



Ejercicios de autocomprobación del Tema 1

1. ¿Cuáles son las principales diferencias entre un sistema de procesamiento de archivos y un sistema de bases de datos?

En el sistema de base de datos si se modifica algo, se modificará en todos los campos que esté contenido, mientras que en el sistema de procesamiento de archivos solo se modificará el archivo en cuestión o, para finalizar, el sistema de procesamiento de archivos no ocupa mucho espacio, mientras que el de base de datos necesita mucho espacio. Además, un sistema de base de datos puede mantener fácilmente con los principios de integridad, consistencia, no redundancia, seguridad y fiabilidad junto a efectividad en búsqueda mientras que para un sistema de procesamiento de archivos es más difícil. Un sistema de procesamiento de archivos solo puede ser accedido por los usuarios que lo tengan, mientras que un sistema de base de datos puede ser accedido por todos (mientras estén autorizados); El primero tiene independencia lógica y física.

2. Características más importantes en un sistema de base de datos. Propiedades más deseables. Explica a tu juicio cual es la propiedad más importante.

Las características más importantes son la de concurrencia para evitar problemas de acceso por múltiples usuarios/administradores: Los principios mencionados anteriormente (Integridad: veracidad de los datos que se introduzcan para evitar fallos; Consistencia: Si una base de datos contiene varias copias de la información del mismo objeto éstas deben ser idénticas; Fiabilidad: Los datos deben estar protegidos contra fallos; Seguridad: No todos los datos deben ser accesibles a todos los usuarios).

En mi opinión la propiedad más importante es la de consistencia e integridad de los datos en una base de datos. Ya que como vivimos en una época de auge de información se ha de buscar que no se repita la información o no sea del todo la realidad de la que se trata.

3. Hemos conocido cuales son las ventajas de utilizar un sistema de bases de datos, ¿podrías encontrar algún inconveniente?

La gestión de una base de datos no es muy “interactivo”, es decir, no es algo alcanzable a muchos sin que se familiaricen con el entorno. Además, necesitamos una gran cantidad de memoria para almacenar la información y actualmente siguen siendo, las memorias, la parte más lenta de un ordenador o computadora a mi entender.

4. Explicar la diferencia entre independencia física e independencia lógica.

La independencia física permite cambiar el esquema interno sin modificar el sistema conceptual, mientras que la independencia lógica permite cambiar el sistema conceptual sin

modificar los externos.

5. Definir brevemente:

- Base de datos: Conjunto de datos comunes a un “proyecto” almacenados sin redundancia para ser útiles a diferentes aplicaciones.
- DBMS (DataBase Management System): Software para describir las estructuras y gestionar la información de la base de datos.
- DBA (DataBase Administrator): Administrador de una o varias bases de datos, gestionando los usuarios y el funcionamiento.
- Acceso concurrente: Posibilidad de acceder a una misma base de datos por varios usuarios de forma simultánea.
- Vista de usuario: Interfaz con la que tiene que lidiar el usuario.

6. Explicar brevemente los conceptos de: Integridad, fiabilidad y seguridad en una base de de datos.

- Ordenarlos por importancia, explicando los criterios utilizados para elaborar el orden.
- ¿En qué etapa de la vida de una base de datos se deben tener en cuenta unos y otros? • ¿Cómo se mantienen en una base de datos?.

Integridad: veracidad de los datos que se introduzcan para evitar fallos;

Consistencia: Si una base de datos contiene varias copias de la información del mismo objeto éstas deben ser idénticas;

Fiabilidad: Los datos deben estar protegidos contra fallos;

Seguridad: No todos los datos deben ser accesibles a todos los usuarios.

Ordenar según importancia: Seguridad y fiabilidad ya que creo que la protección ante fallos y que la información no esté expuesta ante cualquiera son los principios más valiables para una base de datos -> Consistencia para evitar la redundancia y optimizar la búsqueda de información -> Integridad.

Etapas de vida de una base de datos:

- Integridad: En la planificación de una base de datos ya que hay que planear la búsqueda y veracidad de la información recogida.
- Seguridad: En la planificación de la base de datos.
- Fiabilidad: En el diseño, ya que hay que ir planteando los cambios a realizar y los métodos a implementar para proteger la base de datos ante posibles fallos técnicos.

