Prueba técnica DA - Jikkosoft

Aspirante: **Jamith Bolaños Vidal**

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 sistemas.

Respuesta: Ambos, OnLine Transaction Processing OLTP y OnLine Analytical Processing OLAP son sistemas de procesamiento de datos que ayudan a almacenar y analizar datos empresariales, pueden recopilar y almacenar datos de múltiples fuentes, como sitios web, aplicaciones, medios inteligentes y sistemas internos, su diferencia se puede evidenciar en que OLTP se utiliza para manejar y procesar transacciones en tiempo real, está en función de realizar operaciones de lectura y escritura garantizando que las transacciones puedan ser rápidas y precisas, OLAP es empleada para realizar análisis complejos de los datos, permitiendo la exploración y análisis multidimensional de grandes volúmenes de datos, esto está ligado a la estructura de datos, OLTP hace uso de datos altamente normalizados, esto ayuda a minimizar la redundancia y optimizar el rendimiento, esto hace que los datos estén organizados en tablas con relaciones entre ellas, por el contario OLAP hace uso de datos desnormalizados y organizados en esquemas de estrella o copo de nieve esto ayuda a realizar consulta de gran complejidad.

Otra diferencia se puede encontrar que las consultas para OLTP las consultas tienden a ser simples, están orientadas a modificación de datos, por el contrario en OLAP las consultas son complejas, pueden involucrar la combinación de muchas tablas y están orientadas a la agregación y análisis de grandes volúmenes de datos.

Los ejemplos de los usos de cada sistema tenemos que OLTP se puede usar para comercio electrónico, bancos, sistemas de reservas y OLAP para análisis de ventas, inteligencia de negocios, gestión de campañas de marketing; para optimizar un sistema OLTP se puede hace ruso de índices, normalización, control de concurrencia y escalabilidad por medio del uso de particionamiento y replicación para manejar un alto volumen de transacciones, para OLAP se puede hacer uso de desnormalización, almacenamiento columnar, creación de vistas materializadas para pre computar y guardar los resultados de consultas muy complejas, implementar caché en el resultado de las consultas, y utilizar índices y particionamiento adecuado para optimizar el rendimiento de consultas que abracan grandes volúmenes de datos.

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

Respuesta: Este concepto se aplica a las bases de datos relacionales, es un proceso que permite organizar los datos para ayudar a reducir la redundancia y mejorar la integridad de los datos mediante el proceso de dividir las tablas grandes en tablas mucho más pequeñas y haciendo una vinculación entre ellas mediante relaciones definidas como llaves primarias y foráneas. Para ello se aplican varias reglas denominadas Formas Normales FN

La primera Forma Normal 1NF busca eliminar los grupos repetidos, busca que todos los valores de una columna sean atómicos, es decir no divisibles, para ello se debe eliminar los grupos repetidos de las tablas individuales, se debe crear una tabla independiente por cada grupo relacionado y se deben identificar cada conjunto de datos relacionados con una clave principal, por ejemplo si tenemos en una tabla que maneja la información de un inventario, podemos tener el campo proveedor, si para un producto se tiene un segundo proveedor la solución no es agregar un nuevo campo proveedor dos o proveedor tres, de acuerdo al número de proveedores que pueda tener el producto, la solución está en crear una nueva tabla para almacenar los datos de los proveedores y vincular los datos a través de las claves primarias.

La Segunda Forma Normal 2FN consiste en primer lugar en cumplir con la 1FN y se deben eliminar las dependencias parciales, es decir se deben crear tablas independientes para conjuntos de valores que se apliquen a varios registros y se deben relacionar estas tablas por medio de una clave externa, supongamos que tenemos la dirección de un cliente en un sistema de contabilidad, esta dirección también es necesaria en la tabla de pedidos, entregas, facturas, cuentas por cobrar entre otras, entonces en lugar de almacenar el dato en todas las tablas se almacena la dirección del cliente como una entrada independiente de cada una de las tablas ya sea en la tabal cliente o en una tabla de direcciones independiente.

La Tercera Forma Normal 3FN consiste en primer lugar en cumplir con la 2FN y en eliminar los campos que no dependen de la clave, es decir los valores de un registro que no tienen ninguna relación con la clave de ese registro no deben pertenecer a esa tabla, por ejemplo si se cuenta con una tabla empleado y esta tabla tiene un campo relacionada a un departamento de la empresa y también cuenta con un campo jefe del departamento, podemos observar que el jefe de departamento no depende de la tabla empleado, en este caso la 3FN recomienda separar la información en una tabla nueva, sin embargo hay que tener cuidado en el proceso de normalización debido a que si dividimos el modelo en muchas tablas pequeñas podemos estar afectando el rendimiento del modelo, para ello se recomienda aplicar la tercer forma normal a los datos que cambian con más frecuencia, acá podemos encontrar una de los principales problemas en el proceso de normalización, entre más se normalicen las tablas, para poder recuperar los datos se van a requerir más uniones (Joins) en la consulta. Esto afecta el rendimiento para ello se recomienda optimizar mediante índices y desnormalizar parcialmente para mejorar el rendimiento de las consultas.

