

Bases de Datos I
16/09/2020
Preguntas Cap 1 16/09
IS-501
José Inestroza

Preguntas del Capitulo 01 "Base de Datos y Usuarios de Base de Datos" del libro "Fundamentals of Database Systems" de los autores Ramez Elmasri, Shamkant B. Navathe, 7ma edición del año 2015.

- 1. ¿Para qué sirve la normalización?**
R/ Sirve para garantizar coherencia y ahorrar espacio de almacenamiento al almacenar cada elemento lógico de datos en un solo lugar de la base de datos.
- 2. ¿Qué es información persistente?**
R/ Es Información o datos que sobreviven a la finalización de la ejecución y puede luego ser recuperada directamente por otro programa.
- 3. ¿Qué es un DBA y cuál es su labor?**
R/ DBA es un administrador de bases de datos; El DBA es responsable de autorizar el acceso a la base de datos, coordinando y monitoreando su uso, y adquiriendo software y recursos de hardware según sea necesario. El DBA es responsable de problemas como violaciones de seguridad y tiempo de respuesta deficiente del sistema. En organizaciones grandes, el DBA es asistido por un personal que realiza estas funciones.
- 4. ¿Para qué sirve la integridad referencial?**
R/ Es un tipo de restricción de las más frecuentes, que implica especificar que un registro en un archivo debe estar relacionado con otros registros en otros archivos.
- 5. ¿Qué es información y cómo se diferencia de un dato?**
R/ La información son datos con un objetivo particular o datos que tienen un uso en un sistema; Los datos brutos están desorganizados y son inútiles por sí solos. Es solo cuando se a procesan y organizan cuándo comienzan a convertirse en información.
- 6. ¿Qué son las reglas de negocio y cómo se traducen en una base de datos?**
R/ Son un conjunto de restricciones para el manejo de datos, en una base de datos se traducen en restricciones especificadas en el DBMS.

7. ¿Qué lenguaje permite realizar esta tarea?

R/ SQL.

8. ¿Cuáles son los tipos de usuarios finales de un sistema?

R/ Los usuarios finales son las personas cuyos trabajos requieren acceso a la base de datos para realizar consultas, actualizar y generar informes; la base de datos existe principalmente para su uso. Hay varias categorías de usuarios finales:

* Los usuarios finales ocasionales acceden ocasionalmente a la base de datos, pero pueden necesitar información diferente cada vez.

* Los usuarios finales ingenuos o paramétricos constituyen una parte considerable de la base de datos usuarios finales. Su principal función laboral gira en torno a la consulta constante y actualizar la base de datos, utilizando tipos estándar de consultas y actualizaciones llamadas transacciones enlatadas, que han sido cuidadosamente programadas y probado.

* Los usuarios finales sofisticados incluyen ingenieros, científicos, analistas de negocios y otros que se familiaricen completamente con las instalaciones del DBMS para implementar sus propias aplicaciones para satisfacer sus complejos requisitos.

* Los usuarios independientes mantienen bases de datos personales mediante el uso de paquetes de programas listos para usar que brindan recursos basados en menús o gráficos fáciles de usar interfaces.

9. ¿Cuándo no se debe usar un DBMS/SGBD?

R/ 1. Alta inversión inicial en hardware, software y capacitación.

2. La generalidad que proporciona un DBMS para definir y procesar datos.

3. Gastos generales para proporcionar funciones de seguridad, control de simultaneidad, recuperación e integridad.

10. ¿Qué es redundancia?

R/ Se refiere al hecho de almacenar varias veces el mismo dato.

11. ¿Qué es el universo de datos?

R/ Es el lugar donde se almacenan todos los datos que corresponden a una base de datos.

12. ¿Qué es una base de datos?

R/ Una base de datos es una colección de datos ordenados lógicamente, coherentes con algún sentido. Una variedad aleatoria de datos no se puede denominar correctamente base de datos.

13. **¿Qué son las restricciones de integridad?**
R/ Son un conjunto de restricciones para mantener los datos integros es decir, mantener los datos correctos en su correcta posición.
14. **¿Cuáles son los actores que intervienen en la gestión de los datos?**
R/ * Administradores de bases de datos
* Diseñadores de bases de datos
* Usuarios finales
* Analistas de sistemas y programadores de aplicaciones(Ingenieros de software)
* Los diseñadores e implementadores de sistemas DBMS
* Los desarrolladores de herramientas
* Operadores y personal de mantenimiento
15. **¿Qué software permite realizar esta gestión anterior?**
R/ El DBMS o SGBD Sistema gestor de bases de datos como ser: Oracle8, PostgreSQL, Access, MS-SQL Server, MySQL.
16. **¿Qué significa definir una base de datos?**
R/ Significa determinar ciertas características y restricciones que debe tener la base de datos para su correcto uso, como ser:
* Controlar la redundancia
* Restringir el acceso no autorizado
* Proporcionar almacenamiento persistente para objetos de programa
* Proporcionar estructuras de almacenamiento y búsqueda Técnicas para un procesamiento de consultas eficiente
* Proporcionar respaldo y recuperación
* Proporcionar múltiples interfaces de usuario
* Representar relaciones complejas entre datos
17. **¿Qué es un buffer?**
R/ Es un espacio de **memoria**, en el que se almacenan datos de manera temporal, normalmente para un único uso (generalmente utilizan un sistema de cola **FIFO**); su principal uso es para evitar que el programa o recurso que los requiere, ya sea **hardware** o **software**, se quede sin datos durante una transferencia (entrada/salida) de datos irregular o por la velocidad del proceso.
18. **Históricamente hablando ¿Cómo es la relación histórica del procesamiento de datos?**
R/ La mayoría de los primeros sistemas de bases de datos se implementaron en grandes y costosas computadoras mainframe que comenzaron a mediados de la década de 1960 y continuaron durante

las décadas de 1970 y 1980. Los principales tipos de sistemas tempranos se basaron en tres paradigmas principales: sistemas jerárquicos, sistemas basados en modelos de red y sistemas de archivos invertidos. Progresivamente el modelo de datos relacionales también introdujo lenguajes de consulta de alto nivel que proporcionaron una alternativa al lenguaje de programación interfaces, por lo tanto, la abstracción de datos y la independencia de los datos del programa mejoraron mucho en comparación con los sistemas anteriores.

