

Resumen

Hola,

Este documento recoge las respuestas a todas las preguntas evaluables del Connolly. A diferencia de las respuestas que están subidas en el repositorio, estas son las respuestas a las preguntas propias del libro (4^o Edición) (no las de una versión completamente desfasada). Además, se incluyen una serie de cuestiones útiles acerca del desarrollo de la asignatura (que me gustaría que me hubieran dicho el primer día).

Este texto contiene multitud de erratas (gazapos “tipográficos”). Tómese la libertad de corregirlas (e incluso de pasarlo a L^AT_EX para que no dé tanto asco). (Nota obtenida: mucha).

Espero que sean de tu provecho.

Salu2.

1. Estructura del Examen.

El examen consta de las partes mencionadas en la guía docente: definiciones, preguntas cortas, MER, MR, Creación de Tablas y relleno de Consultas. **NO ENTRA NADA PRÁCTICO ACERCA DE NORMALIZACIÓN.** No es necesario aprobar las definiciones ni las preguntas cortas, sin embargo si el resto (ver sección *Criterios de Evaluación*).

El examen consta de, aproximadamente, 18 carillas **GRAPADAS** y no se permiten más folios. A continuación se muestra la estructura de cada una de las preguntas:

1.1. Definiciones formales.

Computa el 5 % de la nota del examen. Son 4 definiciones formales con un espacio predeterminado (2-3 líneas, dependiendo del tamaño de letra). Las definiciones deben ser **formales**: no vale explicar con tus propias palabras. Las definiciones formales son del estilo de los entregables (aunque no se limitan a ellos).

1.2. Preguntas cortas.

Computa el 25 % de la nota del examen. Son 5-6 preguntas extraídas **textualmente del Connolly** (de hecho es el motivo de este documento). Para cada pregunta hay un espacio de media carilla. Simplemente es reproducir lo que pone en este documento. A diferencia del apartado anterior, aquí sí se permite el uso de tus propias palabras.

1.3. Carilla en blanco.

Computa entre el 0.0 % de la nota el examen. Esta carilla se usa para responder a preguntas cortas que no tengan suficiente sitio o para realizar un borrador del MER. En el segundo caso puede computar de forma positiva, nunca negativa (puede ser a lápiz).

1.4. MER.

Computa 30 % de la nota del examen. Es un MER con un enunciado parecido a los de las prácticas (no numismática) de longitud de una carilla. Está dividido en apartados con una puntuación individual para cada uno. Uno de ellos corresponde a la semántica no reflejada al modelo (3 %).¹. El espacio disponible es de una carilla.

1.5. MR.

Computa el 15 % de la nota del examen. Es similar al ejercicio colgado en el repositorio ("*ejercicio de examen*"). No hay ninguna implicación semántica en el modelo, por lo que se requerirá **la aplicación estricta del estándar** (resolución de jerarquías, etc.).

1.6. Creación de tablas.

Computa el 5 % de la nota del examen. Consiste en crear un script SQL que cree la base de datos relacional proporcionada (municipios, hospitales, prestar_servicio). El script se crea por partes (una para municipios, otra para hospitales y otra para prestar_servicio) generando una tabla para cada parte (la última cuenta más). Hay que incluir también las políticas de integridad referencial (foreign key). Los tipos de datos de cada atributo vienen proporcionados.

¹Pro tip: se recomienda realizarlo primero en lápiz en la carilla en blanco y pasarlo a bolígrafo en el espacio de la respuesta

1.7. Consultas.

Computa el 20 % de la nota del examen. Se proporciona una consulta con huecos para rellenar. Hay tres tipos de consultas: 0.8 %, 1 % y 1.5 % de dificultad creciente. Probablemente ponga las consultas de hospitales, municipios y prestar_servicio. En ese caso

- Prima el left join sobre el inner join.
- La última consulta tiene un campo que se rellena exactamente igual que la anterior (viene resuelto).

No es necesario tener la consulta perfecta para que cuente algo. Basta con rellenar algunos de los campos bien.

2. Desarrollo de la asignatura.

2.1. Criterios de Evaluación.

El individuo subirá un pdf con la nota obtenida en cada parte. Como se ha mencionado antes, no es obligatorio aprobar las partes correspondientes a definiciones formales y preguntas cortas. Si es necesario aprobar las otras partes con las siguiente salvedad (criterios obtenidos mediante ingeniería inversa):

- Se requiere un 10.5 % en MER como mínimo.
- Se requiere un 7.5 % en Consultas como mínimo.

Los resultados respecto al MER son, como mucho, del 22 % (2015-2016).

2.2. Presentación Optativa.

El individuo propondrá a finales de cuatrimestre una presentación optativa sobre un tema de elección libre. La presentación no salva del suspenso en una de las partes del examen (menos de los mínimos ya indicados). Puede sumar en la parte de participación.

2.3. Entregables y proyecto inicial.

Tanto los entregables como el proyecto inicial (proyecto de esquí) no serán revisados por el profesor. En su lugar, serán evaluados por los propios alum-

nos y esa será la evaluación recibida final (se ha llegado al extremo de copiar hasta las iniciales del profesor y no ser detectado).²

Todos los entregables y el proyecto final están disponibles en este repositorio (no en el mismo orden).

2.4. Proyecto final.

Más de lo mismo, pero no está claro si son completamente evaluados por los alumnos.

3. Preguntas resueltas:

A continuación se muestran las preguntas cortas. Tienen varias erratas (no corregidas) y el capítulo 16 no ha sido resuelto (no entró en mi examen). Se ha realizado mediante la combinación del libro, las respuestas que están en el repositorio y mi propia cabeza.

Las preguntas que “no entran” fueron determinadas mediante la comparación de la localización de la respuesta en el libro con los capítulos (y apartados) de lectura obligatoria mencionados en la guía docente.

²Pro tip: para asegurar el punto de participación, rellenar todos los campos a la hora de la corrección (pese a que esté correcto y no haya nada que comentar). Al final se puede añadir un comentario breve e insulso sobre el trabajo.