Otra de las dificultades puede ser la complejidad en las consultas para ello se recomienda la utilización de vistas y vistas materializadas que puede hacer más sencillo el acceso a la información de acuerdo a las necesidades evidenciadas.

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

Respuesta: La desnormalización de datos es un proceso que consiste en combinar tablas que están previamente normalizadas, esto con el fin de reducir el número de uniones o Joins para obtener una consulta y mejorar el tiempo de respuesta de las consultas, hay que tener cuidado en su manejo debido a que puede introducir redundancia de datos y problemas de consistencia, este proceso incluye procesos como combinar tablas, agregar campos derivados, por ejemplo si tenemos la tabla clientes, se puede agregar el campo total ventas para introducir un campo que

almacene el valor de todas las ventas del cliente, aunque este dato no hace parte del cliente sino de las ventas, de esta forma se acelera el acceso a este dato, en lo particular considero mejor poder calcular el valor directamente de la tabla ventas, sería examinar el caso y mirar los beneficios y riesgos para poder tomar una decisión de este tipo. Otro ejercicio típico es duplicar datos en dos tablas para evitar los joins.

Dentro de los problemas más comunes que trae un proceso de desnormalización tenemos la duplicidad de los datos y puede arrojar varios problemas si los datos no se actualizan de manera uniforme, como discrepancia en los datos, falta de integridad de los mismo, para ello es necesario ser muy cuidadosos en el proceso de actualización, asegurando que los datos duplicados se mantengan sincronizados, para ello se puede hacer uso de triggers o procedimientos almacenados para realizar el proceso de manera automática, otra dificultad es el costo de almacenamiento, la duplicidad de los datos en el proceso de desnormalización aumenta los requisitos de almacenamiento, para ellos es necesario evaluar si el aumento de rendimiento justifica el aumento en los costos de almacenamiento, si se aplica este proceso es necesario realizar constante monitoreo y auditoría de la consistencia de datos, automatizar procesos de sincronización, evaluación continua del rendimiento y tener una documentación clara y detallada sobre los datos que se han desnormalizado y el por qué se realizó el proceso, esto ayuda a entender y tomar buenas decisiones en el futuro.

4. ¿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.

Respuesta: Un esquema de estrella es un tipo de arquitectura que se utiliza para bases de datos dimensionales, especialmente en almacenes de datos, es utilizado para que las consultas sean eficientes y fáciles de analizar, su diferencia con el esquema copo de nieve se centra principalmente en su estructura y el enfoque de normalización que presenta el esquema copo de nieve y no está presente en el esquema estrella.

En cuanto al esquema de estrella presenta un tabla central conocida también como tabla de hechos que contiene los datos numéricos o métricas del negocio, por ejemplo ventas e ingresos, la tabla hechos está rodeada por varias tablas de dimensiones, que contienen datos descriptivos y cualitativos relacionados a las métricas, por ejemplo de tiempo, ubicación o producto, estas tablas de dimensiones se encuentran desnormalizadas, es decir contienen toda la información en una sola tabla sin divisiones adicionales. La estructura copo de nieve tiene una variación de la estrella, en este caso las tablas de dimensiones si están normalizadas, las tablas dimensiones se dividen en subdimensiones, eliminando redundancia al separar lo datos en más tablas y mantienen relaciones unas tablas con otras, esto hace que visualmente adquiera una forma de copo de nieve.

Ejemplo: Esquema estrella.

Tabla hechos:

Ventas: Id_venta, fecha, di_producto, id_tienda, cantidad, ingresos. Id_cliente

Tablas de dimensiones:

Producto: Id_producto, nombre, categoría, marca

Tienda: Id_tienda, nombre, ubicación, gerente

Fecha: id_fecha, fecha, mes, trimestre, anio

Cliente: id_cliente, nombre, correo

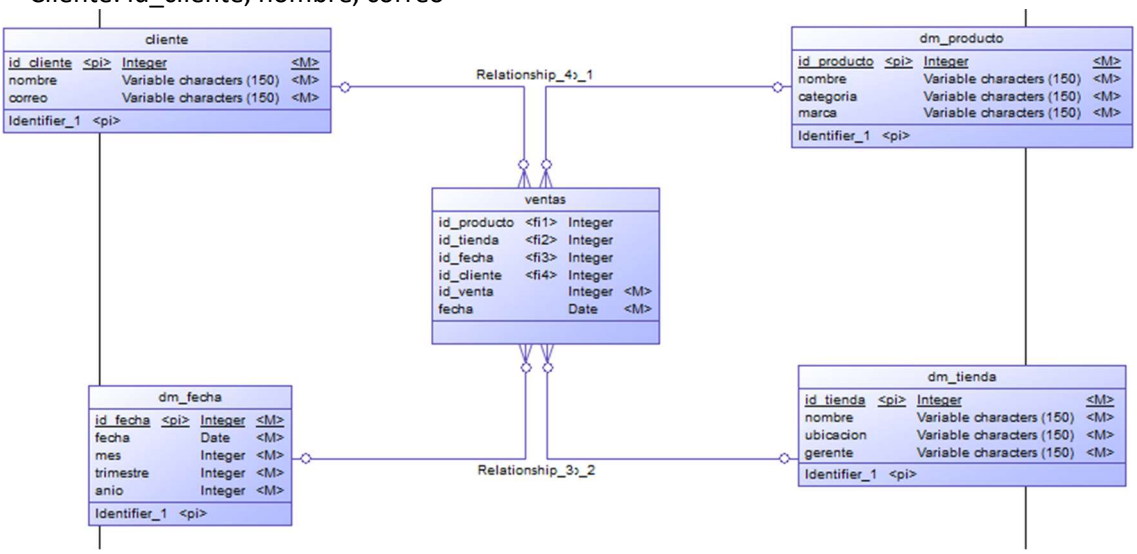

Ilustración 1 Ejemplo diagrama estrella

Ejemplo diagrama Copo de nieve:

Tabla hechos:

Ventas: Id_venta, fecha, di_producto, id_tienda, cantidad, ingresos. Id_cliente

Tablas de dimensiones:

Producto: Id_producto, nombre, id_categoria

Categoría: id_categoria, nombre_categoria, id_marca

Marca: id_marca, nombre_marca

Tienda: Id_tienda, nombre, id_ubicacion

Ubicación: id_ubicacion, ciudad, país, dirección

Fecha: id_fecha, fecha, mes, trimestre, anio

Cliente: id_cliente, nombre, correo

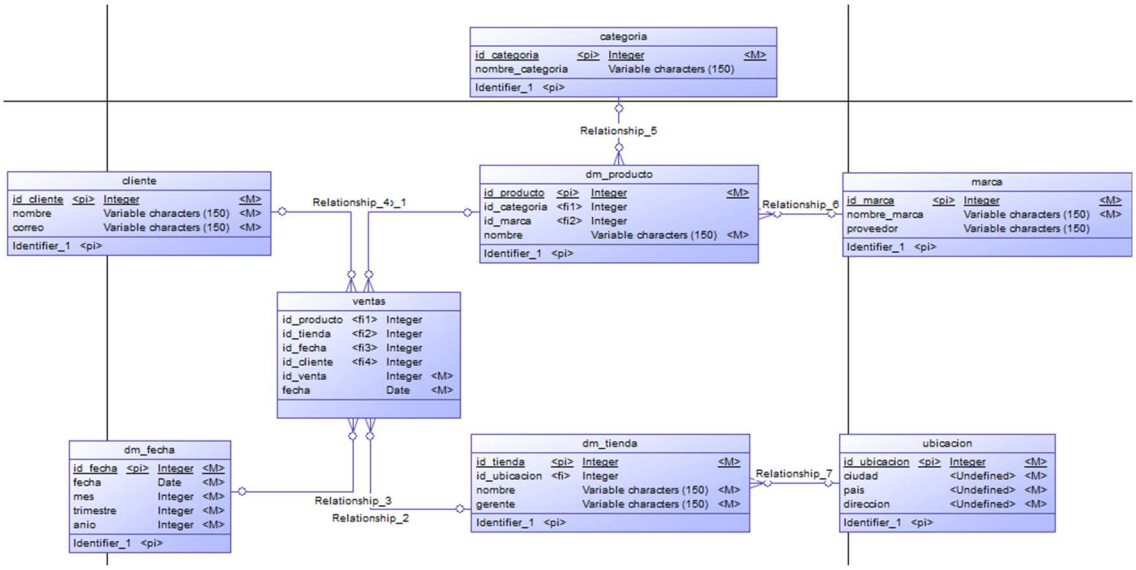

Ilustración 2 Diagrama copo nieve

5. 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 data warehouse es un sistema utilizado para almacenar, organizar y analizar grandes volúmenes de datos históricos que provienen de diversas fuentes, su misión principal es poder apoyar el análisis y toma de decisiones estratégicos para una organización, en cambio las bases de datos están pensadas para ayudar en operaciones transaccionales y de gestión diaria a través de operación CRUD, (crear, leer, actualizar y Eliminar), los almacenes de datos están diseñados para facilitar la consulta y análisis de datos agregados y complejos por medio de generación de reportes, informes para soportar la toma de decisiones a largo plazo, en cuanto a la estructura las bases de datos normalmente se encuentran con una estructura altamente normalizada para minimizar la redundancia y optimizar las operaciones de escritura, por el contrario los almacenes de datos, utilizan estructuras desnormalizadas, como esquemas de estrella o copo de nieve para optimizar las consultas y acceso rápido a grandes volúmenes de datos.

Las bases de datos usan el proceso OLTP, Online Transaction Processing, que se enfoca en procesamiento rápido y eficiente de transacciones en tiempo real, como insertar, actualizar y borrar datos en cambio un almacén de datos usa OLAP Online Analytical Processing que se centra en consulta, agregación y análisis de datos, permitiendo ejecutar consultas complejas que involucren grandes volúmenes de datos, en cuanto al tiempo de retención de datos hay una diferencia, las bases de datos maneja datos actuales y es posible que almacene la información necesaria para las operaciones diarias, el almacén de datos almacena datos históricos que pueden abarcar años, lo que permite analizar tendencias a largo plazo.