La aparición de lenguajes de programación orientados a objetos en la década de 1980 y La necesidad de almacenar y compartir objetos complejos y estructurados condujo al desarrollo de Bases de datos orientadas a objetos (OODB). Ellos también incorporó muchos de los paradigmas útiles orientados a objetos, como abstracción de datos, encapsulación de operaciones, herencia e identidad de objeto, muchos conceptos orientados a objetos se incorporaron en las versiones más nuevas de DBMS relacionales, lo que llevó a sistemas de gestión de bases de datos, conocidos como ORDBMS.

A partir de la década de 1990, el comercio electrónico (e-commerce) surgió como un importante aplicación en la Web. Gran parte de la información crítica sobre el comercio electrónico Web páginas son datos extraídos dinámicamente de DBMS, Se desarrollaron una variedad de técnicas para permitir el intercambio de datos extraídos dinámicamente en la Web EXTended Markup Language (XML) es un estándar para intercambiar datos entre varios tipos de bases de datos y páginas web. XML combina conceptos a partir de los modelos utilizados en sistemas de documentos con conceptos de modelado de bases de datos.

En la primera década del siglo XXI, la proliferación de aplicaciones y plataformas como sitios web de redes sociales, grandes empresas de comercio electrónico, búsqueda web índices, y el almacenamiento / respaldo en la nube llevaron a un aumento en la cantidad de datos almacenados en grandes bases de datos y servidores masivos. Algunos de los requisitos de estos nuevos sistemas fueron no es compatible con los DBMS relacionales SQL (SQL es el modelo de datos estándar y lenguaje para bases de datos relacionales). El término NOSQL generalmente se interpreta como No Solo SQL, lo que significa que en sistemas que gestionan grandes cantidades de datos, algunos de los datos se almacenan usando sistemas SQL, mientras que otros datos se almacenarían usando NOSQL, dependiendo de los requisitos de la aplicación.

19. ¿Qué es una perspectiva de los datos?

R/ Puede ser un subconjunto de la base de datos o pueden contener datos virtuales que se derivan de los archivos de la base de datos pero que no se almacenan explícitamente.

20. ¿Qué es una transacción?

R/ Es un programa o proceso en ejecución que incluye uno o más accesos a bases de datos, como lectura o actualización de registros de bases de datos. Se supone que cada transacción debe ejecutar un acceso a la base de datos, es lógicamente correcto si se ejecuta en su totalidad sin interferencia de otras transacciones.

21. ¿Qué es un índice?

R/ Es un archivo auxiliar utilizado a menudo para proporcionar estructuras de datos especializadas y técnicas de búsqueda para acelerar la búsqueda en disco de los registros deseados. Los índices se basan típicamente en estructuras de datos de árbol o estructuras de datos hash que son convenientemente modificados para la búsqueda de disco.

22. A que se refiere el término “data no tradicional”.

R/ Incluyen información que está disponible públicamente en Internet, pero que a menudo es difícil de obtener en un formato estructurado.

23. A que se refiere las aplicaciones tradicionales de base de datos.

R/ Refiere a la aplicación de las bases de datos en áreas tradicionales del conocimiento, algunos ejemplos son:

- * Aplicaciones científicas que almacenan grandes cantidades de datos
- * Almacenamiento y recuperación de imágenes
- * Almacenamiento y recuperación de videos
- * Aplicaciones de minería de datos que analizan grandes cantidades de datos
- * Aplicaciones espaciales que almacenan y analizan ubicaciones espaciales
- * Aplicaciones de series de tiempo que almacenan información como datos económicos

24. ¿Qué es aislamiento de una transacción?

R/ Es la propiedad de aislamiento la que asegura que cada transacción pueda ejecutarse de forma aislada de otras transacciones, a pesar de que cientos de las transacciones pueden estar ejecutándose al mismo tiempo.

25. ¿Qué es concurrencia?

R/ La concurrencia es un software de control que sirve para asegurar que varios usuarios que intentan actualizar los mismos datos lo hagan de forma controlada para que el resultado de las actualizaciones sea correcto.

26. ¿Qué son los procedimientos almacenados?

R/ Los procedimientos más complicados para hacer cumplir las reglas de deducción para inferir información se denominan popularmente procedimientos almacenados; se vuelven parte de la definición general de la base de datos y se invocan de manera apropiada cuando se cumplen ciertas condiciones.

27. ¿Qué es atomicidad?

R/ La propiedad de atomicidad asegura que o se ejecutan todas las operaciones de la base de datos en una transacción o ninguna.

28. Liste 10 ejemplos de colecciones de datos de la vida real.

- R/**
1. World Data Center for Climate.
 2. National Energy Research Scientific Computing Center.
 3. LexisNexis.
 4. Google.
 5. Sprint.
 6. AT&T.
 7. Youtube.
 8. Amazon.
 9. Central Intelligence Agency (CIA).
 10. Biblioteca del congreso de los Estados Unidos.

29. ¿Cuáles de esos ejemplos anteriores tiene sentido aplicar en un archivo tradicional de procesamiento, en lugar de un acercamiento de base de datos?

R/ Puede realizarse una versión local o de archivo tradicional de procesamiento de LexisNexis almacenando todos los documentos legales para su búsqueda y gestión de forma controlada localmente; de igual forma por la naturaleza de los ejemplos anteriores que mueven cantidades enormes de datos no es conveniente aplicar archivo tradicional de procesamiento puesto la aplicación se volvería muy pesada solo por la cantidad de datos que almacenaría.

30. ¿Cuáles capacidades, por tanto, deberían ser parte del DBMS/SGBD?

R/ Acceso directo a los datos solicitados y predicción de petición personalizada.

Preguntas del Capitulo 02 "Base de Datos y Usuarios de Base de Datos" del libro "Fundamentals of Database Systems" de los autores Ramez Elmasri, Shamkant B. Navathe, 7ma edición del año 2015.

1. ¿Qué es el estado de la base de datos?

R/ Es la abstracción del estado de la base de datos basándose en satisfacer la estructura y las restricciones especificadas en el esquema de la base de datos, determinando así estados validos o no validos para la base de datos.

2. ¿Qué son y cuáles son las herramientas y utilidades que tiene un SGBD?

R/ Las utilidades ayudan a administrar la base de datos, estas son:

- * Cargando. Se utiliza una utilidad de carga para cargar archivos de datos existentes.
- * Copia de seguridad. Una utilidad de respaldo crea una copia de respaldo de la base de datos, generalmente por volcar toda la base de datos en cinta u otro medio de almacenamiento masivo.
- * Reorganización del almacenamiento de la base de datos. Esta utilidad se puede utilizar para reorganizar un conjunto de archivos de base de datos en diferentes organizaciones de archivos y crear un nuevo acceso a caminos para mejorar el rendimiento.
- * Supervisión del desempeño. Dicha utilidad monitorea el uso de la base de datos y proporciona estadísticas al DBA.

