**MÓDULO 7 - Diseño dinámico de bases de datos**

**Objetivos de aprendizaje**

**Descubrir estrategias para crear un proceso ETL que funcione para satisfacer las necesidades organizativas y de las partes interesadas y mantener un proceso ETL de forma eficiente.**

**Comprender qué pueden incluir los distintos procesos y herramientas de almacenamiento y extracción de datos**

**Explicar cómo optimizar la creación de nuevas tablas.**

**Identificar y describir dónde pueden encajar las nuevas tablas en el pipeline.**

**Reconocer los diferentes aspectos de las bases de datos, incluidas las bases de datos OLAP y OLTP, columnares y relacionales, distribuidas y unidomiciliadas.**

**Comprender la importancia del rendimiento y la optimización de las bases de datos.**

**Describir los cinco factores del rendimiento de las bases de datos: carga de trabajo, rendimiento, recursos, optimización y contención.**

**Realizar la depuración de canalizaciones mediante consultas**

**Rendimiento de las bases de datos**

**Data Marts, Data Lakes y el proceso ETL**

**Un Data Warehouse es una gran colección de datos de todos los sistemas de la empresa. Los Data Warehouse eran muy comunes cuando las empresas usaban una sola máquina para almacenar y calcular sus bases de datos relacionales. Sin embargo, con el auge de las tecnologías en la nube y la explosión del volumen de datos, surgieron nuevos patrones para el almacenamiento y la computación de datos. Una de estas herramientas es un data mart.**

**Un Data Mart es una base de datos orientada a temas que puede ser un subconjunto de un almacén de datos más grande.**

Un data mart es una forma conveniente de acceder a los datos relevantes que deben extraerse para un proyecto en particular.

**Un Data Lake es un sistema de base de datos que almacena grandes cantidades de datos sin procesar en su formato original hasta que se necesitan**. **Esto hace que los datos sean fácilmente accesibles, ya que no requieren mucho procesamiento.**

Al igual que un Data warehouse, un Data Lake combina muchas fuentes diferentes, pero los Data Warehouse son jerárquicos con archivos y carpetas para organizar los datos.

Mientras que los Data Lake son planos y aunque los datos se han etiquetado para que sean identificables, no están organizados, son fluidos, por lo que se denomina Data Lake.

*No requieren que los datos se transformen antes del almacenamiento.* Por lo tanto, son útiles si el sistema de BI incorpora muchos tipos de datos diferentes.

Una forma de integrar Data Lakes en un sistema de datos es a través de ELT.

ELT permite a los analistas de BI obtener tantos tipos diferentes de datos en un sistema de almacenamiento tan pronto como esos datos estén disponibles. Solo tienen que transformar los datos que necesitan,

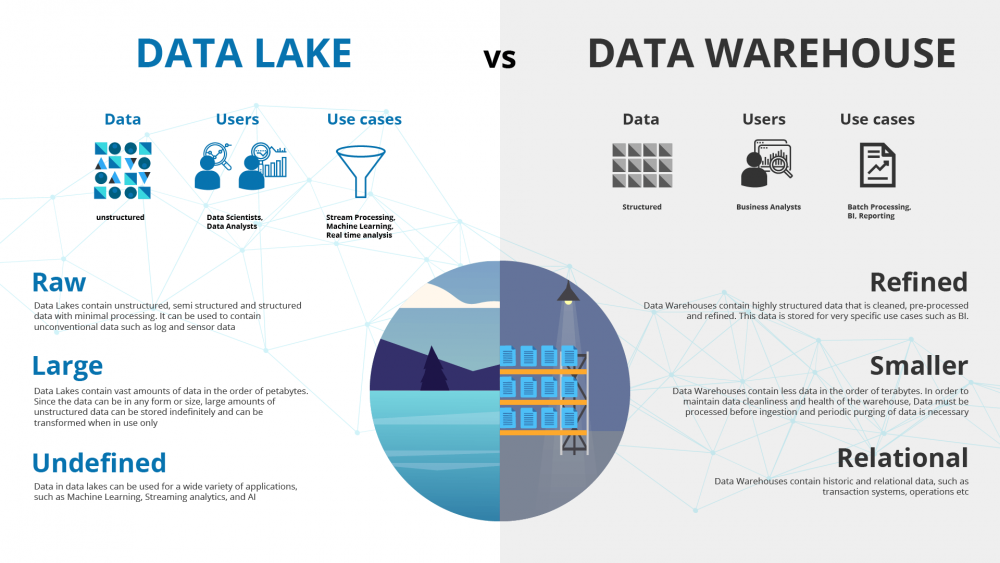
ELT también reduce los costos de almacenamiento y permite a las empresas escalar los recursos informáticos y de almacenamiento de forma independiente..