Para migrar una base de datos transaccional OLTP¨a un almacén de datos OLP se puede realizar un proceso de planificación que puede incluir las siguientes etapas:

- Análisis de requerimiento: identificar los objetivos del almacén de datos, necesidades de reportes, frecuencia de actualización, determinar los datos de la base de datos transaccionales que deben ser migrados al almacén de datos, se debe incluir cuál es la fuente de los datos.
- Diseño de almacén de datos: diseñar el modelo de datos para el almacén, eligiendo el mejor esquema, ya sea estrella o copo de nive, se diseña también el proceso ETL (Extracción, Transformación y Carga) que se encargará de extraer los datos de la base de datos transaccionales, transformarlos según sea necesario y cargarlos al almacén.
- Implementación del proceso ETL: implementar el proceso con cada uno de los pasos para Extracción, transformación aplicando los procesos definidos previamente y la inserción de datos garantizando la integridad de los datos
- Optimización del almacén de datos: crear índices, particiones y vistas materializadas para mejorar el rendimiento de las consultas y de acuerdo a los requerimientos definidos inicialmente
- Validación y pruebas: comparar los datos del almacén de datos con las fuentes originales para asegurar que el proceso se ha realizado de manera correcta y probar las consultas de los reportes en el almacén de datos para asegurar que si cumplen con los requisitos de rendimiento y exactitud en la información suministrada

- Implementación y mantenimiento: implementar el almacén de datos en producción y monitorear el rendimiento, así como establecer procesos de mantenimiento periódico para actualización de datos, realización de backups y ajustar el rendimiento del almacén
- Capacitación y documentación: capacitar a los usuarios finales y analistas de cómo usar el almacén de datos para generar reportes y realizar análisis además de documentar el procesos, este ejercicio debe hacerse por todas las etapas previamente descritas.

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

Respuesta: El teorema CAP o también conocido como teorema de Brewer, nos sugiere que todo sistema de bases de datos distribuido es vulnerable a fallos de conectividad de la red, por esta razón es imposible garantizar simultáneamente las siguientes propiedades: Consistencia: que implica que todos los nodos de un sistema distribuido ven los mismos nodos al mismo tiempo, esto significa que cualquier lectura en un sistema distribuido devuelve el resultado más reciente de una escritura; La Disponibilidad: que implica que cada solicitud recibida por el sistema, ya sea lectura y escritura, debe recibir una respuesta del estado de la solicitud, es decir que el sistema debe estar siempre disponible para aceptar solicitudes y finalmente la tolerancia: que implica que el sistema sigue funcionando a pesar de que ocurra una partición en la red, lo que significa que el sistema sigue operando incluso si se pierden mensajes entre los nodos debido a fallas de la red.

El teorema CAP establece que un sistema distribuido, solo es posible cumplir dos de las tres propiedades mencionadas a la vez, pero no las tres al mismo tiempo, las diferentes combinaciones son:

- CP, Consistencia y Tolerancia a Particiones, en un sistema que prioriza la consistencia y la tolerancia a particiones, el sistema puede garantizar que todos los nodos vean los mismos datos, en este caso es posible que se deba sacrificar la posibilidad durante una partición de red, ejemplo, si ocurre una partición en la red, el sistema puede optar por dejar de aceptar nuevas solicitudes de lectura o escritura hasta que se restablezca la conectividad, garantizando que todos los nodos tengan la misma versión de los datos.
- AP, Disponibilidad y Tolerancia a Particiones, Un sistema que prioriza la disponibilidad y la tolerancia a particiones permite que las operaciones continúen incluso durante una partición de red, pero puede que no garantice la consistencia de los datos, ejemplo, durante una partición, diferentes nodos pueden aceptar y procesar solicitudes, lo que puede llevar a situaciones donde los nodos tengan datos diferentes hasta que se resuelva la partición y se sincronicen los datos.
- CA, Consistencia y Disponibilidad, esta combinación garantiza que todos los nodos vean los mismo datos y que todas las solicitudes sean respondidas, pero no puede tolerar particiones de red. En este caso si ocurre una partición de red, el sistema puede fallar, ya que no puede mantener la consistencia y la disponibilidad al mismo tiempo.

7. ¿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.

Respuesta: Las propiedades ACID son un conjunto de principios que garantizan la fiabilidad y la integridad de las transacciones en un sistema de bases de datos, estas propiedades son fundamentales para asegurar que las operaciones en la base de datos se ejecuten de manera segura, incluso en presencia de errores o fallos en el sistema, las propiedades ACID incluyen, Atomicidad, asegura que la transacción sea todo o nada, es decir todas las operaciones dentro de una transacción deben completarse con éxito, Consistencia: asegura que la transacción lleve la base de datos de un estado válido a otro estado válido, manteniendo la integridad de los datos según las reglas de la base de datos, Aislamiento: El aislamiento asegura que las operaciones de una transacción no sean visibles para otras transacciones hasta que se complete la transacción, esto evita que las transacciones infieran entre sí. Durabilidad: esto asegura que una vez la transacción ha sido confirmada sus cambios son permanentes, incluso si el ocurre un fallo en el sistema. En bases de datos distribuidas, garantizar las propiedades ACID puede ser más complejo debido a la necesidad de coordinar múltiples nodos.

