PRUEBA TÉCNICA ARQUITECTO DE DATOS JIKKOSOFT S.A.S

Sección 1: Conocimiento teórico

1. Explique las diferencias entre los sistemas OLTP y OLAP. Proporcione ejemplos de escenarios de uso y cómo optimizar cada tipo de sistema.

Las diferencias fundamentales entre los sistemas OLTP y OLAP parten principalmente de su objetivo, mientras los sistemas OLTP su propósito es el procesamiento de transacciones individuales pequeñas (pocos registros por transacción), donde la concurrencia suele ser muy alta, así como normalización de los datos para minimizar la redundancia de información, se busca optimizarlos principalmente para la escritura de los datos mediante el particionamiento de las tablas para segmentar los datos (ej. Por país, región, filial, etc.) y creando estrategias de bloqueos de registros con el fin de garantizar la consistencia de los datos. En cuanto a la consulta de datos, la optimización se centra en el indexado de los campos de consulta frecuente, igualmente con el particionado de tablas.

En cuanto a los sistemas OLAP su principal propósito es el de procesar información con fines analíticos, donde las transacciones son grandes (miles, millones de registros por transacción) y la concurrencia de acceso no suele ser alta. La estructura de los sistemas OLTP sacrifican un poco la normalización de los datos, con el fin de priorizar el rendimiento en la consultas de la información. La optimización de los sistemas OLTP se puede orientar a la construcción de cubos multidimensionales que permitan la agregación de los datos de manera pre calculada en dimensiones y medidas que faciliten y mejoren el rendimiento de la consultas complejas, a nivel de diseño se utilizan esquemas estrella o copo de nieve usando desnormalización para reducir la cantidad de cruces entre tablas, a nivel de base de datos se pueden realizar estrategias de particionamiento de tablas (por fechas, país, región, filial, etc.) teniendo el cuenta el uso que los usuarios hacen del sistema, se puede acompañar de vistas materializadas o directamente tablas con información agregada para facilitar la explotación de los datos.

1. Describa el concepto de normalización de datos. Incluya ejemplos de problemas que pueden surgir al aplicar diferentes niveles de normalización y cómo resolverlos.

La normalización es el proceso de diseño de base de datos que permite una vez tenemos el modelo entidad relación llegar al modelo relacional, el cual nos muestra la estructura física en la que serán almacenados los datos, este proceso se basa en reglas llamadas formas normales, donde las principales son:

Primera forma Normal(1FN):

* Cada atributo es atómico (su valor debería ser indivisible, ejemplo: Lugar de nacimiento: ”Bogotá – Colombia”, no cumple primera forma normal, para solucionar debería crearse un atributo para Ciudad de Nacimiento y otro atributo para País de Nacimiento).
* Debe existir un atributo que de identifique los datos, cada registro debe tener un atributo que sea llave primaria e identifique de manera única el registro.
* Todos los atributos que no sean clave, deben depender de la llave primaria

Segunda forma Normal(2FN):

- Cumple con 1FN

- todos los atributos que no sean claves, deben depender exclusivamente de la llave primaria, no pueden tener dependencias parciales, Ejemplo: Si tenemos en una tabla de personas con llave primaria número de identificación de la persona, adicionalmente tenesmos como atributos no clave, el código de País de nacimiento y nombre de país de nacimiento, aunque estos atributos son identificados por la llave primaria cumpliendo la 1FN, el campo nombre país de nacimiento tiene una dependencia parcial con el atributo código de país de nacimiento por que no cumpliría la 2FN, para solucionarlo, se debe construir una nueva tabla de Países en donde se tenga el código gel país y el nombre del País, dejando en la tabla de personas, sólo el código de País de nacimiento y relacionarlo con el identificador de la tabla Países.

Tercera forma normal(@FN):

* Cumple con 2FN
* Los atributos no clave, dependen de manera no transitiva de la clave primaria. Como ejemplo si en una tabla de libros, tenemos como llave primaria el número del libro, y como atributos no claves el nombre del autor y nacionalidad del autor, El nombre del autor depende de la llave primaria, pero la nacionalidad del autor depende del nombre y transitivamente del código del libro. Para solucionarlo se debe llevar a una nueva tabla de autores, el campo Nacionalidad, con la cual la tabla libros cumpliría la 3FN, adicionalmente la nueva tabla de Autores, quedaría incumpliendo la 2FN con lo que sería necesario llevar el campo Nacionalidad a una nueva tabla de Países, garantizando la consistencia del modelo y evitar la duplicidad de información.

Los principales problemas al aplicar la normalización a los modelos de datos son los referentes al rendimiento y a la complejidad de mantener la consistencia de la información.

En cuando al rendimiento, cuando de realiza una normalización estricta, pueden genera modelos de datos extremadamente complejos, con muchas relaciones entre tablas que hace las consultas sean extensas y su rendimiento pueda verse afectado por la combinación de relaciones y cantidad de datos, algunas formas de solucionar estos problemas consisten en la creación de índices que nos ayuden a agilizar las consultas de información, adicionalmente la creación de vistas lógicas de la información que simplifique la lógica del modelo y pueda ser consumida de manera más sencilla, incluso podrían ser vistas materializadas con los resultados de uso más frecuente.