**ETL frente a ELT**

La principal diferencia entre estos dos sistemas de pipelines es el orden en que transforman y cargan los datos. También hay algunas otras diferencias clave en la forma en que se construyen y utilizan:

|  |  |  |
| --- | --- | --- |
| **Diferencias** | **ETL** | **ELT** |
| *El orden de extracción, transformación y carga de datos.* | ***Los datos se extraen, se transforman en un área de ensayo y se cargan en el sistema de destino.*** | Los datos se ***extraen, se cargan en el sistema de destino y se transforman según sea necesario para el análisis*** |
| *Ubicación de las transformaciones* | Los datos se mueven a un área de preparación donde se transforman antes de la entrega | Los datos se transforman en el sistema de destino, por lo que no se requiere un área de preparación |
| *Era de la tecnología* | ***ETL se ha utilizado durante más de 20 años*** y se han desarrollado muchas herramientas para respaldar los sistemas de tuberías ETL. | ELT es una tecnología más nueva con menos herramientas de soporte integradas en la tecnología existente |
| *Acceso a los datos dentro del sistema.* | Los sistemas ETL solo transforman y cargan los datos designados cuando se construyen el almacén y la canalización | Los sistemas ELT cargan todos los datos, lo que permite a los usuarios elegir qué datos analizar en cualquier momento |
| *Cálculos* | Los cálculos ejecutados en un sistema ETL reemplazan o revisan las columnas existentes para enviar los resultados a la tabla de destino | Los cálculos se agregan directamente al conjunto de datos existente |
| *Sistemas de almacenamiento compatibles* | Los sistemas ETL suelen estar integrados con almacenes de datos relacionales y estructurados | Los sistemas ELT pueden ingerir datos no estructurados de fuentes como lagos de datos |
| *Seguridad y cumplimiento* | La información confidencial se puede redactar o anonimizar antes de cargarla en el almacén de datos, lo que protege los datos | Los datos deben cargarse antes de que los datos puedan anonimizarse, lo que los hace más vulnerables |
| *Tamaño de datos* | ETL es excelente para manejar conjuntos de datos más pequeños que deben someterse a transformaciones complejas | ELT se adapta bien a los sistemas que utilizan grandes cantidades de datos estructurados y no estructurados |
| *Tiempos de espera* | Los sistemas ETL tienen tiempos de carga más largos, pero el análisis es más rápido porque los datos ya se han transformado cuando los usuarios acceden a ellos. | La carga de datos es muy rápida en los sistemas ELT porque los datos se pueden ingerir sin esperar a que ocurran las transformaciones, pero el análisis es más lento. |

**Sistemas de almacenamiento de datos**



|  |  |
| --- | --- |
| **Data warehouse** | **Data lake** |
| ***Los datos ya han sido procesados*** y almacenados en un sistema relacional | ***Los datos son crudos y sin procesar*** hasta que se necesitan para el análisis; adicionalmente, puede tener una copia de todo el OLTP o base de datos relacional |
| El propósito de los datos ya ha sido asignado y los datos están actualmente en uso | El propósito de los datos aún no ha sido determinado. |
| Hacer cambios en el sistema puede ser complicado y requerir mucho trabajo. | Los sistemas son altamente accesibles y fáciles de actualizar |

También hay un tipo específico de data warehouse que se puede usar como fuente de datos: data marts. Los data marts son muy similares a los data warehouse en la forma en que están diseñados, excepto que son mucho más pequeños. Por lo general, un data mart es un solo subconjunto de un almacén de datos que cubre datos sobre un solo tema.

Conclusiones clave

Actualmente, los sistemas ETL que extraen, transforman y cargan datos, y los sistemas ELT que extraen, cargan y luego transforman datos son formas comunes en que los pipelines se construyen para mover datos a donde deben ir.



**Los cinco factores del rendimiento de la base de datos**

La optimización exitosa se puede medir por el rendimiento de la base de datos.

**El rendimiento de la base de datos es una medida de la carga de trabajo que puede procesar una base de datos, así como los costos asociados. Consideraremos los factores que influyen en el rendimiento, la carga de trabajo, el rendimiento, los recursos, la optimización y la contención de la base de datos**. Primero, comenzaremos con la carga de trabajo.