Cómo se mantiene una base de datos: La integridad la mantiene los usuarios/administradores con acceso a la base de datos y que pueden modificar sus datos. La seguridad y la fiabilidad el programar que hay planteando e implementado métodos para cumplir que esos objetivos.

7. (requiere del seminario S1) Definir y situar históricamente los siguientes términos:

- COBOL: Common Business-Oriented Language, Lenguaje Común orientado a negocios, 1959, tenía el objetivo de ser un lenguaje de programación universal.

- Acceso secuencial: Lectura/Escritura de datos de forma seguida, una tras otra.
- CODASYL: Conference on Data Systems Languages, 1959, tenía el objetivo de desarrollar un lenguaje de programación que pudiera ser usado en multitud de computadores.
- DBTG: Data Base taskGroup, 1971, dieron a conocer el modelo y lenguaje de red originales.
- SQL: Structures Query Language, 1974, lenguaje específico, diseñado para administrar, y recuperar información de sistemas de gestión de base de datos.
- Ingres: 1970, soporte comercial de código abierto con destino de apoyar a las grandes aplicaciones comerciales y gubernamentales.
- Cálculo Relacional: Lenguaje de consulta que describe la respuesta deseada sobre una base de datos sin especificar cómo obtenerla.
- Consultas recursivas: Consulta en términos de una base de datos a problemas que se solucionan con recursividad, que consiste en que la solución a un problema determinado está expresada en términos del mismo problema.
- Modelo semántico: 1970, describe el significado de sus instancias, es una abstracción que define cómo los símbolos almacenados se relacionan con el mundo real.
- Base de datos orientada a objetos: 1960, cuando se integran las características de una base de datos con las de un lenguaje de programación orientado a objetos, el resultado es un sistema gestor de base de datos orientada a objetos.
- Base de Datos Multimedia: almacenan, además de texto, información de audio y vídeo, y permiten su manipulación y recuperación.
- Base de datos XML: Sistema software que da persistencia a datos almacenados en formato XML
- Base de datos móvil. Base de datos donde los usuarios pueden acceder a la información lejos de donde se encuentra almacenada la base de datos, se hace utilizando una conexión inalámbrica.
- Minería de Datos: 1990 aprox. Campo de la estadística y las ciencias de la computación referido al proceso que intenta describir patrones en grandes volúmenes de conjuntos de datos .

ETS Ing. Informática y de Telecomunicación, C/ Daniel Sancedo Aranda s/n, 18071, Granada (Spain), Tel: +34 958 244019, Fax: +34 958 243317



Ejercicios de autocomprobación del Tema 2

1. Explicar la relación existente entre los niveles de una base de datos y el concepto de independencia.

La relación existente entre los niveles de una base de datos, se explican mediante el concepto de transformación (o correspondencia entre niveles), y es el conjunto de normas que establecen cómo se definen los datos de un nivel en términos de otro. Con este concepto, se establece la independencia lógica y física. La independencia física es la que supone que el diseño de la base de datos es indiferente al almacenamiento real o físico de los datos, permitiendo alterar la estructura física sin alterar la lógica de la aplicación. La independencia lógica es la que persigue que los cambios en el esquema lógico general no afecten a las vistas de usuario de manera que las aplicaciones no necesitan ser modificadas con el objetivo de mejorar la seguridad, fiabilidad y decrementar los problemas.

2. Explicar la diferencia entre esquema externo y aplicaciones de usuario.

Una aplicación de usuario es un programa informático diseñado como herramienta para permitir a un usuario realizar uno o diversos tipos de tareas y el esquema externo es el nivel de abstracción que permite definir las información 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 a nivel conceptual.

3. Explica por qué, a tu juicio, no se han desarrollado DDLs a nivel interno.

El DDL es un lenguaje destinado a la definición de estructura de datos y esquemas en la base de datos. En el nivel conceptual es necesario el DDL, pero en el interno no, porque ahí es donde se ve representada físicamente la base de datos en el ordenador, por lo que no es necesario el DDL ahí.

