

Cuestionario control diseño de bases de datos

Respuestas

1. Se genera una descomposición con pérdida ya que en las entidades generadas no hay un atributo que identifique a la otra entidad de manera única. La propuesta de tal atributo podría ser el nombre, sin embargo los nombres se pueden llegar a repetir por lo tanto ese atributo no nos sirve para relacionar ambas entidades generadas.
2. FALTA
3. Una superclave es la encargada de identificar a un conjunto de atributos de manera única, mientras que una dependencia funcional es un concepto más general para indicar que la existencia de cierto conjunto de atributos depende de otro conjunto de atributos.
4. NO EXISTE
5. Descomponer una entidad en otras entidades basado en las dependencias funcionales nos permite evitar redundancia en los datos y facilitar los procesos de inserción, eliminación y actualización de algún registro en la base de datos.
6. NO EXISTE
7. Corregir datos duplicados y anomalías en la base de datos. Evitar la creación y actualización de relaciones y dependencias no deseadas entre datos. Prevenir borrados indeseados de datos. Optimizar el espacio de almacenamiento. Reducir el tiempo y complejidad de revisión de las bases de datos cuando es necesario introducir nuevos tipos de datos. Facilitar el acceso e interpretación de los datos a los usuarios y aplicaciones que los usan.
8. Puede que existan entidades que por definición se encuentran en la forma Normal de Boyce Codd, sin embargo es notable que estas requieren más normalización como es el caso de la entidad `inst_info` que se menciona en el libro.
9. El poder descomponer las entidades a una forma más atómica y el poder identificar las relaciones entre dichas entidades y así evitar errores en los diseños de bases de datos.
10. Es el conjunto de todas las dependencias incluidas en F , junto con las dependencias que pueden inferirse de F , reciben el nombre de clausuras de F ; está designada mediante F^+ , donde F es el conjunto de todas las dependencias funcionales de un esquema de relación.
11. $R = (A, B, C, G, H, I)$ y $F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$
 - a) $A \rightarrow H$ (transitivity), $AG \rightarrow I$ (augmenting then transitivity), $CG \rightarrow HI$ (augmenting and transitivity)
12. Algoritmo para calcular F^+
 - a) Inicializar F^+ con F
 - b) Para cada dependencia funcional en F^+
 - 1) Aplicar reglas de aumentación y reflexividad y añadir las dependencias funcionales resultantes a F^+
 - c) Para cada par de dependencias funcionales f_1 y f_2 en F^+
 - 1) si f_1 y f_2 pueden ser combinadas usando transitividad entonces añadir la dependencia funcional resultante a F^+
 - d) repetir desde el paso b hasta que F^+ deje de modificarse
13. - Reglas de cerradura adicionales:
 - a) si $a \rightarrow b$ y $a \rightarrow c$, entonces $a \rightarrow bc$
 - b) si $a \rightarrow bc$, entonces $a \rightarrow b$ y $a \rightarrow c$
 - c) si $a \rightarrow b$ y $cb \rightarrow z$, entonces $ac \rightarrow z$
14. Algoritmo para determinar la cerradura de atributos sobre un conjunto de dependencias funcionales A
 - a) Inicializar el resultado como A .
 - b) repetir mientras haya cambios en el resultado
 - 1) para cada dependencia $b \rightarrow c$
 - Si b es un subconjunto del resultado entonces hacer $resultado = resultado \cup c$
15. Verificar la selección de una superclave, verificar las dependencias funcionales y calcular la cerradura de F .

16. NO EXISTE

17. a cubierta canónica es el mínimo conjunto de dependencias funcionales equivalentes a F . Sin dependencias redundantes ni partes de dependencias redundantes.
18. Básicamente si un atributo A se puede eliminar sin afectar la cerradura del conjunto de dependencias funcionales, entonces A es un atributo extraño.
19. Para probar si un atributo A es extraño en $a \rightarrow b$ en F se hace lo siguiente:
 - a) para checar si es extraño en a
 - 1) calcular $(\{a\} - A)^+$ usando dependencias de F
 - 2) si $(\{a\} - A)^+$ contiene a b , entonces A es extraño en a
 - b) para checar si es extraño en b
 - 1) calcular a^+ usando solo dependencias de en $F' = (F - \{a \rightarrow b\}) \cup \{a \rightarrow (b - A)\}$
 - 2) si a^+ contiene a A , entonces A es extraño en b
20. La cubierta canonica para F es un conjunto de dependencias F_c tal que:
 - a) F implica logicamente todas las dependencias de F_c
 - b) F_c implica todas las dependencias de F
 - c) Ninguna dependencia en F_c contiene un atributo extraño
 - d) El lado izquierdo de todas las dependencias funcionales en F_c es único.
21. La descomposición sin pérdidas implica que al dividir una relación en 2 o más relaciones, se puede regresar a la relación original con la unión natural de las relaciones generadas.
22. Para checar si una dependencia $a \rightarrow b$ se preserva en una descomposición de R en R_1, R_2, \dots, R_n aplicamos las siguientes pruebas:
 - a) inicializamos el resultado en a
 - b) repetir mientras haya cambios en el resultado:
 - 1) Para cada R_i en la descomposición