CAPITULO 1

1. Proporcione cuatro ejemplos de sistemas de bases de datos distintos de los enumerados en la Sección 1.1

(Inventar)

2. Explique cada uno de los siguientes términos:

- Datos: para los usuarios finales constituye todos los valores diferentes conectados con los diferentes objetos7entidades que son motivo de preocupación para ellos.
- Base de Datos: colección compartida de datos lógicamente relacionados, junto con una descripción de estos datos, que están diseñados para satisfacer las necesidades de información de una organización.
- Sistema de gestión de bases de datos: un sistema software que permite a los usuarios definir, crear, mantener y controlar el acceso a la base de datos.
- Programa de aplicación de bases de datos: un programa informático que interactúa con la base de datos emitiendo las apropiadas solicitudes (normalmente una instrucción SQL) dirigidas al SGBD.
- Independencia de los datos: separación de las estructuras subyacentes de archivo de los programas que operan en ellos, también llamado independencia de datos del programa.
- Seguridad: capacidad de una base de datos de prevenir accesos no autorizados.
- Integridad: se refiere a la complementación y corrección de los datos en una base de datos.
- Vistas: una tabla virtual que no existe necesariamente en la base de datos pero es generada por el SGBD de las tablas base subyacentes cuando se accede a él. Estas presentan únicamente un subconjunto de la base de datos que es de particular interés para el usuario. Las vistas pueden ser modificadas para los requisitos particulares, por ejemplo, pueden cambiar los nombres de campo, y también proporcionan un nivel de seguridad que impide a los usuarios ver ciertos datos.

1. Describa el enfoque de tratamiento de los datos adoptado en los antiguos sistemas basados en archivos. Indique las desventajas de este enfoque.

El tratamiento de los datos adoptado en los antiguos sistemas basados en archivos es un enfoque descentralizado en el que cada departamento, con la ayuda de personal especializado en procesamiento de datos, almacenaba y controlaba sus propios datos.

Las desventajas son:

- Separación y aislamiento de los datos.
- Duplicación de los datos.
- Dependencias entre los datos.
- Formatos de archivos incompatibles.

- Consultas fijas/proliferación de programas de aplicación.

1. Describa las principales características del enfoque de base de datos y compárelas con la técnica basada en archivos.

Una base de datos es un repositorio centralizado, posiblemente de gran tamaño, compuesto por datos que pueden ser utilizados simultáneamente por múltiples departamentos y usuarios. En lugar de disponer de una serie de archivos desconectados con datos redundantes, todos los elementos están integrados, manteniéndose al mínimo las posibles duplicaciones. La base de datos deja de ser propiedad de un departamento y pasa a ser un recurso corporativo compartido. La base de datos también almacena la descripción de los datos.

(Falta la comparación. Se hace a partir del apartado anterior)

3. Describa los cinco componentes del entorno SGBD y explique cómo se relacionan entre sí.

- Hardware: sistemas informáticos en los que se ejecutan los programas de aplicación y los SGBD. Esto puede ir desde un solo PC hasta una red de computadoras.
- Software: incluye el software de los SGBD y los programas de aplicación, junto con el sistema operativo, incluyendo el software de red si el SGBD se usa en una red.
- Datos: los datos actúan como un puente entre los componentes de hardware y software y los componentes humanos. La base de datos contiene los datos operacionales y los metadatos ("datos sobre los datos")
- Procedimientos: instrucciones y reglamentos que rigen el diseño y uso de la base de datos. Esto puede incluir instrucciones sobre cómo iniciar sesión en el SGBD, hacer copias de seguridad de la BBDD y cómo manejar fallas de hardware y software.
- Personas: incluye a los diseñadores de bases de datos, administradores de bases de datos, los programadores de las aplicaciones y los usuarios finales.

1. (No entra en la guía docente)

4. Explique las ventajas y desventajas de los SGBD.

Algunas ventajas del enfoque de base de datos incluyen el control de redundancia de datos, consistencia de datos, intercambio de datos y mejora de la seguridad e integridad. Algunas de las desventajas incluyen la complejidad, costo y menor rendimiento y alto impacto de una falla.

CAPÍTULO 2

2.1 Explique el concepto de independencia de los datos y explique su importancia en un entorno de base de datos.

El concepto de independencia lógica de los datos hace referencia a la inmunidad de los esquemas externos a las modificaciones que se efectúen en el esquema conceptual.

Su importancia reside en que debe ser posible efectuar cambios en el esquema conceptual, como por ejemplo la adición o eliminación de nuevas entidades, atributos o relaciones, sin necesidad de modificar los esquemas externos existentes ni reescribir los programas de aplicación disponibles.

2.2 Para resolver la cuestión de la independencia de los datos, se propuso la arquitectura en tres niveles de ANSI-SPARC. Compare y contraste los tres niveles de este modelo.

- ☐ Nivel externo: es la vista que los usuarios tienen de la base de datos. Este nivel describe la parte de la base de datos que es relevante para cada usuario. Las diferentes vistas pueden constituir diferentes representaciones de los mismos datos.
- ☐ Nivel conceptual: es la vista comunitaria de la base de datos. Este nivel describe qué datos están almacenados en la base de datos y las relaciones existentes entre los mismos. Este nivel contiene la estructura lógica de toda la base de datos.
- ☐ Nivel interno: representación física de la base de datos en la computadora. Este nivel describe cómo están almacenados los datos en la base de datos. Cubre la implementación física de la base de datos.

2.3 ¿Qué es un modelo de datos? Indique y explique los tipos principales de modelos de datos.

Un modelo de datos es una colección integrada de conceptos para describir y manipular datos, las relaciones existentes entre los mismos y las restricciones aplicables a los datos, todo ello dentro de una organización.

Existen varios tipos de modelos de datos:

- ☐ Modelos de datos basados en objetos: utilizan conceptos tales como entidades, atributos y relaciones. Los tipos más comunes de modelos de datos son: entidad-relación, semántico, funcional y orientado a objetos. El modelo de datos orientado a objetos amplía la definición de entidad para incluir los atributos que describen el estado del objeto, sino también las acciones asociadas con el objeto (comportamiento).
- ☐ Modelos de datos basados en registros: la base de datos está compuesta por una serie de registros de formato fijo. Cada tipo de registros define un número fijo de campos. Existen tres tipos principales de modelos lógicos de datos basados en registros:
 - Modelo de datos relacional: está basado en el concepto de relaciones matemáticas. Los datos y relaciones se representan mediante tablas que tienen filas y columnas.
 - Modelo de datos en red: los datos se representan como colecciones de registros, mientras que las relaciones se representan mediante conjuntos.

