro	80
Pernos	Este	90
Pernos	Oeste	120
Pernos	Centro	140
Lavaderos	Este	20
Lavaderos	Oeste	10
Lavaderos	Centro	30

Una manera muy clara de representar estos datos sería como una matriz bidimensional:

	Este	Oeste	Centro
Tuercas	50	60	100
Tornillos	40	70	80
Pernos	90	120	140
Lavaderos	20	10	30

Estos datos de las ventas son inherentemente una matriz bidimensional (las dos dimensiones que son productos y regiones). Mientras puede llenarse en una tabla relacional de tres campos, encaja mejor en una matriz con dos dimensiones – productos y regiones. En el argot multidimensional, diríamos que esta tabla representa las Ventas dimensionadas por Productos y Regiones.

Ahora hablemos sobre cómo las consultas en estas dos tablas podrían diferir. Si todo lo que fuese a preguntar sea preguntas como ¿Cuáles son las ventas de tuercas en el Este?, ¿Cuáles son las ventas de lavaderos en el Oeste?, y otras consultas que recuperaran un sólo número, entonces no habría ninguna necesidad de poner estos datos en una base de datos multidimensional. Sin embargo, si quisiera hacer las preguntas como ¿Cuáles son las ventas totales de tuercas? o ¿Cuáles son las ventas totales para el Este?, entonces empieza a entrar en preguntas que involucran recuperar números múltiples y acumulados. Si considera bases de datos más grandes dónde podría tener miles de productos, el tiempo que toma para una base de datos relacional recuperar todos los números y acumularlos se vuelve intolerable. Una base de datos relacional típica puede examinar unos cientos de registros por segundo. Una base de datos multidimensional típica puede adicionar números en filas y columnas a una velocidad de 10,000 o más por segundo. Como veremos en el ejemplo siguiente, es fácil generar consultas que podrían tomar minutos u horas para completarse usando la tecnología relacional, pero sólo segundos usando la tecnología multidimensional OLAP.

Preguntas como ¿Cuáles son las ventas totales para las tuercas? o Buscar las ventas totales para el Este involucran aritmética de fila y columna, sólo como una hoja de cálculo. Para obtener una respuesta a las ventas totales para el Este, la base de datos bidimensional simplemente busca la columna llamada Este y suma todos los números en la columna. La misma pregunta en la tabla relacional debe buscar y recuperar los cuatro registros individuales dónde Región = Este y agregar los datos. Una base de datos multidimensional puede localizar la columna entera llamada Este y consolidar sus contenidos en mucho menos tiempo que el tomado en la base de datos relacional para encontrar todos los registros Este. Por tanto, para este tipo de preguntas, una base de datos multidimensional tiene una ventaja enorme de rendimiento.

• CONSOLIDACIÓN:

LA CLAVE PARA UNA RESPUESTA CONSISTENTEMENTE RÁPIDA

El tiempo de respuesta a una pregunta en una base de datos multidimensional, no obstante, todavía depende de cuántos números tienen que ser calculados al vuelo (on-the-fly). Lo que la mayoría personas necesitan de sus aplicaciones es un tiempo de respuesta consistentemente rápido, sin tener en cuenta la pregunta. Así que la única manera de conseguir el tiempo de respuesta consistentemente rápido es pre-agregar (o consolidar) todos los subtotales lógicos y totales. Esto es, de hecho, lo que la mayoría de establecimientos IS también hace con sus tablas relacionadas. La diferencia es que una base de datos multidimensional hace centenares – si no miles – de cálculos aritméticos en filas y columnas más rápidamente que una base de datos relacional y puede consolidar las enormes bases de datos en unos minutos u horas.

Usando nuestro ejemplo anterior, asumamos que queremos obtener el tiempo de respuesta completamente consistente de nuestra aplicación sin tener en cuenta cual era la pregunta. Con las bases de datos relacionales, el tiempo de búsqueda es aproximadamente proporcional al número de archivos recuperados. Así que tomaría cuatro veces como mucho recuperar un total como las Ventas Totales para el Este mas que el que habría para recuperar un solo registro como Lavaderos para el Este. Para calcular las ventas Totales para el Este, cuatro registros tienen que ser recuperados y sumados. Si preguntáramos ¿Cuales son las ventas totales para todas las regiones? tendríamos que calcular el total de los 12 números en la base de datos (cuatro productos medidos en tres regiones). Esto tomaría 12 veces de tiempo.

Para lograr el tiempo de respuesta consistente, la mayoría de los diseñadores de sistemas agrupa los totales y los coloca nuevamente en la base de datos, así:

Producto	Región	Ventas
Tuercas	Este	50
Tuercas	Oeste	60

Tuercas	Centro	100
Tuercas	Total	210
Tornillos	Este	40
Tornillos	Oeste	70
Tornillos	Centro	80
Tornillos	Total	190
Pernos	Este	90
Pernos	Oeste	120
Pernos	Centro	140
Pernos	Total	350
Lavaderos	Este	20
Lavaderos	Oeste	10
Lavaderos	Centro	30
Lavaderos	Total	60
Total	Este	200
Total	Oeste	260
Total	Centro	350
Total	Total	810

Con todos los totales agrupados, podemos responder cualquier pregunta que involucra totales por producto o región accediendo sólo a un único registro. Como se verá brevemente, esto trabaja bien hasta que la base de datos se vuelve demasiado grande – entonces el pre-cálculo de estos totales toma más tiempo que las horas del día.

En el ejemplo anterior, calcular los totales involucra 28 lecturas y ocho escrituras de bases de datos. Una base de datos relacional típica puede leer aproximadamente 200 archivos por segundo y puede escribir quizás 20 nuevos registros por segundo. La agrupación en esta pequeña base de datos tomaría menos de un segundo. Sin embargo, es realmente más típico encontrar bases de datos que podrían tomar días o incluso semanas para consolidar.