En BI, ***la carga de trabajo se refiere a la combinación de transacciones*, consultas, análisis y comandos del sistema que procesa el sistema de base de datos en un momento dado**.

Es común que la carga de trabajo de una base de datos fluctúe drásticamente de un día a otro, según los trabajos que se procesen y cuántos usuarios interactúen con la base de datos.

A menudo se puede predecir estas fluctuaciones. Por ejemplo, puede haber una mayor carga de trabajo a fin de mes cuando se procesan los informes o la carga de trabajo puede ser muy ligera justo antes de las vacaciones.

**El rendimiento es la capacidad general del hardware y el software de la base de datos para procesar solicitudes. *El rendimiento se compone de la velocidad de entrada y salida,* la velocidad de la unidad de procesamiento central, qué tan bien la máquina puede ejecutar procesos paralelos, el sistema de administración de la base de datos y el sistema operativo y el software del sistema**.

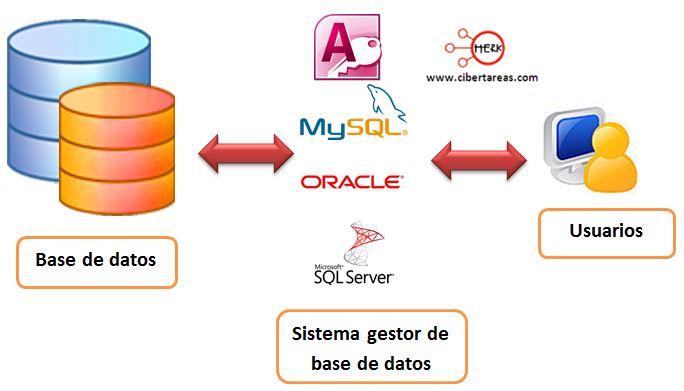
Básicamente, el rendimiento describe un tamaño de carga de trabajo que el sistema puede manejar. Vayamos a los recursos. **En BI, los recursos son las herramientas de hardware y software disponibles para su uso en un sistema de base de datos. Esto incluye el espacio en disco y la memoria.**

**Los recursos son una parte importante de la capacidad de un sistema de base de datos para procesar solicitudes y manejar datos. También pueden fluctuar, especialmente si el hardware u otros recursos dedicados se comparten con bases de datos, aplicaciones de software o servicios adicionales.** Además, los sistemas basados en la nube son particularmente propensos a las fluctuaciones. Es útil recordar que los factores externos pueden afectar el rendimiento.

**La optimización implica *maximizar la velocidad y la eficiencia con* la que se recuperan los datos para garantizar altos niveles de rendimiento de la base de datos. Este es uno de los factores más importantes a los que los analistas de BI vuelven una y otra vez.**

Finalmente, el último factor de rendimiento de la base de datos es la contención. **La contención se produce cuando dos o más componentes intentan utilizar un único recurso de forma conflictiva. Esto realmente puede ralentizar las cosas. Por ejemplo, si hay varios procesos que intentan actualizar el mismo dato, esos procesos están en conflicto. A medida que aumenta la contención, disminuye el rendimiento de la base de datos**. Limitar la contención tanto como sea posible ayudará a garantizar que la base de datos funcione de la mejor manera.

**Una guía de los cinco factores del rendimiento de la base de datos**



|  |  |  |
| --- | --- | --- |
| **Factor** | **Definición** | **Ejemplo** |
| *Carga de trabajo* | La combinación de transacciones, consultas, análisis de almacenamiento de datos y comandos del sistema que procesa el sistema de base de datos en un momento dado. | Diariamente, la base de datos necesita procesar informes de ventas, realizar cálculos de ingresos y responder a solicitudes en tiempo real de las partes interesadas. Todas estas necesidades representan la carga de trabajo que la base de datos debe poder manejar. |
| *Rendimiento* | La capacidad general del hardware y software de la base de datos para procesar solicitudes. | El rendimiento del sistema es la combinación de la velocidad de entrada y salida, la velocidad de la CPU, la capacidad de la máquina para ejecutar procesos paralelos, el sistema de administración de la base de datos y el sistema operativo y el software del sistema. |
| *Recursos* | Las herramientas de hardware y software disponibles para su uso en un sistema de base de datos. | El sistema de base de datos se basa principalmente en la nube, lo que significa que depende de los recursos y el software en línea para mantener la funcionalidad. |
| *Mejoramiento* | Maximizar la velocidad y la eficiencia con la que se recuperan los datos para garantizar altos niveles de rendimiento de la base de datos. | Verificar continuamente que la base de datos funcione de manera óptima es parte de su trabajo como profesional de BI del equipo. |
| *Contención* | Cuando dos o más componentes intentan utilizar un solo recurso de forma conflictiva. | Debido a que este sistema genera informes automáticamente y responde a las solicitudes de los usuarios, hay momentos en los que puede estar intentando ejecutar las consultas en los mismos conjuntos de datos al mismo tiempo, lo que provoca una ralentización para los usuarios. |