- Modelo de datos jerárquico: tipo restringido de modelo en red. Solo permite que un nodo tenga un padre.
- Modelos de datos físicos: describen como se almacenan los datos en la computadora.

2.4 Explique la función y la importancia del modelado conceptual.

El modelado conceptual da soporte a todas las vistas externas y se apoya, a su vez, en el esquema interno. Es el corazón de la base de datos.

La importancia del modelado conceptual es que, si no es una representación completa y precisa de los requisitos de datos de la organización, parte de la información será representada de forma incorrecta o no será representada en absoluto.

2.5 Describa los tipos de servicios que cabe esperar que un SGBD multiusuario proporcione. (No entra)

2.6 De las distintas funciones descritas en respuesta a la Cuestión 2.5, ¿Cuáles cree que no serían necesarias en un SGBD autónomo para PC? Justifique su respuesta. (No entra)

2.7 Explique la función e importancia del catálogo del sistema. (No entra)

2.8 Describa los componentes principales de un SGBD e indique qué componentes podrían ser responsables de cada uno de los servicios identificados en la Cuestión 2.5. (No entra)

2.9 ¿Qué quiere decir el término 'arquitectura cliente-servidor' y cuáles son las ventajas de este enfoque? Compare la arquitectura cliente-servidor con las otras dos arquitecturas. (No entra)

2.10 Compare la arquitectura cliente-servidor en dos niveles para un SGBD tradicional con la arquitectura cliente-servidor en tres niveles. ¿Por qué resulta esta última arquitectura más apropiada para la Web? (No entra)

2.11 ¿Qué es un monitor TP? ¿Qué ventajas aporta un monitor TP a un entorno OLTP? (No entra)

CAPÍTULO 3

3.1 Explique cada uno de los siguientes conceptos en el modelo de datos relacional:

- ☐ Relación: una tabla con filas y columnas
- ☐ Atributo: una columna de nombre de una relación
- ☐ Dominio: conjunto de valores admisibles para uno o más atributos
- ☐ Tupla: fila de una relación
- ☐ Intensión y extensión: la intensión de una relación es la estructura de una relación, junto con la especificación de los dominios y cualesquiera restricciones sobre los posibles valores. Las tulas se denominan extensión (o estado) de una relación.
- ☐ Grado y cardinalidad: el grado de una relación es el número de atributos que contiene. La cardinalidad de una relación es el número de tuplas que contiene.

3.2 Describa la relación entre relaciones matemáticas y relaciones en el modelo de datos relacional.

En las relaciones matematicas, el producto cartesiano de elementos simples daba elementos de cada tupla de un unico valor. Las relaciones en el modelo de datos relacional, cada celda tiene solo un valor.

En una relación, los valores posibles para cada posición están determinados por el conjunto o dominio sobre el que sea posición está definida. En una tabla los valores de cada columna deben provenin del mismo dominio de atributo.

En un conjunto no hay elementos repetidos. En una tabla no hay tuplas repetidas.

Puesto que una relación es un conjunto, el orden de los elementos no importa. En una relación de la base de datos el orden de las tuplas es indiferente.

3.3. Describa las diferencias entre una relacion y un esquema de relación. ¿Qué es un esquema de base de datos relacional?

Un esquema de relación es una relación denominada definida por un conjunto de parejas de atributos y nombres de dominio. Una relación es un conjunto de asignaciones de los nombres de atributos a sus correspondientes dominios.

Un esquema de la base de datos relacional es un conjunto de esquemas de relación, cada uno con un nombre distinto.

3.4. Explique las propiedades de una relación.

Una tabla relacional tiene las siguientes propiedades:

- ☐ La tabla tiene un nombre que es distinto a todas las otras tablas de la base de datos.
- ☐ Cada celda de la tabla contiene exactamente un valor.
- ☐ Cada columna tiene un nombre distinto.
- ☐ Los valores de una columna son todos del mismo dominio.

- ☐ El orden de las columnas no tiene ninguna importancia. En otras palabras, siempre un nombre de columna se mueve junto con los valores de columna, nos podemos intercambiar las columnas.
- ☐ Cada registro es distinto; no hay duplicados.
- ☐ El orden de los registros no tiene ninguna importancia (teóricamente).

3.5. Explique las diferencias entre las claves candidatas y la clave principal de una relación. Explique el concepto de clave externa. ¿Cómo se relacionan las claves externas de una relación con las claves candidatas? Proporcione ejemplos para ilustrar su respuesta.

La clave principal es la clave candidata seleccionada para identificar tuplas únicamente dentro de una relación. Una clave externa es un atributo o conjunto de atributos dentro de una relación que se corresponden con la clave candidata de alguna (posiblemente la misma) relación.

La relación es que ambos atributos son comunes.

3.6. Defina las dos principales reglas de integridad para el modelo relacional. Explique por qué es deseable imponer esas reglas.

- ☐ Integridad de entidad: en una tabla base, ninguna columna de una clave primaria puede ser nulo.
- ☐ Integridad referencial: si existe una clave externa en una tabla, el valor de clave externa debe coincidir con el valor de clave candidata de alguna tupla en su relación de origen o el valor de la clave externa debe ser completamente nulo.

Es deseable imponer estas reglas pues garantizan que los datos son precisos.

3.7. ¿Qué es una vista? Explique la diferencia entre una vista y una relación base.

Una vista es el resultado dinámico de una o más operaciones relacionales que operan sobre las relaciones base para producir otra relación. Una vista es una relación virtual que no tiene por qué existir necesariamente en la base de datos, sino que puede producirse cuando se solicite por parte de un usuario concreto, generándose en el momento de la solicitud.

La diferencia es que una relación base es una relación nominada correspondiente a una entidad del esquema conceptual y cuyas tuplas **están almacenadas físicamente en una base de datos**.

CAPÍTULO 4 (Este capítulo no entra pues lo dijo en clase)

4.1. ¿Cuál es la diferencia entre un lenguaje procedimental y otro no procedimental? ¿Cómo clasificaría el álgebra relacional y el cálculo relacional?

Un lenguaje procedimental especifica que información queremos y no cómo obtenerla. Un lenguaje procedimental especifica exactamente que información queremos y cómo obtenerla.