Un servidor OLAP multidimensional puede realizar las mismas consolidaciones aritmética de fila y de columna. Considerando que una base de datos relacional puede acceder a unos cientos de registros por segundo, un servidor OLAP adecuado debe ser capaz de consolidar de 20,000 a 30,000 celdas (equivalente a registros en la tabla relacional) por segundo, incluyendo el tiempo para escribir los totales en la base de datos – aproximadamente dos a tres más rápido, en orden de magnitud, que la tecnología relacional. Es la capacidad de realizar consolidaciones a velocidades altas la fuente del poder de la base de datos multidimensional. (Las celdas son el resultado de combinaciones, mientras refiriéndose a las combinaciones de productos, regiones, o miembros de otras dimensiones.)

Aquí está cómo las mismas consolidaciones aparecerían en una base de datos multidimensional OLAP. No tiene que saber algo sobre la tecnología de base de datos al observar que esta representación bidimensional de los datos totales en filas y columnas tiene más sentido que la vista relacional anterior. Y, así como la tabla subsiguiente toma un menor espacio en esta página que la tabla relacional anterior, la base de datos multidimensional OLAP tomará menos espacio de disco puesto que los nombres de las regiones y productos

no se repiten en la base de datos multidimensional a diferencia de la tabla relacional. También puede prever maneras de guardar los datos subsiguientes en disco que requerirían menos accesos de disco que con la tabla relacional anterior. El almacenamiento físico de datos en el disco y el esquema de indexado para localizar los datos son claves de la velocidad de la base de datos OLAP.

	Este	Oeste	Centro	
Tuercas	50	60	100	210
Tornillos	40	70	80	190
Pernos	90	120	140	350
Lavaderos	20	10	30	60
Total	200	260	350	810

Aquí hay alguna terminología multidimensional para aprender: Las celdas que contienen los datos de la fuente original (mostrado letras normales) se llaman entradas. Los totales calculados (mostrados en negrita) se llama resultados. Este, Oeste y Centro son miembros de entrada de la dimensión Región. El Total por Región es un miembro resultado de la dimensión Región. Similarmente, las Tuercas, Tornillos, Pernos, Lavaderos y Total son miembros de la dimensión Producto. Los números reales (en este caso digamos que ellos son Cajas) representan una variable. Para esta tabla, se diría que la variable 'Cajas' esta dimensionada por 'Producto' y por 'Región'. Las Variables son típicamente medidas numéricas como Ventas, Costos, Ganancias, Gastos, y similares. El número en la intersección de cada región y producto ocupa una celda, así como en una hoja de cálculo. Las celdas son el resultado de combinaciones. La tabla anterior tiene 20 combinaciones y por tanto 20 celdas.

Las dimensiones son aproximadamente equivalentes a Campos en una base de datos relacional. En la tabla relacional anterior, hay campos llamados Producto y Región. En la base de datos multidimensional, Producto y Región son ambas dimensiones. Las celdas son aproximadamente equivalentes a Registros. En este ejemplo, hay el mismo número de registros en la tabla relacional como hay celdas en la base de datos multidimensional y ellos contienen la misma variable (los números).

• JERARQUÍAS SIMPLES DENTRO DE LAS DIMENSIONES

En el ejemplo anterior, hay una jerarquía simple dentro de las dimensiones Producto y Región. Estas jerarquías simples pueden representarse gráficamente como sigue:

Los productos individuales se acomodan en un Total del Producto y las regiones individuales se acomodan en un Total de Región. Éstas son las jerarquías simples, significando que cada entrada se acomoda en sólo un total. Es posible que una dimensión como Productos pudiera tener formas múltiples de vincular los totales: por ejemplo, los productos podrían vincularse por tamaño, color, planta industrial, y similares. En una sección posterior llamada Todas las dimensiones no se crean igual, veremos ejemplos de estructuras jerárquicas que son mucho más complejas.

Las jerarquías simples pueden contener muchos niveles. Por ejemplo:

En este ejemplo, las ciudades se acomodan a los estados, los estados a las regiones, y así sucesivamente. Si su servidor OLAP no soportara niveles múltiples de jerarquía dentro de una dimensión, tendría que expresar ciudades, estados y regiones como dimensiones separadas en la base de datos.

La razón que necesita para niveles múltiples de jerarquía o dimensiones adicionales es que no puede mezclar ciudades, estados, y regiones en una dimensión a menos que tenga dimensiones jerárquicas. Tome la tabla

siguiente. Los usuarios quieren poder ver las ventas del producto por región o estado. Si no tuviera las jerarquías en sus dimensiones, podría intentar crear una base de datos bidimensional así:

Agregando al lado de las filas trabajos hechos. Puede conseguir un número correcto para las ventas totales en el Este, o Nueva York, por ejemplo. Sin embargo, los totales para un producto particular serán equivocados porque las columnas contienen las ventas por estado y también las ventas por región (qué ya contiene las ventas estatales).

En las bases de datos multidimensionales sin jerarquías, la solución a este problema sería tener dimensiones separadas para región y estado.

Ahora puede sumar el producto por región o por estado y obtener un resultado correcto. Sin embargo, se complica conceptualmente más. Imagine la complejidad si su análisis geográfico tiene tres o cuatro niveles de detalle y los productos tienen varios niveles de jerarquía también. ¡Intente visualizar un cubo de datos de siete u ocho dimensiones!

El otro problema con esta solución es que una base de datos con ciudades y estados en dimensiones separadas crearía una base de datos muy esparcida, significando que muchas de las celdas no contendrían datos. Dado que cada ciudad pertenece a sólo un estado, las celdas en todas las otras intersecciones de esa ciudad y todos los otros estados estarían vacías. Hablaremos después sobre las consecuencias negativas de la dispersión.

