Data Warehouse, Modelo, Conceptos e Implementación orientada a SQL Server (página 2)

Enviado por Erith Eduardo P rez Gallardo

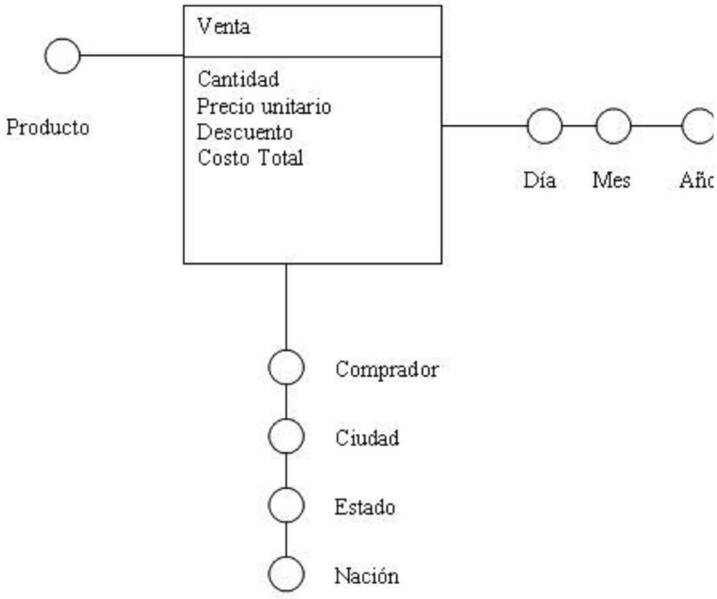
Partes: 1, 2

El Modelo de Hechos Dimensionales (DFM)

El análisis de los datos en un tiempo finito, ha traído consigo estudios sobre la mejor forma de almacenar y repre consultados de una forma más rápida. El uso del Modelo Multidimensional es una de las aproximaciones más ac en estos días. Este se basa en el estudio de los eventos del negocio analizados desde sus distintas dimensiones. Asi

Definición 1: Llamamos evento o **Hecho** a una operación que se realiza en el negocio en un tiempo determinad de decisiones. Se Representan en una caja con su nombre y las medidas que lo caracterizan. (Robert Wrembel &

Ej: Figura 1: Representación gráfica de un Hecho y sus dimensiones



Los **Hechos** están estrechamente relacionados con el tiempo. Los eventos que son estáticos no tiene objetivo de a muy pocos los hechos que no ocurren con determinada periodicidad en un negocio. Los hechos están caracteriza muestra en el ejemplo de la figura 1: la cantidad, el precio unitario, el descuento, etc, son las medidas del Hecho

Nota: Fíjese que el producto que se vende, su costo y la fecha de la venta no son características de esta como lo p relacional. En este caso, esos serían dimensiones de ese Hecho, por las que, puede ser analizado más adelante.

Definición 2: Una **Medida** es una propiedad de un Hecho (casi siempre numérica), que es usada para su análi

Concilia, 2007)

Nota: Un hecho puede no poseer ninguna medida. En ese caso se dice que el Hecho es vacío y solo se usa para cc

Definición3: Una **Dimensión** es una característica de un hecho que permite su análisis posterior, en el proces Wrembel & Christian Concilia, 2007)

Nota: Un hecho debe estar relacionado al menos con una dimensión: "El tiempo".

Es un interés del negocio tomar decisiones sobre los hechos que ocurren en este, pero para esto se necesita su ana semana antes del 14 de Febrero, puede ser un objeto de análisis para un negocio comercial. Para esto se necesita dimensión Tiempo. En este caso en los Días:

7 <= d <= 14. Si se quisiera saber que productos fueron los más vendidos en esos días entonces tendríamos que a análisis, Producto. Así adicionando dimensiones a nuestro estudio se pudieran llegar a conclusiones sobre si el si comprarse más objetos de un producto o menos de otro. Elemento este muy importante para la futura estrategia

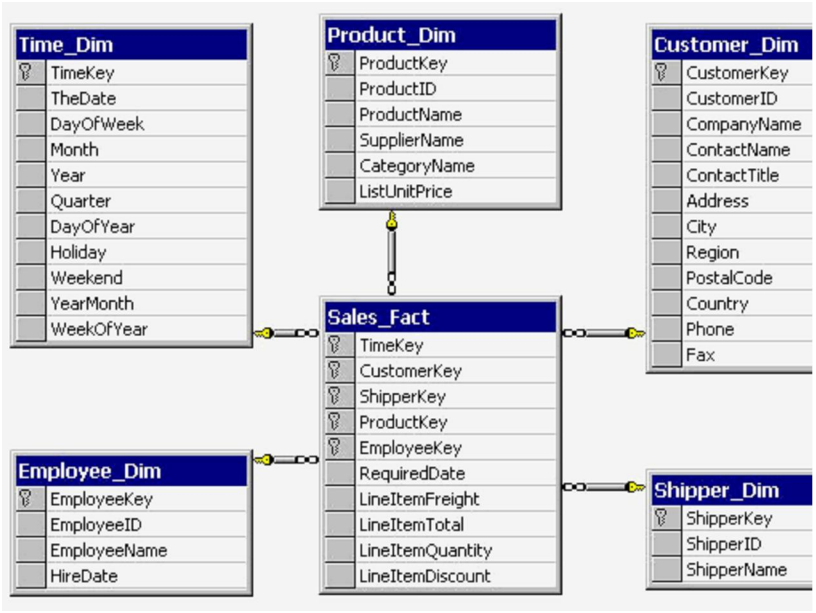
Definición 4: En una empresa pueden existir varios hechos que sean analizados por dimensiones iguales. En es Dimensiones Compartidas. (Robert Wrembel & Christian Concilia, 2007). Un ejemplo de esto es El Hecho Venta dimensiones Tiempo y Producto. Lo mismo ocurre con el Hecho Compras.