El álgebra y el cálculo relacional son lenguajes procedimentales.

4.2. Explique los siguientes términos:

- ☐ Relacionalmente completo: lenguaje que puede usarse para producir cualquier relación que se pueda obtener mediante el cálculo relacional.
- ☐ Cierre de las operaciones relacionales: permite anidar las expresiones en el álgebra relacional pues tanto operandos como operadores son relaciones

4.3. Defina las cinco operaciones básicas del álgebra relacional. Defina las operaciones de combinación, intersección y división en términos de estas cinco operaciones básicas.

4.4. Explique las diferencias existentes entre las cinco operaciones básicas de combinación: combinación theta, equicombinación, combinación natural, combinación externa y semicombinación. Proporcione ejemplos para ilustrar su respuesta.

4.5. Compare y contraste el cálculo relacional de tuplas con el cálculo relacional de dominios. En concreto, explique la diferencia entre las variables de tuplas y de dominio.

4.6. Defina la estructura de una fórmula (bien formada) tanto en el cálculo relacional de tuplas como en el cálculo relacional de dominios.

4.7. Explique por qué una expresión del cálculo relacional puede ser insegura. Ilustre su respuesta con un ejemplo. Explique cómo garantizar que las expresiones del cálculo relacional sean seguras.

CAPÍTULO 5

5.1. ¿Cuáles son los dos componentes principales de SQL y que función tienen?

- ☐ Un lenguaje de definición de datos (DDL) para definir la estructura de base de datos.
- ☐ Un lenguaje de manipulación de datos (DML) para recuperar y actualizar datos.

5.2. ¿Cuáles son las ventajas y desventajas de SQL?

Ventajas:

- ☐ Es no procedimental.
- ☐ Es de formato prácticamente libre, lo que quiere decir que las diversas partes de las instrucciones no tienen por qué ser escritas en ubicaciones concretas de la pantalla.
- ☐ La estructura de los comandos está compuesta por palabras inglesas.
- ☐ SQL puede ser utilizado por diversos tipos de usuarios.
- ☐ Está estandarizado.

Desventajas:

No encontrado.

5.3. Explique la función de cada una de las cláusulas de la instrucción SELECT. ¿Qué restricciones se aplican a estas cláusulas?

- ☐ FROM: especifica la tabla o tablas que se utilizarán.
- ☐ WHERE: filtros de las filas sujeta a algunas condiciones.
- ☐ GROUP BY: grupos de filas con el mismo valor de la columna.
- ☐ HAVING: filtros de los grupos sujetos a algunas condiciones.
- ☐ SELECT: especifica que las columnas que aparecen en la salida.
- ☐ ORDER BY: especifica el orden de la salida.

5.4. ¿Qué restricciones se aplican al uso de instrucciones de agregación dentro de la instrucción SELECT?

Una función de agregación puede ser utilizada solamente en la lista de selección y en la cláusula HAVING.

Aparte de COUNT(*), cada función elimina primero los nulos y funciona solamente con los restantes valores no nulos. Count cuenta con todas las filas de una tabla, independientemente de si se producen valores nulos o valores duplicados.

5.5. Explique cómo funciona la cláusula GROUP BY. ¿Cuál es la diferencia entre las cláusulas WHERE y HAVING?

SQL aplica primero la cláusula WHERE. Luego, conceptualmente, arregla la tabla basándose en las columnas agrupadas. Después, aplica el HAVING y finalmente ordena el resultado según la cláusula ORDER BY.

WHERE filtra filas sujetas a la misma condición. HAVING filtra grupos sujetos a alguna condición.

5.6. ¿Cuál es la diferencia entre una subconsulta y una combinación (JOIN)? ¿Bajo que circunstancias no puede utilizarse una subconsulta?

Con una subconsulta, las columnas especificadas en el SELECT están restringidas a una tabla. De esta manera, no se puede usar una subconsulta si SELECT contiene columnas de más de una tabla.

CAPITULO 6

6.1. Describir los ocho tipos de datos base de SQL.

- ☐ Datos booleanos: están compuestos por los valores de verdad TRUE, FALSE. También soportan el valor de verdad UNKNOWN (como NULL). Los valores son comparables.
- ☐ Datos de caracteres: están compuestos por una secuencia de caracteres extraídos de un conjunto de caracteres definidos por la implementación. Hay dos tipos: CHAR (fijo) y VARCHAR (variable). Si se introducen menos caracteres en un tipo CHAR se rellenan con espacios.
- ☐ Datos de bit: se utilizan para definir cadenas de bits (0 o 1). El formato es BIT [VARYING] [longitud].
- ☐ Datos numéricos exactos: definen números con una representación exacta. Tiene precisión (total de dígitos significativos) y escala (número total de posiciones decimales).
- ☐ Datos numéricos aproximados: se utilizan para definir números que no tiene una representación exacta, como, por ejemplo, los números reales.
- ☐ Datos de fecha y hora: se utilizan para definir instantes temporales con un cierto grado de precisión.
- ☐ Datos de intervalo: se utiliza para representar periodos de tiempo. Cada tipo de datos de intervalo está compuesto por un subconjunto contiguo de los siguientes campos: YEAR, MONTH, DAY, HOUR, MINUTE, SECOND.
- ☐ Datos de gran tamaño: se utilizan para almacenar datos grandes.

6.2. Explique cuál es la funcionalidad y la importancia de la característica de mejora de la integridad.

La función de la mejora de integridad es proteger la base de datos frente a posibles incoherencias.

6.3. Explique cada una de las cláusulas de la instrucción CREATE TABLE.

- Default: cláusula opcional que se especifica para proporcionar un valor predeterminado para cada columna concreta. SQL utilizará este valor predeterminado siempre que se omita en una instrucción INSERT el valor correspondiente a una columna.
- NOT NULL, UNIQUE, CHECK: nada especial. Ver apuntes.
- PRIMARY KEY: especifica la columna o columnas que forman la clave principal de la tabla.
- FOREIGN KEY: especifica una clave externa en la tabla (hijo) y la relación que tiene con otra tabla (padre). Especifica una lista de columnas que formarán la clave externa, una subcláusula REFERENCES que indica la tabla padre, una regla opcional de actualización (ON UPDATE) y una regla opcional de borrado (ON DELETE).
- CHECK y CONSTRAINT: permiten definir restricciones adicionales.