8. Explique el término “ETL” y “ELT”.

Respuesta: Son dos enfoques para el procesamiento de los datos, especialmente en el contexto de la integración de datos en almacenes de datos, o sistemas de análisis de datos, en ambos casos se refiere a procesos utilizado para mover, transformar y cargar datos desde múltiples fuentes hacia un destino común, la gran diferencia esta en el orden en que se realizan los pasos, la extracción los datos se extraen de diversas fuentes, como bases de datos, archivos, sistemas ERP, esto puede ser la recolección de datos estructurados, semiestructurados, o no estructurados. La transformación implica aplicar técnicas sobre los datos para sus análisis, esto puede incluir limpieza de datos, agregaciones, validaciones, uniones y cualquier tipo de manipulación necesaria para asegurar la consistencia y calidad de los datos, finalmente la carga de los datos se realiza en el destino final generalmente un data warehouse, dónde se almacenan para su análisis posterior. En el proceso ELT el proceso cambia, primero se hace la carga de los datos como vienen en su origen y las transformaciones se hacen en el destino.

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

Respuesta: El lago de datos es un sistema o repositorio que permite almacenar datos en su forma bruta o nativa, sin necesidad de estructurarlos o procesarlos previamente, los datos pueden ser estructurados, como tablas de bases de datos, semiestructurados como archivos JSON y XML, y no estructurados como archivos de texto, imágenes y videos, tienen características principales como flexibilidad, escalabilidad, almacenamiento económico, acceso para análisis y transformaciones posteriores, su uso es para análisis de Big Data, Data Science y Machine Learning y Almacenamiento de Datos no estructurados.

El almacén de datos es un sistema de bases de datos diseñados específicamente para el análisis de datos y generación de informes, a diferencia de los lagos datos, los datos en un almacén de datos están estructurados y organizados para un acceso y análisis rápidos y eficientes.

Sección 2: Habilidades prácticas

A. Estructura de consultas SQL

1. Escriba una consulta SQL para encontrar los 5 principales clientes por ingresos del último año en la tabla de ventas.

   ```
   SELECT cliente_id, SUM(ingreso) AS total_ingreso
   FROM ventas
   WHERE YEAR(fecha_venta) = YEAR(CURDATE()) - 1  -- Filtra las ventas del último año
   GROUP BY cliente_id
   ORDER BY total_ingreso DESC
   LIMIT 5;
   ```

2. Escriba una consulta SQL para recuperar el segundo salario más alto de la tabla de empleados.

   ```
   SELECT DISTINCT salario
   FROM empleados
   ORDER BY salario DESC
   LIMIT 1 OFFSET 1;
   ```

3. Optimice una consulta SQL y explique las mejoras de rendimiento
   Consulta original:

   ```
   SELECT cliente_id, SUM(total_venta) AS total_cliente
   FROM ventas
   WHERE YEAR(fecha_venta) = 2023
   AND total_venta > 100
   GROUP BY cliente_id
   ORDER BY total_cliente DESC;
   ```

   Consulta optimizada:

   ```
   CREATE INDEX idx_fecha_total ON ventas (fecha_venta, total_venta);
   SELECT cliente_id, SUM(total_venta) AS total_cliente
   FROM ventas
   WHERE fecha_venta >= '2023-01-01' AND fecha_venta < '2024-01-01'
   AND total_venta > 100
   GROUP BY cliente_id
   ORDER BY total_cliente DESC;
   ```

   Se crea un índice compuesto en las columnas fecha_venta y total_ventas (idx_fecha_total) permite que la consulta filtre rápidamente por fecha de venta y total de venta acelerando la ejecución de la consulta. En la clausula Where se evita la función YEAR(fecha_venta) = 2023 por un rango de fechas, fecha_venta >= ‘2023-01-01’ and fecha_venta <= ‘2024-01-01’ de esta forma utilizar el rango de fechas específico en la función de la función YEAR(), esto permite que se haga un filtrado más eficiente utilizando el índice en fecha_venta sin necesidad de aplicar la función a cada registro.

B. Diseño de bases de datos:

1. Diseñe un esquema de base de datos normalizado para una librería en línea. Incluya tablas para libros, autores, editores, clientes y pedidos. Nota: se aprecia en gran medida incluir requisitos funcionales adicionales como manejo de stock y seguimiento de envíos.

   Para el esquema se tienen en cuenta las siguientes tablas:

   Autores, que contiene toda la información relacionada a los autores de los libros.

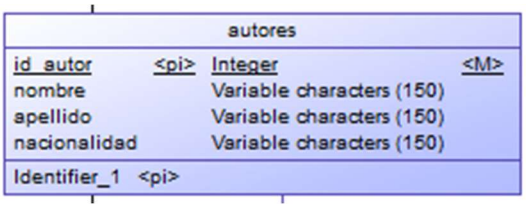

Ilustración 3: tabla autores

Editores, almacena la información sobre los editores de los libros.