Las herramientas son las que sirven para el almacenamiento de los datos algunas serían:

- * Sistema de diccionario de datos ampliado (o repositorio de datos). Además de almacenar información se almacenan decisiones de diseño, estándares de uso, descripciones de programas de aplicación e información del usuario.
- * Diseño y desarrollo de GUI.
- * Sistema integrado de comunicaciones de datos y DBMS se denomina sistema DB / DC.

3. ¿Cómo se clasifican los SGBD?

R/ Clasificación de SGBD por criterios:

- * Modelo de datos → relacional, objeto, objeto-relacional, valor-clave, jerárquico, de red, y NOSQL que utiliza varios modelos: basados en documentos, basados en gráficos, basados en columnas y clave-valor.

- * El número de usuarios admitidos por el sistema → Los sistemas de un solo usuario admiten solo un usuario a la vez y en su mayoría son utilizados con PC. Los sistemas multiusuario, que incluyen la mayoría de DBMS, admiten múltiples usuarios concurrentes.
- * Número de sitios en los que se distribuye la base de datos → está centralizado si los datos se almacenan en un solo sitio de computadora. Un DBMS distribuido (DDBMS) puede tener la base de datos real y software DBMS distribuido en muchos sitios conectados por una red informática. Los grandes sistemas de datos a menudo se distribuyen masivamente, con cientos de sitios.
- * El costo → Los principales productos RDBMS están disponibles como versiones de copia de 30 días de examen gratuito así como versiones personales, que pueden costar menos de \$ 100 y permitir una buena cantidad de funcionalidad. se venden en forma de licencias: las licencias de sitio permiten el uso ilimitado del sistema de base de datos con cualquier número de copias ejecutándose en el cliente sitio. Otro tipo de licencia limita el número de usuarios concurrentes o el número de asientos de usuario en una ubicación.
- * En función de los tipos de opciones de ruta de acceso para almacenar archivos → Una familia conocida de DBMS se basa en estructuras de archivos invertidas.
- * Propósito general o especial → Cuando el rendimiento es una consideración primordial, un DBMS de propósito especial se puede diseñar y construir para una aplicación específica.

4. Liste las categorías de modelos de datos.

1. **El modelo de datos relacionales** representa una base de datos como una colección de tablas, donde cada tabla se puede almacenar como un archivo separado. La mayoría de las bases de datos relacionales utilizan el lenguaje de consulta llamado SQL y admite una forma limitada de vistas de usuario. El modelo de datos de objetos define una base de datos en términos de objetos, sus propiedades y sus operaciones. Los objetos con la misma estructura y comportamiento pertenecen a una clase, y las clases se organizan en jerarquías (o gráficos acíclicos). Las operaciones de cada clase se especifican en términos de procedimientos predefinidos llamados métodos. Los DBMS relacionales han ampliado sus modelos para incorporar bases de datos de objetos. conceptos y otras capacidades; estos sistemas se conocen como relacionales de objetos o sistemas relacionales extendidos. Los sistemas de big data se basan en varios modelos de datos, con los siguientes cuatro modelos más comunes. El modelo de datos clave-valor asocia una clave única con cada valor (que puede ser un registro u objeto) y proporciona un acceso muy rápido a un valor dado su clave. El modelo de datos del documento se basa en JSON (Java Script Notación de objetos) y almacena los datos como documentos, que se

parecen un poco objetos complejos. El modelo de datos de gráficos almacena objetos como nodos de gráficos y las relaciones entre objetos como bordes de gráficos dirigidos. Finalmente, los datos basados en columnas Los modelos almacenan las columnas de filas agrupadas en páginas de disco para un acceso rápido y permitir múltiples versiones de los datos. El modelo XML ha surgido como un estándar para el intercambio de datos en la Web y se ha utilizado como base para implementar varios prototipos de sistemas XML nativos. XML utiliza estructuras de árbol jerárquicas. Combina conceptos de bases de datos con conceptos a partir de modelos de representación de documentos. Los datos se representan como elementos; con el uso de etiquetas, los datos se pueden anidar para crear estructuras de árbol complejas. Este modelo se parece conceptualmente al modelo de objetos, pero utiliza una terminología diferente. Se han agregado capacidades XML a muchos productos DBMS comerciales. modelos de datos importantes, ahora conocidos como modelos de datos heredados, son la red y los modelos jerárquicos. El modelo de red representa los datos como tipos de registro y también representa un tipo limitado de relación 1: N, llamado tipo de conjunto. Una relación 1: N, o uno a muchos, relaciona una instancia de un registro con muchos registros instancias que utilizan algún mecanismo de enlace de puntero en estos modelos. El modelo, también conocido como modelo CODASYL DBTG, 14 tiene un lenguaje de registro en tiempo asociado que debe estar integrado en un lenguaje de programación host. El DML de red se propuso en el Informe del Grupo de tareas de base de datos (DBTG) de 1971 como extensión del lenguaje COBOL. El modelo jerárquico representa los datos como estructuras de árbol jerárquicas. Cada jerarquía representa una serie de registros relacionados. No existe un idioma estándar para modelo jerárquico. Un DML jerárquico popular es DL / 1 del sistema IMS. Dominó el mercado de DBMS durante más de 20 años entre 1965 y 1985. Su DML, llamado DL / 1, fue un estándar industrial de facto durante mucho tiempo.

5. ¿Qué son los esquemas de datos?

R/ La descripción de una base de datos que se especifica durante el diseño de la base de datos y no se espera cambiar con frecuencia. Un esquema mostrado se denomina diagrama de esquema. Un diagrama de esquema muestra solo algunos aspectos de un esquema, como los nombres de tipos de registros y elementos de datos, y algunos tipos de restricciones. Otros aspectos no son especificado en el diagrama de esquema.

6. ¿Qué es la arquitectura Cliente/Servidor de SGBD?

R/ Se desarrolló para hacer frente a entornos informáticos en los que

una gran cantidad de PC, estaciones de trabajo, servidores de archivos, impresoras, servidores de bases de datos, servidores web, servidores de correo electrónico y otro software y los equipos están conectados a través de una red. La idea es definir servidores especializados con funcionalidades específicas. Las máquinas cliente proporcionan al usuario las interfaces adecuadas para utilizar estos servidores, así como con potencia de procesamiento local para ejecutar aplicaciones locales. Algunas máquinas serían solo sitios de clientes Otras máquinas serían servidores dedicados y otras tendrían ambos funcionalidad de cliente y servidor.