Las dimensiones deben ser atómicas y las relaciones entre estas crean jerarquías que permiten un análisis jerárq el Tiempo, que es dividido en tres dimensiones. Día, Mes y Año. Cada uno es una dimensión distinta, pero relacio de 1 a muchos, que permite el análisis del Hecho, por días, meses o años, o la combinación de ellos. Esto da al tra manejan el Tiempo como una propiedad de una entidad, y lo tratan como un todo. Por lo que, como podemos inf convertir las bases de datos de estos sistemas a la nueva filosofía. SQL Server tiene facilidades para esto llamada que permite leer datos desde cualquier SGBDR que posea un driver ODBC o implemente la nueva tecnología OL

Diagrama en Estrella

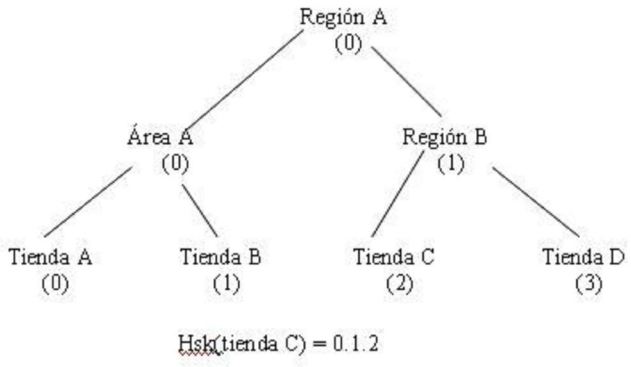
Uno de los tipos de consultas más usadas en las OLAP es la llamada Estrella. Su nombre lo adquiere debido a qu relacional (MOLAP Multidimentional Online Analitical Processing) está dado por varias tablas que almacenan la que contiene el hecho con una relación 1:m con estas tablas de dimensiones. Veamos un ejemplo gráfico:

Figura 2: Diagrama en estrella del Hecho, Ventas. (Microsoft Data Warehouse Training Kit, 2000)



Como podemos ver en la figura las tablas de dimensiones están ligadas a la tabla Hecho, por relaciones. La integ creación de llaves foráneas en la tabla Hecho, que a su ves forman parte de la llave principal de la esta tabla. Es i completas son guardadas en una sola tabla dimensión. Este es el formato no normalizado, existe otro formato qu dimensión. Ejemplo (Time_Dim). Cada tabla dimensión tiene su propia llave que es mantenida por el sistema Da llama "Surrogate Key". Las llaves Surrogate Jerárquicas, no son más que una codificación de cada elemento de l dimensión. Veamos la figura 3 de cómo se logran estas llaves.

Figura 3: Formación de una llave Surrogate Jerárquica (Robert Wrembel & Christian Concilia, 2007)



Vamos a ver ahora como sería una consulta sobre este tipo de diagrama en estrella:

Figura 3: Plantilla de consulta para una consulta en estrella (ad hoc star query) (Robert Wrembel & Christian Co

```
SELECT    <grouping attributes and/or aggregation functio
FROM      <fact table>, D1, D2, …, Dk
WHERE     <star join conditions: equalities on key-f.key> A
          LP1 AND LP2 AND … AND LPk AND
          <restrictions on attributes of the fact table>
GROUP BY  <grouping attributes>
HAVING    <group selection predicate>
ORDER BY  <sorting attributes>
```

Nota: En la figura las D1, D2, .. , Dk significan tablas de dimensión y los LP1, LP2, ...,LPk son los predicados usa
ejemplo siguiente muestra mejor como sería esta consulta: (Robert Wrembel & Christian Concilia, 2007)

```
SELECT L.area, D.month, SUM(F.sales)
FROM SALES_FACT F, LOCATION L, DATE D, PRODU
WHERE F.day = D.day AND F.store_id = L.store_id AND
        F.product_id = P.item_id AND D.year = 1999 A
L.population>1000000 AND P.category = “air condition”
GROUP BY L.area, D.month
```

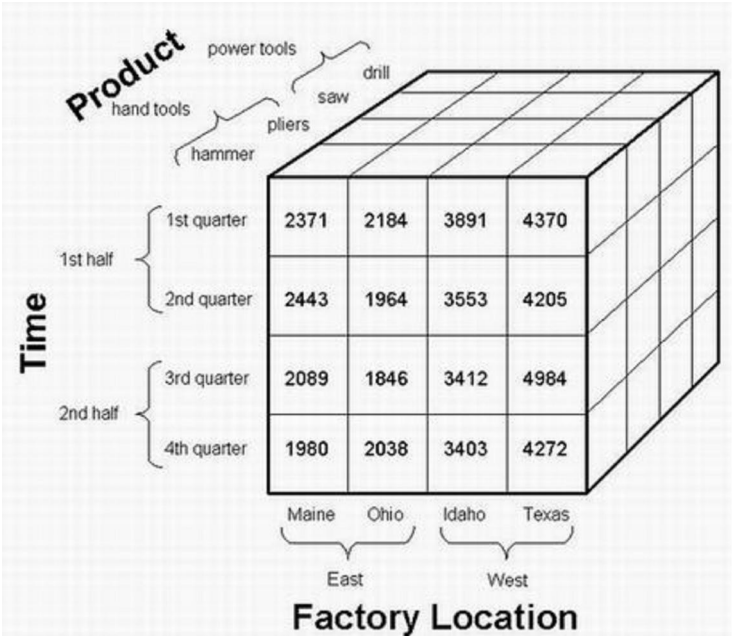
En este tipo de procesamiento el mayor de los problemas es el super join que se crea al procesar las tablas de dim
Hecho, para esto se han hecho varios estudios sobre la mejor forma de hacer este tipo de consultas de forma que
las técnicas mejores probadas es la de reescribir la consulta como lo muestra el siguiente ejemplo que mostramo

Ejemplo: Optimizar la consulta en el Data Warehouse (Robert Wrembel & Christian Concilia, 2007)

```
SELECT dim2.dim2_attr, dim3.dim3_attr, dim5.dim5_attr, fact.fact1
FROM fact, dim2, dim3, dim5
WHERE fact.dim2_key = dim2.dim2_key /* joins */
AND fact.dim3_key = dim3.dim3_key
AND fact.dim5_key = dim5.dim5_key
AND dim2.dim2_attr IN ('c','d') /* dimension restrictions */
AND dim3.dim3_attr IN ('e','f')
AND dim5.dim5_attr IN ('l','m')
is rewritten in the following form:
SELECT … FROM fact
WHERE fact.dim2_key IN (SELECT dim2.dim2_key FROM dim2 WHERE dim2.dir
('c','d'))
AND fact.dim3_key IN (SELECT dim3.dim3_key FROM dim3 WHERE dim3.din
('e','f'))
AND fact.dim5_key IN (SELECT dim5.dim5_key FROM dim5 WHERE AND dim5
('l','m'))
```