4. Explica por qué, a tu juicio no se han desarrollado DMLs a nivel externo.

El DML es un lenguaje en el que podemos introducir datos en los esquemas, modificarlos, eliminarlos y consultarlos. El nivel externo coincide con la vista de usuarios, por lo que sería peligroso la existencia de DMLs a este nivel, ya que podría haber modificaciones no deseadas por parte de todos los usuarios que accedan a la base de datos.

5. Buscar tres ejemplos de lenguajes de cuarta generación. Indicar sus objetivos o funciones.

Shell: Es un programa que permite a los usuarios interactuar con el sistema, procesando las órdenes que se le indican.

Perl: Significa lenguaje práctico de extracción y reporte (Practica Extraction and Report

language). Su objetivo era simplificar las tareas habituales a realizar en el sistema operativo

Python: Es un lenguaje de scripting independiente de plataforma y orientado a objetos, preparada para realizar cualquier tipo de programa, desde aplicaciones Windows a servidores de red o incluso, páginas web. Es un lenguaje interpretado.

6. ¿Cuál es el enfoque actual del concepto de lenguaje anfitrión? Dar ejemplos de lenguajes anfitrión.

El lenguaje anfitrión es un lenguaje principal a partir del cual se desarrolla la actividad necesaria con la base de datos. Los programas de aplicación se escriben normalmente en lenguaje anfitrión. Para acceder a la base de datos, las instrucciones necesitan ser ejecutadas desde el lenguaje anfitrión. Algunos ejemplos de lenguaje anfitrión son Cobol, C, C++, Python o Java.

7. ¿Qué elementos conciernen al nivel interno de una base de datos?

El nivel interno es la representación física de la base de datos en el ordenador y cómo están almacenados los datos, así que todo lo que esté relacionado con la gestión de memoria sería un elemento importante en el nivel interno.

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

Gestión de credenciales y permisos sobre la escritura/lectura de datos organizados en partes. Bloqueando e informando de los intentos fallidos de acceso.

9. Explicar las ventajas de la arquitectura cliente/servidor a tres niveles.

Mayor flexibilidad para el usuario, que puede acceder desde varios dispositivos y reducción de costes, porque la gestión de las aplicaciones (instalación, actualización,...) se hacen al servidor, y no de cliente en cliente.

ETS Ing. Informática y de Telecomunicación, C/ Daniel Saucedo Aranda s/n, 18071, Granada (Spain), Tlf: +34 958 244019, Fax: +34 958 243311



Ejercicios de autocomprobación del Tema 3a y 3b

1. ¿Cuándo aparece el concepto de modelo de datos? ¿Por qué?

Un modelo de datos es un lenguaje orientado a hablar de una base de datos. Típicamente un modelo de datos permite describir las estructuras de datos de la base, es decir, los tipos de datos que hay en la base y la forma en la que se relacionan. Además permite hablar de las restricciones de integridad y sobre las operaciones de manipulación de datos que podamos necesitar realizar.

Otro enfoque es pensar que un modelo de datos permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

2. Diferencias entre los modelos de datos semánticos y los modelos de datos implementables.

El modelo de datos semántico es un modelo conceptual de datos en el que se incluye información semántica. Esto significa que el modelo describe el significado de sus instancias. Tal modelo de dato semántico es una abstracción que define cómo los símbolos almacenados (los datos de la instancia) se relacionan con el mundo real. Los hechos son típicamente expresados por relaciones binarias entre elementos de datos, mientras que las relaciones de orden superior se expresan como colecciones de relaciones binarias **“Objeto -<Tipo relación> - Objeto”**. Es una manera de estandarizar las relaciones. Facilitando así la creación de las bases de datos mediante un modelo de datos implementable mediante un lenguaje tipo sql.

3. Defina entidad y proporcione un ejemplo. Distinguir entre conjunto de entidades y entidad concreta.

Una entidad es un objeto concreto o abstracto que presenta interés para el sistema y sobre el que se recoge información la cual va a ser representada en un sistema de base de datos. La mayoría de las entidades modelan objetos o eventos del mundo real, por ejemplo, clientes, productos o llamadas de pedidos.

