

UT2 – Unidad 2 Acceso a la información.

Wuke Zhang

2-ASIR

Responde a las siguientes preguntas.

1. ¿Qué se entiende por software libre considerando aspectos como:

- **Gratuidad:** El software libre no siempre es gratuito, aunque muchas veces lo es. La principal característica del software libre no es su precio, sino su libertad de uso, modificación y distribución.
- **Código fuente:** En el software libre, el código fuente está disponible para que los usuarios lo vean, lo estudien, lo modifiquen y lo adapten a sus necesidades. Esto permite que la comunidad contribuya al desarrollo del software y pueda adaptar soluciones específicas.
- **Uso comercial:** El software libre permite el uso comercial, lo que significa que las empresas pueden utilizarlo, modificarlo y venderlo si desean, siempre y cuando respeten la licencia bajo la cual está publicado (por ejemplo, la GPL, que exige compartir los cambios realizados).

2. Ventajas e inconvenientes de los productos de pago respecto a los libres

- **Ventajas:**
 - **Soporte técnico:** Los productos de pago suelen ofrecer soporte técnico especializado y continuo, lo cual es fundamental en entornos empresariales.
 - **Actualizaciones regulares y planificadas:** Los productos de pago suelen seguir un plan de actualización establecido, lo que da estabilidad en el entorno empresarial.
 - **Garantía de calidad y seguridad:** Las empresas que desarrollan software de pago realizan pruebas exhaustivas para asegurar la calidad y seguridad del producto.
- **Inconvenientes:**
 - **Costo elevado:** Los productos de pago pueden tener un costo alto, tanto en licencias como en soporte adicional, lo que puede ser una limitación para algunas organizaciones.
 - **Dependencia del proveedor:** Usar software propietario de pago puede generar dependencia de un solo proveedor, lo que puede dificultar la migración a otro sistema.
 - **Menor flexibilidad:** Los productos de pago no siempre permiten modificaciones o adaptaciones, por lo que los usuarios deben ajustarse a lo que el software permite.

3. ¿Qué tiene que ver la administración de un SGBD con el diseño de bases de datos?

La administración de un Sistema de Gestión de Bases de Datos (SGBD) incluye tareas como la configuración, el monitoreo del rendimiento, la seguridad y las copias de seguridad. Aunque estas tareas se enfocan en la operación diaria, están relacionadas con el diseño de la base de datos porque un buen diseño facilita la administración, el rendimiento y la escalabilidad del sistema. La administración debe comprender el diseño para optimizar consultas, gestionar índices y asegurar una estructura eficiente de almacenamiento.

4. Ventajas de usar bases de datos frente a los tradicionales sistemas de ficheros

- **Consistencia de datos:** Los SGBD mantienen la integridad y consistencia de los datos, evitando redundancias y errores.
- **Acceso concurrente:** Permiten que múltiples usuarios accedan y modifiquen los datos simultáneamente sin causar conflictos.
- **Seguridad:** Los SGBD permiten definir permisos y restricciones, protegiendo los datos de accesos no autorizados.

5. Tres objetos típicos de una base de datos y su función

- **Tabla:** Estructura que almacena datos en filas y columnas, representando una entidad como "clientes" o "productos".
- **Índice:** Estructura que mejora la velocidad de búsqueda de datos en una tabla, similar al índice de un libro.
- **Vista:** Consulta almacenada que permite ver datos de una o varias tablas, mostrando información relevante sin revelar toda la estructura de la base de datos.

6. ¿Qué es una base de datos distribuida?

Una base de datos distribuida es aquella en la que los datos se almacenan en varios servidores ubicados en distintos lugares geográficos. Estos servidores funcionan como una unidad lógica, lo que permite a los usuarios acceder a los datos como si estuvieran en un solo lugar, aunque realmente estén distribuidos en múltiples ubicaciones.

7. Fases involucradas en el desarrollo de una base de datos

- **Recolección de requisitos:** Identificar las necesidades del negocio y las especificaciones del sistema.
- **Diseño conceptual:** Crear un modelo conceptual que represente las entidades y sus relaciones.
- **Diseño lógico:** Transformar el modelo conceptual en un modelo lógico compatible con el SGBD.
- **Diseño físico:** Definir cómo se almacenarán los datos físicamente en el sistema.
- **Implementación:** Crear las estructuras de la base de datos y poblarla con datos.
- **Pruebas:** Verificar que la base de datos cumple con los requisitos.
- **Puesta en marcha:** Implementar la base de datos en el entorno de producción.
- **Mantenimiento:** Realizar ajustes y mejoras conforme cambian las necesidades.

8. ¿Para qué sirve un disparador en un SGBD?

Un disparador (o "trigger") es un tipo de programa que se ejecuta automáticamente en respuesta a ciertos eventos en una base de datos, como inserciones, actualizaciones o eliminaciones en una tabla. Los disparadores se utilizan para mantener la integridad de los datos, auditar cambios, o llevar a cabo acciones automáticas, como registrar quién modificó un dato o actualizar otras tablas relacionadas.

9. Explica con tus palabras qué es el diccionario de datos en un SGBD

El diccionario de datos es un conjunto de metadatos, es decir, información sobre la estructura y los elementos de la base de datos. Este diccionario almacena detalles sobre las tablas, columnas, índices, vistas y otros objetos de la base de datos, así como sobre permisos y restricciones. Es esencial para que el SGBD conozca y administre su propio contenido y permite a los administradores y desarrolladores entender mejor la estructura de la base.

10. Problema y posible solución en dos escenarios

- **Caso 1: Una base de datos con miles de usuarios y centrada en consultas (como un buscador):**
 - **Problema:** Baja velocidad de respuesta debido a consultas intensivas.

- **Solución:** Implementar índices en las columnas más consultadas y utilizar una arquitectura de replicación de base de datos para distribuir la carga entre varios servidores. También, se podrían emplear técnicas de caching para acelerar el acceso a los datos más frecuentemente solicitados.
- **Caso 2: Una base de datos de venta online con miles de usuarios y operaciones por segundo:**
 - **Problema:** Problemas de concurrencia y bloqueos debido a las constantes operaciones de escritura.
 - **Solución:** Configurar una estrategia de particionamiento (sharding) de la base de datos para distribuir los datos en varios servidores. Esto reduce la carga de cada servidor individual y mejora la velocidad de acceso. También, implementar índices adecuados en las tablas de transacciones para optimizar las consultas y usar réplicas de solo lectura para balancear la carga de lectura.