La manera correcta de resolver este problema es usar las dimensiones jerárquicas. Los Estados en la Región del Este Oriental estarían un nivel por debajo (o en la jerga OLAP sólo debajo del Este), las ciudades simplemente estarían debajo de los estados, y así en adelante:

Con una dimensión geográfica que contiene regiones y estados organizados jerárquicamente, ahora podemos consultar la base de datos para que retorne las ventas del producto por región o estado y los totales siempre serán correctos. La base de datos sabe que la aritmética de la columna no combina a los miembros de la dimensión región que está en niveles diferentes de jerarquía. Por ejemplo, puede sumar todas las ciudades, todos los estados, o todas las regiones, pero sabe que la adición de estados y regiones en conjunto produce un resultado erróneo porque los números de estados ya están incluidos en los números de la región.

La noción de niveles de jerarquía es muy útil al formular las consultas OLAP. Por ejemplo, si el usuario quisiera ver una matriz con los productos al lado y regiones abajo, podría especificar si desea ver todos los niveles de la dimensión región, solo ciudades, sólo estados, y así sucesivamente.

También podemos usar las jerarquías para profundizar (drill down) a niveles sucesivos de detalle. Por ejemplo, podríamos buscar números sólo debajo del Este y obtener las ventas del producto para los estados en la región Oriental. Muchas aplicaciones OLAP usan la profundización como una manera de navegar a través de las sucesivas capas de detalle. Frecuentemente esto se realiza de manera que el usuario simplemente puede pulsar el botón en un producto en la fila en la pantalla y la aplicación automáticamente es llevada al próximo nivel de detalle. En la sección titulada Dentro de los Datos Relacionales veremos por qué puede tener sentido el usar un servidor OLAP para realizar las consolidaciones, pero todavía mantener datos de bajo nivel de detalle en una base de datos relacional.

• VARIABLES

Las variables son las medidas numéricas, similares a los campos valor en una base de datos relacional, como Ventas, Costos, Precio y afines. Algunos servidores OLAP tratan las variables como una dimensión especial y hay algunas razones muy buenas para esto. Piense en variables como dimensionadas por ciertas dimensiones en su base de datos. Por ejemplo, Ventas podría ser dimensionada por Región, Producto y Tipo de Cliente. Precio, por otro lado, podría ser idéntico para todas las Regiones y Tipos de Cliente y por consiguiente sólo

podría necesitar ser dimensionado por Producto. Si las variables fueran simplemente una dimensión normal, se forzaría a la dimensión Precio por todas las otras dimensiones, y habría entonces muchas celdas innecesarias en la base de datos. Tratando las variables como un caso especial de dimensión, puede seleccionar sólo las dimensiones relevantes para cada variable.

Este concepto se llama las Variables Dimensionadas Independientemente y es una herramienta esencial para optimizar el funcionamiento de una base de datos multidimensional, reduciendo su tamaño al mínimo lógico, y reduciendo la complejidad de cargas de la base de datos. No todos los servidores OLAP soportan las variables dimensionadas independientemente.

Las variables (en algunos servidores OLAP) pueden definirse como tener relaciones matemáticas complejas a otras variables. A veces tales variables se llaman variables complejas. En una dimensión normal (Regiones, por ejemplo), sólo pueden expresarse las relaciones entre los miembros de la dimensión con la suma. Por ejemplo, Nueva York sería la suma simple de las ciudades en Nueva York. Una variable debe ser capaz de definir las relaciones matemáticas muy complejas entre las variables. Estas relaciones pueden incluir operaciones aritméticas complejas, promedios calculados, relaciones atrasadas en el tiempo, e incluso ecuaciones simultáneas. Es un valor que se busca en las capacidades de un servidor OLAP debido a que una falta de tales capacidades normalmente significará mucha programación externa que tendrá que ser hecha para definir estas relaciones externas a la propia base de datos.

Las variables también son especiales porque ellas deben incorporar varias reglas para la agrupación. Por ejemplo, cuando las ventas se enlazan con los Productos para el Total del Producto, las cantidades se suman aritméticamente. El Precio, por otro lado, no es aditivo pero es promediado o calculado usando fórmulas más complejas. Similarmente, al convertir los datos de una periodicidad a otra (de diario a por semana), se tratan las variables diferentemente. La conversión de Ventas diarias a Ventas semanales es hecha sumando los días. La conversión del precio de diarias a semanal no se hace ciertamente sumando los precios diarios.

Las variables también pueden contener información sobre cómo ellas se convierten de una moneda a otra (por ejemplo, Ventas e Inventario casi siempre usan diferentes reglas de conversión de moneda), descripciones largas, definiciones de unidad, y así sucesivamente. Todos estos Atributos de Variable normalmente se guardan en un diccionario de datos.

Un concepto más que debe introducirse a estas alturas es la variable deducida. Una variable deducida es una variable que desde el punto de vista del usuario parece ser una variable en la base de datos pero que realmente se calcula en realidad en tiempo de ejecución. Por ejemplo, una base de datos podría contener las variables para Ingresos y Gastos. Se podría crear una variable llamada Margen Bruto substrayendo los Gastos de los Ingresos y guardando esta variable en la base de datos. Alternativamente, podría definir Margen Bruto como una variable deducida, significando que el valor es calculado on-the-fly usando la fórmula $\text{Margen Bruto} = \text{Ingresos} - \text{Gastos}$ (asumiendo que su servidor OLAP soporta variables deducidas). Las Variables Derivadas, por supuesto, no toman ningún espacio en la base de datos por lo que ellas son una manera sumamente útil para reducir el tamaño de una base de datos y reducir tiempos de consolidación al precio de una cantidad pequeña de carga en tiempo de ejecución siempre que una consulta involucre una variable deducida. Desde el punto de vista de un usuario que consulta la base de datos, sin embargo, las variables deducidas simplemente son como cualquier otra variable.

Tenga presente que una base de datos que no trate variables como una dimensión especial con las capacidades mencionadas previamente, probablemente requerirá considerablemente más trabajo en el lado de desarrollo de aplicación. Además, el rendimiento en tiempo de ejecución puede comprometerse seriamente.