6.4. Explique las ventajas y desventajas de las vistas.

6.5. Describa cómo funciona el proceso de resolución de vistas. (No entra)

El proceso de resolución de la vista combina la consulta con la consulta que define la vista:

- Los nombres de columna de la vista contenidos en la lista SELECT se traducen a sus correspondientes nombres de columna en la consulta de definición.
- Los nombres de vista en la cláusula FROM se sustituyen por las correspondientes listas FROM de la consulta de definición.
- La cláusula WHERE de la consulta del usuario se combina con la cláusula WHERE de la consulta de definición utilizando el operador lógico AND.
- Las cláusulas GROUP BY y HAVING se copian de la consulta de definición

6.6. ¿Qué restricciones son necesarias para garantizar que una vista sea actualizable? (No entra)

6.7. ¿Qué es una vista materializada y cuáles son las ventajas de mantener una vista materializada, en lugar de utilizar el proceso de resolución de vistas? (No entra)

6.8. Describa la diferencia existente entre el control de acceso discrecional y el control de acceso obligatorio. ¿Qué tipo de mecanismo de control soporta SQL? (No entra)

6.9. Describa cómo funcionan los mecanismos de control de acceso SQL. (No entra)

CAPÍTULO 7 (No entra)

CAPÍTULO 8 (No entra)

CAPÍTULO 9

9.1. Describa los componentes principales de un sistema de información.

Los sistemas de información incluyen las bases de datos, el software de base de datos, el software de aplicación, el hardware informático y el personal que utiliza y desarrolla el sistema.

9.2. Explique la relación existente entre el ciclo de vida de los sistemas de información y el ciclo de desarrollo de los sistemas de bases de datos.

Un sistema de información son los recursos que habilitan la recolección, administración, control y diseminación de datos/información a través de una compañía. La base de datos es un componente fundamental de un sistema de información. El ciclo de vida de un sistema de información está inherentemente asociado al ciclo de vida de la base de datos que la soporta.

Tipicamente, las fases del ciclo de vida de los sistemas de información incluyen: planificación, análisis de requisitos, diseño, prototipado, implementación, testeo, conversión y mantenimiento operativo. Como una base de datos es un componente fundamental de el sistema de información más largo, el ciclo de vida del sistema de desarrollo de base de datos está inherentemente asociado al ciclo de vida de los sistemas de información

9.3. Describa el propósito y actividades principales asociados con cada etapa del ciclo de desarrollo de un sistema de base de datos.

- Planificación de la base de datos: actividades de gestión que permiten llevar a cabo las distintas etapas del ciclo de vida del desarrollo de sistemas de base de datos de la forma más eficiente y efectiva posible.
- Definición del sistema: describe el ámbito y los límites de la aplicación de base de datos y las principales vistas de usuario.
- Recopilación y análisis de requisitos: el proceso de recopilar y analizar la información acerca de la parte de la organización a la que el sistema de base de datos tenga que dar soporte, y utilizar esta información para identificar los requisitos relativos al nuevo sistema.
- Vista de usuario: define qué es lo que se requiere de un sistema de base de datos desde la perspectiva de un determinado rol de la organización o de un área de aplicación empresarial. Hay tres enfoques de tratar con múltiples vistas de usuario:
- Diseño de la base de datos: el proceso de creación de un diseño que dé soporte a la misión y a los objetivos de la misión de la empresa para el sistema de base de datos requerido.
- Selección del SGBD: la selección de un SGBD apropiado para soportar el sistema de base de datos.

- Diseño de la aplicación: el diseño de la interfaz de usuario y de los programas de aplicación que permiten utilizar y procesar la base de datos.
- Prototipado: construcción de un modelo operativo de un sistema de base de datos.
- Implementación: la realización física del diseño de la base de datos y el diseño de las aplicaciones.
- Conversión y carga de los datos: transferencia de los datos existentes a la nueva base de datos y conversión de las aplicaciones existentes para que se ejecuten con la nueva base de datos.
- Pruebas: el proceso de operar el sistema de base de datos con la intención de localizar posibles errores.
- Mantenimiento operativo: el proceso de monitorizar y mantener el sistema de base de datos después de la instalación.

9.4. Explique lo que representa una vista de usuario en el contexto de un sistema de base de datos.

Una vista de usuario define que se requiere de un sistema de base de datos desde la perspectiva de un trabajo particular o un área de aplicación de la empresa.

9.5. Explique las técnicas principales para la gestión del diseño de un sistema de base de datos que tenga múltiples vistas de usuario.

Enfoque centralizado: los requisitos de cada vista de usuario se combinan en un único conjunto de requisitos para el nuevo sistema de base de datos. Durante la etapa de diseño de la base de datos se crea un modelo de datos que representa todas las vistas de usuario.

Enfoque de integración de vistas: los requisitos de cada vista de usuario se mantienen en listas separadas. Durante la etapa de diseño de la base de datos se crean y combinan los modelos de datos que representan cada una de las vistas de usuario.

Combinación de las dos anteriores.

9.6. Realice una comparación entre las tres fases del diseño de una base de datos.

Diseño conceptual de la base de datos: proceso de construcción de un modelo de los datos utilizados en una empresa, de forma independiente de todas las consideraciones físicas.

Diseño lógico de la base de datos: proceso de construcción de un modelo de los datos utilizados en una empresa basándose en un modelo de datos específico, pero de forma independiente de un SGBD concreto y de cualquier otra consideración física.

Diseño físico de la base de datos: el proceso de generar una descripción de la implementación de la base de datos en el almacenamiento secundario; describe las relaciones base, la organización de los archivos y los índices utilizados para conseguir un acceso eficiente a los datos, así como cualesquiera medidas de seguridad y restricciones de integridad asociadas.

9.7. ¿Cuáles son los objetivos principales del modelado de datos? Identifique los criterios aplicables a un modelo de datos óptimo.

9.8. Identifique la etapa o etapas en las que resulta apropiado seleccionar un SGBD y describa una técnica para elegir el mejor SGBD.

Las etapas apropiadas para seleccionar un SGBD son entre las fases de diseño conceptual y lógico de la base de datos.