**Optimizar el rendimiento de la base de datos**

**La partición de datos es el proceso de dividir una base de datos en distintas partes lógicas para mejorar el procesamiento de consultas y aumentar la capacidad de administración.** La distribución de datos dentro del sistema es extremadamente importante. Asegurarse de que los datos se han particionado de manera adecuada y consistente también es parte de la optimización.

El siguiente problema son los datos fragmentados. **Los datos fragmentados ocurren cuando los datos se dividen en muchas partes que no se almacenan juntas. A menudo como resultado del uso frecuente de los datos o de la creación, eliminación o modificación de archivos.**

***Índices, particiones y otras formas de optimizar la lectura de datos***

Una de las tareas continuas de una base de datos es la lectura de datos. La lectura es el proceso de interpretar y procesar datos para que estén disponibles y sean útiles para los usuarios.

La optimización de la lectura es una de las formas principales en que puede mejorar el rendimiento de la base de datos para los usuarios

**Índices**

Los índices en las bases de datos son básicamente los mismos: usan las claves de las tablas de la base de datos para buscar muy rápidamente en ubicaciones específicas en la base de datos en lugar de todo. Por eso son tan importantes para la optimización de la base de datos: cuando los usuarios realizan una búsqueda en una base de datos totalmente indexada, la información puede devolverse mucho más rápido. Por ejemplo, una tabla con las columnas ID, Nombre y Departamento podría usar un índice con los nombres e ID correspondientes.

## **Índices Agrupados**

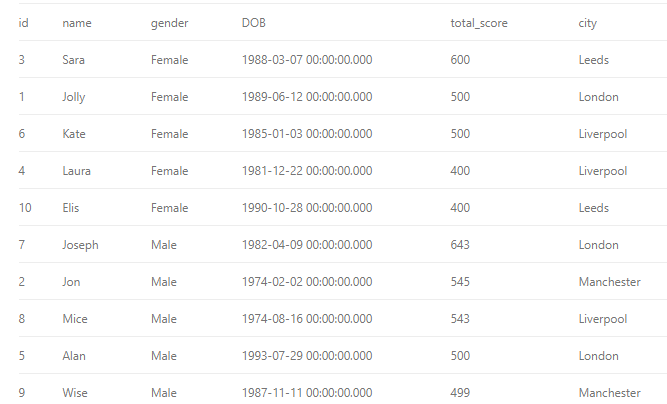
Un índice agrupado define el orden en el cual los datos son físicamente almacenados en una tabla. Los datos de las tablas pueden ser ordenados sólo en una forma, por lo tanto, sólo puede haber un índice agrupado por tabla. En SQL Server, la restricción de llave primaria crea automáticamente un índice agrupado en esa columna en particular.

para crear un nuevo índice agrupado, ejecute el siguiente script:

|  |  |
| --- | --- |
|  | use schooldb  **CREATE CLUSTERED INDEX** IX\_tblStudent\_Gender\_Score  ON student(gender ASC, total\_score DESC) |

El script de arriba crea un índice llamado “IX\_tblStudent\_Gender\_Score” en la tabla student. Este índice es creado en las columnas “gender” y “total\_score”. Un índice que es creado en más que una columna es llamado “índice compuesto.

El índice de arriba primero ordena todos los registros en *orden ascendente de gender*. Si gender es lo mismo para dos o más registros, los registros son ordenados en *orden descendiente* de los valores en su columna *“total\_score”.* Usted puede crear un índice agrupado en una sola columna también. Ahora, si usted selecciona todos los registros desde la tabla student, ellos serán recuperados en el siguiente orden:



