1. **ANSI-SPARC**
   1. **Definir cada uno de los 3 niveles ANSI SPARC**

Niveles ANSI-SPARC:

* Nivel Interno: Es el de mayor relación con el SO. Contiene la administración de los archivos en almacenamiento secundario. Maneja la estructura interna, su organización y los métodos de acceso.
  + *Mapping interno-conceptual: relaciona el nivel interno y el conceptual. Es decir, cómo los registros de las tablas del modelo lógico del nivel conceptual se almacenan en los distintos archivos del nivel interno. Garantiza la independencia física de datos.*
* Nivel Conceptual: Administra la estructura lógica de los datos, es decir, la representación del modelo de datos lógico de la base de datos y maneja cuestiones de estructuras, relaciones, seguridad e integridad. Oculta los detalles de implementación física del sistema y se enfoca en describir entidades, atributos, tipos, relaciones y restricciones.
  + *Mapping externo-conceptual: relaciona el nivel externo y el conceptual. Es decir, cómo las vistas o visiones del nivel externo se relacionan con el modelo de datos lógico del nivel conceptual. Garantiza la independencia lógica de datos.*
* Nivel Externo: Se relaciona con el usuario final. Contiene la administración de las vistas o visiones de los distintos usuarios o roles que utilizan los programas de aplicación. Pueden existir muchas vistas, cada una de las cuales abarca una porción del modelo de datos lógico. Brindan una abstracción de la base de datos.
  1. **¿Cuál de los niveles es relacional y cual no? ¿Por qué?**

Tanto el nivel externo como el conceptual son relacionales, mientras que el interno no lo es. Esto se debe a que, en los niveles externo y conceptual, los datos son representados y mantenidos en tablas, las cuales son relaciones y se pueden explicar mediante la teoría de conjunto. En cambio, en el nivel interno, los datos no se observan como tablas, sino como archivos en memoria secundaria compuestos por registros.

* 1. **¿En que se relaciona ANSI SPARC con independencia física y lógica?**

ANSI-SPARC garantiza la independencia física y lógica de los datos definiendo dos ‘niveles’ intermedios conocidos como mappings: mapping interno-conceptual (garantiza independencia física de datos) y mapping externo-conceptual (garantiza independencia lógica de datos).

1. **Transacciones**
   1. **Defina transacción y sus 4 propiedades**

Una transacción es una unidad de trabajo lógica que comprende una o varias sentencias SQL, las cuales transforman un estado consistente de la base de datos en otro estado consistente. Es una unidad de trabajo porque todas las operaciones dentro de la transacción se toman como un todo. Es también una unidad de recuperación porque si alguna operación falla, toda la transacción se vuelve atrás.

Una transacción comienza con un BEGIN TRANSACTION y puede finalizar con

* COMMIT: marca la finalización de una transacción satisfactoria. Indica que la base de datos está en un estado consistente y que las actualizaciones producidas pueden ser ahora confirmadas o persistidas.
* ROLLBACK: marca la finalización de una transacción insatisfactoria. Indica que algo ha salido mal, que la base de datos puede encontrarse en un estado inconsistente y que es necesario que todas las modificaciones producidas sean deshechas.

Las transacciones tienen 4 propiedades principales (ACID):

* Atomicidad: las transacciones son unidades atómicas: se hace todo o no se hace nada.
* Consistencia: las transacciones transforman un estado de consistencia de la base de datos en otro estado también consistente.
* Aislamiento: las transacciones concurrentes no interfieren entre sí en su procesamiento y todas deben mantener la consistencia de la base de datos.
* Durabilidad: una vez confirmada una transacción, los cambios producidos serán permanentes, más allá de que ocurra una caída posterior del sistema.
  1. **¿Una sentencia es una transacción en si misma? ¿Por qué?**

Una sentencia es una transacción ya que es una unidad de trabajo que puede implicar una acción sobre más de un registro. Ejemplo: DELETE con una condición que incluya a más de un registro.