Una técnica para seleccionar un SGBD es: definir los términos de referencia del estudio, seleccionar entre dos o tres productos candidatos, evaluación de los productos y la recomendación de un producto y su correspondiente generación de un informe.

9.9 El diseño de aplicaciones implica el diseño de las transacciones y el diseño de la interfaz de usuario. Describa el propósito y las actividades principales asociados con cada una de esas tareas.

El diseño de aplicaciones y de la base de datos son actividades paralelas en el ciclo de vida del desarrollo de bases de datos. En la mayoría de los casos, no se puede completar el diseño de la aplicación hasta que el diseño de la base de datos ha tenido lugar. Por otra parte, la base de datos existe para dar apoyo a las aplicaciones y, por lo tanto, debe haber un flujo de información entre el diseño de la aplicación y el diseño de la base de datos.

Las dos actividades principales asociadas con el diseño de la aplicación son el diseño de la interfaz de usuario y de las aplicaciones que usarán y procesarán la base de datos.

Nos debemos asegurar de que toda la funcionalidad que se dice en los requerimientos esté presente en el diseño de la aplicación para la base de datos. Esto implica diseñar la interacción entre el usuario y los datos, lo que denominamos diseño de transacciones. Además de diseñar como conseguir la funcionalidad requerida, tenemos que diseñar una interfaz de usuario apropiada para el sistema de base de datos.

9.10. Explique por qué las pruebas no pueden demostrar la ausencia de fallos, sino sólo la presencia de los mismos.

9.11. Describa las principales ventajas de utilizar la técnica de prototipado a la hora de construir un sistema de bases de datos.

El propósito de desarrollar un prototipo del sistema de base de datos es permitir a los usuarios usar el prototipo para identificar las características del sistema que funcionan bien o mal y, si es posible, sugerir mejoras o incluso nuevas funcionalidades para el sistema de base de datos. De esta manera, podemos clarificar en gran medida los requerimientos y evaluar la posibilidad e realizabilidad de un diseño particular de un sistema. Los prototipos deben tener la gran ventaja de ser relativamente baratos y rápidos de construir.

9.12. Defina el propósito y las tareas asociados con la administración de datos y la administración de bases de datos.

Administración de datos: gestión de los recursos, lo que incluye la planificación de la base de datos, el desarrollo y el mantenimiento de estándares, políticas y procedimientos, así como el diseño conceptual y lógico de la base de datos.

Administración de bases de datos: la gestión de la implementación física de un sistema de bases de datos, lo que incluye el diseño físico de la base de datos y su implementación, la configuración de los controles de seguridad e integridad, la monitorización de las prestaciones del sistema y la reorganización de la base de datos según sea necesario.

CAPÍTULO 10 (No entra)

CAPÍTULO 11

11.1. Describa qué es lo que representan los tipos de entidad en un modelo ER y proporcione ejemplos de entidades con existencia física o conceptual.

Tipo de entidad es un conjunto de objetos con las mismas propiedades, los cuales son identificados por un usuario o compañía como poseedoras de una existencia independiente. Cada objeto, el cual debería ser unívocamente identificable en un conjunto, se denomina una instancia de entidad. Un tipo de entidad tiene existencia independiente y puede representar objetos con una existencia física o conceptual.

Existencia física: miembro, video, rama.

Existencia conceptual: rol, renta, registro.

11.2. Describa qué es lo que representan los tipos de relación en un modelo ER y proporcione ejemplos de relaciones unarias, binarias, ternarias y cuaternarias.

Un tipo de relación es un conjunto de asociaciones significativas entre entidades. Como con las entidades, cada asociación debería ser unívocamente identificable en un conjunto. Cada relación tiene un nombre que describe su función.

Las entidades involucradas en una relación particular se denominan participantes. El número de participantes en una relación se denomina grado y indica el número de entidades involucradas en una relación. En una relación de grado uno se denomina unaria, también llamada recurvisa. Una relación recursiva describe una relación donde la misma entidad participa con varios roles. Ej: supervisar.

Una relación con un grado mayor que 2 se denomina relación compleja. Una relación de grado 3 se denomina ternaria.

11.3. Describa qué es lo que representan los atributos en un modelo ER y proporcione ejemplos de atributos simples, compuestos, univaluados, multivaluados y derivados.

Un atributo es una propiedad de una entidad o una relación.

Los atributos representan que queremos saber sobre las entidades. Representan también la principal fuente de datos almacenados en la base de datos.

- Atributo simple: es un atributo compuesto de un solo componente. No se pueden subdividir. Ejemplo: precio
- Atributo compuesto: es un atributo compuesto de varios componentes. Se pueden dividir para dar componentes más pequeñas de existencia independiente. Ej: nombre se puede subdividir en nombre y apellidos.
- Atributo univaluado: atributo que tiene un sólo valor para una instancia de entidad. La mayoría de atributos son univaluados para una instancia de entidad. Ej: dni.

- Atributo multivaluado: es un atributo que tiene varios valores para una instancia de entidad. Algunos atributos tienen varios valores para una instancia de entidad. Ej: entidad video, atributo categorías (infantil, comedia).
- Atributo derivado: representa un atributo que se puede derivar del valor de un atributo relacionado o conjunto de atributos no necesariamente en la misma entidad. Algunos atributos pueden estar relacionados para una entidad particular. Ej: edad = fAct - fNac.

11.4. Describa qué representa la restricción de multiplicidad para un tipo de relación.

Es el número de instancias de una entidad que pueden relacionarse con una instancia de la entidad asociada.

11.5. ¿Qué son las restricciones empresariales y cómo modela la multiplicidad estas restricciones?

La multiplicidad restringe el número de instancias que se pueden relacionar con otra entidad a través de una relación. La multiplicidad es la representación de las políticas establecidas por el usuario o compañía y se refiere como restricción empresarial. Asegurarse de que todas las restricciones empresariales son identificadas y representadas es una parte importante de modelar una compañía.

La multiplicidad para una relación binaria puede ser 1:1, 1:* o *:*.

11.6. ¿Cómo representa la multiplicidad tanto la cardinalidad como la restricción de participación en un cierto tipo de relación?

La multiplicidad realmente consiste en dos restricciones separadas conocidas como la cardinalidad y la participación.