Una entidad puede ser concreta y real como una persona o un libro, o puede ser concreta y abstracta como un préstamo, unas vacaciones o un concepto. Mientras que un conjunto de entidades es un conjunto de entidades del mismo tipo que comparten las mismas propiedades o atributos.

4. Para el ejemplo de entidad antes definida, defina atributos y proporcione un ejemplo. Identificar la clave primaria de ésta.

Atributo es una característica o rasgo de un tipo de entidad que describe la entidad, por ejemplo, el tipo de entidad “Persona” tiene el atributo “Fecha de Nacimiento”.

Persona(**DNI**, Nombre y Apellidos, Fecha de nacimiento, nacionalidad);

5. Poner un ejemplo de atributo compuesto relacionado con la entidad definida.

Si suponemos que Persona([...], **Dirección**,[...]); Podemos ver que Dirección es claramente un atributo compuesto ya que podemos dividirlo en varios atributos independientes como lo es “Ciudad”, “Estado”, “Código Postal”....;

6. Definir relación y dar un ejemplo que involucre a la entidad antes definida.

Las relaciones de bases de datos son asociaciones entre tablas que se crean utilizando sentencias de unión para recuperar datos. Una relación funciona haciendo coincidir datos en columnas de clave, normalmente columnas que tienen el mismo nombre en ambas tablas. En la mayoría de los casos, la relación conecta la clave principal, o la columna de identificador único para cada fila, de una tabla a un campo de otra tabla.

Una “Persona” se relaciona con un “Banco” mediante una relación de tipo “Préstamo”;

7. Dar un ejemplo de relación de grado mayor que dos referente a la entidad antes definida.

Una “**Persona**” se relación con “**Padre**” y “**Madre**”;

8. Definir cardinalidad máxima y mínima y dar dos ejemplos: uno referente a una relación uno a muchos y otro referente a una relación muchos a muchos.

La cardinalidad mínima es el mínimo número de asociaciones que una instancia (fila) de una entidad (tabla) puede presentar en una relación conicidad con otra entidad.

La cardinalidad máxima es análogo y contrario, el máximo número de asociaciones....

Una relación (m:M) tiene de cardinalidad mínima “n” y máxima “M”, ejemplo obreros y tareas;

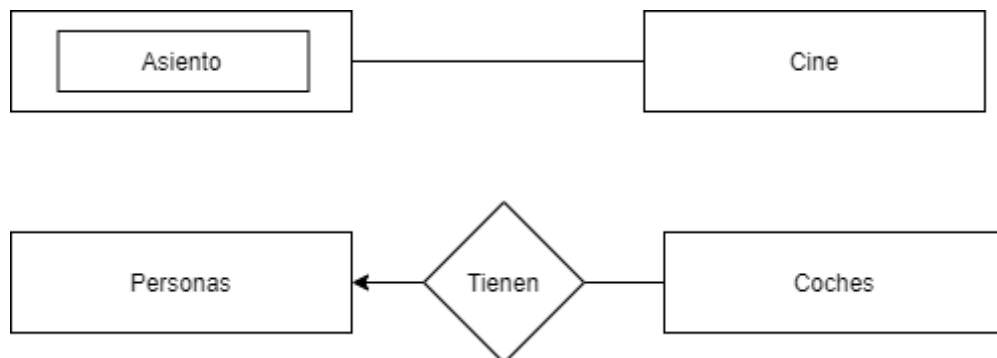
Una relación (1:M) tiene una cardinalidad mínima “0” y máxima “M”, ejemplo personas y coches;

9. Definir entidad débil, dar dos ejemplos diferentes de los estudiados en clase.

Es aquella cuya existencia depende de alguna otra entidad ya que este por sí sola no posee atributos suficientes para formar una clave primaria.

“Cuenta Bancaria” y “Transacciones”; “Sala Cine” y “Asientos”.

10. Distinguir mediante ejemplos el concepto de entidad débil de el de relación uno a muchos.



11. Establecer la conexión entre el concepto de entidad débil y el de atributo compuesto multivaluado. Ilustrar esta idea mediante dos ejemplos.

