Participante

Edward Neftalí Liriano Gómez 2022-0437

Profesor

Francis Ramírez

Asignatura

Electiva 1 (Big Data)

Tema

Diseño e implementación de un Data Warehouse

Índice

Diseño e implementación de un Data Warehouse

Introducción al Diseño del Almacén de Datos: El diseño de un almacén de datos es un componente crucial en la arquitectura de cualquier sistema de inteligencia empresarial. Antes de comenzar a diseñar tablas y relaciones, es fundamental comprender en profundidad los objetivos y requisitos del negocio, así como los principios y conceptos fundamentales del diseño de almacenes de datos.

Modelo Dimensional: El modelo dimensional, desarrollado por Ralph Kimball, es una técnica popular para diseñar almacenes de datos. En este modelo, los datos se organizan en tablas de hechos y dimensiones. Las tablas de hechos contienen medidas cuantitativas que se analizan, mientras que las dimensiones proporcionan contexto y describen los atributos por los cuales se puede analizar y filtrar la información.

Esquema Estrella vs. Esquema Copo de Nieve: El esquema estrella es una estructura simple en la que las tablas de hechos se conectan directamente a las dimensiones. Por otro lado, el esquema copo de nieve permite una mayor normalización de las dimensiones, dividiéndolas en subdimensiones. Aunque el esquema copo de nieve puede parecer más organizado, puede complicar las consultas y afectar el rendimiento del almacén de datos.

Proceso de Diseño del Almacén de Datos: El proceso de diseño del almacén de datos sigue una serie de pasos estructurados que incluyen la identificación de requisitos analíticos y de informes, la definición de dimensiones y hechos, la exploración y perfilado de datos, la conformidad de dimensiones entre diferentes procesos comerciales y la priorización de procesos para diseñar modelos dimensionales específicos.

Consideraciones para las Claves de Dimensión: Las claves de dimensión son identificadores únicos que se utilizan para relacionar tablas de hechos con dimensiones. Se recomienda el uso de claves de sustitución (surrogate keys) en lugar de claves de negocio, ya que garantizan la unicidad y facilitan la integración de datos de múltiples fuentes.

Atributos y Jerarquías de Dimensiones: Los atributos de dimensión proporcionan detalles contextuales sobre las entidades de negocio, mientras que las jerarquías permiten una navegación estructurada a través de los datos. Es importante seleccionar cuidadosamente los atributos relevantes y definir jerarquías lógicas que reflejen la estructura de los datos y las necesidades de análisis del negocio.

Consideraciones sobre valores "Desconocido" o "Ninguno": Es esencial comprender el significado semántico de los valores NULL en los datos de origen y diseñar el almacén de datos de manera que minimice su presencia y manejo. Identificar si NULL representa "Ninguno" o "Desconocido" en el contexto de los datos es fundamental para tomar decisiones informadas sobre su tratamiento en el almacén de datos.

Diseño Físico para un Almacén de Datos:

Después de haber establecido el esquema lógico para el almacén de datos, el siguiente paso crucial es implementarlo físicamente en una base de datos. Este proceso implica una planificación cuidadosa para varios aspectos, incluyendo la ubicación de archivos, las estructuras de datos como particiones e índices, y la compresión de datos.

Actividad de E/S del Almacén de Datos:

Antes de diseñar la base de datos física, es fundamental comprender los tipos de cargas de trabajo que debe soportar y los datos que debe almacenar el almacén de datos. La actividad de entrada/salida (E/S) en la base de datos se genera principalmente por las operaciones de carga y consulta de datos, así como por las operaciones de mantenimiento como copias de seguridad. Esta sección explora en profundidad cómo estas diferentes cargas de trabajo impactan en la actividad de E/S del almacén de datos y cómo pueden ser gestionadas de manera eficiente.

Particionamiento de Tablas:

El particionamiento de tablas es una estrategia crucial para mejorar el rendimiento y la gestión de grandes volúmenes de datos en un almacén de datos. Distribuir los datos en particiones basadas en una función definida puede ayudar a optimizar las consultas, agilizar las operaciones de carga y facilitar el mantenimiento de la base de datos.

Consideraciones para los Índices:

Los índices son elementos fundamentales para maximizar el rendimiento de las consultas en una base de datos. Sin embargo, diseñar e implementar índices de manera efectiva en un almacén de datos puede ser un desafío, ya que deben equilibrar la necesidad de mejorar el rendimiento de las consultas con el impacto en las operaciones de inserción, actualización y mantenimiento de datos. En esta sección, se exploran las mejores prácticas para diseñar índices eficientes tanto para tablas de dimensiones como de hechos, incluyendo el uso de índices de columnas y otras estrategias avanzadas.