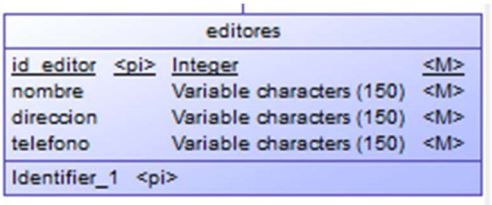

Ilustración 4. tabla editores

Libros, cuenta con la información sobre los libros

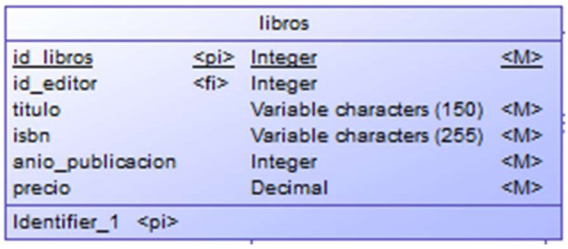

Ilustración 5. Tabla libros

Libro-autor, es la tabla resultante de la relación muchos a muchos entre libros y autores

Ilustración 6. libro autor

Clientes, almacena la información sobre los clientes.

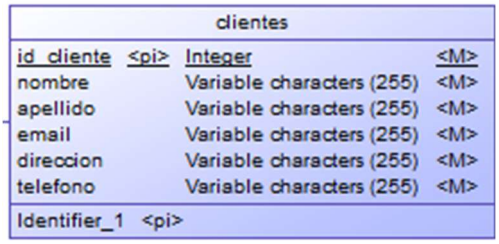

Ilustración 7. Tabla clientes

Pedidos, en esta tabla se encuentran los pedidos realizados por los clientes.

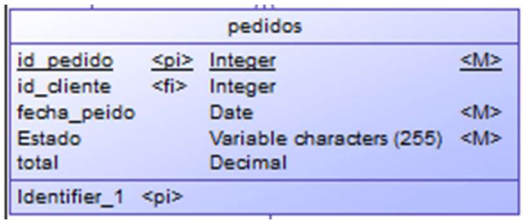

Ilustración 8. Tabla pedidos

Detalle pedido, en esta tabla se relaciona los detalles de cada pedido, incluido los libros comprados

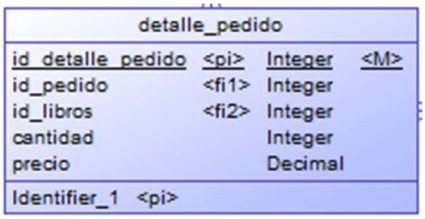

Ilustración 9. Tabla Detalle Pedido

Stock, es en esta tabla dónde se va a almacenar la información de la información de stock de cada libro