Para cerrar con broche dorado este tema es necesario hacer alusión a los llamados Cubos de datos: Estos no son
las tablas Dimensión y la tabla Hecho que al final dan una vista en forma de Cubo cuyas celdas están compuestas
Esta es la base de las aplicaciones OLAP. El cubo de datos es lo que hace que los reportes sean obtenidos con un l
análisis de los datos pueda ser tan diverso, pues cada cara del cubo se refiere a un análisis distinto de las medidas
gráfico del cubo:

Figura 4: Cubo de datos (Microsoft Books Online, 2000)



Como podemos ver en el ejemplo la cantidad de producción puede ser analizada por producto, teniendo en cuent por Localización de las Industrias o en su conjunto por todas ellas a la vez o cualquier combinación de estas. Esto una amplia gama de posibilidades de las cuales puede tomar ventaja. En nuestro caso de estudio de las ventas. E Sales_Fac en conjunto con las restantes tablas de Dimensión nos permite analizar las ventas por Empleado, por

Extracción y Transformación de los Datos

SQL puede conectarse a cualquiera de los formatos creados por Microsoft Office, archivos texto y a bases de dato usadas por múltiples sistemas de escritorio que resuelven problemas importantes en muchas empresas. Además (Sistema Gestor de Bases de Datos) que posea un driver ODBC compatible con Microsoft o implemente la tecnol páginas de Excel con datos resumidos o exportarlas a este, para que los usuarios más avanzados puedan crear gr SQL en la herramienta perfecta para crear sistemas Data Warehouse. Para realizar estas tareas Microsoft cuenta administración del SQL Enterprise Manager, que contiene un área de diseño con los elementos que se necesitan transformación de datos (DTS Package).

Para crear un paquete solo se hace clic derecho sobre el área vacía de esa rama y se selecciona la opción New Pac ventana con un área de diseño que permite empezar a diseñar el paquete en cuestión. Hay tres elementos fundam paquete:

1. Las conexiones: que permiten conectarse a cualquier fuente de datos como las que relacionamos anterior
2. Las tareas: que permiten transformar los datos de cada conexión antes de que sean copiados a otra conexi de ActiveX, pueden transferir otros archivos desde o hacia un sitio FTP, pueden enviar un mensaje a un op datos, crear nuevas tablas en la BD resultante, etc.
3. Los flujos de trabajo: que permiten definir hacia donde irán los datos luego de que se le apliquen las respe la nueva forma deseada.

Así usando estos tres elementos fundamentales se crea una especie de script gráfico que sigue una secuencia lógi transformación de los datos almacenados de un formato a otro como SQL Server desde donde podrán ser analiza

Aunque desde el Enterprise Manager de SQL Server se puede ejecutar los paquetes que se crean haciendo clic de Execute o en un Schedule. Existen varias herramientas adicionales que pueden usarse desde la ventana de coma más interesante para el uso de scripts es dtsrun. Para obtener la ayuda completa de esta podemos ejecutar dtsru con SQL Server instalado.

Ejemplo: Uso de la herramienta dtsrun desde un script de windows

```
@echo off

Copy \\srvaplics\aplics\*.mdb d:\convertsql /Y

if (%ERRORLEVEL% NEQ 0) echo "Error al copiar la base de datos"

else dtsrun /S SASSQL /U sa /P Pepe2006 /N loadaccessdb
```

Como podemos ver a dtsrun se le pasan como parámetros el nombre del servidor el usuario con derechos para ej el nombre del paquete a ejecutar. Existen otra serie de opciones que pueden ser consultadas en la ayuda de la he puede estar relacionado como vemos en el ejemplo con la copia de bases de datos hacia lugares desde donde es

A continuación mostraremos un ejemplo práctico de cómo diseñar un paquete para obtener el balance de compr exportado por el sistema contable SENTAI.

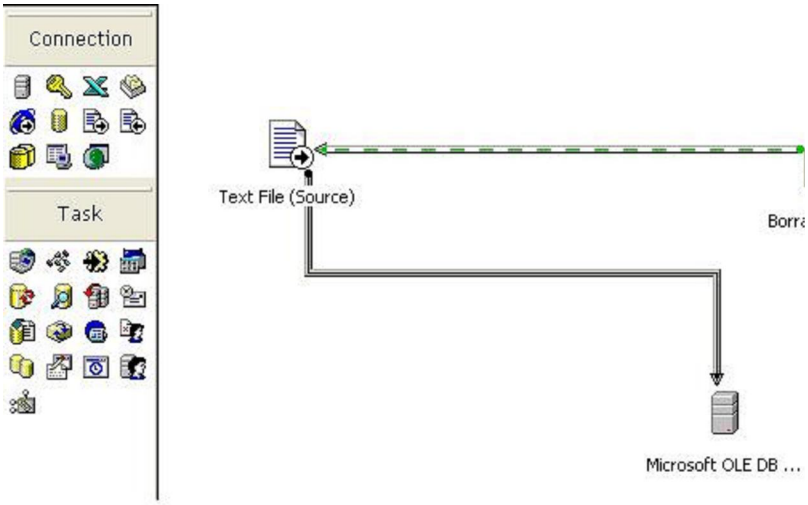
Ejemplo: Obtención de Balance de SENTAI

1. Creación de un nuevo paquete desde el SQL Enterprise Manager
2. Arrastrar hacia el área de diseño la conexión hacia un archivo TXT

3. Configurar la conexión especificando el archivo y su formato
4. Arrastrar hacia el área de diseño una conexión al servidor SQL al que se va a importar el estado de las cue
5. Configura esta conexión especificando la base de datos a usar
6. Hacer clic sobre la conexión al archivo texto para seleccionarlo.
7. Hacer clic sobre la acción (trasform data task) y seguido se hace clic sobre la conexión al Server SQL del á
8. Hacemos clic derecho sobre la línea que une a ambos ahora y editamos las propiedades de la transformaci acuerdo a nuestros objetivos.
9. Presionamos el botón de salvar para guardar el paquete.
10. Escriba el nombre del paquete y cierre la ventana de diseño.

