1.- ¿Cuáles son las responsabilidades de un DBA y de un diseñador de bases de datos?

“Los diseñadores de bases de datos son responsables de identificar los datos que se almacenarán en la base de datos y para elegir estructuras apropiadas para representar y almacenar esta información. Estas tareas se realizan principalmente antes de que la base de datos se implemente y se llene realmente con datos. Es responsabilidad de los diseñadores de bases de datos comunicarse con todos los posibles usuarios de la base de datos para comprender sus requisitos y crear un diseño que cumple con estos requisitos.”(Elmarsi & Navathe, 2011, p15).

Opino que hay que tomar muy en cuenta estas responsabilidades ya que la información que ingresemos a la base de datos se vuelve nuestra responsabilidad ya que somos responsables de que esa información sea segura y que no cualquiera tenga acceso a esta información.

2.- ¿Cuáles son las diferencias básicas entre modelo relacional, modelo orientado a objetos y modelo XML?

“**El modelo de datos relacionales**, así como los modelos de datos heredados -la red y los modelos jerárquicos- que se han utilizado ampliamente en el pasado.

[..] Podemos considerar el modelo de datos de **objeto** como un ejemplo de una nueva familia de modelos de datos de implementación de alto nivel que están más cerca de los modelos de datos conceptuales.

[…] XML (eXtended Markup Language), que es un modelo de datos estructurado por árbol (jerárquico) varias relaciones comerciales han utilizado XML para el almacenamiento de sus productos.”(Elmarsi & Navathe, 2011, p31).

La diferencia entre las 3 Es su jerarquía ya que en uno se mira cómo se han estado utilizando los datos, en otro como se implementan y en el último como se almacenan.

3.- ¿Cuál es la diferencia entre lenguaje de manipulación de datos procedural y no procedural?

“”Hay dos tipos principales de LMD. Un LMD de alto nivel o no procedural puede ser utilizado por sí solo para especificar operaciones de bases de datos complejas de forma concisa. Un nivel bajo o DML procedural debe estar integrado en una programación de propósito general de idioma. Este tipo de DML normalmente recupera registros individuales u objetos de la base de datos y procesos cada uno por separado. Recupera y procesa un conjunto de registros.”(Elmarsi & Navathe, 2011, p37).

Básicamente un no procedural especifica las operaciones en una base de datos complejas y un procedural está ligado al idioma de la programación para recuperar registros y trabajarlos por separado.

4.- ¿Cuál es la diferencia entre independencia lógica de datos e independencia física de datos, indique cuál es más difícil de lograr y por qué?

“La independencia de los datos lógicos es la capacidad de cambiar el esquema conceptual sin tener que cambiar esquemas externos o programas de aplicación. La independencia de los datos físicos es la capacidad de cambiar el esquema interno sin tener que cambiar el esquema conceptual. Por lo tanto, el externo los esquemas no necesitan ser cambiados también.”(Elmarsi & Navathe, 2011, pp35-36).

Nosotros podemos cambiar el esquema conceptual para expandir la base de datos (agregando un tipo de registro o elemento de datos), para cambiar las restricciones o para reducir la base de datos (eliminando un tipo de registro o elemento de datos).

En general, la independencia de los datos físicos existe en la mayoría de las bases de datos y entornos de archivos donde los detalles físicos como la ubicación exacta de los datos en el disco y el hardware detalles de codificación de almacenamiento, ubicación, compresión, división, fusión de registros, etc., están ocultos para el usuario. Las aplicaciones no tienen conocimiento de estos detalles.

27.- ¿Cuál es la propiedad de reunión sin pérdida de una descomposición?

“Otra propiedad que debe tener una descomposición D es la unión no aditiva propiedad, que asegura que no se generan tuplas espurias cuando un NATURAL JOIN se aplica a las relaciones resultantes de la descomposición.  
Definición. Formalmente, una descomposición D = {R1, R2,..., Rm} de R tiene la pérdida (no aditiva) unir propiedad con respecto al conjunto de dependencias F en R si, para cada estado de relación r de R que satisface F, se cumple lo siguiente, donde \* es el UNIÓN NATURAL de todas las relaciones en D: \* (πR1 (r),..., πRm (r)) = r.”(Elmarsi & Navathe, 2011, pp553-554).

Pérdida se refiere a la pérdida de información y no a la pérdida de tuplas. Si no se tiene la propiedad de unión si pérdida, y después de aplicar la proyección (π) y Natural Join (\*) esas tuplas adicionales representan información errónea.