• ARITMÉTICA DEL VECTOR

Los datos que son intrínsecamente organizados en arreglos pueden ser manipulados mucho mas rápidamente y

fácilmente que los mismos datos llenados en una tabla relacional. Por ejemplo, podemos substraer fácilmente el nivel por el real del presupuesto para crear un nivel de variación:

En un servidor datos multidimensional OLAP, esta aritmética del vector puede expresarse en una operación. En el caso de la representación relacional, cada registro en la base de datos tendría que ser accedido, el real substraído del presupuesto, y la variación almacenada en un nuevo campo. Esta operación podría tomar instrucciones de mucho más magnitud de tiempo.

La aritmética del vector permite cálculos consistentemente rápidos de variables deducidas. Por ejemplo, debido a que podemos substraer el 'real' del 'presupuesto' rápidamente para conseguir la 'variación', puede que no exista ninguna necesidad de guardar realmente los números de 'variación' en la base de datos.

• BASES DE DATOS N-DIMENSIONALES

Una base de datos bidimensional es fácil de entender. Ahora extendamos el concepto a tres o más dimensiones. Suponga que tomamos el ejemplo anterior y agregamos los números del presupuesto para cada combinación de producto y región. En la tabla relacional, hacemos esto agregando un nuevo campo llamado Real/Pspto y cada registro en la base de datos se designa como Real o Presupuesto en este campo. En una representación relacional normalizada, agregar este campo duplica el número de registros:

Producto	Región	Act/Bud	Ventas
Tuercas	Este	Actual	50
Tuercas	Oeste	Actual	60
Tuercas	Centro	Actual	100
Tornillos	Este	Actual	40
Tornillos	Oeste	Actual	70
Tornillos	Centro	Actual	80
Pernos	Este	Actual	90
Pernos	Oeste	Actual	120
Pernos	Centro	Actual	140
Lavaderos	Este	Actual	20
Lavaderos	Oeste	Actual	10
Lavaderos	Centro	Actual	30
Tuercas	Este	Budget	50
Tuercas	Oeste	Budget	60
Tuercas	Centro	Budget	100
Tornillos	Este	Budget	40

Tornillos	Oeste	Budget	70
Tornillos	Centro	Budget	80
Pernos	Este	Budget	90
Pernos	Oeste	Budget	120
Pernos	Centro	Budget	140
Lavaderos	Este	Budget	20
Lavaderos	Oeste	Budget	10
Lavaderos	Centro	Budget	30

En la representación multidimensional, simplemente convertimos la matriz bidimensional a una matriz tridimensional:

Esta matriz de 4x3x2 tiene 24 celdas que corresponden a los 24 registros en la representación relacional. Dado que un análisis deseado puede requerir que cualquier combinación de dimensiones sea comparada con otra, se necesita ser capaz de rotar la vista del cubo de datos. En la ilustración anterior, la vista que se está mostrando (en otros términos, encarándolo) es Producto vs Región.

Si rotamos el cubo 90 grados, la cara que se mostrara será Producto vs Real/Pspto. Si rotamos el cubo de nuevo, la cara que se mostrara será Real/Pspto vs Región. Un arreglo tridimensional tiene un total de seis caras, o vistas. Un arreglo cuatri-dimensional tiene doce vistas. Un arreglo n-dimensional tiene $n * (n-1)$ vistas. La capacidad de rotar los datos del cubo es la técnica principal para el reporte multidimensional y a veces se llama cortar y jugar a los dados. (slice and dice).

• LIMITACIONES PRÁCTICAS EN EL TAMAÑO DE LA BASE DE DATOS

Hay un concepto erróneo común en el mercado sobre que el tamaño de la base de datos OLAP está principalmente limitado por el número máximo de dimensiones soportadas. La limitación real, sin embargo, casi siempre es el número de celdas, no el número de dimensiones. Además, no todas las dimensiones se crean igual. Algunos vendedores soportan las jerarquías simples dentro de las dimensiones – otros soportan jerarquías complejas múltiples dentro de las dimensiones. Se detallara mas en la sección titulada Todas las dimensiones no se crean igual. Basta decir que una base de datos ocho-dimensional que usa un producto OLAP puede reducirse a sólo tres o cuatro dimensiones con otro.

En general, como el número de dimensiones aumenta, el número de celdas en la base de datos se incrementa exponencialmente. Por ejemplo, una base de datos bidimensional con 100 Productos y 100 Regiones tendría 10,000 celdas. Si agregamos una tercera dimensión para Tiempo con 52 semanas, tenemos ahora 520,000 celdas. Agregando una cuarta dimensión para Real, Presupuesto, Variación y la Pronostico nos lleva a 2,080,000 celdas. Agregando una quinta dimensión para guardar 10 Tipos de Cliente tenemos el total de 20,800,000. ¡Una base de datos de 16 dimensiones con sólo cinco miembros en cada dimensión tendrían encima de 152 mil millones (152,587,890,625) de celdas!.

La mayoría de los servidores OLAP comerciales acierta el límite de celdas mucho tiempo antes de que ellos corran fuera de dimensiones. Por ejemplo, un servidor OLAP comercial proclama soportar 32 dimensiones, pero tiene un límite de aproximadamente dos mil millones de celdas. Con sólo dos miembros en cada dimensión, una base de datos de 32 dimensiones tendría 232 (o 4.3 mil millones) de celdas. Así, aun cuando

cada dimensión tenga sólo dos miembros, todavía no podría usar todas las 32 dimensiones debido a la limitación de dos mil millones de celdas. En la práctica, la mayoría de las dimensiones (como Productos y Regiones) tienen muchos más de dos miembros.