Una entidad débil está ligada directamente con una entidad fuerte al igual que el atributo compuesto multivaluado y ambos se traducen en nuevas filas y columnas en el modelo de tablas de la base de datos permitiendo añadir información extra.

12. Defina el concepto de subtipo, poner dos ejemplos que no se hayan explicado en clase.

Los subtipos son subconjuntos de un conjunto de datos. Generalmente se usan cuando se quiere tener datos de un tipo de dato, pues, aunque se diferencian en algunos aspectos, no dejan de ser de dicho tipo de dato. Básicamente es una agrupación de instancias dentro de un tipo de entidad, que debe representarse explícitamente debido a su importancia para el diseño o aplicación.

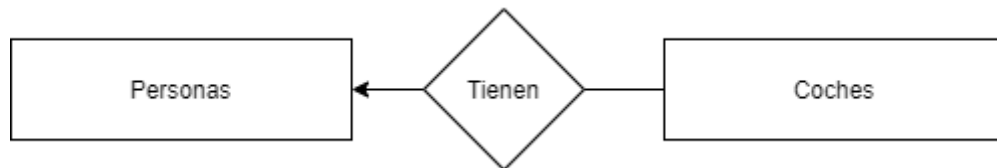
Ejemplo: Vehículo - [Coche, moto, turismo...]; Objeto Ikea - [Piezas, clavos, herramientas...]

13. Defina el concepto de herencia. ¿Como se relaciona con el caso anterior?.

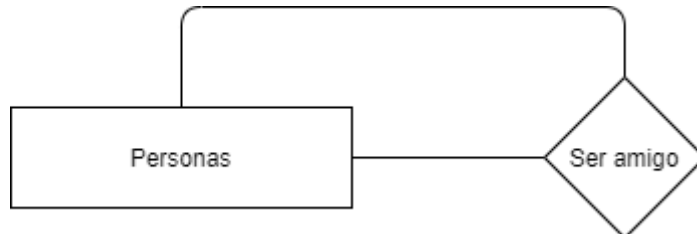
Un subtipo hereda todos los atributos del supertipo, y toda relación en la que participa el supertipo. Básicamente los “hijos” de un supertipo tendrán toda la información que el supertipo más un extra añadido por su propia particularidad.

15. Poner un ejemplo mediante el diagrama entidad/relación de cada una de las siguientes situaciones:

a) Relación con atributos uno a muchos.



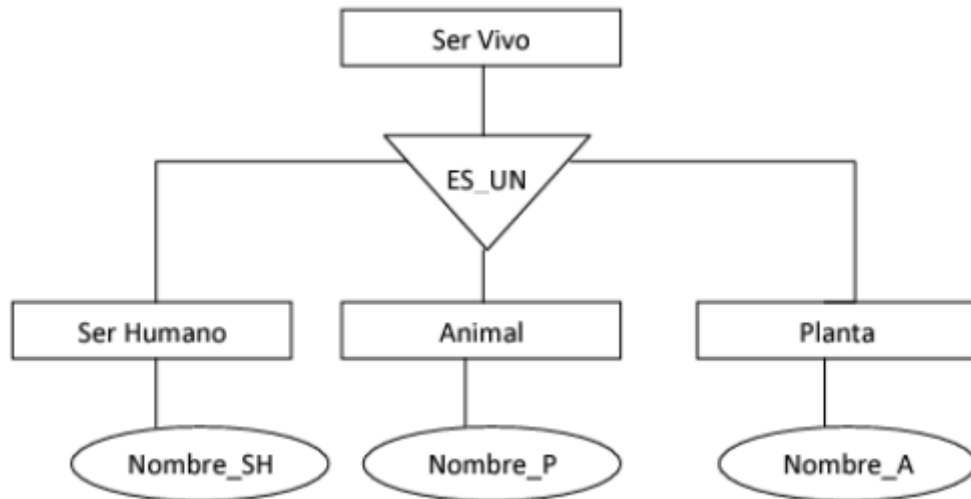
b) Relación involutiva muchos a muchos.