## 

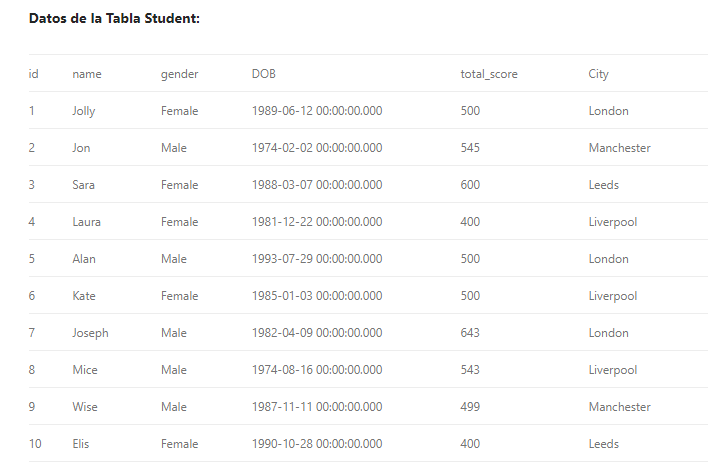
## **Índices no Agrupados**

Un índice no agrupado no ordena los datos físicos dentro de la tabla. De hecho, un índice no agrupado es agrupado en un solo lugar y los datos de la tabla son almacenados en otro lugar. Esto es similar a un libro de texto donde *el contenido del libro* está localizado en un lugar y *el índice* está localizado en otro. Esto permite tener más de un índice no agrupado por tabla.

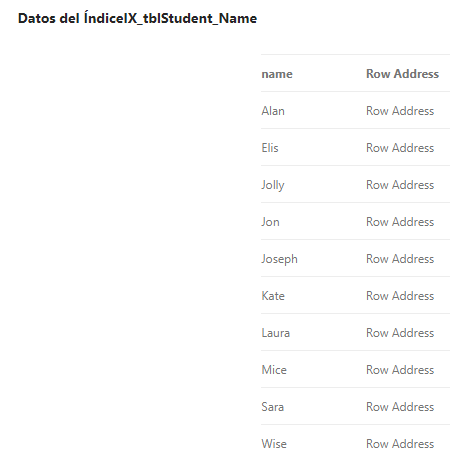
use schooldb

**CREATE NONCLUSTERED INDEX** IX\_tblStudent\_Name

ON student(name ASC)

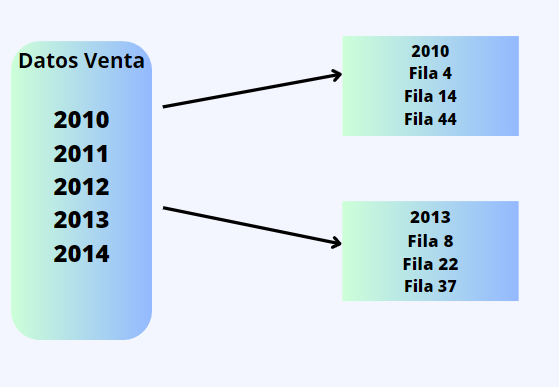


Luego de la creación del índice no agrupado, resulta:



**Particiones**

**La partición de datos es otra forma de acelerar la recuperación de la base de datos. Hay dos tipos de particiones: verticales y horizontales**. La partición horizontal es la más común e implica el diseño de la base de datos para *que las filas se organicen por agrupaciones lógicas* en lugar de almacenarse en columnas. Las diferentes filas se almacenan en diferentes tablas; esto reduce el tamaño del índice y facilita la escritura y recuperación de datos de la base de datos.



En lugar de crear una tabla de índice para ayudar a la base de datos a buscar los datos más rápido, las particiones dividen las tablas más grandes y difíciles de manejar en tablas mucho más manejables y más pequeñas.

En este ejemplo, la tabla de ventas más grande se divide en tablas más pequeñas; estas tablas más pequeñas son más fáciles de consultar porque la base de datos no necesita buscar tantos datos a la vez.

Otros métodos de optimización

Además de facilitar la búsqueda en su base de datos con índices y particiones, también puede optimizar sus búsquedas reales para mejorar la legibilidad o usar la memoria caché de su sistema para ahorrar tiempo recuperando datos de uso frecuente.

Consultas

Las consultas son solicitudes de datos o información de una base de datos. En muchos casos, es posible que tenga una colección de consultas que ejecuta regularmente; estas pueden ser consultas automáticas que generan informes o búsquedas regulares realizadas por los usuarios.

Si estas consultas no están optimizadas, pueden tardar mucho tiempo en devolver resultados a los usuarios y consumir recursos de la base de datos en general. Hay algunas cosas que puede hacer para optimizar las consultas:

**Evitar usar SELECT\* y SELECT DISTINCT: El uso de SELECT\* y SELECT DISTINCT hace que la base de datos tenga que analizar una gran cantidad de datos innecesarios.** En su lugar, puede optimizar las consultas seleccionando campos específicos siempre que sea posible.

