Ejercicios de autocomprobación del tema 2:

1) **Características más importantes en un sistema de base de datos. Propiedades más deseables. Explicar a tu juicio cual es la propiedad más importante.**
   Podemos esquematizar las características más importantes como sigue:
   - No redundancia:
     - Los datos no deben estar duplicados.
     - Correcta gestión de accesos concurrentes (eficientemente y con seguridad).
   - Consistencia:
     - Los datos deben ser consistentes (lo que quiere decir que no contendrán errores lógicos).
     - Correcta aplicación de ciertos mecanismos de integridad.
   - Fiabilidad:
     - Los datos están protegidos contra fallos catastróficos.
     - Correcto funcionamiento de mecanismos de mantenimiento, recuperación y relanzamiento de transacciones.
   - Seguridad: No todos los datos serán accesibles por todos los usuarios:
     - Correcto funcionamiento de mecanismos de gestión de usuarios y privilegios.
     - Correcto funcionamiento de mecanismos de protección de información.

   A mi manera de ver son todos realmente importantes, pero si me tuviera que quedar con alguna de las características anteriores, aseguraría primero la consistencia de mis datos (lo que implicaría en cierta manera la no redundancia de mis datos) dado que por muy seguro y fiable que sea en los accesos a mis datos y en el almacenamiento de los mismos, si los datos son erróneos o contienen fallos lógicos nuestra base de datos no nos servirá para nada.

2) **Explicar la relación existente entre los niveles de una base de datos y el concepto de independencia.**
   La transformación o correspondencia entre niveles es un conjunto de normas que establece cómo se definen los datos de un nivel en términos de otro. Se puede entender como el mecanismo fundamental para el establecimiento de la independencia, tanto lógica como física.
   - Transformación Conceptual/Interna:
     - Cómo se organizan las entidades lógicas del nivel conceptual en términos de registros y campos almacenados en el nivel interno.
     - Independencia Física:
       - Cambio en el nivel interno.
       - Se cambia la correspondencia.
       - No varía en el nivel conceptual.
   - Transformación Externa/Conceptual:
     - Describe un esquema externo en términos del esquema conceptual subyacente.
     - Independencia Lógica:
       - Cambios en el nivel conceptual.
       - Se cambia la correspondencia
       - No varía el nivel externo.
     - No siempre es posible.

- Transformación Externa/Externa:
    - Algunos SGBDs permiten describir esquemas externos en términos de otros esquemas externos.
    - Independencia Lógica:
        - Cambios en el esquema externo subyacente.
        - Se cambia correspondencia.
        - No varia el esquema externo dependiente.

3) Explicar la diferencia entre esquema externo y aplicaciones de usuario.

En el esquema externo se definen las vistas que los usuarios tendrán de la base de datos, pero el usuario no hará uso directo de este esquema, sino que será necesario que se implemente en una aplicación que será la que si utilice el usuario (a modo de intermediario).

Una aplicación de usuario es un programa informático diseñado como herramienta para permitir a un usuario realizar uno o varios tipos de tareas. Mientras que el esquema externo es el nivel de abstracción que permite definir las informaciones a las que pueden acceder los usuarios o aplicaciones de una base de datos, y en el que las vistas que se definen vienen en función de las entidades y relaciones definidas en el nivel conceptual.

4) Diferencias existentes entre lenguaje anfitrión en una base de datos y lenguaje de comandos o "en línea".

El lenguaje "en línea" o huésped es el LMD (Lenguaje de Manipulación de Datos) de bajo nivel, cuya función es el de la manipulación física de los datos. Llamado así porque suele estar alojado en algún otro lenguaje de programación de propósito general.

El lenguaje anfitrión es un lenguaje principal a partir del cual se desarrolla la actividad necesaria con la base de datos. Es independiente. Los programas de aplicación se escriben normalmente en lenguaje anfitrión (Cobol, C, C++, Python, Java, etc…). Para acceder a la base de datos, las instrucciones LMD necesitan ser ejecutadas desde el lenguaje anfitrión.

5) ¿Qué se entiende por lenguajes "de cuarta generación"? ¿Cuáles son sus principales funciones?

Los lenguajes de cuarta generación son lenguajes que se relaciones menos con los procedimientos y que son aún más parecidos al inglés que los lenguajes de tercera generación. Algunas características incluyen capacidades de consulta y base de datos, de creación de códigos y capacidades gráficas. Son entornos de desarrollo de aplicaciones construidos por un conjunto de herramientas integradas. Se centran principalmente en las fases de construcción e implementación del ciclo de vida del desarrollo software. Usan comandos de alto nivel para recuperar y formatear datos. Generación automática de código de programa.

Los principales objetivos de estos lenguajes son:

- Acelerar el proceso de construcción de aplicaciones.
- Crear aplicaciones fáciles y rápidas de mantener, reduciendo así el costo en mantenimiento.
- Minimizar los problemas de depuración.
- Capaz de generar código "libre de errores" a partir de expresiones de alto nivel de requerimientos.
- Crear lenguajes fáciles de usar por el usuario.