En cuanto a la consistencia del modelo, lo aconsejable sería evitar al maximo las operaciones de inserción, borrado o actualización directamente sobre las tablas, todas estas operaciones deberían realizarse mediante procedimientos almacenados que garanticen las validaciones correspondientes antes de realizar los cambios sobre los datos.

1. Describa el concepto de desnormalización de datos. Incluya ejemplos de problemas que pueden surgir al aplicar diferentes niveles de normalización y cómo resolverlos.

Es el concepto contrario al de Normalización, lo que se busca es reducir la complejidad de los modelos de datos con el fin de facilitar las labores de consumo de la información y mejorar el rendimiento de las consultas, esto es especialmente útil en modelos de bodegas de datos y sistemas OLAP.

La problemática más evidente es la de redundancia de información que nos lleva a tener información repetida, aumento del espacio de almacenamiento y complejidad al momento del mantenimiento de los datos (actualizaciones, borrados).

Estos problemas pueden mitigarse mediante el uso de procedimientos almacenados o triggers que garanticen la actualización de los datos en las tablas correspondientes reduciendo la complejidad de los procesos de actualización.

Otro problema es el rendimiento de la escritura de lo registros, al tener que escribir más datos en diferentes lugares, así como posiblemente tener que indexar estos, los procesos de escritura se harán menos eficientes, por que se rían necesarios adicionar estrategias que mitiguen esta problemática, como son el particionamiento de tablas, realizar los procesos de escritura en horarios de menos carga (cuanto la naturaleza del sistema lo permita), o hacer mayores inversiones a nivel de hardware.

1. ¿Qué es un esquema de estrellas y cómo se diferencia de un esquema de copo de nieve? Diseñe un ejemplo concreto y compare las implicaciones de rendimiento y mantenimiento.

Los esquemas estrella y de copo de nieve, son modelos utilizados para estructurar el diseño de base de datos de Bodegas de datos, son también conocidos como modelos dimensionales.

El esquema estrella busca simplificar la estructura de la base de datos, alrededor de una tabla de hechos se relacionan directamente las dimensiones del modelo, sin más ramificaciones a partir de estas, lo que logra contar de una estructura simple y eficiente en la consulta de datos, brindando un mejor rendimiento, pero sacrificando en gran medida la normalización de los datos, conviviendo con redundancia de información.

Un esquema de copo de nieve estructura las tablas de hechos y a su alrededor las dimensiones, pero estás últimas pueden tener dimensiones adicionales a manera de ramificaciones (normalización), lo que permite reducir la redundancia, garantizar la consistencia, pero sacrificando complejidad y rendimiento.

Pendiente Diseño

En cuanto al rendimiento, existirá

1. Explique el concepto de un almacén de datos y cómo se diferencia de una base de datos. Describa cómo manejaría una migración de una base de datos transaccional a un almacén de datos.

Un Datawarehouse es una estructura de información cuyo propósito es almacenar grandes volúmenes de datos de una empresa, esta información suele provenir de diferentes fuentes, como son los sistemas transaccionales (nomina, inventarios, contabilidad, producción, etc), archivos de trabajo ( informes financieros, presupuestos, Proyecciones), incluso de fuentes externas como pueden ser, datos de análisis del mercado, información del clima, etc…

La información en los almacenes de datos es procesada y catalogada con el objetivo fundamental de proveer los datos para procesos de análisis de datos, por lo que es normal que se encuentre almacenada con cierto grado de agregación y desnormalización para incrementar el rendimiento y reducir la complejidad.

A diferencia de las bases de datos, que suelen estar limitadas al sistema transaccional al que pertenecen, poseen una normalización más estricta y su uso es mayormente transaccional.

En mi criterio, el concepto de migración, cuando hacemos referencia a llevar una base de datos a un almacén de datos es difícil de encajar, ya que sus objetivos son totalmente distintos. Lo plantearía en lugar de migración, como el progreso de agregar la información del área de negocio al que pertenece la base de datos al almacén de datos, técnicamente conocido como DataMarts.

El procesos que plantearía para llevar estos datos al almacén sería el siguiente:

Análisis:

* Analizar la estructura funcional de la base de datos.
* Identificar con el área de negocio los datos que se requieren para el análisis de información (identificación de tablas de hechos y dimensiones).
* Identificar si existen las dimensiones o requieren crearse en el almacén de datos.

Diseño del almacén de datos:

* Definir el nivel de agregación que se requiere almacenar en el almacén de datos.
* Diseñar las estructuras de tablas en el almacén de datos, que recibirán la información de la base de datos.
* Garantizar que la estructura sea consistente a alas necesidades de análisis requeridas del almacén de datos.