Usar INNER JOIN en lugar de subconsultas: el uso de subconsultas hace que la base de datos analice una gran cantidad de resultados y luego los filtre, lo que puede llevar más tiempo que simplemente unir tablas en primer lugar.

Además, se pueden usar consultas agregadas previamente para aumentar la funcionalidad de lectura de la base de datos. Básicamente, la agregación previa de datos significa ensamblar los datos necesarios para medir ciertas métricas en tablas para que no sea necesario volver a capturar los datos cada vez que se ejecuta una consulta en ellos.

Más información sobre la optimización de consultas:

<https://blog.devart.com/how-to-optimize-sql-query.html>

Almacenamiento en caché

Finalmente, el caché puede ser una forma útil de optimizar la legibilidad de la base de datos. Esencialmente, el caché es una capa de memoria a corto plazo donde se pueden almacenar tablas y consultas. Al consultar el caché en lugar del propio sistema de base de datos, se puede ahorrar recursos.

Por ejemplo, si se accede con frecuencia a la base de datos para obtener informes de ventas anuales, se puede guardar esos informes en la memoria caché y extraerlos directamente de la memoria en lugar de pedirle a la base de datos que los genere una y otra vez.

**ACTIVIDAD**

**Estudio de caso: Deloitte - Optimización de sistemas de bases de datos obsoletos**

Antecedentes de la empresa

Deloitte colabora con firmas independientes para proporcionar servicios de auditoría y aseguramiento, consultoría, riesgo y asesoría financiera, gestión de riesgos, impuestos y servicios relacionados a clientes selectos. La visión de marca de Deloitte es ser un estándar de excelencia dentro del campo y defender los valores de su marca a medida que desarrollan estrategias líderes y herramientas de vanguardia para que los clientes faciliten sus negocios. Estos valores incluyen integridad, brindar un valor excepcional a los clientes, compromiso con la comunidad y fortaleza de la diversidad cultural.

El reto

Debido al tamaño de la empresa y sus necesidades de datos en constante evolución, la base de datos creció y cambió para adaptarse a los problemas actuales sin tiempo para considerar el rendimiento a largo plazo. Debido a esto, la base de datos eventualmente se convirtió en una colección de tablas no administradas sin uniones claras o un formato consistente. Esto dificultó consultar los datos y transformarlos en información que pudiera guiar de manera efectiva la toma de decisiones.

La necesidad de optimización apareció gradualmente, ya que el equipo tenía que realizar un seguimiento continuo de los datos y tenía que probar y probar repetidamente la validez de los datos. Con una base de datos recientemente optimizada, los datos podrían entenderse, confiarse y usarse más fácilmente para tomar decisiones comerciales efectivas.

Principalmente, esta base de datos contenía datos financieros y de marketing que idealmente se utilizarían para conectar campañas de marketing y oportunidades de ventas para evaluar qué campañas tuvieron éxito. Pero debido al estado actual de la base de datos, no había una forma clara de relacionar los éxitos con campañas de marketing específicas y evaluar su desempeño financiero. El mayor desafío de esta iniciativa fue programar las fuentes de datos externas para alimentar datos directamente en la nueva base de datos, en lugar de en las tablas anteriores que estaban programadas para quedar obsoletas. Además, el diseño de la base de datos necesitaba tener en cuenta las tablas que representaban el ciclo de vida de los datos y se diseñaron con uniones que pudieran admitir de manera fácil y lógica diferentes consultas y solicitudes de datos.

El enfoque

Debido a la escala del proyecto y las necesidades específicas de la organización, el equipo de BI decidió diseñar su propio sistema de base de datos que pudieran implementar en toda la organización. De esa forma, la arquitectura de la base de datos realmente capturaría sus necesidades de datos y conectaría las tablas cuidadosamente para que fueran más fáciles de consultar y usar.

Por ejemplo, el equipo quería poder conectar fácilmente la estimación inicial del éxito financiero de una campaña de marketing con su valor final y qué tan bien los procesos internos podían predecir el éxito de una campaña. Los aumentos de la estimación inicial fueron buenos, pero si las estimaciones fueron con frecuencia mucho más altas que los resultados reales, podría indicar un problema con las herramientas utilizadas para desarrollar esas estimaciones. Pero en el estado actual de la base de datos, había docenas de tablas en los grupos de contabilidad que estaban creando problemas de acceso que impedían que se hicieran estos conocimientos. Además, los diferentes grupos de contabilidad tenían mucha superposición que el equipo esperaba estructurar de manera más cuidadosa para un uso a más largo plazo.

