

Todas las preguntas de exámenes que han caído en Bases de Datos

1. Conjunto DBTG en las bases de datos en red.

Un conjunto DBTG está compuesto por un tipo de registro owner y un tipo de registro member y un tipo de set que mantiene entre ambos tipos de registros una relación 1:1 o 1:N. Los tipos de registros representan a los objetos del dominio del problema representados por sus atributos, y el tipo de set representa el conjunto de relaciones existentes en el dominio del problema entre esos objetos.

2. Tipo de registro ficticio en una base de datos jerárquica.

Dado a que el modelo de datos jerárquico no permite la representación de relaciones N:N, en ocasiones los tipos de registros deben de ser duplicados. Un tipo de registro ficticio es la duplicación de un tipo de registro existente en el modelo jerárquico cuyos atributos son punteros a los registros existentes en el tipo de registro original.

3. Tipo de registro ficticio (dummy) en una base de datos en red.

En el modelo en red, las relaciones muchos a muchos son mantenidas mediante un tipo de registro dummy, que puede mantener atributos (o no) que mantiene dos relaciones una a muchos con dos tipos de registros que en el problema mantienen una relación muchos a muchos.

4. Transacción.

Una transacción es un conjunto de operaciones constituidas por un conjunto de acciones que se ejecutan contra una base de datos y que deben satisfacer las reglas **ACID**.

Una acción es cualquier acción que acceda a la base de datos (leer, escribir).

Una operación es un conjunto de acciones que se realizan contra el mismo gránulo de la base de datos.

- Atomicidad**: se ejecutan todas las sentencias o ninguna.

- Preservación de la consistencia**: la ejecución de una transacción deja la BD en un estado consistente.

- Aislamiento**: una transacción no muestra los cambios que produce hasta que finaliza.

- Persistencia**: una vez que finaliza, sus efectos perduran en la BD.

- Seriabilidad**: el efecto de ejecutar transacciones concurrentemente debe ser el mismo que se produciría la ejecutarlas por separado en un orden secuencial según van entrando en el sistema.

5. Autonomía de la base de datos distribuida.

La autonomía de la base de datos distribuida es la capacidad que tienen los DBA locales de actuar sobre el esquema de la base de datos distribuida, actualizando o modificando dicho esquema definiendo nuevos objetos o modificando los existentes.

6. Checkpoint.

Los checkpoint son puntos de control que se realizan en las copias de seguridad incrementales de las bases de datos. Cada vez que se produce un checkpoint se pasa a definitivas y almacenamiento estable las transacciones realizadas desde el último checkpoint. Esto se realiza sobre los ficheros logs/redo tanto si el almacenamiento es por adelanto o diferido de las transacciones de la base de datos.

7. Para el SBD explique:

a. Diferencias entre SGBD y BD

Una base de datos es una colección de archivos relacionados que almacenan tanto una representación abstracta del dominio de un problema del mundo real cuyo manejo resulta de interés para una organización y un SBD es una colección de programas de aplicación que proporcionan al usuario de la base de datos los medios necesarios para definir, manipular, mantener la integridad y controlar la privacidad de la base de datos.

b. Cada uno de los componentes del SGBD

Lenguaje de definición de datos (DDL): es un lenguaje artificial basado en un determinado modelo de datos que permite la representación lógica de los datos.

Lenguaje de definición de almacenamiento de los datos (DSDL): se definen los datos correspondientes al dominio del problema a los dos niveles de abstracción, y a esta definición de los datos se le denomina *Esquema de la Base de Datos*.

Lenguaje de manipulación de datos (DML): es un lenguaje artificial con el que se realizan dos funciones bien diferentes en la gestión de los datos:

1.La definición del nivel externo o de usuario de los datos.

2.La manipulación de los datos.

Dependiendo del modelo de datos, existen dos tipos de DML:

Procedimentales: (hay que decirlo todo, acciones y operaciones)

No procedimentales: (se le dice los datos y que se quiere obtener)

El diccionario de datos: Es uno o un conjunto de archivos que contienen información acerca de los datos que pueden ser almacenados en la base de datos. En el diccionario de datos se encuentran almacenado:

El esquema lógico de la base de datos.

El esquema físico de la base de datos.

Los subesquemas de la base de datos.