• Tipo de Datos Serie de Tiempo

El tiempo probablemente es la dimensión más común en las bases de datos OLAP. Casi todos queremos mirar las tendencias – las tendencias de las ventas, tendencias financieras, tendencias del mercado, y tan similares. Los usuarios quieren mirar las tendencias en todos los aspectos de su negocio, comparados con los periodos de los años anteriores, convirtiendo el periodo actual en año a fecha, y así sucesivamente. Una serie de números que representan una variable particular (como las ventas) con el tiempo se llama una serie de tiempo. Por ejemplo, 52 números de las ventas semanales es una serie de tiempo, como lo es 12 meses de números de ganancia, cinco días de balance del dinero en efectivo, etc.

En una celda de la hoja de cálculo, puede guardar simplemente un solo número. Suponga que puede guardar 10 años de historia diaria en cada celda. Esa es la idea de un tipo de datos serie de tiempo.

La suma de un tipo de datos serie de tiempo (ahora disponible en unos servidores OLAP) le permite guardar una cadena entera de números (representada, por ejemplo, por puntos de datos diario, semanal o mensualmente) en cada celda. Si su servidor OLAP tiene un tipo de datos serie de tiempo, puede guardar toda su información histórica en cada celda, en lugar de tener que especificar una dimensión separada para tiempo.

A diferencia de las otras dimensiones, el tiempo tiene muchas reglas y características especiales. En primer lugar, una serie de tiempo siempre tiene una periodicidad particular, significando el intervalo de tiempo entre los números en la serie. Las periodicidades comunes son diarias, semanales, mensuales, trimestral, etc. En segundo lugar, los datos serie de tiempo debe incluir reglas para la conversión a otras periodicidades. Éstos se llaman atributos de la serie de tiempo.

Antes de entrar en detalles sobre el tipo de datos serie de tiempo, veamos cómo tendríamos que tratar con datos serie de tiempo en ausencia de un tipo de datos serie de tiempo especial. La mayoría de los servidores OLAP todavía no tiene los tipos de datos serie de tiempo.

Si no tiene un tipo de datos serie de tiempo, debe definir explícitamente uno de sus dimensiones en la base de datos como tiempo. Debido a que querrá su aritmética de fila y de columna para trabajar correctamente, tendrá que escoger una periodicidad para la base de datos entera (como mensualmente) y expresar todo en esta periodicidad. Los miembros de la dimensión meses se nombrarían explícitamente – como Enero, Febrero, Marzo, etc.

Múltiples periodicidades como diario, semanal, mensual, trimestral u otras periodicidades deben representarse como jerarquías dentro de la dimensión tiempo. Esto requerirá programación y mantenimiento amplios, sobre todo si la organización no tiene una estructura simple del año fiscal. Además de la complejidad agregada, esta representación jerárquica de tiempo aumentará también el tamaño de su base de datos por un orden de magnitud.

Convertir todos los datos a una sola periodicidad puede ser una solución poco satisfactoria, también. En primer lugar, si convierte semanas a meses precedentes para cargar la base de datos, por ejemplo, pierde para siempre la capacidad de ver los datos semanales. En segundo lugar, tales conversiones de los datos son complicadas y agregan un paso extra a la preparación de los datos. En tercer lugar, como nuevos puntos de datos se agregan, más columnas se requerirán en la matriz y eventualmente la matriz se tornará demasiado grande.

La solución al problema de tiempo es no usar tiempo como una dimensión en absoluto, pero usar un tipo de

datos serie de tiempo que le permitirá guardar más de un número en cada celda. Esto permite la conversión inteligente de datos serie de tiempo automáticamente de un tipo a otro por el servidor OLAP.

Una celda de datos serie de tiempo puede contener mucho mas información comparada con una celda numérica simple, o incluso un registro completo en una base de datos relacional. Por ejemplo, la información contenida en la serie de tiempo Ventas podría contener los atributos siguientes:

- Fecha de Inicio = 1/1/94
- Periodicidad = Diario, sólo días comerciales
- Conversión = Suma Total
- Descripción larga = Variable=Ventas, Producto=Tuercas, Region=Este,
- Tipo de Datos = Numérico, simple precisión
- Dispersión = No esparcido
- Calendario = 445 Año Fiscal
- Puntos de Datos = 708,800,821,743,779,856,878,902,799,...

La fecha de inicio es la fecha que corresponde al primer punto de datos.

La periodicidad puede ser intervalos diarios, semanales, mensuales, trimestrales, anuales, de cada hora, 15 minutos, periodos contables 4-4-5, periodo 13, o periodicidad personalizada. El software entiende años calendario y periodos fiscales, tal como año fiscal, las semanas comerciales, etc.

El método de conversión describe cómo convertir, por ejemplo, datos diarios a datos semanales. La suma total de los días para obtener semanas. El último Periodo usaría el valor del último día de la semana (para cosas como balance bancario o dinero en efectivo). Los promedios toman el valor medio durante la semana (frecuentemente usado para convertir cantidades de inventario), etc.

Los tipos de datos pueden ser valores numéricos de simple o doble precisión, cadenas de texto, o fechas.

Los datos esparcidos serían cualquier serie de tiempo dónde el mismo número se repite una y otra vez. Por ejemplo, un precio sólo podría cambiar una vez por año. Definiendo esta serie de tiempo como esparcida causaría que la base de datos guarde sólo las fechas en que el precio cambia y los nuevos valores. El calendario puede ser año fiscal, el año calendario o puede personalizar el periodo.

Los puntos de datos deben poder guardar series de tiempo largas, como 10 años de datos diarios.

Cualquier servidor OLAP que usa un tipo de datos serie de tiempo debe tener una comprensión completa de los calendarios. No sólo debe poder convertir, por ejemplo, semanas en meses, sino que debe poder saber efectuar asignaciones por haber cambiado a otra forma, como convertir meses en semanas. Debe entender la diferencia entre el año calendario y el año fiscal. Debe saber sobre los periodos contables, como los años lunares, periodos contables 4-4-5, etc. Debe saber sobre años bisiestos y festividades. Debe saber asignar meses en semanas, semanas en días, días en horas de trabajo, y muchas reglas necesarias para la exacta conversión de periodicidad. Por ejemplo, convertir datos semanales a datos diarios requiere conocer si está tratándose de una semana de trabajo de cinco días, una semana de trabajo de seis días, etc. Y necesita saber si los días de fin de semana tienen el mismo peso como los días de la semana.