Ahora podrá ver el nombre del paquete y si desea ejecutarlo solo debe hacer clic derecho sobre este y seleccionar serán insertados en la nueva base de datos de contabilidad de la empresa.

Ejemplo: Vista gráfica del paquete creado



Técnicas de desarrollo del soporte de Hardware

Antes de definir una estrategia en la cual se desarrollará el hardware que soporta el Data Warehouse, es necesari pueden estar detenidas las aplicaciones que usan este. En nuestros días muchas empresas necesitan que estos sis año, por lo que no hay mucho margen de error. La desconexión de estas aplicaciones por solo una hora puede ha usuarios que nunca más visitarán el sitio Web pensando que está fuera de servicio o porque se buscaron otro pro acostumbrado, para que esto no ocurra antes de echar a andar un sistema de tanta importancia hay que prever c soporte.

Una de las técnicas clásicas para desarrollar hardware siempre ha sido aumentar memoria, capacidad de proces almacenamiento, al servidor que soporta el sistema. Esta técnica es comúnmente llamada "**Scalling Up**". En es subir su capacidad de procesamiento (Adicionar otro Procesador o cambiar el existente por uno más rápido) en s 500; o HP Proliant ML350 y ML370; tiene que apagar el sistema para hacerlo, al igual que si desea aumentar la que muchos de estos servidores pueden durar años en fallar y por tanto otra de las situaciones muy comunes es q necesita aumentar sus capacidades técnicas es muy difícil encontrar piezas compatibles en el mercado por el con hardware, por esto es muy necesario comprar piezas con vista a estas situaciones a tiempo, cosa esta muy difícil funcionando correctamente. Muchos de los directivos de estas grandes empresas no ven con muy buenos ojos la todavía no sueña con fallar o que resuelve los problemas en la actualidad con una buena calidad de respuesta cor para esto es una gran responsabilidad del Administrador de Sistemas, gestión que el futuro cuando ocurran los p reconocimiento por parte de la directiva de la empresa. En el caso de la capacidad de almacenamiento, si tenem hace falta apagar el sistema. Estos se pueden conectar en caliente sin necesidad de formatear siquiera, el hardwa

A pesar de que los discos SCSI siguen siendo muy fiables y rápidos. En estos días alternativas muy buenas puede Sistemas tome otra decisión. Existe en el mercado tarjetas RAID de discos Serial ATA, que permiten hacer arreg facilidad. Estos discos son menos fiables que los SCSI, pero tiene un menor precio y su relación precio/capacidad un ejemplo: Un disco de 72 GB SCSI, la empresa Tecun lo comercializa en Cuba con un costo de: $370.00, mient costar $75.00. Si decimos que el disco solo durará un año de explotación lo que es muy raro en ambientes climat protegidos por baterías. Al año siguiente se podrá comprar un disco Serial ATA por el mismo precio, pero de 160 anterior. Si sumamos los costos de los dos discos en los dos años, no llegará a ser igual al costo del SCSI, que por problemas otros 4 años. Sin embargo habrás aumentado tu capacidad de almacenamiento al doble cosa esta que que aumentan su colección de datos exponencialmente, como es el caso de un Data Warehouse.

Es importante destacar que aunque hablo de un solo disco estoy asumiendo que al menos existe otro que está cor usado en caso de que falle el principal. No es concebible un sistema tan importante que no tenga este tipo de resp haber respaldo, lo que incrementa los costos.

En caso de que no se posea hardware RAID se pueden usar las facilidades del Sistema Operativo Windows 2k3 q administrador de discos. Este tipo de arreglos a pesar de ser menos rápido que el manejado vía hardware. Es una

Ejemplo de crear una partición espejo para el sistema:

1. Abra el administrador de discos luego de haber instalado físicamente los dos discos.

2. Haga clic derecho sobre el cuadro que dice disco 0 y haga clic en convertir a disco dinámico.
3. Le aparece una ventana que le indica marcar los discos que desea convertir. Marque los dos discos el 0 y e
4. El sistema le preguntara si desea continuar con la operación a pesar que el disco 0 tiene el sistema operati va areiniciar para completar la operación. Acepte y espere a que el sistema se reinice.
5. Abra el administrador de discos y ya con los discos siendo dinámicos haga clic derecho sobre la partición c adicionar espejo.
6. En la ventana que aparece seleccione el disco 1 para usar como espejo y acepte.
7. El sistema creará una partición idéntica en el segundo disco y empezará a duplicar la información existent

Nota: Para que un disco pueda ser espejo total del primero deben ser de capacidades idénticas. No puede haber usar discos de varias marcar como Ejemplo: Maxtor y Segate en el mismo sistema.

A pesar de que los servidores profesionales son muy fiables, en empresas medianas y pequeñas usarlos, puede im Una alternativa puede ser usar una PC Moderna con una memoria bastante grande 4 GB pudriera ser suficiente. "Pentium D" que poseen procesamiento en paralelo y una velocidad de más de 3 GHz que lo hacen muy compete característica importante y es que este tipo de configuración puede hacer uso de otra técnica de desarrollo llama hace falta mejorar la capacidad de este servidor porque el negocio se ha desarrollado y ahora se procesa una may otro servidor o varios iguales o mejores en dependencia de la disponibilidad del mercado y del presupuesto perm distribuyendo así la carga de trabajo. Windows 2k3 permite la configuración de este tipo de soporte con gran faci aumentado en cantidad de servidores y mejorando en prestaciones, capacidad de procesamiento y tolerancia a fa días este tipo de configuraciones son mucho más fiables y económicas. El uso de varios servidores en forma de cl característica de SQL Server 2000 que permite la actualización de los datos de todos estos servidores según las n de SQL Server 2000 no solo se puede aplicar sobre servidores configurados como cluster también puede usarse p un subconjunto del Data Warehouse a un servidor de una entidad de la empresa. Estaríamos hablando de un pos clásico de empresa comercial estaríamos hablando de replicar el Data Mart de las ventas de Ciego, al servidor de podrían sacar reportes muy provechosos por parte de los comerciales sin necesidad de congestionar la Red WAN Teniendo en cuenta que los enlaces WAN entre provincias son bastante costosos y lentos en comparación con los