El gestor de la base de datos (Intérprete): Este componente es el encargado de proporcionar una interfaz entre los datos almacenados y los programas de aplicación que los manejan.

“Todas las peticiones deben ser permitidas por el gestor”. Es responsable de:

-Garantizar la privacidad, seguridad, integridad y acceso ocurrencia de los datos.

El administrador de la base de datos: Se trata de un componente humano, entre sus tareas asignadas se encuentran:

-La definición del esquema canónico.

-La definición del esquema físico.

-La definición de los subesquemas o visiones.

-El control de la privacidad.

Los usuarios de la base de datos: entre los que podemos considerar:
Usuarios terminales, técnicos, especializados, críticos...

8. Compatibilidad de relaciones: dos relaciones R1 y R2 son compatibles si ambas tienen el mismo grado y el atributo i-ésimo de R1 está definido en el mismo dominio que el atributo i-ésimo de R2.

9. ¿Qué problemas evita formalizar en FN2 en la operación de Inserción?

Redundancia de valores de atributos dependientes, ya que se repite su valor para un mismo valor del atributo clave. Inconsistencia de valores de atributos dependientes, ya que pueden tener distinto valor para un mismo valor del atributo clave. No se puede almacenar la dependencia funcional completa hasta que no se haya insertado una tupla.

10. Operadores Básicos

Operador resta: Dadas dos relaciones compatibles R1 y R2, la diferencia es una nueva relación R3, compatible con R1 y R2 cuya extensión será todas aquellas tuplas de R1 que no estén presentes en la extensión de R2.

Operador división: Dadas dos relaciones R1 con esquema (x_a, x_b, \dots, x_z) y R2 con subesquema de R1 (x_l, x_j, x_n) , la división entre R1 y R2, es una relación R3 con esquema igual a la diferencia de los esquemas de R1 menos R2 y con extensión igual a todas aquellas tuplas sin repetición de R1 para las cuales en R1 está presente toda la extensión R2.

Operador reunión: La reunión de dos relaciones R1 y R2 (no necesariamente compatibles), pero en las que existe al menos un atributo con el dominio común, sobre una cualificación Q, es una nueva relación R3 cuya intención está formada por la concatenación de las intenciones R1 y R2, y cuya extensión está formada por las tuplas que resultan del producto cartesiano de $R1 \times R2$ que satisfacen la cualificación Q.

Operador Unión: La unión de dos relaciones compatibles R1 y R2, es una nueva relación R3 también compatible, cuyo esquema es igual al de R1 y R2 y cuya extensión está formada por la agrupación, **sin repetición**, de las extensiones R1 y R2.

11. ¿Por qué la FN2 no requiere que una relación esté en FN2 o FN3?

Porque la FN2 se basa en el concepto de determinante funcional que incorpora el concepto de dependencia funcional completa. Dado que los determinantes funcionales sólo pueden ser claves candidatas de una relación, en su definición está implícita que las dependencias de los atributos de la clave deben ser completas (FN2) y que no puede haber dependencias funcionales entre atributos que no sean clave (FN3)

12. Base de datos: es una colección de archivos relacionados que almacenan tanto una representación abstracta del dominio de un problema del mundo real, como los datos correspondientes a la información acerca del mismo. Tanto la representación como los datos



están sujetos a una serie de restricciones, las cuales forman parte del dominio del problema y también están almacenadas en esos ficheros.

13. Granularidad: Al nivel de detalle en que pueden ser descritas las representaciones externas derivadas de la representación lógica se le denomina Granularidad. "Una mayor granularidad proporciona una mayor independencia".

14. Ligadura: Para garantizar la integridad es necesario que en algún momento los procedimientos que manipulan la representación de un nivel tengan en cuenta como se representa otro nivel, a este proceso de vinculación se le denomina ligadura. Hay de dos tipos:

Ligadura lógica: externa y lógica.

Ligadura física: lógica y física.

15. Propiedades de las dependencias funcionales:

1. Reflexiva: todo conjunto de atributos de una relación es funcionalmente dependiente de sí mismo y de cualquiera de sus posibles agregados.
2. Aumento: $R.a \rightarrow R.b$ entonces $R.(a+c) \rightarrow R.(b+c)$.
3. Transitiva: $R.a \rightarrow R.b$ y $R.b \rightarrow R.c$ entonces $R.a \rightarrow R.c$.

A estas tres reglas se les denomina Axiomas de Armstrong.