Aunque podría parecer simple a la vista, un algoritmo de conversión de periodicidad completo y robusto requiere un conocimiento incrustado amplio de reglas en un calendario y es una obra de software altamente compleja. Una vez que la tiene, sin embargo, simplifica grandemente todo el desarrollo de las aplicaciones y almacenamiento de datos futuros porque todos los datos pueden guardarse en su periodicidad nativa pero pueden usarse inmediatamente en cualquier otra periodicidad. Esto elimina el tener que construir dimensiones de tiempo desde el principio y escribir rutinas externas para convertir de una periodicidad a otra.

Probablemente es solo el más grande ahorrador de tiempo desde el punto de vista de un diseñador que puede obtener con un producto OLAP multidimensional.

• DATOS ESPARCIDOS

Cuando agregamos dimensiones a una base de datos multidimensional, el número de puntos de datos o celdas crecen rápidamente. Considere, sin embargo, que no vendemos cada producto en cada tienda todos los días. De hecho, nuestras tiendas más pequeñas pueden llevar sólo 20 por ciento de nuestros productos. Para estas tiendas, 80 por ciento de las celdas estarán vacías. En la práctica, tales bases de datos de mercadeo pueden tener más de 95 por ciento de las celdas vacías o las celdas pueden contener valores cero. Es en situaciones dónde menos del 10 por ciento de las celdas tienen cualesquiera datos en ellas, se dice que la base de datos es escasamente poblada o simplemente esparcida.

Otra categoría de datos esparcidos se crea cuando muchas celdas contienen el mismo número. Si la matriz del minorista tuviera una dimensión precio, por ejemplo, el mismo precio podría emplearse a través de todas las 1,200 tiendas. Para cada uno de los 3,000 productos habría 1,200 puntos de datos que contendrían lo mismo. Cuando se tiene en cuenta los días, la situación podría ponerse aun peor. Si no cambia el precio de un producto todos los días, tendría el mismo precio para las tiendas 1,200 veces el número de días que el precio permanece inalterable. En lugar de repitiendo el número una y otra vez en la base de datos, la misma información puede capturarse guardando el número una vez, junto con el número de días que el número se repite secuencialmente.

Una tabla relacional no sabría si un precio permaneció siendo el mismo durante 200 días consecutivos porque no es organizada junto con las dimensiones. Debido a ello, llenaría ciegamente registro tras registro con información duplicada. Esto tiene el efecto de usar el espacio del disco, pero más importante reduce la velocidad de las consultas. Un servidor OLAP que entiende sobre datos esparcido puede saltar encima de ceros, datos perdidos, y cadenas de datos duplicadas.

11. TODAS LAS DIMENSIONES NO SE CREAN IGUAL

Entre los vendedores de servidores OLAP, algunos ofrecen sólo dimensiones muy simples, sin jerarquías o tipos de datos especiales. Otros ofrecen dimensiones muy sofisticadas con jerarquías múltiples, abundantes tipos de datos y una serie de otras capacidades. Asegúrese que sus datos encajarán en las características de su servidor OLAP fácilmente sin requerir mucha programación externa. Esta tarea puede complicarse porque las características técnicas para los servidores OLAP pueden ser confusas.

Por ejemplo, tomemos la especificación para el tamaño de la base de datos. Esto puede referirse al número máximo de celdas en una base de datos o al máximo espacio de disco físico ocupado por la base de datos. Como discutimos previamente en la sección llamada Limitaciones Prácticas en las Dimensiones, el número máximo de celdas en una base de datos es uno de los criterios más importantes al seleccionar un servidor OLAP. Pero incluso esta especificación puede ser confusa. Digamos que hay dos bases de datos comerciales cada una con una restricción de tamaño de 100 mil millones de celdas. Si una base de datos soporta un tipo de datos serie de tiempo y la otra no lo hace, la comparación carece de sentido. Dado que un tipo de datos serie de tiempo puede guardar miles de números en cada celda, una base de datos con un tipo de datos serie de tiempo puede tener una capacidad que es 1,000 veces (o más) mayor que una que no tiene un tipo de datos serie de tiempo.

Hay dos otros problemas que se relacionan al tamaño de la base de datos que los valores mencionados:

- Dependiendo de la velocidad de consolidación, el tamaño de la base de datos puede no ser un factor clave. Una base de datos con capacidad muy grande pero con velocidad de consolidación lenta puede no ser más útil que una base de datos con capacidad limitada.

- Puede haber una gran diferencia entre el número de celdas en una base de datos y el número de celdas que realmente contienen datos. Una base de datos podría tener 100 mil millones de posibles combinaciones (o celdas) de productos, regiones, y así sucesivamente, pero sólo uno por ciento de esas celdas podría contener realmente datos. Esto se llama dispersión. Una base de datos multidimensional OLAP puede tener dos limitaciones del tamaño:

- El número de combinaciones
- El número de combinaciones que contienen los datos (que se resumen a un problema de espacio de disco).

Una advertencia con respecto al tamaño de la base de datos: algunos vendedores intentan superar las limitaciones del tamaño inherentes proporcionando uniones en tiempo de ejecución o consolidaciones entre múltiples tablas. Como en la tecnología relacional, una operación en tiempo de ejecución afecta el compromiso de rendimiento seriamente. El tamaño de la base de datos sólo debe referirse a la capacidad de una sola tabla con un solo índice.

Las dimensiones son más confusas aun. Cualquier intento por igualar el número de dimensiones soportadas por una base de datos con la capacidad de una base de datos es vano debido a las diferentes definiciones que una dimensión debe hacer. Una base de datos ocho dimensional en un producto de un vendedor podría representarse por sólo una o dos dimensiones en otro.