Datos distribuidos

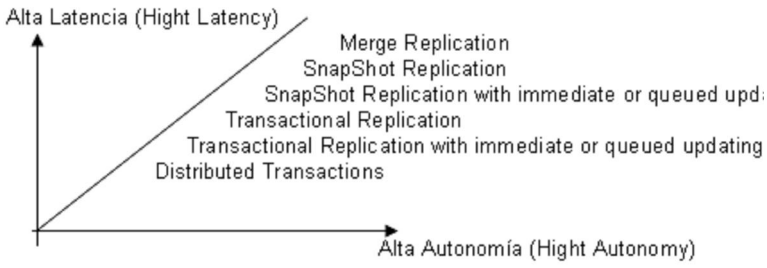
La distribución de los Datos a través de varios servidores es toda una teoría de la que nosotros solo vamos a menc haciendo énfasis en la replicación de SQL Server. Para comenzar debemos decir que existen dos técnicas fundam

1. Transacciones Distribuidas
2. Replicación

La primera necesita que todos los servidores involucrados estén funcionando correctamente y en red (online). Es actualizadas todas las bases de datos involucradas en todo momento. Todos los servidores tienen que estar sincro transacción falla completa. Teniendo que revertir el proceso en todos los servidores con su correspondiente consu su parte puede mantener un margen de tiempo que permite a los servidores no tener que estar sincronizados en t uso de una u otra técnica está dado principalmente por dos factores: Latencia entre servidores y Autonomía de ca por ejemplo: Si el Data Mart ubicado en Ciego es actualizado a partir de la información recolectada del sistema d comerciales de esta provincia. Entonces este Servidor posee una autonomía. Estos datos no tienen por qué ser rep Santiago de Cuba o Matanzas. En este caso esa información solo sería necesaria en El servidor central de la Empr Warehouse central. Así mismo estos pueden ser replicados en el horario nocturno evitando que los enlaces de cor día que hay un mayor tráfico de correo, Messenger, sistemas contables y otros. Es decir en nuestro ejemplo la rep implementación para poder mantener los datos distribuidos con autonomía a través de todas las sucursales del p

Estas dos técnicas de trabajo poseen una serie de variantes para su implementación que mostramos en forma de

Figura 5: Formas de implementación de datos distribuidos (Microsoft Training, 2000)



SQL Server 2000 tiene tres formas básicas para implementar la distribución de datos:

1. Snapshot Replication
2. Transactional Replication
3. Merge Replication

SanapShot Replication: Replica todos los datos en la publicación hacia el o los subscriptores. En el tiempo especi es muy usado en ambientes con una gran autonomía y donde se necesitan los datos publicados con una latencia r

Transactional Replication: En este tipo de replicación solo se replican los cambios incrementales. Este tipo de rep la latencia de actualización debe ser baja, puede ser muy útil además en ambientes conectados por enlaces de po los datos replicados es muy bajo. El agente que permite realizar este trabajo tiene además la opción de replicar lo

Merge Replication: En este tipo de replicación se analizan los artículos publicados y se crea uno solo con la inforr estos en cada servidor. Al final cada servidor se actualiza con el artículo resultante. En este tipo de replicación pu mismas bases de datos con un nivel de autonomía alto. Estas colisiones pueden ser resultas programando reglas

SQL Server usa una filosofía llamada Publicador-Distribuidor-Subscriptor. Análogamente a la vida real el Public que no son más que objetos de una base datos como: Tablas, Vistas, Procedimientos almacenados; o Porciones d tabla determinada o una selección de una o varias tablas relacionadas y a partir de esa publicación es que un sub las actualizaciones de esta publicación que le llega a través del distribuidor. Tenga en cuenta que el distribuidor p publica o puede estar en otro diferente para eliminar carga al publicador y buscar un balance en el procesamient tres tipos básicos de implementación física de los servidores:

1. Publicador Central y Muchos Suscriptores
2. Muchos Publicadores y un solo suscriptor
3. Muchos Publicadores y muchos suscriptores

En nuestro caso particular necesitaríamos publicadores en todas las sucursales del país y un subscriptor central transaccional (Tansactional Replication) que se ejecutara en el horario de la noche. Es importante destacar que l Lotus Notes. Implementan este sistema de bases de datos distribuidas con replicación con muy buena calidad, pr banda en los enlaces WAN lo que las hace muy competitivas en el mundo empresarial.

Consultas distribuidas

Existen dos formas fundamentales de hacer una consulta distribuida usando SQL Server. La primera consiste en OPENDATASOURCE, que permite ejecutar una consulta sobre cualquier SGBDR que soporte OLE DB o tenga u Microsoft. Este tipo de procesamiento es llamado en la literatura como: "ad hoc query". La otra vía es usando ser soporten también la tecnología OLE DB de Microsoft. Esta última es usada para hacer consultas frecuentes a dif consultas con períodos de tiempo largos. (Microsoft Training, 2000)

Sintaxis de OPENROWSET:

OPENROWSET('*provider_name*'

{'*data -source*'; '*user_id*' ; '*password*' | '*provider_string*'},

{[*catalog*.][*schema*.]*object* | '*query*'})

Descripción de Parámetros:

Provider_name: Nombre del proveedor OLEDB. Ejemplo MSDASQL(Para ODBC)

Data-source: Fuente de donde se obtendrán los datos. Como veremos más adelante no es más qu contiene la BD.

User_id: nombre de usuario de acceso a la BD

Passord: Contraseña de acceso a la BD

Provider_string: En caso de ser necesaria una cadena de conección a la fuente de datos.

Catalog: No es más que el nombre de la base de datos.

Schema: Nombre del usuario dueño de los objetos de la BD. Ejemplo en SQL dbo.

Object: Tabla, procediemiento, función, u otro objeto de la base de datos.

Query: Consulta en cuestión.