16. Define el concepto de dependencia funcional y dependencia funcional completa. Explique cada una de las propiedades de la dependencia funcional.

Dependencia Funcional: Dada una relación R, se dice que el atributo R.y perteneciente a R es funcionalmente dependiente de otro atributo R.x perteneciente a R y se expresa de la forma $R.x \rightarrow R.y$ si, y solo si, cada valor de R.x tiene asociado a él exactamente un valor de R.y para cualquier extensión de la relación R.

Dependencia Funcional Completa: Se dice que el atributo R.y perteneciente a R es funcionalmente dependiente y de forma completa de otro atributo R.x perteneciente a R si, y solo si, depende funcionalmente de R.x y no de ningún subconjunto de los atributos que formen parte del atributo R.x.

Propiedades:

Reflexiva: Dados los atributos a y b de una relación R, para los que se cumple que R.b de R.a entonces en la relación R, está presente una dependencia funcional de la forma $R.a \rightarrow R.b$

Aumento: Dados los atributos a y b de una relación R en la que está presente la dependencia funcional $R.a \rightarrow R.b$ entonces también estará presente la dependencia funcional $R.(a+c) \rightarrow R.(b+c)$, siendo cualquier otro atributo que forme parte de la intención de la relación R.

Transitiva: Dados los atributos a, b y c de una relación R en la que está presente las dependencias funcionales $R.a \rightarrow R.b$, y $R.b \rightarrow R.c$, entonces también estará presente la dependencia funcional $R.a \rightarrow R.c$.

Unión: Dados los atributos a, b y c de una relación R, en la que están presente las dependencias funcionales $R.a \rightarrow R.b$ y $R.a \rightarrow R.c$, entonces también estará presente la dependencia funcional $R.a \rightarrow R.(b+c)$

Pseudo-transitiva: Dados los atributos a, b, c y d de una relación R en la que están presentes las dependencias funcionales $R.a \rightarrow R.b$ y se cumple que $R.c \rightarrow R.b$ y $R.(b+c) \rightarrow R.d$ entonces también estará presente la dependencia funcional $R.(a+c) \rightarrow R.d$

Descomposición: Dados los atributos a, b y c de una relación R en la que esta presente la dependencia funcional $R.a \rightarrow R.b$ y se cumple que $R.c \rightarrow R.b$ entonces también estará presente la dependencia funcional $R.a \rightarrow R.c$.

17. Explique todas las variantes posibles de transformación del modelo conceptual al modelo relacional para una relación que tiene una cardinalidad máxima 1:N

Existen dos tipos que son RTECAR-3.1 y RTECAR-3.2 en los cuales RTECAR-3.1 es un tipo de interrelación binaria 1:N ambos tipos de entidad participan de forma total, o el tipo de entidad que interviene con cardinalidad máxima muchos participa de forma parcial, entonces, cada tipo de entidad se transforma en una tabla por aplicación de la regla RTECAR-1, y el identificador del tipo de entidad que participa con cardinalidad máxima uno pasa a formar con cardinalidad máxima uno pasa a formar parte de la tabla correspondiente al tipo de entidad que participa con cardinalidad máxima muchos. Este atributo será definido como clave foránea de esta tabla manteniendo una referencia con la tabla correspondiente al tipo de entidad que participa con cardinalidad máxima uno. Si el tipo de interrelación tuviera atributos asociados, estos atributos pasan a formar parte de la tabla correspondiente al tipo de entidad que participa con cardinalidad máxima muchos.

RTECAR-3.2: Si en un tipo de interrelación binaria 1:N ambos tipos de entidad participan de forma parcial o únicamente el tipo de entidad que interviene con cardinalidad máxima uno participa de forma parcial, entonces, cada tipo de entidad se transforma en una tabla correspondiente al tipo de interrelación. Esta tabla estará formada por los identificadores de los tipos de entidad que se intervienen en el tipo de interrelación y por todos los atributos asociados al tipo de interrelación. La clave principal de esta tabla será el atributo identificador correspondiente al tipo de entidad que interviene con cardinalidad máxima muchos, y será necesario definir como claves foráneas los atributos identificadores correspondientes a los dos tipos de entidad.