Aquí hay una lista de algunos de las características importantes soportadas por algunos servidores OLAP que pueden reducir la complejidad del diseño de la base de datos y pueden simplificar el desarrollo de aplicaciones de usuario:

- El tipo de datos especial serie de tiempo
- Dimensión especial para las variables
- Jerarquías múltiples dentro de una dimensión
- Clases dentro de una dimensión
- Variables deducidas
- Variables Dimensionadas Independientemente
- Alias (o seudónimos)
- Velocidad de Consolidación

Por ejemplo, una base de datos que no tiene un tipo de datos serie de tiempo podría usar tres dimensiones o niveles diferentes dentro de una dimensión para guardar datos diarios, semanales y mensuales. Si la base de datos incluye un tipo de datos serie de tiempo y la funcionalidad tiempo inteligente (por ejemplo, la capacidad de convertir automáticamente de una periodicidad a otra), esas tres dimensiones no se requerirían.

Algunas bases de datos tienen un número máximo de dimensiones absoluto. Otras tienen un número máximo de dimensiones para cada variable. En otras palabras, las Ventas podrían ser dimensionadas por un juego diferente de dimensiones que el Precio.

Algún software de base de datos requiere que los subconjuntos de miembros de la dimensión, como tamaño del producto, colores, etc., se definan como dimensiones separadas. Otras bases de datos soportan miembros de clases dentro de una dimensión, por ello eliminan la necesidad por dimensiones adicionales. Y como se apunta en la siguiente sección, las jerarquías múltiples dentro de una dimensión pueden eliminar la necesidad de colocar niveles diferentes de detalle en diferentes dimensiones.

12. JERARQUÍAS MÚLTIPLES Y CLASES DENTRO DE LAS DIMENSIONES

El factor más importante al determinar cuántas dimensiones necesitará para una base de datos particular es la

existencia de jerarquías múltiples y clases dentro de las dimensiones. Por ejemplo, una base de datos de ventas de champú podría querer vincular las ventas del producto por tamaño (6 onz., 15 onz. y así sucesivamente), por tipo (cabello seco, cabello graso, cabello normal), y posiblemente por otros atributos, como fragancia/sin fragancia, la marca de fábrica etc. Si su servidor OLAP soporta jerarquías múltiples dentro de una dimensión, todas estas relaciones pueden expresarse con una dimensión. Una jerarquía vincularía las ventas del producto por tamaño, una por tipo, etc. Si no tiene la capacidad de jerarquías múltiples, entonces tendría que tener una dimensión separada para el tamaño, tipo, y así por el estilo, qué complicaría grandemente la conceptualización de la base de datos y multiplicaría el tamaño (el número de combinaciones) de la base de datos muchas veces.

Otro uso común para las jerarquías múltiples está en la dimensión geográfica. Suponga que los datos de las ventas se almacenan por cliente. Los clientes individuales podrían vincularse en ciudades, estados, etc. También podrían vincularse por representante de ventas, distrito de ventas, y región de ventas dónde estos distritos y regiones no tienen nada que hacer con la ciudad o límites del estado.

Sin las jerarquías múltiples, esta base de datos tendría que ser representada con dimensiones separadas para cada vinculación:

Una de las herramientas más poderosas para simplificar una base de datos multidimensional y reducir el número de dimensiones son las clases dentro de una dimensión. Las clases son típicamente los atributos como tamaño, color, las cuentas de la casa y otras características que definen un subconjunto de los miembros de una dimensión.

13. PROFUNDIZANDO EN LOS DATOS RELACIONALES

La mayoría de las organizaciones ha estandarizado las bases de datos relacionales para sus almacenes de datos (data warehouses). Hay casos dónde no es necesario ni deseable replicar todos los datos detallados en una base de datos multidimensional. Por ejemplo, suponga que tenemos una base de datos de ventas en un RDBMS para 50,000 clientes. Y suponga que los 50,000 clientes se acomodan en 500 ciudades, 50 estados, cinco regiones y un gran total.

Obtener un número simple sin consolidar fuera de una base de datos multidimensional no es más rápido que obtenerlo de una base de datos relacional. Por consiguiente, no puede haber ningún punto colocando a los clientes individuales en la base de datos multidimensional. Obtener el Total requeriría la adición de 50,000 registros – evidentemente algo que no queríamos hacer con las bases de datos relacionales. Obtener los totales por región requeriría adicionar aproximadamente 10,000 clientes por región – todavía demasiado para consultar a la tecnología relacional. Cuando profundiza en las ciudades, estará probablemente accediendo aproximadamente a 100 clientes por ciudad – todavía un valor que vale la pena en la base de datos multidimensional. Sin embargo, si quiere ver las ventas para un cliente específico, estará accediendo a sólo un registro, que puede obtenerlo rápidamente de la base de datos relacional como de la base de datos multidimensional. Así que, en esencia, si desea profundizar hasta el fondo de su base de datos multidimensional correctamente dentro de los detalles de la base de datos relacional. Algunos servidores OLAP comerciales soportan esta característica.

La razón por la que esta técnica puede ser útil es que la mayoría del volumen de datos reside a nivel de detalle. En el ejemplo anterior, hay 50,000 registros en la base de datos relacional, pero sólo 556 miembros en la base de datos multidimensional. Así literalmente el 90 por ciento del volumen de los datos permanece en la base de datos relacional, pero todavía obtiene todos los beneficios de la velocidad para las consolidaciones de la base de datos multidimensional.

14. SEGURIDAD Y ROBUSTEZ

La seguridad es un problema importante con cualquier base de datos que es compartida por múltiples

usuarios. La seguridad de la base de datos tiene dos propósitos principales:

- Impedir que los usuarios no autorizados manipulen los datos,
- Control de acceso a porciones de la base de datos sobre un principio de usuario por usuario.