6) **Buscar cuatro ejemplos de lenguajes de cuarta generación. Indicando sus objetivos o funciones.**

   i. MATLAB→ Su objetivo es proporcionar un programa de cálculo matemático técnico que nos permite manipular matrices, representar datos y funciones, implementar algoritmos…
   ii. SQL→ Su objetivo es el acceso a bases de datos relacionales permitiendo una gran variedad de operaciones sobre las mismas.
   iii. PostScript→ Su objetivo es usar un lenguaje de programación completo para describir una imagen de impresión.
   iv. Clipper→ Herramienta líder de desarrollo de aplicaciones de bases de datos relacionales bajo un sistema operativo MS-DOS, sobre todo programas de gestión, contabilidad y facturación, agendas comerciales y programas de testificación.
   v. QBE→ (Query By Examples) Lenguaje de consulta de bases de datos relacionales similar al lenguaje de consulta estructurado SQL. Se refiere a una familia de lenguajes que implementan las ideas del cálculo relacional de dominios, un lenguaje formal desarrollado para las bases de datos relacionales por IBM.

7) **¿Cuál es el enfoque actual del concepto de lenguaje anfitrión? Dar tres ejemplos de los mismos.**

   En la actualidad se tiende a realizar acoplamientos entre lenguaje anfitrión y el DSL, ya sea usando una API provista por el SGBD, para que el programador pueda acceder a la base de datos desde el código fuente, o también usando una inmersión en el código fuente del lenguaje anfitrión, desarrollando un código fuente híbrido entre lenguaje anfitrión y DSL.

   Un ejemplo podría ser JDBC de Java, que permite la ejecución de operaciones sobre bases de datos desde un programa realizado en Java, pero sin importar el sistema operativo en el que se ejecutará o la base de datos a la que se accede.

8) **Explica, a tu juicio, por qué no se han desarrollado DDLs a nivel interno.**

   Debido al acoplamiento entre lenguaje anfitrión y los lenguajes de datos, no se haría necesario un DDL específico a nivel interno.

   El DDL (Data Definition Languages) es un lenguaje destinado a la definición de estructuras de datos y esquemas en la base de datos. El nivel interno es una representación física de la base de datos en el ordenador, es en el nivel conceptual donde es necesario el DDL y en el nivel interno no es necesario debido a su funcionalidad.

9) **Explica, a tu juicio, por qué no se han desarrollado DMLs a nivel externo.**

   Por el mismo motivo por el que no se han desarrollado DDLs a nivel interno, por el acoplamiento existente actualmente entre lenguaje anfitrión y los lenguajes de datos.

   Dicho de otra forma, por definición el DML es el lenguaje por el que podemos introducir datos en los esquemas, modificarlos, eliminarlos y consultarlos. También debe permitir consultar la estructura de los esquemas definidos en la base de datos. El nivel externo es una vista de usuario, el cual no debería de tener acceso a los datos ni a los esquemas de la base de datos, pues no tiene por qué tener conocimientos de bases de datos y podría causar daños tanto a la estructura como a la integridad de los datos.

10) ¿Qué elementos conciernen al nivel interno de una base de datos?

Un DDL que describa el esquema interno (cómo se organizan las entidades lógicas del nivel conceptual en términos de registros y campos almacenados en el nivel interno) y un DML que permita el acceso a los datos almacenados en el esquema interno.

(El DDL se usa para la definición de datos y para definir estructuras y esquemas en la base de datos, mientras que el DML sirve para la manipulación de datos y permite consultar la estructura de los esquemas, otro punto por el que no se han desarrollado DML para el nivel interno, ya que, para consultar sobre una estructura, debe estará previamente creada).

11) ¿Qué cuestiones deben cubrir a tu juicio una buena herramienta de gestión de privilegios de usuarios?

Debe ser capaz de permitir realizar distinciones en varios grupos de usuarios con diferentes grupos de privilegios, delimitando con la mayor exactitud posible el nivel de acceso permitido para cada uno de los grupos (debe restringir lo que se añade, quien accede, a qué pueden acceder los usuarios, quienes pueden acceder…).

12) Indicar qué elementos deben formar parte del catálogo de una base de datos.

Contiene información sobre:

- Estructura de la base de datos.
- Cada uno de los ficheros.
- El tipo de fichero.
- El formato de almacenamiento de cada elemento.
- Restricciones de cada dato.

Los elementos son: tablas, columnas, filas, privilegios de las filas y columnas, definición de vistas, usuarios, sinónimos de usuarios…

13) Explicar a tu juicio cual es la propiedad más importante que se le debe pedir a un sistema de gestión de bases de datos.

Podemos agrupar en tres clases las funciones provistas por un SGBD:

-Consulta y actualización de datos
-Mantenimiento de esquemas
-Manejo de transacciones

Desde mi punto de vista, el cumplimiento de la normativa ACID es la propiedad más importante que debe poseer un SGBD, esto es: Atomicidad (transacciones atómicas completas), Consistencia (o integridad, cualquier transacción llevará a la base de datos de un estado valido a otro igual de valido), Isolation (o aislamiento, para asegurarse que una operación no puede afectar a otras), Durabilidad (o persistencia de datos).

14) Explicar las ventajas de la arquitectura cliente-servidor a tres niveles.

La arquitectura a 3 niveles nos va a permitir realizar una implementación mucho más eficiente de nuestra base de datos, porque para que podamos mantener una independencia física y también lógica, la mejor forma de controlarlo es disponiendo de varios niveles encargados de cumplir dichas restricciones, pudiendo relacionarse entre sí a la vez. Además, podremos diferenciar también a un nivel más externo para las perspectivas de usuarios, porque son elementos independientes, que para nada deben relacionarse con el nivel interno.

Esto se traduce en una reducción de costes significativa en cuanto al mantenimiento de los clientes ya que la instalación, configuración y actualización de las aplicaciones es realizada en el servidor y no en cada cliente. También nos brinda mayor facilidad y

flexibilidad para el usuario ya que podría acceder desde cualquier dispositivo sin necesidad de grandes costes en niveles inferiores.