18. Definición Base de datos

Una base de datos es una colección de archivos relacionados que almacenan tanto una representación abstracta del dominio de un problema del mundo real cuyo manejo resulta de interés para una organización, como los datos correspondientes a la información acerca del mismo. Tanto la representación como los datos están sujetos a una serie de restricciones, las

cuales forman parte del dominio del problema y cuya descripción esta también almacenada en esos ficheros.

19. Compatibilidad de relaciones

Dos relaciones R1 y R2 se dice que son compatibles si ambas relaciones tienen el mismo grado y el atributo enésimo de R1 esta definido en el mismo dominio que el atributo enésimo de la relación R2 si bien el nombre de los atributos puede ser diferente.

20. Responde a las siguientes cuestiones

a. Conjunto DBTG en las bases de datos en red.

b. Tipo de registro ficticio en las bases de datos jerárquicas.

c. Transacción.

d. Autonomía en las bases de datos distribuidas.

a. Un conjunto DBTG está formado por un tipo de registro owner y un tipo de registro member y un tipo de set que mantiene entre ambos tipos de registros una relación 1:1 o 1:N. Los tipos de registros representa a conjuntos de objetos del dominio del problema representados por un conjunto de atributos y el tipo de set representa el conjunto de relaciones existentes en el dominio del problema entre esos objetos.

b. Dado que el modelo jerárquico no permite representar relaciones N:N, en ocasiones los tipos de registros deben ser duplicados. Un tipo de registro ficticio es la duplicación de un tipo de registro existente en el modelo cuyos atributos son punteros a los registros existentes en el tipo de registro original.

c. Una transacción es un conjunto de operaciones constituidas por un conjunto de acciones que se ejecutan contra una base de datos y que debe satisfacer las reglas ACID.

Una acción es cualquier acción que acceda a la base de datos (leer, escribir).

Una operación es un conjunto de acciones que actúa sobre el mismo gránulo de la base de datos. Las reglas ACID: Atomicidad, definitividad, aislamiento y definitividad y serializabilidad. (Explicar cada una de ellas)

d. La autonomía en las bases de datos distribuidas es la capacidad que tienen los DBA locales de actuar sobre el esquema de la base de datos distribuida, actualizando o modificando dicho esquema definiendo nuevos objetos o modificando los existentes.

21. Integridad de las bases de datos relacionales:

a. Definición y Tipos de integridades.

b. Reglas, definición y ejemplos.

a. El término integridad de una base de datos hace referencia a la corrección, exactitud, validez y precisión de los datos con respecto al problema del mundo real que está representado. Una

base de datos es íntegra cuando está representado fielmente, sin error, ni ambigüedad el estado actual del problema del mundo real.

Podemos considerar que existen dos tipos de integridades:

- Integridad semántica que considera: integridad de clave, referencial y de dominio.
 - Integridad funcional: asertos y triggers.
- **Integridad de clave:** en toda tabla de una base de datos debe existir al menos uno o un conjunto de atributos que identifiquen sin ambigüedad a cada tupla de la tabla. Estos atributos no podrán tomar valores nulos. De entre todos los atributos o conjuntos de atributos que cumplen esta propiedad uno deberá seleccionarse como clave principal, y los otros como clave secundaria.
Ejemplo: PRIMARY KEY (atributo)
 - **Integridad de referencia:** Dados dos atributos x, y pertenecientes al mismo dominio primario D y existentes en la misma o diferentes relaciones R, y podrá tomar valores nulos o existentes para y en la extensión de R, si y es clave principal de R.

Ejemplo: `CONSTRAINT fk_ejemplo FOREIGN KEY (atributo) REFERENCE Tabla(atributo)`

- **Integridad de dominio:** los atributos de una relación deben estar definidos en dominios que garanticen el conjunto de valores que pueden tomar para cualquier tupla de una relación.
Ejemplo: `CONSTRAINT CHECK chk_1 atributo NOT NULL`
Ejemplo. `CONSTRAINT CHECK chk_2 atributo values "H", "M".`
- **Asertos:** un aserto es un predicado que expresa una condición que se desea que la base de datos satisfaga siempre. Las restricciones de dominio y las de integridad referencial son formas especiales de los asertos.
Ejemplo: `create assertion <nombre-aserto> check <predicado>`
`create assertion restricción-suma check`
`(not exists (select * from sucursal`
`where (select sum(importe) from préstamo`
`where préstamo.nombre-sucursal = sucursal.nombre-sucursal)`
`>= (select sum (importe) from cuenta`
`where préstamo.nombre-sucursal = sucursal.nombre-sucursal)))`
- **Triggers:** es una orden que el sistema ejecuta de manera automática como efecto secundario de la modificación de la base de datos. Para diseñar un mecanismo disparador hay que cumplir dos requisitos:
 - Especificar las condiciones en las que se va a ejecutar el disparador. Esto se descompone en un evento que causa la comprobación del disparador y una condición que se debe cumplir para ejecutar el disparador.