7. ¿Qué es la arquitectura de 3 esquemas y la independencia de datos?

R/ El objetivo de la **arquitectura de tres esquemas**, es separar las aplicaciones de usuario de la base de datos física. En esta arquitectura, los esquemas pueden definirse en los tres niveles siguientes: **1. El nivel interno** tiene un esquema interno, que describe el físico estructura de almacenamiento de la base de datos. **2. El nivel conceptual** tiene un esquema conceptual, que describe la estructura de toda la base de datos para una comunidad de usuarios. **3. El nivel externo** o de vista incluye una serie de esquemas externos o puntos de vista. Cada esquema externo describe la parte de la base de datos que le interesa a un grupo de usuarios en particular y oculta el resto de la base de datos de ese grupo de usuarios.

La independencia de los datos: la capacidad de cambiar el esquema en un nivel de un sistema de base de datos sin tener que cambiar el esquema en el siguiente nivel superior nivel. Podemos definir dos tipos de independencia de datos: **1. La independencia lógica** de los datos es la capacidad de cambiar el esquema conceptual sin tener que cambiar esquemas externos o programas de aplicación. 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).

2. La independencia de los datos físicos es la capacidad de cambiar el esquema interno sin tener que cambiar el esquema conceptual. Por tanto, tampoco es necesario cambiar los esquemas externos.

8. ¿Qué son los Lenguajes de los SGBD?

R/ Son los encargados de especificar esquemas conceptuales e internos para la base de datos y cualquier mapeo entre los dos. un lenguaje, llamado lenguaje de definición de datos (DDL), es utilizado por el DBA y por los diseñadores de bases de datos para definir ambos esquemas. Los DBMS tendrá un compilador DDL cuya función es procesar declaraciones DDL en para identificar descripciones de las construcciones del esquema y almacenar el

esquema descripción en el catálogo DBMS. el lenguaje de definición de almacenamiento (SDL), se utiliza para especificar el esquema. Las asignaciones entre los dos esquemas se pueden especificar en cualquiera de estos lenguajes. Estos permiten el personal de DBA para controlar las opciones de indexación y la asignación de datos al almacenamiento. Para la arquitectura de tres esquemas, necesitaríamos un tercer lenguaje, la definición de vista lenguaje (VDL), para especificar las vistas del usuario y sus asignaciones al concepto esquema, pero en la mayoría de los DBMS, el DDL se utiliza para definir tanto conceptual como externo esquemas.

9. ¿Qué son las interfaces de los SGBD?

R/ Son las que proporcionan la conexión entre el cliente y la base de datos haciendo uso del modulo del cliente diseñado típicamente para ejecutarse en un dispositivo móvil, estación de trabajo de usuario o computadora personal (PC).

10. ¿Qué son los ambientes de los SGBD y cómo se diferencia de las interfases y los lenguajes?

R/ Los ambientes están diseñados para el desarrollo y diseño de bases de datos, el desarrollo de GUI, las consultas y actualización y desarrollo de programas de aplicación para las cuales el DBA. Las interfaces están dirigidas a los clientes y los lenguajes son para especificar esquemas conceptuales e internos de la base de datos.

11. ¿Qué es la arquitectura Centralizada de SGBD?

R/ Un DBMS está centralizado si los datos se almacenan en un solo sitio de computadora. Centralizado DBMS puede soportar múltiples usuarios, pero el DBMS y la base de datos residen totalmente en un solo sitio informático. Las arquitecturas más antiguas usaban computadoras centrales para proporcionar el procesamiento principal de todas las funciones del sistema, incluida la aplicación del usuario programas y programas de interfaz de usuario, así como toda la funcionalidad DBMS.

12. ¿Qué son las instancias dentro de los conceptos de sistemas de base de datos?

R/ Teniendo en cuenta que la abstracción y el uso de objetos están orientados la realización de transacciones en la base de datos, una instancia de dicho objetos sería una serie de operaciones a realizar para concretar transacciones.

13. ¿Qué son los modelos de datos?

R/ Es una colección de conceptos que se puede utilizar para describir la estructura de una base de datos: proporciona los medios

necesarios Por estructura de una base de datos nos referimos a los tipos de datos, relaciones y restricciones que se aplican a los datos.

Bases de Datos I
18/09/2020
Preguntas Cap 3 18/09
IS-501
José Inestroza

Preguntas del Capitulo 03 "Base de Datos y Usuarios de Base de Datos" del libro "Fundamentals of Database Systems" de los autores Ramez Elmasri, Shamkant B. Navathe, 7ma edición del año 2015.

1. ¿Qué son las llaves y los atributos?

R/ Las llaves son conocidas también como restricciones de unicidad de los atributos, un tipo de entidad generalmente tiene uno o más atributos cuyos valores son distintos para cada entidad individual en el conjunto de entidades, dicho atributo se denomina atributo llave y sus valores se pueden utilizar para identificar cada entidad de forma única.

Los atributos describen cada una de las entidades también conocidas como las propiedades particulares que describen a las entidades.

2. ¿Qué son los tipos de relación, los conjuntos de relación, los roles, y las restricciones estructurales?

R/ Tipos de relaciones: Relaciones entre tipos de entidades distintos o similares.

Conjuntos de relaciones: Define un conjunto de asociaciones, o un conjunto de relaciones, entre entidades de estos tipos de entidad.

Los roles: El rol que juega una entidad participante del tipo de entidad en cada instancia de relación, y ayuda a explicar qué significa la relación.

Las restricciones estructurales: Son las restricciones que permiten el estructuramiento correcto del modelo de la base de datos.

3. ¿Qué son los atributos nulos y para qué sirven?

R/ Los Atributos nulos son aquellos atributos que aún no han sido definidos o no se sabe de la existencia de ese atributo y es aplicado una entidad en particular puede no tener un valor aplicable para un atributo.

4. ¿Qué significa UML?

R/ Lenguaje de modelado unificado: Se especifican operaciones sobre objetos, además especificando la estructura del esquema de la base de datos. Las operaciones se pueden utilizar para especificar

los requisitos funcionales durante el diseño de la base de datos.

5. ¿Qué son las entidades, los grupos de entidades y los tipos de entidades?

R/ Entidades: El concepto básico que representa el modelo ER es una entidad, que es una cosa u objeto en el mundo real con una existencia independiente. Una entidad puede ser un objeto con existencia física (por ejemplo, una persona, automóvil, casa o empleado en particular) o puede ser un objeto con existencia conceptual (por ejemplo, una empresa, un trabajo o un curso universitario). Cada entidad tiene atributos: las propiedades particulares que lo describen.