Para poder entender mejor la sintaxis vamos a ver un ejemplo de cómo podemos hacer una consulta sobre un ser sobre una base de datos Access y uno sobre como integrar una consulta de este tipo en una normal. (Microsoft Tr

Ejemplo 1: Encuesta a un Servidor SQL

SELECT a.*

FROM OPENROWSET('SQLOLEDB', 'LONDON1';

'newcustomer';'mypassword',

'SELECT ProductID, UnitPrice

FROM Northwind.dbo.Products ORDER BY UnitPrice')

AS a

Ejemplo 2: Encuesta a un servidor Aceess

SELECT a.*

FROM OPENROWSET('Microsoft.Jet.OLEDB.4.0'

'C:\MSOffice\Access\Samples\Northwind.mdb';

'newcustomer'; 'mypassword',

Orders)

AS a

Ejemplo 3:

USE Northwind

SELECT cust.* ord.*

FROM Customers as cust JOIN

OPENROWSET('Microsoft.Jet.OLEDB.4.0'

'C:\MSOffice\Access\Samples\Northwind.mdb";

'newcustomer'; 'mypassword',

Orders)

AS ord

On cust.customerid = ord.customerid

Como vemos este tipo de consultas puede ayudarnos mucho a la hora de extraer datos desde los sistemas que nut hemos visto que SQL tiene servicios (DTS) y otras herramientas gráficas que pueden extraer datos y transformar permite controlar el trabajo usando sentencias de programación que fácilmente pueden ser combinadas con otra profesionales. Otra de las ventajas que nos ofrece esta filosofía de trabajo es que no tenemos que importar los dat ocasiones en que no queremos importar esa base de datos porque solo se consulta una sola vez al año o por otras podemos consultarla sin mucho esfuerzo.

Existen varios tipos de proveedores de OLE DB que pueden ser consultados en la ayuda de SQL. Aquí queremos (Microsoft Training, 2000)

1. SQL Server: N'SQLOLEDB'
3. Microsoft OLE DB Provider for Access (Jet): 'Microsoft.Jet.OLEDB.4.0'
4. Microsoft OLE DB Provider for Oracle: 'MSDAORA' *data_source* is the SQL*Net alias name for the Oracl server
5. OLE DB Provider for ODBC (Using *data_source* parameter): *provider_name* is 'MSDASQL' *data_source*
6. OLE DB Provider for ODBC (Using *provider_string* parameter): p*rovider_name* is 'MSDASQL' *provider* SERVER=*servername* UID=*login*;PWD=*password*;'

Nota: Para hacer consultas distribuidas es necesario que estén activadas las siguientes opciones. ANSI_NULLS Query Analyzer ya están activadas por defecto, pero en otros como osql no.

Por otra parte el uso de servidores en lazados permite que las consultas sean ejecutadas con el previo conocimien hacer esta. Este tipo de consultas es usada cuando su periodo de ejecución es relativamente bajo. En el caso nues hablando de una consulta hecha para obtener datos de las ventas de los Data Mart de cada servidor SQL de cada empresa podría estar enlazado con los restantes servidores y desde este se podrían crear este tipo de consultas o t veremos más adelante.

Para enlazar un servidor con otro se usa el procedimiento almacenado **sp_addlinkedserver**. Vamos a ver su u

Ejemplo 4: Uso de sp_addlinkedserver (Microsoft Training, 2000)

-- USando SQL Server

EXEC sp_addlinkedserver

@server = 'AccountingServer',

@svrproduct = 'SQL Server'

-- Usando Oracle

EXEC sp_addlinkedserver

@server = 'OracleFinance',

@svrproduct = 'Oracle',

@provider = 'MSDAORA',

@datasrc = 'OracleDB'

Sintaxis de ejecución del procedimiento: (Microsoft Training, 2000)

sp_addlinkedserver [**@server =**] '*server*'

[**,** [**@srvproduct =**] '*product_name*']

[**,** [**@provider =**] '*provider_name*']

[**,** [**@datasrc =**] '*data_source*']

[**,** [**@location =**] '*location*']

[**,** [**@provstr =**] '*provider_string*']

[**,** [**@catalog =**] '*catalog*']

@server Name of the linked server to create

@svrproduct Product name of the OLE DB data source

@provider The unique, friendly name for the OLE DB provider corresponding to

this data source

@datasrc Name of the data source as interpreted by the OLE DB provider

@location Location of the database as interpreted by the OLE DB provider

@server Name of the linked server to create

@svrproduct Product name of the OLE DB data source

@provider The unique, friendly name for the OLE DB provider corresponding to

this data source

@datasrc Name of the data source as interpreted by the OLE DB provider

@location Location of the database as interpreted by the OLE DB provider

En el caso de servidores como SQL Server que necesitan de un usuario y una contraseña para poder conectarse a dos condiciones:

1. El usuario con que se ejecuta la consulta esté definido en los dos servidores y tenga derechos para ejecutar
2. Se ejecute previamente un mapeo de usuarios usando el procediemiento: **sp_addlinkedsrvlogin.**

Sintaxis del procedimiento sp_addlinkedsrvlogin: (Microsoft Training, 2000)

sp_addlinkedsrvlogin [**@rmtsrvname** =] '*rmtsrvname*'

[, [**@useself** =] '*useself*']

[, [**@locallogin** =] '*locallogin*']

[, [**@rmtuser** =] '*rmtuser*']

[, [**@rmtpassword** =] '*rmtpassword*'

@rmtsrvname Name of a linked server to which the login mapping applies.

@useself Determines whether SQL Server login accounts use their own

credentials or the values of the **@rmtuser** and **@rmtpassword**

arguments to connect to the server specified by the **@rmtsrvname**

argument. A value of TRUE for **@useself** is invalid for a Windows

Authenticated login account.

@locallogin An optional login account on the local server. If used, **@locallogin**

must already exist on the local server. If this value is null, then all

login accounts on the local SQL Server will be mapped to the

account on the remote server specified by **@rmtuser**.

@rmtuser The optional user name for connection to **@rmtsrvname** when

@useself is FALSE.

@rmtpassword The optional password associated with **@rmtuser**.