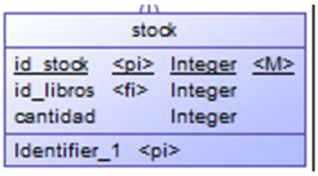

Ilustración 10. Tabla stock

Envios, se registra toda la información relacionada a los envíos de los libros

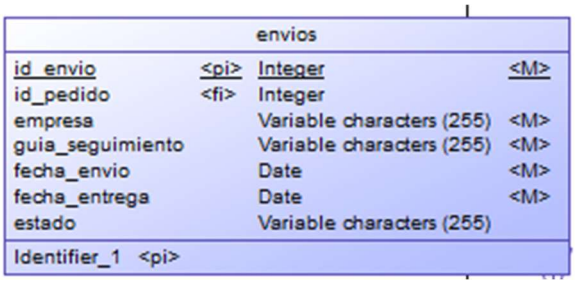

Ilustración 11. Tabla envíos

2. Explique sus decisiones de diseño y cree un diagrama de ER para el esquema diseñado

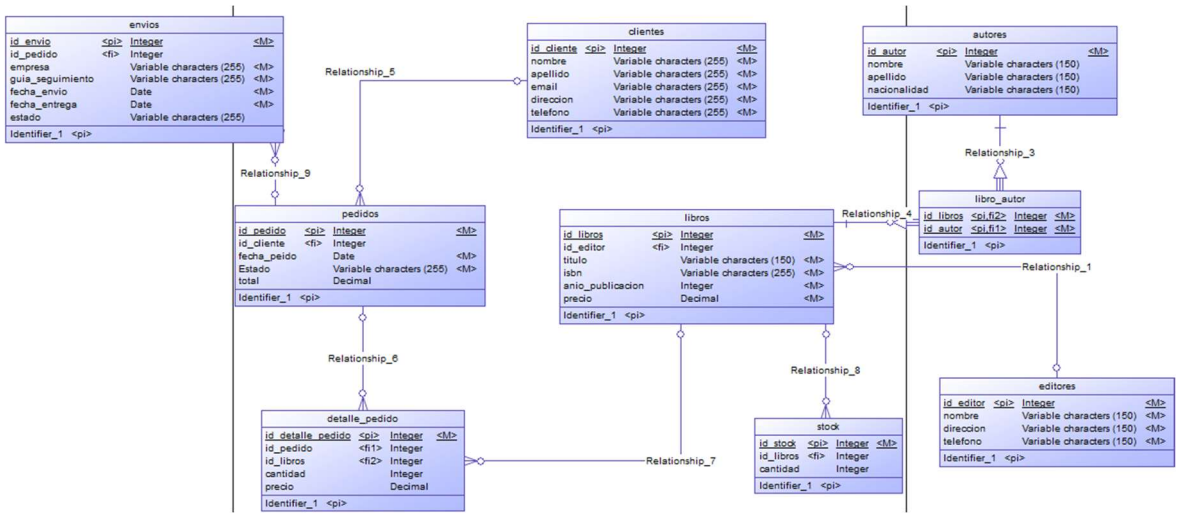

Ilustración 12 Diagrama Entidad relación

El esquema tiene una normalización hasta la 3FN, en la 1FN cada columna tiene valores atómicos, y todas las tablas tienen una clave foránea, en la 2FN no existen dependencias parciales, cada tabla no relacionada con la clave primaria ha sido separada, por ejemplo la relación libro, autor maneja una relación de muchos a muchos y en la 3FN no existen las dependencias transitivas, toda la información está relacionada únicamente con las llaves primarias, esto permitirá que se desarrolle un manejo eficiente de la información evitando redundancia y garantizando la integridad de los datos, esto es muy importante para aplicaciones de comercio electrónico como una librería. El archivo con el modelo ER se anexa en la carpeta entregada en el repositorio.

C. Modelado de datos:

1. Desarrolle un esquema de estrellas para un almacén de datos de ventas al por menor. Identifique la tabla de hechos y las tablas de dimensiones, incluyendo múltiples tablas de hechos y dimensiones compartidas.

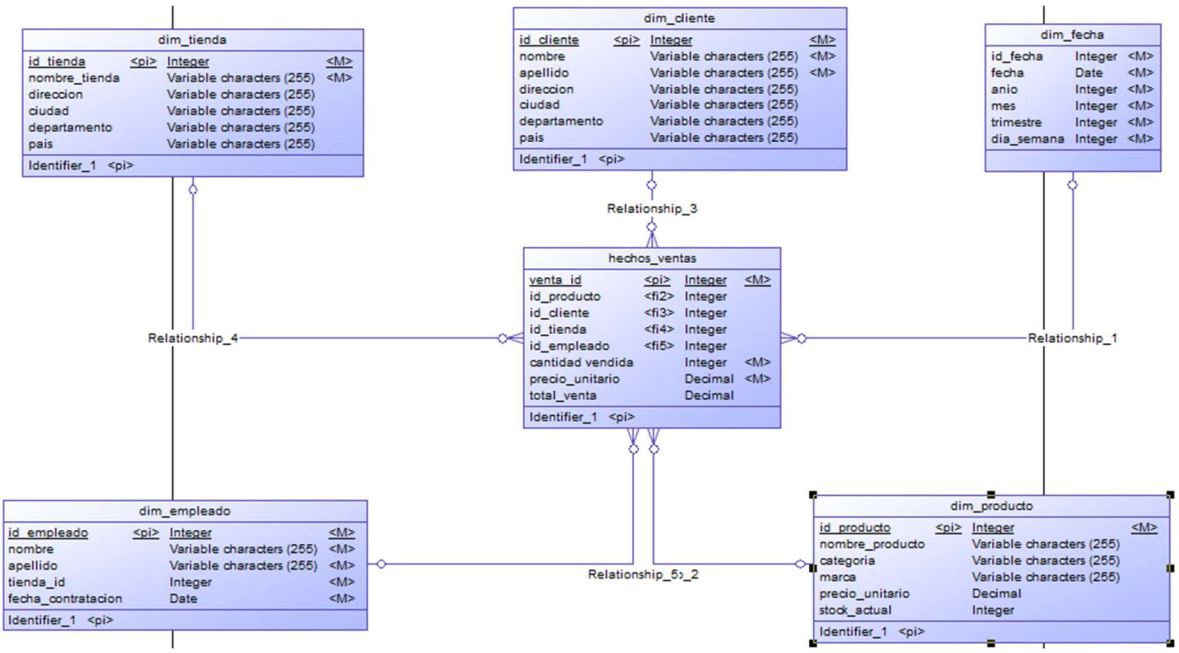

Ilustración 13. Diagrama estrella

D. Transformación de datos.
   1. Proporcione un flujo de trabajo de muestra de ETL para extraer los datos de los clientes de una base de datos transaccional, transformarlos mediante la limpieza y la agregación, y cargarlos en un almacén de datos.

Respuesta: El proceso de ETL debe contener sus tres actividades principales, las cuales se describen a continuación:

1. Extracción: el objetivo es tener lo datos relevantes en las diversas fuentes de los que se puedan obtener ya sean bases de datos relacionales con sus tablas o archivos en formato CSV, JSON, EXCEL, entre otros.
2. Transformación: en este paso se deben limpiar y transformar los datos para adaptarlos a las necesidades identificadas, en el proceso de limpieza se pueden aplicar acciones para remover duplicados, manejar valores nulos, estandarizar valores, convertir los datos a un formato único, identificar valores atípicos; también se pueden realizar acciones para agregar datos que puedan ser computados con los datos obtenidos, como cálculo total de ventas o construir nuevos campos como segmento de clientes de acuerdo a las compras que haya hecho cada cliente.
3. Carga: se preparan el almacén de datos, por medio de la creación o actualización de las tablas de hechos y dimensiones, se deben cargar los datos limpios a las nuevas estructuras y realizar el proceso de validación de que los datos se hayan cargado de manera correcta.