Grupos de entidades: Son los que conforman las bases de datos, suelen ser grupos de entidades similares.

Tipos de entidades: Define una colección (o conjunto) de entidades que tienen los mismos atributos. Cada tipo de entidad en la base de datos se describe por su nombre y atributos.

6. ¿Cómo se comparan y se diferencian los atributos de un único valor de los multivalor?

R/ La mayoría de los atributos tienen un solo valor para una entidad en particular; tales atributos se denominan de valor único. Un atributo puede tener un conjunto de valores para la misma entidad, estos atributos se denominan multivalor, este atributo puede tener límites inferior y superior para restringir el número de valores permitidos para cada entidad individual. En los diagramas ER los atributos multivalor se muestran en óvalos dobles y los atributos compuestos y multivalor pueden anidarse arbitrariamente.

7. ¿Qué son los diagramas entidad relación y qué problemas de diseño y convenciones de nomenclatura posee?










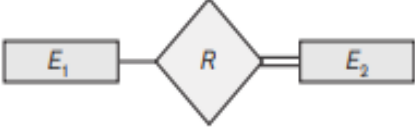

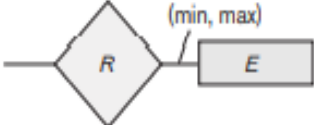
R/ Diagramas ER: Las instancias de entidad individuales en un conjunto de entidades y las instancias de relación individuales en un conjunto de relaciones. En los diagramas ER, el énfasis está en representar los esquemas en lugar de las instancias.

Problemas de diseño: Elección de nombres para tipos de entidad, atributos, tipos de relación y (en particular) roles.

Problema de entidades dependientes el cual surge porque cada relación binaria se puede describir a partir de cualquiera de los dos tipos de entidades participantes,

Convenciones de nomenclatura: Nombres de tipo de entidad y tipo de relación están en letras mayúsculas, los nombres de atributos tienen su letra inicial en mayúscula y los nombres de roles están en letras minúsculas.

8. ¿Cuál es la simbología relacionada en el diseño entidad relación?

Symbol	Meaning
	Entity
	Weak Entity
	Relationship
	Identifying Relationship
	Attribute
	Key Attribute
	Multivalued Attribute
	Composite Attribute
	Derived Attribute
	Total Participation of E_2 in R
	Cardinality Ratio 1: N for $E_1 : E_2$ in R
	Structural Constraint (min, max) on Participation of E in R

R/

9. ¿Cómo se comparan y se diferencian los atributos compuestos de los atributos simples?

R/ Los atributos compuestos pueden ser divididos en subpartes más pequeñas, que representan atributos más básicos con significados independientes. Atributos que no son divisibles se denominan atributos simples o atómicos. Los atributos compuestos pueden formar una jerarquía; El valor de un atributo compuesto es la concatenación de los valores de su componente simple atributos. Los atributos compuestos son útiles para modelar situaciones en las que un usuario a veces se refiere al atributo compuesto como una unidad, pero en otras ocasiones se refiere específicamente a sus componentes.

Bases de Datos I

22/09/2020

Preguntas Cap 1 y 2 22/09

IS-501

José Inestroza

1. Liste las 6 actividades principales para las cuales se usa SQL.

- Definición de datos SQL permite al usuario definir la estructura y organización de los datos almacenados y las relaciones entre los elementos de datos almacenados.
- Recuperación de datos SQL permite a un usuario o un programa de aplicación recuperar datos almacenados de la base de datos y utilizarlos.
- La manipulación de datos SQL permite que un usuario o un programa de aplicación actualice la base de datos agregando datos nuevos, eliminando datos antiguos y modificando datos almacenados previamente.
- El SQL de control de acceso se puede utilizar para restringir la capacidad de un usuario de recuperar, agregar y modificar datos, protegiendo los datos almacenados contra el acceso no autorizado.
- SQL para compartir datos se utiliza para coordinar el intercambio de datos entre usuarios simultáneos, lo que garantiza que los cambios realizados por un usuario no borren inadvertidamente los cambios realizados casi al mismo tiempo por otro usuario.
- Integridad de datos SQL define las restricciones de integridad en la base de datos, protegiéndola de la corrupción debido a actualizaciones inconsistentes o fallas del sistema.

2. Liste los 7 roles que tiene SQL.

- SQL es un lenguaje de consulta interactivo. Los usuarios escriben comandos SQL en un programa SQL interactivo para recuperar datos y mostrarlos en la pantalla, proporcionando una herramienta conveniente y fácil de usar para consultas de bases de datos ad hoc.
 - SQL es un lenguaje de programación de bases de datos. Los programadores incorporan comandos SQL en sus programas de aplicación para acceder a los datos en una base de datos. Tanto los programas escritos por el usuario como los programas de utilidad de base de datos (como los redactores de informes y las herramientas de entrada de datos) utilizan esta técnica para acceder a la base de datos.
 - SQL es un lenguaje de administración de bases de datos. El administrador de la base de datos responsable de administrar una base de datos de miniordenador o mainframe utiliza SQL para definir la estructura de la base de datos y controlar el acceso a los datos almacenados.
 - SQL es un lenguaje cliente / servidor. Los programas de computadora personal usan SQL para comunicarse a través de una red con servidores de bases de datos que almacenan datos compartidos. Esta arquitectura cliente / servidor es utilizada por muchas aplicaciones populares de clase empresarial.
 - SQL es un lenguaje de acceso a datos de Internet. Los servidores web de Internet que interactúan con los datos corporativos y los servidores de aplicaciones de Internet utilizan SQL como lenguaje estándar para acceder a las bases de datos corporativas, a menudo incorporando el acceso a la base de datos SQL en lenguajes de scripting populares como PHP o Perl.
 - SQL es un lenguaje de base de datos distribuido. Los sistemas de administración de bases de datos distribuidas utilizan SQL para ayudar a distribuir datos en muchos sistemas informáticos conectados. El software DBMS en cada sistema usa SQL para comunicarse con los otros sistemas, enviando solicitudes de acceso a datos.
 - SQL es un lenguaje de puerta de enlace de base de datos. En una red informática con una combinación de diferentes productos DBMS, SQL se utiliza a menudo en una puerta de enlace que permite que una marca de DBMS se comunique con otra marca.
3. **¿Cuáles componentes de SQL me servirán para extraer los datos necesarios para cada escenario de un problema en particular? Liste las palabras claves que le permitirán ordenar, agrupar y/o sumarizar datos. Busque ¿cómo se llama la categoría en SQL que agrupa estas palabras claves? ¿existen otras palabras claves de la misma categoría o complementarias? si existen más, ¿cuáles son esas palabras?.**

Capítulo 5 y 6 Fundamentals

1. ¿Qué es el Modelo Relacional (MR)?