Para lograr estos objetivos, el equipo elaboró una estrategia de arquitectura, desarrolló puntos de control para determinar si la funcionalidad requerida podía desarrollarse y eventualmente ampliarse, y creó un sistema iterativo en el que se podían realizar actualizaciones periódicas del sistema de base de datos para continuar refinándolo en el futuro.

Para considerar que el proyecto de optimización de la base de datos fue un éxito, el equipo de BI quería abordar los siguientes problemas:

¿Se consolidaron las tablas y columnas necesarias de una manera más útil?

¿El nuevo esquema y las claves abordaron las necesidades de los equipos de analistas?

¿Qué tablas se consultaban repetidamente y eran accesibles y lógicas?

¿Qué consultas de muestra podrían promover la confianza en el sistema para los usuarios?

Una variedad de socios y partes interesadas tuvieron que participar en el proyecto de optimización porque muchos usuarios en toda la organización se verían afectados. Los administradores de la base de datos y los ingenieros que trabajaron con el equipo de BI fueron particularmente clave para este proyecto porque dirigieron el diseño y la creación de la base de datos, mapearon el ciclo de vida de los datos a medida que maduraban y cambiaban con el tiempo y lo usaron como marco para construir un diseño de flujo de datos lógicos.

Luego, estos ingenieros realizaron entrevistas con varias partes interesadas para comprender

y los requisitos comerciales para los equipos en toda la organización, capacitó a un equipo de analistas en el nuevo sistema y descartó las tablas antiguas que no funcionaban.

Los resultados

El equipo de BI de Deloitte reconoció que, si bien la base de datos se había actualizado continuamente para abordar las necesidades comerciales en evolución, se había vuelto más difícil de administrar con el tiempo. Para promover un mayor rendimiento de la base de datos y garantizar que su base de datos pudiera satisfacer sus necesidades, el equipo de BI colaboró con los ingenieros y administradores de bases de datos para diseñar una arquitectura de base de datos personalizada que abordara cuidadosamente las necesidades comerciales de la organización. Por ejemplo, la nueva estructura de la base de datos ayudó a crear conexiones entre las tablas que rastrean las campañas de marketing a lo largo del tiempo y sus éxitos, incluidos los datos de ingresos y las ubicaciones regionales.

Este esfuerzo de optimización de la base de datos tuvo muchos beneficios. El mayor beneficio fue la capacidad de la organización para confiar en sus datos: el equipo de analistas no tuvo que dedicar tanto tiempo a validar los datos antes de usarlos porque las tablas ahora estaban organizadas y unidas de formas más lógicas. La nueva arquitectura también promovió consultas más simples. En lugar de tener que escribir cientos de líneas de código complicado para obtener respuestas sencillas, la nueva base de datos se optimizó para consultas más sencillas y cortas que tardaban menos tiempo en ejecutarse.

Esto proporcionó beneficios para los equipos de toda la organización:

El equipo de marketing pudo obtener mejores comentarios sobre el valor creado por campañas específicas.

El equipo de ventas podría acceder a información específica sobre sus regiones y territorios, brindándoles información sobre posibles debilidades y oportunidades de expansión.

El equipo de estrategia pudo cerrar la brecha entre los equipos de marketing y ventas y, en última instancia, crear OKR (Objectives and Key Results: objetivos y resultados clave) accionables para el futuro.

La optimización de la base de datos es un proceso iterativo. El trabajo del equipo de BI no se detuvo una vez que implementaron el nuevo diseño de la base de datos. También designaron un equipo para supervisar el gobierno de datos para garantizar la calidad de los datos y evitar que vuelvan a ocurrir los mismos problemas. De esta manera, los datos permanecen organizados y también este equipo puede continuar refinando las bases de datos desarrolladas en función de las necesidades comerciales en evolución.

Conclusión

Las bases de datos donde la organización almacena sus datos son una parte clave de los procesos y herramientas de BI que se crean; si la base de datos no funciona bien, afectará a toda la organización y hará que sea más difícil proporcionar a las partes interesadas los datos que necesitan. tomar decisiones comerciales inteligentes. La optimización de su base de datos promueve un alto rendimiento y una mejor experiencia de usuario para todos los miembros de su equipo.