Ejemplo 5: Uso de mapeo de usarios: (Microsoft Training, 2000)

EXEC sp_addlinkedsrvlogin

@rmtsrvname = 'AccountingServer',

@useself = 'false',

@locallogin = 'Accountwriter',

@rmtuser = 'rmtAccountWriter',

@rmtpassword = 'financepass'

EXEC sp_addlinkedsrvlogin

@rmtsrvname = 'AccountingServer',

@useself = 'false',

@rmtuser= 'allcustomers'

Como podemos ver aquí el procedimiento enlaza el usuario que ejecuta la consulta localmente con el que tiene lo servidor remoto. Así cuando se ejecute la consulta, el servidor usará el usuario adecuado para ejecutar la parte q credenciales diferentes. En nuestro caso de estudio, el servidor de la Sucursal Ciego no tiene que tener la misma restantes servidores de las demás sucursales, para que nuestro Data Warehouse funcione. Así en el servidor cent

servidor de cada sucursal. Esto aumenta la seguridad, pues si un hacker se hace de una de las contraseñas de alg no podrá acceder a las restantes ni al servidor central. Incluso sin pensar en hacker. En el caso de que algún adm una sucursal diferente a la que administra esto le sería imposible debido a esta técnica, lo que daría la posibilida de la empresa tuvieran acceso a reportes que incluyan varios territorios.

Hasta aquí ya sabemos como enlazar los servidores, pero no sabemos como hacer la consulta sobre estos. La úni distribuida sobre servidores enlazados, de una consulta sobre un solo servidor es que la primera tiene que usar n nombrado de cuatro partes completo incluye además de los elementos que ya estamos acostumbrados a usar cor del servidor que se va a encuestar. Así, si el Server de Ciego se llama sassql el nombre de cuatro partes completo sassql.datamartventas.dbo.ventasfact. Ahora mostraremos un ejemplo de cómo se aplica esto en la consulta.

Ejemplo 6: Consulta sobre servidores enlazados (Microsoft Training, 2000)

```
SELECT CompanyName, Phone

INTO PhoneList

FROM AccountingServer.NorthwindRemote.dbo.Suppliers

SELECT CompanyName

FROM AccountingServer.NorthwindRemote.dbo.Suppliers

SELECT ProductName, CompanyName

FROM Products p JOIN

AccountingServer.NorthwindRemote.dbo.Suppliers

ON p.supplierid = s.supplierid
```

Existen algunas restricciones a la hora de hacer consultas sobre servidores distribuidos. Estas son:

1. No se pueden usar sentencias create, alter, drop.
2. No sepuede usar goup by cuando las tablas contienen objetos largos como TEXT, NTEXT, etc.
3. No se pueden usar las sentencias: readtext, writetext, updatetext

Otra de las cosas más importantes que posee esta técnica es que normalmente las consultas son ejecutadas en el modificado para que se hagan en el servidor remoto. Esto puede mejorar en gran medida el rendimiento del serv hace en el otro extremo. Imagine que necesita una consulta sobre las ventas de todas las sucursales. Esta puede s tiempo, dado la cantidad de datos a procesar. Si se divide el trabajo entre los servidores de cada sucursal el tiemp medida, para usar esta técnica se hace uso de la sentencia openquery, que veremos con un ejemplo: (Microsoft T

Sintaxis:

OPENQUERY (*linked_server*, '*query*')

Ejemplo 7: Uso de la sentencia OPENQUERY

```
SELECT * FROM OPENQUERY

(AsiaServer, 'SELECT ProductID, Royalty

FROM Northwind.dbo.ProductInfo')
```

Como podemos ver solo debemos especificar el servidor que va a ejecutar la consulta y luego la consulta en cuesti retornará los resultados:

Existen dos elementos más de alta importancia a la hora de trabajar con servidores enlazados. El primero de ello almacenados en servidores remotos. Para esto se usa la misma forma de nombrado de cuatro partes ejemplo: sassql.datamartventas.dbo.sp_listarVetas. El otro elemento importante es el uso de transacciones distribuidas. E permite hacer tareas que necesitan ejecutarse como un todo en servidores distribuidos como una única transacci la sentencia BEGIN DITRIBUTED TRANSACTION <NAME>, vamos a ver esto con un ejemplo:

Ejemplo 8: Uso de transacciones distribuidas

```
SET XACT_ABORT ON

BEGIN DISTRIBUTED TRANSACTION

EXEC Savingsdb.dbo.withdraw 1234, 100

EXEC Centralserver.Checkingdb.dbo.deposit 1234, 100

COMMIT TRAN
```

Nota: Recuerde que un rollback de la transacción puede llevar este proceso a cabo en todos los servidores involu rendimiento de los mismos. Además si no está la primera sentencia del ejemplo no se hace un rollback completo. activar esa opción (XACT_ABORT) para en última instancia hacer un rollback completo y que la BD no quede in

Salva y restaura de datos

Una de las cosas más importantes cuando usted ya posee un Data Warehouse es mantenerlo y para esto salvar su Para ilustrarlos con un ejemplo les puedo decir que el Data Warehouse que administro generaba todos los días 8 datos se hace engorroso computacionalmente y también a la hora guardar estos datos en un lugar con espacio di

pueden almacenar hasta 80 GB en cinta el proceso de salvar un casete de este tipo diario puede ocupar gran part horario nocturno y requiere de 365 casetes para todo el año lo que es un gasto considerable. Si por el contrario de más barata, necesitaría nada más que la suma aproximada de 23 DVD diarios para hacer la salva de estos datos, persona solo para cambiar los CD cuando se llenen. Todo esto sin tener en cuenta que alguno de estos pueda tene usar compactadores hace posible que este gasto disminuya en una gran medida pues usando la máxima compres el tamaño se reduce aproximadamente a solo 3 GB lo que haría posible salvar todo esta información en solo un D hace que el procesamiento de un servidor profesional (Hacer Altos 600 de dos procesadores 900 MHz, 750 MB d agote al punto de demorar más de 14 horas para compactar toda esa información. Disminuyendo así el procesam este proceso inviable. Sin embargo si usamos una compresión menor en este compactador podemos lograr una r computacional muy buena. Por ejemplo usando un modo de compresión menor podemos lograr un tamaño de so que comparado con los 3 GB en 14 horas es realmente un gran avance.

Otra de las técnicas más usadas y necesarias en nuestro caso para salvar servidores Data Warehouse es la salva i servidor SQL Server. Esta salva incremental necesita ante todo que ya se tenga una salva completa de la BD, perc salva de los datos que han sido modificados o adicionados, lo que disminuye en gran medida este volumen de salv salvados con una salva incremental es mucho más rápida y fiable que restaurar una salva completa de la BD. Est cuando se habla de Data Warehouse, ya que los datos de un sistema así no cambian con tanta frecuencia como lo el volumen de la salva incremental es muy pequeño y por otra parte las adiciones de datos se hace en muchos cas bastante grande.