Un sistema de seguridad de base de datos completo es protegido por contraseña y cada usuario tiene un único id de usuario y contraseña. Los usuarios pueden consolidarse y los privilegios de la base de datos pueden controlarse para un grupo entero haciendo la vida más fácil al administrador de la base de datos. Los usuarios individuales o grupos pueden tener acceso limitado a cualquier subconjunto de bases de datos. Por ejemplo, si soy el Gerente de Ventas para la Región Oriental, mi seguridad de base de datos podría restringirse a Seleccionar Región bajo el Este. Para que cuando yo inicie una sesión, mi vista se restrinja sólo a los datos para la Región Oriental.

La robustez cubre un rango de características de la base de datos para la copia de recuperación, recuperación, y mantenimiento de la base de datos. Por ejemplo, si alguien jala el enchufe de su computadora en medio de una carga de la base de datos, ¿su base de datos se adulterará? ¿Puede recuperarse fácilmente de semejante falla?

En general, las transacciones de la base de datos deben poder ser asociadas de tal manera que todo se ejecute con éxito o ningún cambio sea hecho en la base de datos. Si hay una falla de alguna clase, la base de datos debe deshacer cualquier transacción incompleta y debe regresar al estado que estaba antes de ocurrido el fracaso. Si su servidor OLAP tiene robustez real de base de datos, la única cosa que podría causar una corrupción en la base de datos es una falla física del disco.

El mantenimiento automático de la dimensión es muy importante cuando las bases de datos se vuelven grandes. Por ejemplo, si tiene una base de datos con 30,000 clientes, la asignación de esos clientes a los distritos, regiones, o cualquier cosa no debe ser una tarea manual. Si la transacción de la base de datos desde donde el servidor OLAP es cargado contiene el nombre del distrito o región a la que pertenece el cliente (como usualmente es el caso), entonces el servidor OLAP debe poder leer esa información directamente del almacén de datos y automáticamente construir las jerarquías OLAP. Las herramientas de mantenimiento de dimensión manuales que le permiten arrastrar miembros de la dimensión de un lugar de la jerarquía a otro son buenas para mantener bases de datos muy pequeñas. Sin embargo, desplazarse a través de 30,000 miembros en su pantalla de la computadora puede ser muy tedioso por decir lo menos, ya que tales herramientas no son muy usadas para mantener grandes bases de datos.

Para resumir este punto, agregar miembros, eliminar miembros, o reasignar miembros a padres diferentes debe ser un proceso automático, no un proceso manual. La sincronización de la base de datos OLAP con el almacén de datos subyacente debe asegurarse por tales procesos automatizados.

La arquitectura abierta es diferente de cliente/servidor. Mucho software cliente/servidor no es abierto. Ellos pueden tener una pieza cliente y una pieza servidor, pero estas piezas sólo trabajan juntas y ninguna de ellas puede reemplazarse por software de terceros. Un producto cliente/servidor abierto le permite al cliente ser usado con diferentes servidores, y viceversa. Hay muchos beneficios para el usuario de una arquitectura abierta, incluyendo la capacidad para mezclar y hacer corresponder las mejores herramientas de ambos tipos.

15. EL LENGUAJE DE CONSULTA MULTIDIMENSIONAL MDSQL

Así como las bases de datos relacionales tienen un lenguaje estructurado de consultas, las bases de datos multidimensionales requieren un lenguaje que les permita enunciar las consultas multidimensionales. Pilot le ha propuesto su lenguaje MDSQL como la base para un lenguaje estándar en la industria. Mientras varios comités de estándares optarán indudablemente por varias modificaciones antes de establecer un estándar en la industria MDSQL, el MDSQL de Pilot (usado en el Servidor LightShip) proporciona un ejemplo operativo de

un lenguaje de consulta multidimensional totalmente funcional. Como el SQL relacional, este lenguaje de consulta multidimensional es como el Inglés. Aquí hay algunos ejemplos:

Para listar ventas de fondos mutuos sin carga en el Sur para Julio–Diciembre 1994:

- Seleccione la Dimensión Producto 'sin carga'
- Seleccione la Dimensión Región Sur
- Seleccione Ventas
- A través de tiempo navegue por Región, Producto, Variables,
- Liste el periodo Julio 94 – Diciembre 94

Para crear un conjunto de tiempo durante el último mes, el mismo mes el año pasado, y la variación del porcentaje:

- Tiempo Cambio_en_Porcentaje
- Ingresar el último mes como Mes_Actual
- Ingresar el último mes menos 12 como Mes_Año_Previo
- Salida Variacion_Mensual
- $Variacion_Mensual = (Mes_Actual - Mes_Año_Previo) \% Mes_Año_Previo$

Para crear una jerarquía de componentes de computadora:

ENTRADA

'286' '286 CPU Chip', '386' '386 CPU Chip', '486' '486 CPU Chip', VGA 'VGA monitor a color', CGA 'CGA monitor a color'

SALIDA

Chips 'Total Chips Procesador', Monitores 'Total Monitores a Color'

RESULTADO

Total 'Total de Equipo'

NIVEL

Dispositivo, Categoría

Chips = suma '286' a '486'

Monitores = suma VGA a CGA

Total = suma Chips a Monitores

Para reportar las cifras sólo para chips, seleccione el Producto sólo debajo de Chips

CONCLUSION FINAL

La esencia de la tecnología del servidor OLAP es el resumen y análisis de datos rápido y flexible. Mientras las bases de datos SQL van a continuar dominando el procesamiento de transacciones en línea (por la necesidad de un proceso registro por registro), los servidores OLAP son una tecnología superior para aplicaciones

Inteligentes del Negocio. El análisis de los datos eficaz y flexible requiere la capacidad de resumir los datos de múltiples maneras y perspectivas en tiempos cortos. Los servidores OLAP y las bases de datos relacionales pueden trabajar en armonía para crear un ambiente servidor que puede enviar rápidamente los datos a los usuarios y pueda permitirles realizar el análisis necesario para tomar las mejores decisiones de negocio.