- Especificar las acciones que se van a realizar cuando se ejecute el disparador.
- Este modelo de disparadores se denomina modelo evento-condición-acción.
Ejemplo: CREATE TRIGGER <trigger name>
<BEFORE|AFTER> <INSERT|DELETE|UPDATE> ON <relation name>
FOR EACH <ROW|STATEMENT>
EXECUTE PROCEDURE <procedure name>
(<function args>);

CREATE TRIGGER contar AFTER INSERT ON Estudiantes WHEN (new.edad < 18) FOR
EACH ROW BEGIN cont := cont + 1; END

22. Responde a las siguientes cuestiones (2.5 puntos):

- Definir y describir los componentes del SGBD.
- Tipo de registro ficticio (dummy) en las bases de datos en red.
- Transacción.
- Checkpoint.

a. DDL (explicar), DSDL (explicar), DCI (explicar). DML (explicar).

Diccionario de Datos (explicar).

DBA (explicar)

Usuarios (explicar)

b. En el modelo en red las relaciones muchos a muchos son mantenidas a través de la definición de un tipo registro dummy, que puede tener atributos (o no) que mantiene dos relaciones 1 a muchos con los dos tipos de registro que en el problema mantienen una relación muchos a muchos.

c. Una transacción es un conjunto de operaciones constituidas por un conjunto de acciones que se ejecutan contra una base de datos y que debe satisfacer las reglas ACID. Una acción es cualquier acción que acceda a la base de datos (leer, escribir). Una operación es un conjunto de acciones que actúa sobre el mismo gránulo de la base de datos. Las reglas ACID: Atomicidad, definitividad, aislamiento y definitividad y serializabilidad. (Explicar cada una de ellas)

d. Los checkpoint son puntos de control que se realizan en las copias de seguridad incrementales de las bases de datos. Cada vez que se produce un checkpoint se pasa a definitivas y almacenamiento estable las transacciones realizadas desde el último checkpoint. Esto se realiza sobre los ficheros logs/redo tanto si el procedimiento es con almacenamiento por adelantado o diferido de las transacciones en la base de datos.

23. Resolver las siguientes cuestiones breve y correctamente: (2.5 puntos)

- Compatibilidad de relaciones
- ¿Qué problemas evita normalizar en FN2 en las operaciones de Inserción?
- Operador resta

d. Operador división

e. ¿Por qué la FNBC no requiere que una relación esté en FN2 o FN3)

a. Dos relaciones R1 y R2 son compatibles si ambas tienen el mismo grado y el atributo i-ésimo de R1 está definido en el mismo dominio que el atributo i-ésimo de R2

b. Redundancia de valores de atributos dependientes, ya que se repite su valor para un mismo valor del atributo clave.

Inconsistencia de valores de atributos dependientes, ya que pueden tener distinto valor para un mismo valor del atributo clave.

No se puede almacenar la dependencia funcional completa hasta que no se haya insertado una tupla

c. Dadas dos relaciones compatibles R1 y R2, la diferencia de la relación R2 sobre R1 es una nueva relación R3, compatible con R1 y R2, y cuya extensión está formada por todas aquellas tuplas de R1 que no están presentes en la extensión de R2.

d. Dadas dos relaciones R1 con esquema (xa, xb, ..., xz) y R2 con subesquema de R1 (x1, xj, xn), la división entre R1 y R2, es una relación R3 con esquema igual a la diferencia de los esquemas de R1 menos R2 y con extensión igual a todas aquellas tuplas sin repetición de R1 para las cuales en R1 está presente toda la extensión de R2

e. Porque la FNBC se basa en el concepto de determinante funcional que incorpora el concepto de dependencia funcional completa. Dado que los determinantes funcionales sólo pueden ser claves candidatas de la relación, en su definición está implícita que las dependencias de los atributos de la clave deben ser completas (FN2) y que no puede haber dependencias funcionales entre atributos que no sean clave (FN3)