**Los cinco factores en acción, caso práctico**

****

Revisaremos un sistema de cadena de cine. Hay algunas cosas que debemos tener en cuenta durante la optimización. Primero pensemos para qué se está utilizando esta base de datos. En este caso, la cadena de salas de cine utiliza datos relacionados con la compra de boletos, los ingresos y las preferencias de la audiencia para tomar decisiones sobre qué películas proyectar y posibles promociones. En segundo lugar, consideraremos de dónde provienen los datos. En este ejemplo, se envía desde varias fuentes a un sistema OLAP donde se lleva a cabo el análisis. Además, la base de datos utiliza datos de los sistemas OLTP de los cines individuales para explorar tendencias y ventas de entradas para diferentes géneros y horarios de películas. Los sistemas OLTP que administran datos de transacciones utilizan el modelo de base de datos de copos de nieve. En el centro hay una tabla de datos que captura la información más importante sobre los boletos, como si se ha reservado o no un asiento específico, el tipo de reserva, la identificación de proyección. El ID de empleado de quien ingresó la reserva y el número de asiento.

Para capturar detalles sobre estos hechos, el modelo también incluye varias tablas de dimensiones conectadas a la tabla de hechos con información sobre empleado, película, proyección, auditorio, asiento y reserva. Esta base de datos es bastante sencilla y permite que cada sala de cine registre datos en estas diferentes tablas y evita que se reserve accidentalmente el mismo asiento dos veces. Sin embargo, estos sistemas OLTP individuales no están diseñados para el análisis, por lo que es necesario extraer los datos al sistema OLAP de destino.

Allí, los usuarios pueden acceder a él y explorarlo para obtener información y tomar decisiones comerciales. Bien, ahora que sabemos un poco más sobre la base de datos, averigüemos cómo influyen los cinco factores del rendimiento de la base de datos.

En primer lugar, la carga de trabajo es una combinación de transacciones, consultas, análisis de almacenamiento de datos y comandos del sistema que procesa el sistema de base de datos en un momento dado.

En este caso, la mayor parte de la carga de trabajo es el procesamiento de solicitudes de los usuarios, como la generación de informes programados o el cumplimiento de consultas. Si la base de datos no puede manejar la carga de trabajo, podría provocar que el sistema se bloquee e interrumpa la capacidad del usuario para acceder y utilizar los datos.

Tal vez el informe requiera muchos recursos para generar o puede haber un número creciente de analistas accediendo a estos datos. Pero sabemos que a menudo es posible predecir los momentos de mayor carga de trabajo.

Entonces podemos hacer ajustes para asegurarnos de que el sistema pueda manejar estas solicitudes. Ahora, exploremos el rendimiento qué es la capacidad general del hardware y software de la base de datos para procesar solicitudes. Debido a que el sistema de cine se centra principalmente en el análisis de datos de bases de datos OLTP, trabajaremos con una base de datos OLAP que utiliza principalmente almacenamiento en la nube.

Los procesos de almacenamiento de la base de datos y las computadoras dentro del sistema acceden a los datos de la nube. Debe ser capaz de manejar la carga de trabajo de los cines, especialmente cuando el sistema de base de datos se usa mucho. El hardware y el software que componen el rendimiento del sistema son los recursos. Por ejemplo, las salas de cine pueden usar un disco controlador de caché para ayudar a la base de datos a administrar el almacenamiento y la recuperación de datos de los sistemas de memoria.

A continuación tenemos la optimización, sobre la que ya ha aprendido mucho. Idealmente, los usuarios deberían poder acceder a los datos de transacciones que se han obtenido desde muchos otros sistemas de bases de datos. Si la recuperación se ralentiza, puede llevar más tiempo obtener los datos y proporcionar información a las partes interesadas. Por eso es importante mantener la base de datos optimizada incluso después de haberla configurado.

El último factor, el rendimiento de la base de datos es la contención. La compañía de salas de cine tiene un equipo con muchos analistas diferentes que acceden y usan estos datos.

Todas estas solicitudes pueden terminar compitiendo entre sí y causar controversia. Y esto puede ser potencialmente problemático si el sistema procesa múltiples solicitudes al mismo tiempo, esencialmente realizando las mismas actualizaciones una y otra vez. Para limitar esto, la base de datos procesa las consultas y el orden en que se realizan las solicitudes.

Caso de Estudio

Desarrollá sobre optimizaciones de Bases de Datos.