R/ Representa la base de datos como una colección de relaciones. De manera informal, cada relación se asemeja a una tabla de valores o, hasta cierto punto, a un archivo plano de registros. En la terminología del modelo relacional formal, una fila se llama tupla, un encabezado de columna se llama atributo y la tabla se llama relación. El tipo de datos que describe los tipos de valores que pueden aparecer en cada columna está representado por un dominio de valores posibles.

2. ¿Tiene diferencias el MR del ER? ¿Cuáles son esas diferencias?

R/ Si

- La diferencia básica entre el Modelo E-R y el Modelo Relacional es que el modelo E-R trata específicamente con las entidades y sus relaciones. Por otro lado, el Modelo Relacional se ocupa de las Tablas y de la relación entre los datos de esas tablas.
- Un Modelo E-R describe los datos con conjuntos de entidades, conjuntos de relaciones y atributos. Sin embargo, el modelo relacional describe los datos con las tuplas, atributos y dominio del atributo.
- Uno puede entender más fácilmente la relación entre los datos en el Modelo E-R en comparación con el Modelo Relacional.
- El Modelo E-R tiene la Cardinalidad del Mapeo como una restricción mientras que el Modelo Relacional no tiene tal restricción.

3. Según el modelo relacional, defina:

- **Atómico.** → Refiere a valores indivisibles en el dominio del modelo relacional.
- **Cálculo relacional.** → Se utiliza para definir una consulta de forma declarativa sin dar un orden específico de operaciones.
- **Concepto de modelado del MR.** → Cuando se piensa en una relación como una tabla de valores, cada fila de la tabla representa una colección de valores de datos relacionados. Una fila representa un hecho que normalmente corresponde a una entidad o relación del mundo real. El nombre de la tabla y los nombres de las columnas se utilizan para ayudar a

interpretar el significado de los valores en cada fila. datos. En la terminología del modelo relacional formal, una fila se llama tupla, un encabezado de columna se llama atributo y la tabla se llama relación. El tipo de datos que describe los tipos de valores que pueden aparecer en cada columna está representado por un dominio de valores posibles. Ahora definimos estos términos —dominio, tupla, atributo y relación— formalmente.

- **Flat Structure.** → El modelo relacional representa la base de datos como una colección de relaciones. De manera informal, cada relación se asemeja a una tabla de valores o, hasta cierto punto, a un archivo plano de registros. Se llama archivo plano porque cada registro tiene una estructura lineal o plana simple.
- **Tabla.** → Cada fila de la tabla representa una colección de valores de datos relacionados. Una fila representa un hecho que normalmente corresponde a una entidad o relación del mundo real. El nombre de la tabla y los nombres de las columnas se utilizan para ayudar a interpretar el significado de los valores en cada fila.
- **Tupla.** → Una fila se llama tupla, cada tupla de la relación representa una entidad (u objeto) de estudiante en particular.
- **Dominio.** → Es un conjunto de valores atómicos. . Un método común de especificar un dominio es especificar un tipo de datos del cual se extraen los valores de datos que forman el dominio.
- **Formato.** → Refiere al formato de los datos que tiene cada dominio.
- **Esquema relacional.** → Se usa para describir una relación. un esquema de relación $R = \{A_1, A_2, \dots, A_n\}$ es un conjunto de atributos (en lugar de una lista ordenada de atributos), y un estado de relación $r(R)$ es un conjunto finito de asignaciones $r = \{t_1, t_2, \dots, t_m\}$, donde cada tupla t_i es una asignación de R a D , y D es la unión (indicada por \cup) de los dominios de atributos; es decir, $D = \text{dom}(A_1) \cup \text{dom}(A_2) \cup \dots \cup \text{dom}(A_n)$. En esta definición, $t[A_i]$ debe estar en $\text{dom}(A_i)$ para $1 \leq i \leq n$ para cada mapeo t en r . Cada mapeo t_i se llama tupla.

4. Características de ER.

R/ Las tuplas de una relación no tienen ningún orden en particular, siempre hay un orden entre los registros. . Cada valor de una tupla es

un valor atómico, los atributos compuestos y multivalor no están permitidos. los atributos multivalor deben representarse mediante relaciones separadas y los atributos compuestos se representan solo por sus atributos componentes simples en el modelo relacional básico. los valores NULL, que se utilizan para representar los valores de atributos que pueden ser desconocidos o que pueden no aplicarse a una tupla.

5. Notación del MR. →

- Un esquema de relación R de grado n se denota por R (A1, A2, ..., An).
- Las letras mayúsculas Q, R, S denotan nombres de relación.
- Las letras minúsculas q, r, s denotan estados de relación.
- Las letras t, u, v denotan tuplas.

6. Restricciones de MR, y violación de restricciones.

- **Actualización y cómo se implementan en SQL.** → La operación Actualizar (o Modificar) se usa para cambiar los valores de uno o más atributos en una tupla (o tuplas) de alguna relación R. Es necesario especificar una condición sobre los atributos de la relación para seleccionar la tupla (o tuplas)) ser modificado.

■ **Operación:**

Actualice el salario de la tupla EMPLEADO con Ssn = '999887777' a 28000.

Resultado: aceptable.

■ **Operación:**

Actualice el Dno de la tupla EMPLOYEE con Ssn = '999887777' a 1.

Resultado: aceptable.

■ **Operación:**

Actualice el Dno de la tupla EMPLOYEE con Ssn = '999887777' a 7.

Resultado: Inaceptable, porque viola la integridad referencial.

■ **Operación:**

Actualice el Ssn de la tupla EMPLOYEE con Ssn = "999887777" a "987654321".

Resultado: Inaceptable, porque viola la restricción de clave primaria al repetir un valor que ya existe como clave principal en otra tupla; viola las restricciones de integridad referencial porque hay otras relaciones que se refieren al valor existente de Ssn.

- **Eliminado y cómo se implementan en SQL.** → La operación Eliminar solo puede violar la integridad referencial. Esto ocurre si las claves externas de otras tuplas de la base de datos hacen referencia a la tupla que se está eliminando. Para especificar la eliminación, una condición en los atributos de la relación selecciona la tupla (o tuplas) que se eliminarán. Aquí hay unos ejemplos.