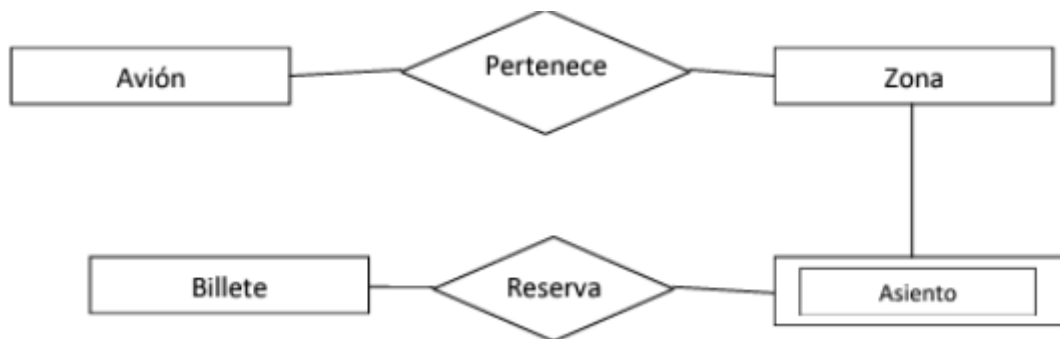
c) Una relación ternaria con algún atributo adicional.



d) Una relación ES-UN donde las subclases tengan atributos adicionales.



e) Entidades fuerte y débil donde esta última se conecte con otras entidades del diagrama.



16. Decidir de forma razonada cuándo se debe incluir una entidad nueva junto con una relación, o un atributo en cada uno de los siguientes casos. En el caso de que se trate de una entidad con una relación, especificar las características de la misma:

- a) La fecha de edición de un libro.
Atributo
- b) Los autores de un libro.
Entidad
- c) El título de una película.
Atributo
- d) Los actores de una película.
Entidad
- e) La valoración psicológica de un alumno.
Atributo
- f) La descripción de un accidente.
Atributo
- g) Los cuartos de baño que tiene un apartamento.
Atributo
- h) El programa de una asignatura.
Atributo
- i) El responsable de una investigación criminal.
Entidad
- j) La fecha de un crimen.
Atributo
- k) Los datos laborales de los trabajadores.
Atributo
- l) El horario de una asignatura.

Atributo

18. ¿Hasta que punto el modelo de datos DBTG es un modelo de datos físico?. Analizar las diferencias entre el modelo de datos físico y dicho modelo.

Modelo de datos físico: es un modelo específico de bases de datos que representa objetos de datos relacionales y sus relaciones.

Modelo de datos DBTG: un consorcio de industrias informáticas formada en 1959 con el objeto de regular el desarrollo de un lenguaje de programación estándar que pudiera ser utilizado en multitud de ordenadores.

19. Poner ejemplos de los distintos casos que aparecen cuando se pasa de una conexión en el modelo E/R a una tabla en el modelo relacional. Estudiar las distintas posibilidades de clave primaria que pueden aparecer.

1:1 La clave primaria será una de las dos claves primarias según conveniencia.

1:n La clave primaria es de la entidad “muchos”.

n:M la clave primaria es el conjunto de las claves primarias.

20. ¿Qué significa la expresión “navegar a puntero”?

Se refiere a una estructura de control utilizada para el recorrido (y potencial procesamiento) de los registros del resultado de una consulta.

21. ¿Porqué decimos que los modelos basados en grafos no mantienen realmente la independencia física?

Si se elimina un nodo, puede eliminar todos los hijos (si es la única vía de acceso a ellos) y por tanto cambiar la estructura.

22. ¿Qué significa la afirmación de que el modelo relacional es simétrico con respecto a las relaciones muchos a muchos?

Esto es, cuando se desea que no se pierda ninguna tupla de la tabla que está a la derecha, será necesario crear tuplas en la primera tabla, utilizándose el operador de combinación externa derecha. Puede que en algún caso sea necesario crear tuplas nulas en ambas relaciones, denominándose en este caso el operador combinación externa simétrica.

23. Poner tres ejemplos de conexiones en el modelo E/R que no se trasladen bien al modelo relacional.

Cuando hay una clave primaria (DNI) en varias entidades distintas.

Algunas entidades débiles que tengan atributos compuestos.

Herencia del tipo es_un.