* 1. **¿Qué mecanismo utiliza para evitar la concurrencia?**

Para el control de concurrencia se utiliza un mecanismo denominado bloqueo. El mismo consiste en: si una transacción desea asegurarse de que un objeto sobre el cual trabajará no sufra modificaciones mientras lo está utilizando, debe adquirir un bloqueo sobre el mismo. Un bloqueo funciona inhibiendo a todas las demás transacciones sobre el objeto dado, evitando así que cambie.

Los sistemas soportan dos tipos de bloqueo:

* Bloqueo exclusivo o de escritura, el cual rechazará cualquier otra petición de bloqueo sobre el objeto.
* Bloqueo compartido o de lectura, el cual rechazará una petición de escritura pero aceptará las de lectura.

Se define un protocolo de acceso a datos:

* Si una transacción desea recuperar una tupla, deberá adquirir previamente un bloqueo compartido sobre la misma.
* Si desea actualizar, deberá adquirir previamente un bloque exclusivo.
* Si una petición de una transacción es rechazada porque entra en conflicto con otra petición de otra transacción, la primera pasará a un estado de espera hasta que se libere el bloqueo.
* Un bloqueo exclusivo no será liberado hasta el final de la transacción.

1. **Recovery**
   1. **¿Qué tipo de fallas pueden ocurrir?**

Un sistema debe estar preparado y ser capaz de recuperarse de fallas locales y globales.

* Locales: ocurren cuando una transacción en particular finaliza con un ROLLBACK.
* Globales: ocurren cuando todo el sistema de la base de datos se ve afectado. Pueden ser:
  + Soft Crash (falla del sistema): implican una caída del sistema de base de datos que involucra a todas las transacciones, pero no hay compromiso de los medios de almacenamiento y sólo se pierde el contenido de la memoria RAM.
  + Hard Crash (falla del medio): implica una caída en la que hay compromiso de los medios de almacenamiento y parte de la base de datos se destruye físicamente (rotura de discos).
  1. **Explique el proceso de recuperación**
* Locales: implica deshacer todas las actualizaciones producidas por la transacción en cuestión. Para ello, el sistema se posiciona en el t-log en el momento del ROLLBACK y comienza a retroceder, revirtiendo uno a uno los comandos de la transacción hasta llegar a su comienzo.
* Globales:
  + Soft Crash: al iniciarse el sistema ejecuta un proceso de recovery general, durante el cual los datos no son accesibles. Para ello, se posiciona en el t-log en el último check point anterior al fallo y:
    1. Crea dos listas vacías: DESHACER y REHACER. En la lista de deshacer coloca todas las transacciones activas en ese momento, y deja vacía la lista de rehacer. Luego, comienza a recorrer el t-log hacia adelante, en sentido del fallo.
    2. Si encuentra un BEGIN TRANSACTION, coloca la transacción correspondiente en la lista de deshacer.
    3. Si encuentra un COMMIT, mueve la transacción correspondiente de la lista de deshacer a la de rehacer.
    4. Cuando termina, elimina de la lista de deshacer aquellas transacciones que se iniciaron en un tiempo posterior al del último check point, ya que las modificaciones de las mismas no se confirmaron y tampoco fueron persistidas en disco.
    5. Finalmente, el sistema deshace las transacciones de la lista de deshacer y rehace las de la lista de rehacer, teniendo en cuenta la necesidad de efectuar rollbacks parciales (las transacciones a rehacer se deshacen desde el check point y se rehacen desde el principio).
  + Hard Crash: implica restaurar una copia de seguridad de la base de datos creada en otro soporte. Una vez restaurada:
    - Si el medio de almacenamiento del t-log no se vio afectado, se lo utiliza para efectuar un recovery normal. El mismo se iniciará desde el último check point anterior a la creación de la copia de seguridad.
    - Si el medio de almacenamiento del t-log se vio afectado, las modificaciones y datos cargados en horarios sin copia de seguridad se pierden.

1. **Independencia:**
   1. **Defina en forma breve los conceptos de independencia lógica de los datos e independencia física de los datos.**

Independencia de datos: inmunidad de los programas para soportar cambios en la presentación y el modo de acceso a los datos.