■ Operación: elimine la tupla WORKS_ON con Essn = '999887777' y Pno = 10.

Resultado: esta eliminación es aceptable y elimina exactamente una tupla.

■ Operación: Elimine la tupla EMPLOYEE con Ssn = '999887777'.

Resultado: esta eliminación no es aceptable, porque hay tuplas en WORKS_ON que se refieren a esta tupla. Por lo tanto, si se elimina la tupla en EMPLEADO, se producirán violaciones de la integridad referencial.

■ Operación: Elimine la tupla EMPLOYEE con Ssn = "333445555".

Resultado: esta eliminación resultará en violaciones de integridad referencial aún peores, porque la tupla involucrada es referenciada por tuplas de las relaciones EMPLOYEE, DEPARTMENT, WORKS_ON y DEPENDENT.

- Inserción y cómo se implementan en SQL. → La operación Insertar proporciona una lista de valores de atributo para una nueva tupla t que se insertará en una relación R. Insertar puede violar cualquiera de los cuatro tipos de restricciones. Las restricciones de dominio se pueden violar si se proporciona un valor de atributo que no aparece en el dominio correspondiente o no es del tipo de datos apropiado. Las restricciones de clave se pueden violar si un valor clave en la nueva tupla t ya existe en otra tupla en la relación r (R). La integridad de la entidad se puede violar si cualquier parte de la clave primaria de la nueva tupla t es NULL. La integridad referencial se puede violar si el valor de cualquier clave externa en t se refiere a una tupla que no existe en la relación referenciada. A continuación, se muestran algunos ejemplos para ilustrar esta discusión.

■ Operación: Inserte <'Cecilia', 'F', 'Kolonsky', NULL, '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4> en EMPLOYEE. Resultado: esta inserción viola la restricción de integridad de la entidad (NULL para la clave principal Ssn), por lo que se rechaza.

■ Operación: Inserte <'Alicia', 'J', 'Zelaya', '999887777', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, '987654321', 4> en EMPLEADO. Resultado: esta inserción viola la restricción de clave porque ya existe otra tupla con el mismo valor de Ssn en la relación EMPLEADO, por lo que se rechaza.

■ Operación: Inserte <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windswept, Katy, TX', F, 28000, '987654321', 7> en EMPLOYEE. Resultado: esta inserción viola la restricción de integridad referencial especificada en Dno en EMPLOYEE porque no existe una tupla referenciada correspondiente en DEPARTMENT con Dnumber = 7.

■ Operación: Inserte <'Cecilia', 'F', 'Kolonsky', '677678989', '1960-04-05', '6357 Windy Lane, Katy, TX', F, 28000, NULL, 4> en EMPLOYEE.
Resultado: esta inserción satisface todas las restricciones, por lo que es aceptable.

7. **Concepto de transacción.** → Una transacción es un programa en ejecución que incluye algunas operaciones de la base de datos, como leer de la base de datos o aplicar inserciones, eliminaciones o actualizaciones a la base de datos. Al final de la transacción, debe dejar la base de datos en un estado válido o consistente que satisfaga todas las restricciones especificadas en el esquema de la base de datos. Una sola transacción puede involucrar cualquier número de operaciones de recuperación y cualquier número de operaciones de actualización.

Bases de Datos I
14/10/2020
Preguntas Cap 14 14/10
IS-501
José Inestroza

Capítulo 14 Fundamentals

- **¿Qué es la información redundante en los SGBD?**
R/ Es información repetida en la base de datos, dado que es uno de los objetivos implícitos en la actividad de diseño de la base de datos, minimizar la redundancia implica minimizar el almacenamiento redundante de la misma información y reducir la necesidad de múltiples actualizaciones para mantener la coherencia entre múltiples copias de la misma información.
- **¿Qué son las tuplas falsas?**
R/ Se denominan tuplas falsas porque a las tuplas que representan información falsa que no es válida, para lo cual se deben evitar las relaciones que contengan atributos concidentes que no sean combinaciones (clave externa, clave primaria) porque la unión de dichos atributos puede producir tuplas falsas.
- **¿Qué es la semántica en las bases de datos?**
R/ La semántica de una relación se refiere a su significado resultante de la interpretación de los valores de los atributos en una tupla, por lo tanto la semántica en las bases de datos hace referencia al significado de cada una de las tuplas que la componen.

- **¿Qué son las anomalías de actualización?**
R/ Son las anomalías o problemas que genera el almacenamiento de combinaciones naturales de relaciones base.
- **¿Qué son las anomalías de eliminación?**
R/ Son los problemas generados al vaciar los registros de una base de datos que al no poder todos los valores de los atributos en NULL por violaciones de integridad a la información se pierden datos inadvertidamente de la base de datos.
- **¿Qué son las anomalías de modificación?**
R/ Son los problemas que genera actualizar o modificar un registro dado que si tiene relaciones asociadas o dependencias, es necesario también modificar cada uno de los registros que representen una relación o dependencia.
- **¿Por qué la implementación de NULL no es una buena práctica?**
R/ Porque Si muchos de los atributos no se aplican a todas las tuplas en la relación, terminamos con muchos NULL en esas tuplas. Esto puede desperdiciar espacio a nivel de almacenamiento y también puede dar lugar a problemas para comprender el significado de los atributos y para especificar operaciones JOIN en el nivel lógico. Las operaciones SELECT y JOIN implican comparaciones; si hay valores NULL presentes, los resultados pueden volverse impredecibles. Además, los NULL pueden tener múltiples interpretaciones.
- **¿Cuáles son las 4 guías de diseño dictadas en el capítulo y qué problemáticas resuelven?**
R/ **Directriz 1.** Diseñar un esquema de relación para que sea fácil de explicar su significado. No combine atributos de varios tipos de entidades y tipos de relaciones en una sola relación. Intuitivamente, si un esquema de relación corresponde a un tipo de entidad o un tipo de relación, es sencillo explicar su significado. De lo contrario, si la relación corresponde a una mezcla de múltiples entidades y relaciones, se producirán ambigüedades semánticas y la relación no podrá explicarse fácilmente.
Directriz 2. Diseñar los esquemas de relación base de modo que no existan anomalías de inserción, eliminación o modificación en las relaciones. Si se presentan anomalías, anótelas claramente y asegúrese de que los programas que actualizan la base de datos funcionarán correctamente. Es importante tener en cuenta que, en ocasiones, es posible que sea necesario infringir estas directrices para mejorar el rendimiento de determinadas consultas.