24. Describa cada uno de los etapas y tareas del análisis y diseño de una base de datos relacionales, indicando en cada ítem su objetivo y actividades que se deben llevar a cabo

0 Descripción y Análisis: La actividad se fundamenta en la descripción detallada del problema, y el análisis de requisitos que sirven de base al modelado de la información

1 Elicitación de los tipos de entidad: Descripción de los objetos del mundo real que representan. Nominarlos

2 Elicitación de los atributos: Descripción de las propiedades de los objetos del mundo real que representan. Nominarlos. Definir el dominio y restricciones.

3 Elicitación de los tipos de interrelaciones: Descripción de las relaciones entre los tipos de entidades. Nominarlas. Definir el significado y cardinalidades mínimas y máximas. Definir si

existen debilidades. Definir los atributos que las caracterizan, nominarlos, establecer su dominio y restricciones.

4 Elicitar toda la semántica: Describir la existencia de relaciones jerárquicas, exclusividades, reflexividades ocualquier otro tipo de semántica avanzada. Caracterizar atributos y las característicasde las relaciones.

5 Construcción del modelo EE/R: Construir el modelo EE/R y validar con la descripción de los elementos.

6 Validación de la información: Validar el modelo EE/R con arreglo a los requisitos de información.

7 Validación funcional: Navegar sobre el modelo validando los requisitos funcionales.

8 Simplificación: Refinar el modelo, simplificándolo para su más fácil, eficiente y completa traducción al modelo lógico. Eliminar atributos múltiples, compuestos y jerarquía.

9 Traducción al Modelo relacional: Aplicar las reglas de transformación del modelo EE/R al modelo Relacional.

10 Normalizar: Normalizar el modelo

11 Validar con requisitos: Incluir todas aquellas restricciones que satisfagan los requisitos. Asertos, triggers, Checks, etc.

12 Refinar y afinar: Refinar el modelo, si fuera necesario, para una más eficiente implementación de la base de datos. Definir índices, tablespaces, etc.

13 Generar la BD: Definir el script de la BD, depurar y validar.

25. Independencia de las bases de datos

Lógica: La modificación de la representación lógica del dominio del problema no afecta a los programas de aplicación, siempre que no elimine ningún ítem de datos que el programa requiera.

Física: La distribución de los datos de las unidades de almacenamiento es independiente de los cambios a nivel lógico y, por ende, de los programas de aplicación

26. Características de las bases de datos

Versatilidad para representar la información: ya que existen varias visiones de esa información.

Desempeño: Las bases de datos deben asegurar un tiempo de respuesta adecuado, permitiendo el acceso simultáneo a la BD por un mismo o distintos procedimientos.

Capacidad de acceso: Ya que los usuarios están constantemente reclamando información sobre los datos almacenados.

Mínima redundancia: Ya que es precisamente el motivo por el que surgieron las bases de datos, cuando las estructuras planas presentaban una gran redundancia.

Simplicidad: Las BD deben estar basadas en representaciones lógicas sencillas que permitan su verificación.

Integridad: Hace referencia a la veracidad de los datos almacenados con respecto a la información del dominio del problema.

Seguridad y privacidad: Se trata de la protección de los datos contra la pérdida de los mismos por fallos del sistema o accesos accidentales o mal intencionados.

Afinación: Hace referencia a la organización física de la información en la BD, lo que determina el tiempo de respuesta en los procedimientos y el cumplimiento de las demás propiedades.

Interfaz con el pasado y el futuro: El problema cambia con el tiempo y con ello, su dominio.

27. Visiones de los datos de una base de datos

Externa: Es la visión que tienen los usuarios finales.

Conceptual: Es la visión del problema tal y como se representa en la realidad.

Física: Es la representación de como se organiza la información en los dispositivos de almacenamiento.



28. Recuperación de las bases de datos

El objetivo de la recuperación es proteger la base de datos contra fallos lógicos y físicos que destruyan los datos, y pueden afectar a la memoria volátil o la memoria secundaria.

29. Modelo de datos

Es un conjunto de reglas de acuerdo con las cuales puede ser descrito un fenómeno.