Independencia física de datos: capacidad del sistema para soportar cambios en los archivos. La brinda el DBMS.

Independencia lógica de datos: capacidad del sistema para soportar cambios en el modelo de datos lógico. La brinda el DBMS.

* 1. **Explique cuál es la clave para que un gestor de base de datos pueda lograr los mismos.**

Mappings.

1. **¿Qué función cumple el optimizador o navegador del gestor de base de datos?**

El optimizador es un componente del DBMS que se encarga de analizar las peticiones y determinar una forma eficiente de satisfacerlas, teniendo en cuenta las tablas referenciadas, su tamaño, índices, etc. Decide, por tanto, cómo realizar la navegación por la base de datos. Las bases de datos tienen una ‘navegación automática’, ya que es el propio sistema quien recorre los datos para dar respuesta a las peticiones. El usuario sólo se limita a determinar qué información desea.

1. **Explique el concepto de Seriabilidad (incluya en la explicación el teorema de bloqueo de dos fases y el protocolo de bloqueo de dos fases)**

La seriabilidad es el grado de corrección que tiene la ejecución de un conjunto de transacciones. Un conjunto de transacciones es seriable si la ejecución concurrente de las mismas produce el mismo resultado que su ejecución en serie.

El teorema de bloqueo de dos fases enuncia que, para un conjunto de transacciones que cumple con el protocolo de bloqueo de dos fases, todos los planes de ejecución alternada son seriables.

El protocolo de bloqueo de dos fases consiste en que, una vez que una transacción libera un bloqueo, no puede volver a adquirir otro. Las transacciones que cumplen con este protocolo tienen dos fases: una de adquisición del bloqueo y otra de liberación.

1. **Explique el concepto de nivel de aislamiento. ¿Cómo se relaciona con el concepto de Seriabilidad?**

El nivel de aislamiento se refiere a la cantidad de interferencia que una transacción es capaz de tolerar por parte de las demás transacciones concurrentes. Mientras mayor sea el nivel de aislamiento, menor será la concurrencia y viceversa. Para garantizar la seriabilidad, no es posible tolerar ninguna cantidad de interferencia.

1. **Explique cómo trabaja el bloqueo por aproximación.**

El protocolo de bloqueo por aproximación indica que antes de solicitar una petición de bloqueo sobre una tupla, una transacción debe adquirir un bloqueo por aproximación sobre la tabla que la contiene. Existen tres tipos:

* Aproximación de bloqueo compartido: la transacción pretende adquirir un bloqueo compartido sobre tuplas individuales de la tabla.
* Aproximación de bloqueo exclusivo: la transacción pretende adquirir un bloqueo exclusivo sobre tuplas individuales de la tabla.
* Aproximación de bloqueo compartido exclusivo: la transacción pretende adquirir un bloqueo exclusivo sobre tuplas individuales de la tabla y puede soportar lectores concurrentes pero no actualizadores.

1. **Explique concurrencia y bloqueo mortal.**

El sistema de base de datos permite que muchas transacciones accedan de forma simultánea a una base de datos. Es por ello que se requiere algún mecanismo de control de concurrencia para evitar que las transacciones concurrentes interfieran entre sí.

Tres posibles problemas:

* Problema de la actualización perdida: se produce cuando dos transacciones recuperan el valor de una tupla en tiempos consecutivos y luego ambas lo actualizan de forma consecutiva. La primera de estas actualizaciones será sobreescrita por la segunda, quien escribirá el nuevo valor sin tener en cuenta la actualización anterior.
* Problema de la dependencia no confirmada: se produce cuando una transacción recupera o actualiza el valor de una tupla que ha sido actualizada por otra transacción aún no confirmada. Si dicha transacción original produce un rollback, deberá deshacerse todo cambio producido por la misma, lo que implicará que la otra transacción leyó y/o actualizó un valor aún no confirmado.
* Problema del análisis inconsistente: se produce cuando una transacción opera sobre el valor de varias tuplas y, en medio del procesamiento, otra transacción modifica los valores de las mismas. Así, la transacción original habrá visto un estado inconsistente de la base de datos y habrá producido, por tanto, un análisis inconsistente.