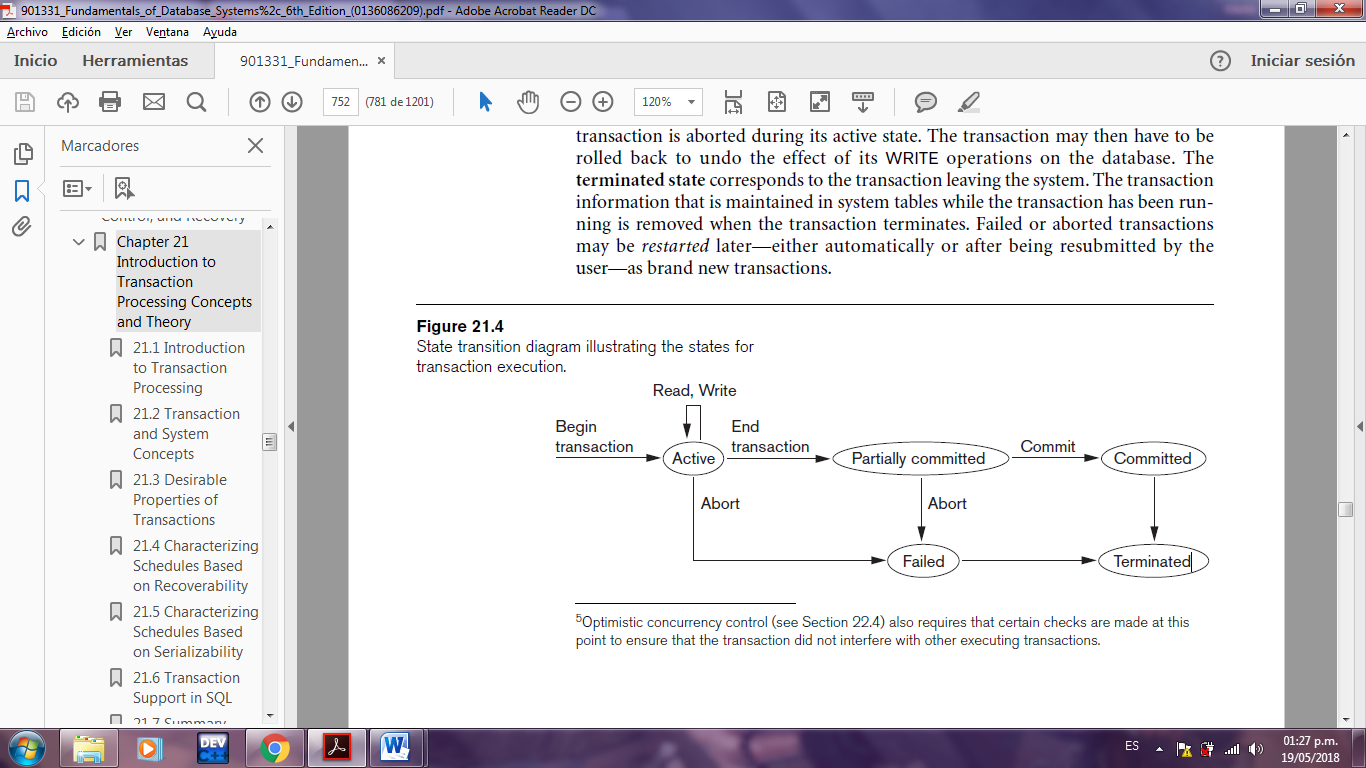
28.- ¿Qué significa el término optimización heurística? Describa las principales heurísticas que se aplican durante la optimización de una consulta.

Navathe define una optimización heurística como: “[...] Una representación de consulta optimizada, que corresponde a la estrategia de ejecución de consulta.

Una de las principales reglas heurísticas es aplicar operaciones SELECT y PROJECT antes aplicando JOIN u otras operaciones binarias, porque el tamaño del archivo resultante de una operación binaria, como JOIN, suele ser una función multiplicativa del tamaño de los archivos de entrada. Las operaciones SELECT y PROJECT reducen el tamaño de un archivo y, por lo tanto, debe aplicarse antes de una combinación u otra operación binaria.”(Elmarsi & Navathe, 2011, pp700-701).