Sistema de Gestión de BD	Modelo de Datos
Jerárquicos	Jerárquicos
En red	Plex
Relacionales	Relacional
Orientados a Objetos	Orientados a Objetos

30. Intención y extensión

La intención es la descripción del tipo de objeto y la extensión es la descripción de las instancias de objetos.

31. Claves

El identificador o superclave es el conjunto de atributos que indentifican de forma única a cada entidad. Cada una de las superclaves formadas por el mínimo número de campos posibles se llama clave candidata.

32. Tipos de entidades: fuertes y débiles

En el caso de un tipo de entidad fuerte, su existencia no depende de la existencia de ningún otro tipo de entidad. Mientras que la débil depende de la existencia de un tipo de entidad fuerte.

33. Compatibilidad de relaciones

Dos relaciones R1 y R2 son compatibles si ambas tienen el mismo grado y el atributo i-ésimo de R1 esta definido en el mismo que el atributo i-ésimo de R2

34. Definición de Formas normales

FN1: Una relación R satisface la primera forma normal si y solo si todos los dominios subyacentes de la relación R contienen valores atómicos. En otras palabras, prohíbe a un campo contener mas de un dato de su dominio de columna. Tambien exige que todas las tablas tengan una clave primaria. Por último, indica que una tabla no debe tener atributos que acepten valores nulos.

FN2: Una relación R satisface la segunda forma normal si y solo si satisface la FN1 y cada atributo de la relación depende funcionalmente de forma completa de la clave primaria de esa relación.

FN3: Una relación R satisface la segunda forma normal si y solo si satisface la FN1 y cada atributo de la relación no depende funcionalmente de forma transitiva de la clave primaria de esa relación. Es decir, no pueden existir dependencias entre los atributos que no forman parte de la clave primaria.

FNBC: Una relación se encuentra en FNBC, si y solo si, se encuentra en FN1 y toda la clave candidata es un determinante funcional de la relación R.

35. Interrelaciones reflexivas

Son las relaciones unarias, por lo que consideran que la interrelación solo hay involucrado un único tipo de entidad.

36. Abstracción

Es la capacidad mediante la cual una serie de objetos se categorizan en un nuevo objeto, al que llamamos tipo de objeto, mediante una función de pertenencia. Todos los elementos categorizados en esta clase tienen propiedades comunes.

37. Eliminar super tipo y subtipo

Para eliminar los supertipos, se usa PRTCAR-3, pero es conveniente usarla si las interrelaciones son totales y exclusivas, y el número de atributos e interrelaciones del supertipo es pequeño. Se borra el supertipo de entidad y se transfieren sus atributos a cada uno de los subtipos de entidad, así como las interrelaciones en las que participaba el supertipo.

Para eliminar los subtipos se borran los tipos de entidad y se transfieren los atributos al supertipo así como sus relaciones. Si la jerarquía es exclusiva, el supertipo participará con las mismas cardinalidades con las que participaban los subtipos. Los identificadores formarán parte de la clave solo si la jerarquía es inclusiva y solo podrá tomar valores nulos en caso de jerarquía parcial.

Para eliminar la jerarquía se transforma la relación en tantas relaciones 1:1 como subtipos haya, manteniéndose las relaciones intactas. En las nuevas relaciones, los subtipos participarán parcialmente si la jerarquía es exclusiva y total o parcialmente si es inclusiva, y el supertipo con cardinalidad 1:1

38. Bases de datos distribuidas

una Base de Datos Distribuida es una colección de datos que pertenecen lógicamente a un solo sistema, pero se encuentra físicamente distribuido en varios computadores o servidores de datos en una red de computadoras. Un sistema de bases de datos distribuidas se compone de un conjunto de sitios lógicos, conectados entre sí, mediante algún tipo de red de comunicaciones, en el cual.

39. Bases de datos activas

El modelo ECA (Evento-Condición-Acción) se usa para especificar las reglas de las bases de datos activas que permiten realizar acciones de forma automática. Estas reglas se denominan triggers (disparadores) y su finalidad es convertir una base de datos pasiva en activa.

40. Bases de datos replicadas

La replicación consiste en la copia y mantenimiento de objetos de la base de datos en múltiples bases de datos que forman un SBDD. Permiten un acceso a los datos en todo lugar y en todo momento. Los cambios aplicados a un sitio se capturan y se almacenan localmente antes de ser enviados a cada uno de los otros sitios.