Nota: Recuerde siempre que los datos salvados deben ser llevados a un dispositivo externo y guardados fuera de física buena, pues si son obtenidos por otras personas estas podrán saber muchos datos de la empresa.

A continuación vamos a ver un ejemplo de cómo usar el WINRAR en un script para salvar los datos de un servido

Ejemplo 9: Salva de datos de un servidor Data Warehouse.

```
rem Script para las salvas diarias de los servidores
@echo off
echo "salva del dia:" >> F:\Tareas\Backup\log.txt
date /T >> F:\Tareas\Backup\log.txt
time /T >> F:\Tareas\Backup\log.txt
del /f /q E:\Shared\Backup\bk*.rar
FOR /F "usebackq tokens=2,3,4 delims=/ " %%i IN (`date /t`) DO @SET datebackup=%%k%%i%%j
FOR /F "usebackq tokens=1,2 delims=: " %%i IN (`time /t`) DO @SET timebackup=%%i%%j
FOR /F "usebackq tokens=1 delims= " %%i IN (`date /t`) DO @SET dayname=%%i
SET DAYTIME=%datebackup%%timebackup%00
if EXIST daytimelastbackup.txt goto NoFirstTime
rem primera vez que se salvan los datos
rem Salvar SQL
dir E:\SqlData\MSSQL\BACKUP\*.bak /B /L /S /A:-D > F:\Tareas\Backup\filelistaplics2.lst
dir E:\SqlData\MSSQL\BACKUP\*.trn /B /L /S /A:-D >> F:\Tareas\Backup\filelistaplics2.lst
"C:\Program Files\WinRAR\rar.exe" a E:\Shared\Backup\bkaplics1.rar @F:\Tareas\Backup\filelistaplics1.lst
"C:\Program Files\WinRAR\rar.exe" a -m1 E:\Shared\Backup\bkaplics2.rar @F:\Tareas\Backup\filelistaplics
rem recopilar toda la salva bajo un solo nombre
"C:\Program Files\WinRAR\rar.exe" a -m0 E:\Shared\Backup\sas9%datebackup%.rar E:\Shared\Backup\bk*
goto eof
:NoFirstTime
rem ################################ Empieza la salva incremental
for /F "tokens=1" %%i in (daytimelastbackup.TXT) do @SET LastBackup=%%i
rem Salvar SQL
dir E:\SqlData\MSSQL\BACKUP\*.bak /B /L /S /A:-D > F:\Tareas\Backup\filelistaplics2.lst
dir E:\SqlData\MSSQL\BACKUP\*.trn /B /L /S /A:-D >> F:\Tareas\Backup\filelistaplics2.lst
"C:\Program Files\WinRAR\rar.exe" a -ta%LastBackup% E:\Shared\Backup\bkaplics1.rar @F:\Tareas\Backu
"C:\Program Files\WinRAR\rar.exe" a -ta%LastBackup% -m1 E:\Shared\Backup\bkaplics2.rar @F:\Tareas\B
rem recopilar toda la salva bajo un solo nombre
"C:\Program Files\WinRAR\rar.exe" a -m0 E:\Shared\Backup\sas9%datebackup%.rar E:\Shared\Backup\bk*
:eof
```

```
rem Fin del Script de Salva

echo %DAYTIME% > daytimelastbackup.TXT

net use x: \\sas2\salvasdia /user:xxxx "xxxxxxx"

if %ERRORLEVEL% EQU 0 (copy /Y E:\Shared\Backup\sas9*.rar x:\%dayname%\) else (echo "Error al Conec
\log.txt)

if %ERRORLEVEL% EQU 0 (del /F /Q E:\Shared\Backup\sas9*.rar) else (echo "Error al copiar el archivo al se

if %ERRORLEVEL% EQU 0 (echo "Salva Exitosa" >> F:\Tareas\Backup\log.txt)

net use x: /d

del /f /Q E:\Shared\Backup\bk*.rar

time /T >> F:\Tareas\Backup\log.txt
```

Conclusiones

1. El uso de sistemas Data Warehouse es una poderosa estrategia para administrar empresas.
2. Los resultados que arrojan los análisis de los datos obtenidos y consolidados en el Data Warehouse pueden corrija las estrataegias hasta ahora trazadas y mejore así las ganancias.
3. El mantenimiento de un Sistema Data Warehouse es algo complejo, que requiere de recursos monetarios y
4. El modelo dimensional brinda una forma muy sencilla de representación de los datos y mejora así el tiemp
5. Los sistemas de transformación de datos de SQL Server brindan una poderosa herramienta a quienes se ir Warehouse sobre este gestor de Bases de datos.

Bibliografía

1. Robert Wrembel & Christian Concilia, DATA WAREHOUSES AND OLAP Concepts, Architectures and Sol
2. Microsoft, Microsoft SQL Server 7.0 Data Warehousing Training Kit, Microsoft Press, 2000.
3. Microsoft, Microsoft Training and Certification, Course 2073A, Programing a Microsoft SQL Server 2000
4. Microsoft, Microsoft Training and Certification, Course 2072A, Administering a Microsoft SQL Server 20(
5. Microsoft, SQL Server 2000 Books Online, Microsoft Press, 2000

Autor

Erith Eduardo Pérez Gallardo

Administrador Red Sucursal Cimex Ciego de Ávila

Ing. Informática y Aspirante Master en Informática Aplicada

Partes: 1, 2

Página anterior

Volver al principio del trabajo

Comentarios

Para dejar un comentario, regístrese gratis o si ya está registrado, inicie sesión.

Trabajos relacionados

Estudio sobre los lenguajes de programación para la robótica
Origen de la palabra robot y su significado. Propiedades características de los robots. El robot y su funcionamiento. Cl...

Estructura de un objeto. Encapsulamiento y ocultación. Organización de los objetos. Actualmente una de las áreas más ca...

Sistemas de Procesamiento de Datos Programación Orientada a Objetos
Rupturas de Informe Definición de una Ruptura de Informe. Especificación de Opciones de Proceso. Una Ruptura de Informe se usa para dividir...

Ver mas trabajos de Programacion