Almacenamiento y Configuración de Archivos:

La planificación adecuada del almacenamiento y la configuración de archivos es esencial para garantizar un rendimiento óptimo y una gestión eficiente de la base de datos del almacén de datos. Esta sección ofrece directrices detalladas sobre cómo configurar y distribuir archivos de datos, archivos de preparación (staging), tempdb y archivos de registro de transacciones para satisfacer las necesidades específicas del almacén de datos y optimizar el rendimiento del sistema en general.

En resumen, proporciona una visión completa y detallada del diseño físico para un almacén de datos, abordando aspectos críticos como la actividad de E/S, el particionamiento de tablas, el diseño de índices y la configuración de archivos. Estas consideraciones son fundamentales para garantizar un rendimiento óptimo y una gestión eficiente de la base de datos en entornos de almacén de datos de gran escala.

Introducción a los Columnstore Indexes

El módulo se adentra en los índices de columnas en SQL Server 2016, que representan una innovación importante en la gestión de bases de datos. Comienza explicando las diferencias entre los índices de columnas y los índices basados en filas, resaltando cómo los índices de columnas almacenan datos en un formato columnar y hacen un uso agresivo de la compresión para reducir la E/S de disco al responder a las consultas.

Se destacan dos tipos principales de índices de columnas: no agrupados y agrupados. Los índices de columnas no agrupados pueden ser copias completas o parciales de los datos subyacentes, lo que les permite ser combinados con otros índices basados

en filas en la misma tabla. Por otro lado, los índices de columnas agrupados almacenan todas las columnas de la tabla y están optimizados para el rendimiento y la compresión.

El módulo explica detalladamente cómo crear ambos tipos de índices utilizando tanto SQL Transact como SQL Server Management Studio. Además, se destacan las nuevas características introducidas en SQL Server 2016, como los índices de columnas no agrupados filtrados, que permiten filtrar las filas incluidas en el índice, reduciendo así el impacto en el rendimiento de tener un índice de columnas en una tabla de procesamiento de transacciones en línea (OLTP).

Se resalta la importancia de los índices de columnas en el contexto de almacenes de datos, donde pueden mejorar significativamente el rendimiento de consultas complejas, especialmente aquellas que implican agregaciones y selecciones de un subconjunto de columnas. Además, se proporciona información sobre cómo los índices de columnas agrupados pueden utilizar un deltastore para reducir el impacto de la fragmentación y mejorar el rendimiento.

Enfocándose en la gestión eficiente de la inserción de datos en tablas de columnas y en la consideración de la fragmentación. Se introduce una característica nueva en SQL Server 2016 que permite almacenar tablas de columnas en memoria, lo que posibilita realizar análisis operacionales en tiempo real.

La gestión de índices de columnas implica consideraciones similares a las de los índices basados en filas, aunque se debe prestar especial atención a las operaciones de manipulación de datos (DML). SQL Server administra un deltastore para que los datos puedan ser manipulados y modificados en una tabla de columnas. El deltastore recopila hasta 1,048,576 filas antes de comprimirlas en un grupo de filas comprimido y marcarlo como cerrado. Luego, el proceso de fondo "tuple-mover" agrega el grupo de filas cerrado de nuevo al índice de columnas.

Para asegurar que los datos se inserten directamente en el índice de columnas, se deben cargar en lotes de entre 102,400 y 1,048,576 filas. Esto se puede lograr mediante métodos de inserción masiva como la utilidad bcp, SQL Server Integration Services y declaraciones de inserción Transact-SQL desde una tabla de preparación.

Cuando se trata de la fragmentación de índices, se proporcionan herramientas y técnicas similares a las utilizadas para los índices basados en filas. Se pueden utilizar tanto SQL Server Management Studio como comandos Transact-SQL para examinar y manejar la fragmentación de índices.

En cuanto a la optimización de memoria, SQL Server 2016 permite tener tanto un índice de columnas agrupado como uno basado en filas en una tabla en memoria. Esta combinación de índices permite realizar análisis operacionales en tiempo real sin necesidad de implementar procesos ETL para mover datos a tablas analíticas. Además, se detallan los niveles de durabilidad disponibles para las tablas en memoria, con opciones que van desde SCHEMA_ONLY hasta SCHEMA_AND_DATA.