Es necesario tener en cuenta algunas consideraciones, como procesos de automatización para que se corran automáticamente las ETLs de acuerdo a las necesidades, también tener en cuenta monitorear el proceso para detectar y corregir errores.

Sección 3: Habilidades blandas y liderazgo

1. Describa un momento en el que tuvo que dirigir un equipo multifuncional para entregar un proyecto de datos. ¿A qué desafíos se enfrentó y cómo los superó?

Respuesta: En La Comisión para el esclarecimiento de la verdad, **Comisión de la Verdad**, una entidad gubernamental que se creó por tres años para investigar sobre los orígenes del conflicto, tuvimos que afrontar varios desafíos para procesar el gran volumen de información que incluían entrevistas, notas de prensa, informes de diversas entidades, investigaciones, expedientes, bases de datos, casos de violencia, todo esto con el fin de catalogar y disponer esta información en un meta buscador que fue la base para los investigadores que crearon el informe final, uno de los desafíos más grandes que enfrentamos como equipo fue hacia el final, pues de antemano todos los colaboradores sabían que la entidad se liquidaría en poco tiempo y muchos de ellos iniciaron a buscar empleo antes de que se terminara nuestra labor, eso hizo que se sobre cargaran tareas en el equipo y contar cada vez con menos personas para la labor titánica a la que nos enfrentamos, esto lo solucionamos con un buen trabajo en equipo, aplicamos la metodología Scrum para hacer seguimiento diario a las tareas y buscar cómo resolverlo, esto ayudo a cumplir de manera acertada con cada tarea.

2. ¿Cómo se mantiene actualizado con las últimas tendencias y tecnologías en arquitectura de datos y gestión de datos? Explique cómo aplicó una tecnología o tendencia reciente en un proyecto real.

Respuesta: Me gusta mantenerme al tanto del avance de nuevas tecnologías, utilizo el auto estudio para poder aportar a los procesos de las entidades en las que trabajo, actualmente estoy cursando último semestre de la Maestría en Ciencia de Datos en la Pontificia Javeriana de Cali, lo cual me ha permitido adquirir nuevo conocimiento para aplicarlo en mi vida laboral, actualmente estoy trabajando en un clasificador de noticias indagando el uso de Transformers y modelos pre entrenados en procesamiento de lenguaje natural NLP, como BERT (Bidirectional Encoder Representations from Transformers) estos modelos han venido transformando al forma en la que se realizan la actividades de clasificación de textos, incluyendo la clasificación de noticias.

3. Proporcione un ejemplo de un problema complejo que resolvió en su papel anterior como arquitecto de datos. Explique su enfoque y el resultado, incluyendo un problema técnico específico relacionado con arquitectura de datos.

Respuesta: En la Comisión de la verdad, trabajé en el procesamiento de grandes volúmenes de datos provenientes de diversas fuentes. Estos datos se almacenaban en un lago de datos, para ellos contábamos con procesos ETL para procesar y mover esta información a una base de datos MongoDB, donde los datos eran consultados y analizados, al principio las ETLs funcionaban mujy bien, pero a medida que se sumaba más información al lago de datos y el volumen de datos crecía, el tiempo de ejecución de las ETLs se incrementó significativamente. Esto no solo ralentizó el flujo de trabajo, sino que también afectó la capacidad de la Comisión para analizar datos críticos en plazos razonables.

Al analizar los cuellos de botella en el rendimiento de los ETLs existentes pude encontrar que Pentaho no estaba escalando eficientemente con el creciente volumen de datos. La plataforma no estaba optimizada para manejar operaciones en un entorno de big data distribuido, para poder solventar el problema se decidió migrar a un enfoque basado en bases de datos distribuidas utilizando Hadoop, que es más adecuado para manejar grandes volúmenes de datos de manera eficiente. Hadoop nos permitió distribuir las cargas de trabajo de procesamiento de datos a través de múltiples nodos, reduciendo significativamente el tiempo de procesamiento, también se tomó la decisión de migrar los procesos ETL a Python, aprovechando bibliotecas como PySpark para la manipulación y procesamiento de datos en un entorno distribuido. Esto no solo mejoró la escalabilidad, sino que también nos dio mayor flexibilidad para optimizar y personalizar los procesos de ETL, esto ayudó significativamente en el proceso y disminuyó en cerca del 70% el tiempo de ejecución de la migración de datos.

Sección 4: Codificación y secuencias de comandos

1. Escriba un script de Python para conectarse a una base de datos y recuperar datos de una tabla. Maneje errores y asegure conexiones. Incluir operaciones adicionales, como inserción y actualización de datos.

Respuesta: Se adiciona el script en el repositorio

2. Cree un script ETL simple utilizando un lenguaje de su elección (por ejemplo, Python, Java) para extraer datos de un archivo CSV, transformarlo eliminando duplicados y cargarlo en una base de datos. Incluya transformaciones más complejas, como mapeo de campos y agregaciones.

Respuesta: Se elije realizar el Scritpt en Python, se adiciona el script en el respositorio junto con los archivos csv utilizados para el ejercicio los cuales son: customers.csv, products.csv, sales_data.csv