Siento que éste tipo de optimización es muy eficiente ya que al final de cada JOIN el archivo generado puede ser muy grande y usando la heurística reducen dicho tamaño y es más optimizado ya que al momento de buscar un dato, se encontrará de manera más rápida porque la información es menor.

29.- ¿Elabore un diagrama de estados que describa los estados típicos de las transacciones durante su ejecución?



“Diagrama de transición del estado que ilustra los estados para ejecución de transacción.

■ BEGIN\_TRANSACTION. Esto marca el comienzo de la ejecución de la transacción.

■ READ OR WRITE. Estos especifican operaciones de lectura o escritura en la base de datos elementos que se ejecutan como parte de una transacción.

■ END\_TRANSACTION. Esto especifica que las operaciones de transacción READ AND WRITE han terminado y marca el final de la ejecución de la transacción. Sin embargo, en este punto, puede ser necesario verificar si los cambios introducidos por la transacción se puede aplicar permanentemente a la base de datos (comprometida) o si la transacción debe abortarse porque viola la serializabilidad o por alguna otra razón.

■ COMMIT\_TRANSACTION. Esto señala un final exitoso de la transacción por lo que cualquier cambio (actualización) ejecutado por la transacción puede ser seguro comprometido con la base de datos y no se deshará.

■ ROLLBACK (o ABORT). Esto indica que la transacción ha finalizado sin éxito, para que cualquier cambio o efecto que la transacción pueda haber aplicado a la base de datos debe deshacerse. “(Elmarsi & Navathe, 2011, pp751-752).

En mi opinión un diagrama muy eficiente ya que es muy cómodo para el entendimiento de lo que ocurre en una transacción sin que quede duda.

30.- ¿Qué es un sistema de bitácora de uso? ¿Cuáles son los registros típicos en el sistema de bitácora? ¿Qué son los puntos de confirmación de transacciones y porqué son importantes?

“El registro es un archivo secuencial, de solo un agregado, que se guarda en el disco, por lo que no se ve afectado por ningún tipo de falla a excepción del disco o falla catastrófica.

Típicamente, uno (o más) búferes de memoria principal contienen la última parte del archivo de registro, por lo que el registro las entradas se agregan primero al búfer de memoria principal. […]Además, el archivo de registro del disco se respalda periódicamente para archivar almacenamiento (cinta) para proteger contra fallas catastróficas.

Los puntos de confirmación de transacciones solo se utilizan para identificar cada transacción.

Los protocolos para la recuperación que evitan retrocesos en cascada que incluyen casi todos los protocolos prácticos, no requiere que las operaciones de LECTURA estén escritas al registro del sistema. Sin embargo, si el registro también se usa para otros fines, como auditar (realizar un seguimiento de todas las operaciones de la base de datos), entonces tales entradas pueden ser incluidas. “(Elmarsi & Navathe, 2011, p753).

Es importante saber que en cómo en cualquier procedimiento, se debe llevar un registro de lo que se está haciendo, como en este caso las transacciones ya en que este registro, nos daremos cuenta de qué está ocurriendo en el proceso y si hay errores cómo proseguir para corregirlos.

31.- Describa las propiedades ACID de una transacción.

“[..] Deben ser aplicados por el control de simultaneidad y los métodos de recuperación de laDBMS.   
■ Atomicidad. Una transacción es una unidad atómica de procesamiento; debe ser cualquiera realizado en su totalidad o no realizado en absoluto.

■ Conservación de consistencia. Una transacción debe preservar la coherencia, lo que significa que si se ejecuta completamente de principio a fin sin interferencia de otras transacciones, debe tomar la base de datos de un estado consistente a otro.

■ Aislamiento. Una transacción debería aparecer como si se estuviera ejecutando de forma aislada de otras transacciones, a pesar de que muchas transacciones se están ejecutando de forma aislada de otras transacciones, aunque muchas transacciones se estén ejecutando simultáneamente. Es decir, la ejecución de una transacción no debe interferirse con cualquier otra transacción que se ejecute simultáneamente.

■ Durabilidad o permanencia. Los cambios aplicados a la base de datos por un cometido la transacción deben persistir en la base de datos. Estos cambios no deben ser perdidos debido a cualquier falla. “(Elmarsi & Navathe, 2011, pp754-755).

Siento que es muy importante conocer estas propiedades ya que al ser aplicadas en la realidad podrían ayudar a saber qué hacer en cada caso y como resolver un problema aplicando todos los conocimientos adquiridos acerca de este tema.