- Cardinalidad: describe el número de posibles relaciones para cada entidad participante.
- Participación: determina si todas o sólo algunas entidades participan en una relación. La cardinalidad de una relación binaria es lo que se refiere como 1:1, 1:* o *:. La restricción de participación representa si si todas las instancias de entidad están involucradas en una relación particular o sólo en algunas.

11.7. Proporcione un ejemplo de tipo de relación con atributos.

Relación actual entre actor y vídeo. Tiene un atributo llamado personaje que indica el personaje representado por el actor.

11.8. Indique en qué difieren los tipos de entidad fuertes y débiles y proporcione un ejemplo de cada uno.

Los tipos de entidad se clasifican como fuertes o débiles. Una entidad fuerte no depende de la existencia de otra entidad para su clave primaria. Una entidad débil es parcial o totalmente dependiente de la existencia de otra entidad o entidades para su clave primaria. Ej: actor y video son entidades fuertes y rol es débil de actor y vídeo.

11.9. Describa cómo pueden aparecer las trampas multiplicativas y las trampas de corte en un modelo ER y cómo se pueden resolver dichos problemas.

Una trampa multiplicativa puede ocurrir cuando dos entidades tienen una relación 1:* que se abren en abanico de una tercera entidad pero las dos entidades deberían tener una relación directa entre ellas para proveer la información necesaria. Una trampa multiplicativa puede ser resuelta mediante la adición de una relación directa entre las dos entidades que originalmente estaban separadas por una tercera entidad.

Una trampa de corte puede ocurrir cuando en un ER se sugiere la existencia de una relación entre entidades pero el camino no existe entre ciertas instancias de entidad. Específicamente, una trampa de corte puede ocurrir cuando hay una relación con participación opcional que forma parte del camino entre entidades que están relacionadas. De nuevo, una trampa de corte puede ser resuelta por la adición de una relación directa entre dos entidades que originalmente estaban relacionadas a través de un camino que incluía participación opcional.

CAPÍTULO 12

12.1. Describa lo que representan una superclase y una subclase.

Superclase: un tipo de entidad que incluye uno o más subgrupos diferentes de sus instancias, los cuales es preciso representar en un modelo de datos.

Subclase: un subgrupo diferenciado de instancias de un tipo de entidad, que necesita ser representado en un modelo de datos.

12.2. Describa la relación entre una superclase y su subclase.

Cada miembro de una subclase es también miembro de la superclase, i.e., la entidad contenida en la subclase es la misma que la contenida en la superclase, aunque tiene un papel distinto. La relación entre una superclase y una subclase es de tipo 1:1 y se denomina relación superclase/subclase. Algunas superclases pueden contener subclases solapadas.

12.3. Describa el proceso de herencia de atributos e ilústrelo con un ejemplo.

El proceso de herencia de atributos consiste en que una subclase posee sus propios atributos específicos además de todos los atributos asociados a la superclase.

Ejemplo: empleadoPorHoras hereda todos los atributos de empleado además de tener sus propios atributos específicos.

12.4. ¿Cuáles son las principales razones para introducir los conceptos de superclases y subclases en un modelo ER?

Hay dos principales razones para introducir los conceptos de superclases y subclases en un modelo ER: en primer lugar nos evita describir conceptos similares más de una vez, ahorrando tiempo al diseñarlos y haciendo que el diagrama ER sea más legible. En segundo lugar, añade más información semántica al diseño en una forma que resulta familiar para muchas personas.

12.5. Describa qué es lo que representa una subclase compartida y cómo se relaciona este concepto con el de herencia múltiple.

Una subclase con más de una superclase se denomina subclase compartida. En otras palabras, un miembro de una subclase compartida debe ser miembro de las superclases asociadas. Como consecuencia, la subclase compartida hereda los atributos de todas las superclases, además de poder tener sus propios atributos adicionales (herencia múltiple).

12.6. Describa los procesos de especialización y generalización e indique sus diferencias.

Especialización: es el proceso de maximizar las diferencias entre miembros de una entidad identificando sus características distintivas.

Generalización: es el proceso de minimizar las diferencias entre entidades identificando sus características comunes.

La principal diferencia es que la especialización va de 'arriba a abajo' y la generalización de 'abajo a arriba'.

12.7. Describa las dos restricciones principales que se aplican a las relaciones de especialización/generalización.

Restricción de participación: determina si todo miembro de la superclase debe participar como miembro de una subclase (obligatoria o opcional).

Restricción de disyunción: describe la relación entre miembros de las subclases e indica si es posible que un miembro de una superclase sea miembro de una subclase o de más de una.

12.8. Describa los conceptos de agregación y composición, indique en qué se diferencian y proporcione un ejemplo de cada uno de esos conceptos. (no entra).

CAPÍTULO 13

13.1 Describa el propósito del proceso de normalización de los datos.

El propósito de la normalización es identificar un conjunto de relaciones que soporten adecuadamente los requisitos de datos de la organización.

13.2. Explique las formas alternativas en que puede usarse la normalización como ayuda al diseño de una base de datos.

La normalización puede usarse en el diseño de una base de datos de dos maneras: la primera es usar la normalización como un enfoque de abajo a arriba en el diseño; la segunda es usar la normalización en conjunto con el modelado ER.

Usando la normalización como un enfoque de abajo a arriba implica analizar las asociaciones entre atributos y, basándose en ese análisis, agrupar los atributos juntos para formar tablas que representan entidades y relaciones. Sin embargo, este enfoque se vuelve difícil con un número grande de atributos. De forma alternativa, se puede usar un enfoque de arriba a abajo en el diseño de la base de datos. En este enfoque, se usa el modelo ER para crear un modelo de datos que representa las entidades principales y las relaciones. Es ahí entonces cuando traducimos el modelo ER en un conjunto de tablas que representan esos datos. Es en este punto cuando usamos la normalización para comprobar que las tablas están bien diseñadas.

13.3. Describa los tipos de anomalías de actualización que puede introducirse en una normalización que tenga datos redundantes.

Las tablas que tienen datos redundantes pueden tener problemas de actualización, los cuales son clasificados como de inserción, borrado o modificación.

Inserción: si insertamos un dato en una tabla es posible que se inserte un nulo en una tabla con atributo de clave primaria, lo que no es posible. Asimismo, insertar en una tabla puede crear inconsistencias entre los datos de distintas tablas.