Directriz 3. En la medida de lo posible, evite colocar atributos en una relación base cuyos valores puedan ser NULL con frecuencia. Si los NULL son inevitables, asegúrese de que se apliquen solo en casos excepcionales y no se apliquen a la mayoría de las tuplas en la relación. Usar el espacio de manera eficiente y evitar uniones con valores NULL son los dos principales criterios que determinan si incluir las columnas que pueden tener NULL en una relación o tener una relación separada para esas columnas (con las columnas clave apropiadas).

Directriz 4. Diseñar esquemas de relaciones para que puedan unirse con condiciones de igualdad en atributos que estén apropiadamente relacionados (clave primaria, clave externa) pares de manera que garantice que no se generen tuplas falsas. Evite las relaciones que contengan atributos coincidentes que no sean combinaciones (clave externa, clave primaria) porque la unión de dichos atributos puede producir tuplas falsas.

Problemas que resuelve: Resuelve la ambigüedad semántica de las relaciones, permitiendo que se expliquen fácilmente; también resuelve las principales anomalías de inserción, eliminación o modificación en las relaciones teniendo en cuenta que se puede infringir en estas anomalías por únicamente conseguir mejor rendimiento; también resuelve la problemática de los atributos múltiples con valor NULL y por ultimo también resuelve la problemática de crear tuplas falsas al evitar las relaciones que contengan atributos coincidentes que no sean combinaciones.

- **¿Qué es una dependencia funcional?**

R/ Es una restricción entre dos conjuntos de atributos de la base de datos. Una dependencia funcional $X \rightarrow Y$ es una dependencia funcional completa si la eliminación de cualquier atributo A de X significa que la dependencia ya no se mantiene; es decir, para cualquier atributo $A \in X$, $(X - \{A\})$ no determina funcionalmente Y. **Una dependencia funcional** es una propiedad del esquema de relación R, no de un estado de relación legal particular r de R. Por lo tanto, una FD no puede inferirse automáticamente de una extensión de relación dada r, sino que debe ser definida explícitamente por alguien que conozca la semántica de la relación.

- **¿Qué son las normas formales sobre las llaves primarias?**

R/ Son en las que se da un conjunto de dependencias funcionales para cada relación, y que cada relación tiene una clave primaria designada. Esta información combinada con las pruebas (condiciones) para las formas normales impulsa el proceso de normalización para el diseño de esquemas relacionales.

- **¿Qué es la normalización de datos?**

R/ Puede considerarse un proceso de análisis de los esquemas de relación dados en función de sus FD y claves primarias para lograr las propiedades deseables de (1) minimizar la redundancia y (2) minimizar las anomalías de inserción, eliminación y actualización, el procedimiento de normalización proporciona a los diseñadores de bases de datos lo siguiente:

- Un marco formal para analizar esquemas de relaciones basados en sus claves y en las dependencias funcionales entre sus atributos.
- Una serie de pruebas de forma normal que se pueden realizar en esquemas de relaciones individuales para que la base de datos relacional se pueda normalizar al grado deseado.

- **¿En qué consiste la primera forma normal?**

R/ , se definió para no permitir atributos multivalor, atributos compuestos y sus combinaciones. Establece que el dominio de un atributo debe incluir solo valores atómicos (simples, indivisibles) y que el valor de cualquier atributo en una tupla debe ser un valor único del dominio de ese atributo. Por lo tanto, 1NF no permite tener un conjunto de valores, una tupla de valores o una combinación de ambos como valor de atributo para una sola tupla. En otras palabras, 1NF no permite relaciones dentro de relaciones o relaciones como valores de atributo dentro de tuplas. Los únicos valores de atributo permitidos por 1NF son valores atómicos únicos (o indivisibles). La primera forma normal también rechaza atributos multivalor que son compuestos en sí mismos.

- **¿En qué consiste la segunda forma normal?**

R/ La segunda forma normal (2NF) se basa en el concepto de dependencia funcional completa. Una dependencia funcional $X \rightarrow Y$ es una dependencia funcional completa si la eliminación de cualquier atributo A de X significa que la dependencia ya no se mantiene; es decir, para cualquier atributo $A \in X$, $(X - \{A\})$ no determina funcionalmente Y. Una dependencia funcional $X \rightarrow Y$ es una dependencia parcial si algún atributo $A \in X$ puede eliminarse de X y la dependencia aún se mantiene; es decir, para algunos $A \in X$, $(X - \{A\}) \rightarrow Y$.

Un esquema de relación R está en 2NF si cada atributo no principal A en R es completamente funcionalmente dependiente de la clave primaria de R.

- **¿En qué consiste la tercera forma normal?**

R/ La tercera forma normal (3NF) se basa en el concepto de dependencia transitiva. Una dependencia funcional $X \rightarrow Y$ en un

esquema de relación R es una dependencia transitiva si existe un conjunto de atributos Z en R que no es una clave candidata ni un subconjunto de cualquier clave de R, 11 y tanto $X \rightarrow Z$ como $Z \rightarrow Y$ sostener.

De acuerdo con la definición original de Codd, un esquema de relación R está en 3NF si satisface 2NF y ningún atributo no primario de R depende transitivamente de la clave primaria.

Forma Normal	Prueba	Normalización
Primera forma normal (1FN)	La relación no debe tener atributos de valores múltiples ni relaciones anidadas.	Forme nuevas relaciones para cada atributo multivalor o relación anidada.
Segunda forma normal (2FN)	Para las relaciones en las que la clave principal contiene varios atributos, ningún atributo que no sea de clave debe ser funcionalmente dependiente de una parte de la clave principal.	Descomponga y configure una nueva relación para cada clave parcial con sus atributos dependientes. Asegúrese de mantener una relación con la clave primaria original y cualquier atributo que sea funcionalmente dependiente de ella.
Tercera forma normal (3FN)	La relación no debe tener un atributo no clave determinado funcionalmente por otro atributo no clave (o por un conjunto de atributos no clave). Es decir, no debería haber dependencia transitiva de un atributo que no sea de clave en la clave principal.	Descomponer y establecer una relación que incluya los atributos no clave que determinan funcionalmente otros atributos no clave.

- **¿Qué remedia/corriges la normalización?**
R/ Corrige la redundancia de datos, las tuplas falsas y los atributos con valores en NULL que dañan la integridad de la información.
- **¿Los ejercicios hechos en clase síncrona consideran tablas de bases normalizadas? ¿Por qué de su respuesta?**
R/ No, porque tenemos múltiples relaciones en una sola tabla y porque los datos presentan dependencias funcionales y transitivas entre relaciones.