Bloqueo mortal: situación en la que dos o más transacciones se encuentran en estados simultáneos de espera, esperando a que alguna otra libere un bloqueo para continuar.

Ante la ocurrencia de un bloqueo mortal, es preferible que el sistema lo detecte y lo rompa. Para detectarlo, puede recurrir al reconocimiento de un ciclo en un grafo de espera, aunque algunos sistemas implementan un mecanismo de tiempo. Para romper un bloqueo mortal, el sistema selecciona a una de las transacciones y la deshace, liberando así su bloqueo.

1. **Defina base de datos. ¿Cuáles son los beneficios de las bases de datos con respecto a los archivos convencionales?**

Base de datos: sistema computarizado cuya finalidad es almacenar información y permitir a los usuarios recuperar y actualizar la misma en base a peticiones.

La diferencia y el beneficio principal de un sistema de base de datos frente a los archivos convencionales es que, al recuperar la información, la base de datos devuelve los datos y también su estructura.

Otros beneficios:

* Compactación: el DBMS se encarga de resolver problemas de redundancia y fragmentación.
* Velocidad: las bases de datos son más rápidas para obtener la información de negocio requerida.
* Productividad: las bases de datos permiten a los programadores centrarse en qué información de negocio es necesaria, sin tener que conocer los detalles técnicos acerca de cómo obtenerla.
* Seguridad: las bases de datos trabajan con una granularidad más fina, lo que permite mayor minuciosidad en las reglas de seguridad.
* Información actualizada: los cambios y actualizaciones producidas son visibles al instante.
* Manejo centralizado de la información: los modelos de datos se crean y administran de forma centralizada.
* Integridad: es el grado de corrección de los datos.
* Transaccionabilidad: el DBMS procesa las transacciones como unidades atómicas.
* Recuperación ante fallos: la base de datos está preparada para soportar un fallo eventual en el sistema.

1. **Defina y explique las funciones del DBA y DBMS**

DBA (administrador de la base de datos): personal técnico responsable del funcionamiento y la disponibilidad del sistema de base de datos.

Funciones del DBA:

* Respaldo: debe asegurarse de que se cuenta con un mecanismo y una política de respaldo o backup.
* Integridad: es el encargado de definir las reglas de integridad.
* Seguridad: debe establecer los controles físicos y lógicos para el acceso a los datos.
* Disponibilidad: debe cumplir con los contratos de servicio.
* Desempeño: debe asegurarse de que el sistema funcione lo más rápido y de la mejor manera posible.
* Soporte: debe brindar asesoramiento en materia de base de datos.

DBMS (sistema de administración de base de datos): es una pieza de software colocada entre los programas de aplicación y el sistema operativo. Es el encargado de procesar todas las peticiones de acceso a la base de datos.

Funciones del DBMS:

* Definición de datos: debe ser capaz de aceptar definiciones de datos en forma fuente y transformarlas en su forma objeto correspondiente.
* Manipulación de datos: debe ser capaz de procesar peticiones para recuperar, actualizar, eliminar o agregar datos a la base de datos.
* Optimización y ejecución: el componente optimizador del DBMS debe encargarse de interceptar cada petición y determinar una forma eficiente de implementarla.
* Seguridad e integridad: debe inspeccionar cada petición y rechazar aquellas que intenten violar las restricciones de seguridad o integridad definidas por el DBA.
* Recuperación y concurrencia: el administrador de transacciones debe imponer controles de recuperación y concurrencia.
* Diccionario de datos: debe proveer la funcionalidad de diccionario de datos, el cual contiene toda la información de los diferentes niveles del esquema de la base de datos.
* Rendimiento: debe realizar las tareas descritas de la manera más eficiente posible.

Objetivo general: proporcionar una interfaz de usuario para la base de datos.