Procesos de ETL

* Definir las reglas de calidad de datos que se requiere aplicar a la información de la base de datos para cumplir con los requerimientos del almacén de datos.
* Construir los paquetes ETL que se encarguen de extraer la información del origen, realicen la calidad de datos requerida
* Realizar la trasformación (homologaciones, agregaciones, Redundancia, etc) definida para preparar los datos de acuerdo al diseño.
* Construir los paquetes de carga de los datos a la estructura del Almacén de datos.
* Verificar consistencia de los datos.

1. Describa el teorema de CAP y sus implicaciones para las bases de datos distribuidas.

Un sistema distribuido tierne tres características deseadas, Coherencia, Disponibilidad y tolerancia a la partición (consistency, availability and partition tolerance). El teorema de CAP nos dice que los sistemas distribuidos sólo pueden ofreces dos de las tres características la mismo tiempo.

Por lo que implica a nivel de análisis y diseño determinar cuales son las necesidades reales de la solución a construir y los recursos disponibles para determinar el modelo que deberá ser implementado para dar cumplimiento a los requerimiento teniendo en cuenta las restricciones del teorema.

A niveles prácticos no se puede garantizar al 100% que no se presenten particiones, por lo que es necesario tener presente las restricciones y las posibilidades de los sistemas actuales cuando de busque una solución CA.

1. ¿Cuáles son las propiedades de ACID en un sistema de base de datos? Proporcione ejemplos de cómo estas propiedades se aplican y se garantizan en bases de datos distribuidas.

Las propiedades ACID hacen referencia a la Atomicidad (Atomicity), Consistencia (Consistency), Aislamiento (Isolation) y Durabilidad (Durability).

Atomicidad: Una transacción es atómica cuando se ejecuta completamente o no se ejecuta en absoluto, por ejemplo un pedido de un producto en línea que comprende la generación del pedido, elaboración del envío y la descarga del inventario. Si una o más de estas tres acciones no puede completarse, debe reversarse las que fueron completadas, antes de confirmar la transacción.

Consistencia: Las transacciones de la base de datos parten de una base consistente y cuando finalizan deben dejar la base de datos en estado consistente, para esto deben garantizar que se cumplan las reglas de integridad de los datos. Muchas de estas reglas pueden ser definidas en la estructura (llaves primarias y foráneas, actualizaciones en cascada, restricciones de valores en columnas etc.) otras reglas deben ser garantizadas a nivel de implementaciones, como por ejemplo, garantizar que no se produzcan pedidos de productos con inventario cero, No se generen despachos a destinos a los que no se tiene cobertura, no generar instalación de un servicio de internet a un sector donde no se tienen redes.

Aislamiento: Una transacción en ejecución no puede ser visible para otras transacciones hasta el momento que se complete. A manera de ejemplo sistema de cine que cumple la propiedad de aislamiento no permitirá que dos transacciones asignen el mismo a dos personas diferentes.

Durabilidad: Busca asegurar que una transacción que ha sido completada exitosamente, sus cambios persistan en la base de datos incluso si el sistema falla.

1. Explique el término “ETL” y “ELT”.

Estos procesos hacen referencia a los procesos encargados de mover los datos masivamente de un origen de datos a un destino de datos diferente, mayormente usado (pero no exclusivo) en los procesos de cargue a Almacenes de datos.

La diferencia fundamente está en el orden en que se realizan los procesos:

ETL ( Extracción, transformación y Carga): Se usa regularmente cuando se cuenta con un área temporal independiente (Staging) para almacenar la información y procesarla de manera que no se ocupen los recursos tanto del sistema origen como el sistema destino, una vez la data está preparada se realiza la carga al destino. Es utilidad tanto para poblar Datawarehouse, como para realizar integraciones masivas entre sistemas transaccionales.

ELT (Extracción, Carga y Transformación): Se usa cuando no se cuenta con un área temporal independiente para el almacenamiento de los datos extraídos, también en escenarios donde se define que las reglas de preparación de los datos para garantizar la consistencia e integridad de la información no puede ser delegada por el sistema destino, esto es más usado en integraciones masivas de sistemas transaccionales. Igualmente puede usarse en los procesos para poblar los almacenes de datos, cuando se quiere aprovechas las capacidades de procesamiento del Almacén.

Es una decisión de diseño cual usar de acuerdo a los escenarios de negocio y los recursos con que se cuenten.

1. ¿Qué es un lago de datos y cómo se diferencia de un almacén de datos? Proporcione ejemplos de casos de uso.

Un Lago de datos al igual que un Almacén de datos es un repositorio de grandes volúmenes de información, con la diferencia fundamental que el lago de datos permite almacenar datos en formato nativo, sean estos estructurados, semi estructurados o no estructurados, danto la posibilidad de procesamiento futuro de estos y usarlos para distintos tipos de análisis o ser proveedores de otros sistemas que puedan requerirlos.

Como ejemplos de casos de usos, puede usarse para almacenar archivos multimedia (video, imágenes, grabaciones de voz de un call center, de encuestas, etc).

Almacenamiento de información de Sensores IOT.

Almacenamiento de información de eventos de seguridad en sistemas informáticos o de seguridad.

Datos de navegación.

Información del clima, fenómenos naturales, etc.