Borrado: si se elimina un dato de una tabla que representa el último dato de su tipo es posible que se eliminen todos los detalles sobre información asociada a ese dato.

Modificación: si se quiere cambiar el valor de un dato particular de una columna se tienen que actualizar todos los valores que sean iguales al dato particular (i.e, que estén relacionados).

13.4. Describa el concepto de dependencia funcional.

Dependencia funcional: describe la relación entre atributos de una relación. EJ: si A y B son atributos de la relación R, B será funcionalmente dependiente de A ($B \rightarrow A$) si cada valor de A está asociado con exactamente un valor de B.

13.5. ¿Cuáles son las características principales de las dependencias funcionales que se utilizan para la normalización?

- Hay una relación 1:1 entre los atributos del lado izquierdo (determinante) y los del lado derecho de la dependencia funcional.
- Se cumplen en todo instante de tiempo.
- El determinante tiene el número mínimo de atributos necesario para mantener la dependencia con los atributos del lado derecho. En otras palabras, debe haber una dependencia funcional completa entre los atributos del lado izquierdo y los del lado derecho de la dependencia.

13.6. Describa cómo identifica normalmente un diseñador de base de datos el conjunto de dependencias funcionales asociadas con una relación.

Generalmente se identifica mediante conversaciones con los usuarios o a través de documentación apropiada. Si no hay disponibles usuarios o documentación, el diseñador de la base de datos deberá utilizar su sentido común y su experiencia.

13.7. Describa las características de una tabla no normalizada (UNF, Unnormalized Form) y describa cómo puede convertirse dicha tabla en una relación en primera forma normal (1NF).

La regla para primera forma normal es que una tabla en la cual la intersección de cada columna y tupla debe contener un solo valor. En otras palabras, una tabla que contiene más de un valor atómico en la intersección de una o más columnas no está en 1NF. La tabla en UNF se puede convertir en 1NF reestructurando la tabla original eliminando la columna con los multivalores junto con una copia de la clave primaria para crear una nueva tabla. La ventaja de esta aproximación es que las tablas resultantes pueden estar en formas normales mayores que la 1ª.

13.8. ¿Cuál es la mínima forma normal que una relación debe satisfacer? Proporcione una definición para dicha forma normal.

Sólo la 1NF es crítica para crear tablas adecuadas para bases de datos relacionales. Todas las subsecuentes formas normales son opcionales. Sin embargo, para evitar las anomalías de actualización es recomendable llegar hasta 3NF.

1NF es una tabla en la cual la intersección de cada columna y fila contiene un solo valor.

13.9. Describa las dos técnicas existentes para convertir una tabla en forma no normalizada (UNF) en una relación o relaciones en primera forma normal.

- Introduciendo datos apropiados en las columnas vacías de las filas que contiene los datos repetidos. En otras palabras, llenamos los blancos duplicando los datos no repetitivos siempre que sea necesario.
- Colocando los datos repetitivos, junto con una copia de los atributos clave originales en una relación independiente. Algunas veces la tabla no normalizada puede contener más de un grupo repetitivo o puede contener grupos repetitivos dentro de otros grupos repetitivos. En tales casos, se aplica esta técnica de forma sucesiva hasta que no quede ningún grupo repetitivo.

13.10. Describa el concepto de dependencia funcional completa e indique cómo se relaciona este concepto con la forma 2NF. Proporcione un ejemplo para ilustrar su respuesta.

La definición formal de 2NF es una tabla que está en primera forma normal y en la que cada columna que no sea de clave primaria es dependiente funcionalmente de forma completa de su clave primaria. La dependencia funcional completa indica que si A y B son columnas de la tabla, B es dependiente funcionalmente de forma completa de A si B no es dependiente de ningún subconjunto de A. Si B es dependiente de un subconjunto de A, esto se denomina dependencia parcial. Si existe una dependencia parcial en la clave primaria, la tabla no puede estar en 2NF. La dependencia parcial debe ser eliminada para alcanzar 2NF.

13.11. Describa el concepto de dependencia transitiva e indique cómo se relaciona este concepto con la forma 3NF. Proporcione un ejemplo para ilustrar su respuesta.

La definición formal de 2NF es una tabla que está en primera y segunda forma normal y en la cual no hay columnas que no sean de clave primaria dependientes de forma transitiva de la clave primaria. Una dependencia transitiva es un tipo de dependencia funcional que ocurre cuando en un tipo particular de relación se mantiene entre las columnas de una tabla. Por ejemplo, considera una tabla con columnas A, B y C. Si B es funcionalmente dependiente de A, y C es funcionalmente dependiente de B entonces C es transitivamente dependiente de A a través de B. Si existe una dependencia funcional transitiva en la clave primaria, la tabla no está en 3NF.

13.12. Explique cómo difieren las definiciones de las formas 2NF y 3NF basadas en las claves principales de las definiciones generales de formas 2NF y 3NF. Proporcione un ejemplo para ilustrar su respuesta.

La diferencia reside en que en la definición general se utilizan todas las claves candidatas para dar la definición de 2NF y 3NF mientras que en el caso de las claves principales sólo se usa la clave principal (primaria).

CAPÍTULO 14

14.1. Describa el propósito de utilizar reglas de inferencia para identificar dependencias funcionales en una relación dada.

El propósito de utilizar reglas de inferencia es reducir el conjunto de dependencias funcionales a uno más pequeño. Para ello, se identifican las dependencias funcionales implícitas.

14.2. Explique el propósito de los Axiomas de Armstrong.

El propósito de los Axiomas de Armstrong es inferir nuevas dependencias funcionales a partir de otras dadas.

14.3. Explique el propósito de la forma normal de Boyce-Codd (BCNF) y explique cómo difiere la forma BCNF de la forma 3NF. Proporcione un ejemplo para ilustrar su respuesta.

14.4. Describa el concepto de dependencia multivaluada y explique cómo se relaciona este concepto con la forma 4NF. Proporcione un ejemplo para ilustrar su respuesta.

14.5. Describa el concepto de dependencia multivaluada y explique cómo se relaciona este concepto con la forma 5NF. Proporcione un ejemplo para ilustrar su respuesta.

CAPÍTULO 15 (No entra)

CAPÍTULO 16.