$$a' \ t = (\text{resultado} \cap R)^+ \cap R_i$$

$$b' \ \text{resultado} = \text{resultado} \cup t$$
 - c) si el resultado contiene todos los atributos en b , entonces la dependencia funcional $a \rightarrow b$ se preserva.
23. Para checar si una dependencia no trivial $a \rightarrow b$ causa una violación a la BCNF
 - a) calcular a^+
 - b) verificar que incluya todos los atributos de R , es decir, que sea una superclave en R .
24. Algoritmo BCNF
 - a) Inicializar *resultado* con R
 - b) Inicializar una variable $DONE = \text{falso}$
 - c) repetir mientras $DONE$ sea *falso*
 - 1) Si existe un esquema R_i en *resultado* que no esta en BCNF, entonces para cada dependencia $a \rightarrow b$ en R_i , tal que $a \rightarrow R_i$ no este en R^+ hacer:

$$a' \ \text{resultado} = (\text{resultado} - R) \cup (R_i - b) \cup (a, b)$$
 - 2) sino existe hacer $DONE = \text{verdadero}$
25. $R = (A, B, C), F = \{A \rightarrow B, B \rightarrow C\} \text{key} = \{A\}$
 - a) R no esta en BCNF ($B \rightarrow C$, B no es una superclave)
 - b) descomponemos en R_1 y R_2
 - 1) $R_1 = (B, C)$
 - 2) $R_2 = (A, B)$

26. Ejemplo de descomposición en 3NF

a) Esquema:

1) cust_banker_branch = (customer_id, employee_id, branch_name, type)

b) Dependencias Funcionales:

1) customer_id, employee_id \rightarrow branch_name, type

2) employee_id \rightarrow branch_name

3) customer_id, branch_name \rightarrow employee_id

c) primero calculamos la cubierta canónica

1) branch_name es extraño en el lado derecho de la primer dependencia, porque podemos determinar branch_name con solo employee_id

2) Obtenemos Fc =

a' customer_id, employee_id \rightarrow type

b' employee_id \rightarrow branch_name

c' customer_id, branch_name \rightarrow employee_id

d) El bucle genera el siguiente esquema 3NF:

1) (Customer_id, employee_id, type)

2) (employee_id, branch_name)

3) (customer_id, branch_name, employee_id)

e) El esquema simplificado resultante es:

1) (customer_id, employee_id, type)

2) (customer_id, branch_name, employee_id)

27. El propósito de una base de datos relacional es:

a) BCNF

b) Descomposición sin pérdidas

c) Preservación de las dependencias

28. Las dependencias multivaluadas se generan cuando un atributo A determina a otro, pero este no siempre tiene los mismos valores debido a A . Un ejemplo seria:

Nº Curso	Profesor	Material
17	Eva	1
17	Eva	2
17	Julia	1
17	Julia	2
25	Eva	1
25	Eva	2
25	Eva	3

Figura 1: Ejemplo de dependencias multivaluadas

29. Formas en las que se utilizan dependencias multivaluadas:

a) Para determinar si las relaciones son legales bajo un conjunto de dependencias funcionales y dependencias multivalor.

b) Para especificar reglas en un conjunto de relaciones legales.

30. Características de las teorías de dependencias

a) Podemos calcular D^+ usando D , usando definiciones formales de dependencias funcionales y multivalor

b) Podemos arreglarnoslas para dependencias muy simples.

c) Para dependencias más complejas es mejor razonar usando un sistema de inferencia de reglas.

31. Un esquema de relaciones R está en 4NF con respecto al conjunto D , de dependencias funcionales y multivalor si para todas las dependencias multivalor en D^+ de la forma $a \twoheadrightarrow b$ donde a y $b \subseteq R$, al menos una de las siguientes condiciones se cumple:
- $a \twoheadrightarrow b$ es trivial (i.e. $b \subseteq R$ ó $a \cup b = R$)
 - a es una superclave en el esquema R
 - si una relación está en 4NF está en BCNF
32. $R = (A, B, C, G, H, I)$ $F = \{A \twoheadrightarrow B, B \twoheadrightarrow HI, CG \twoheadrightarrow H\}$
- $R1 = (A, B)$
 - $R2 = (A, C, G, H, I)$
 - $R3 = (C, G, H)$
 - $R4 = (A, C, G, I)$
 - $R5 = (A, I)$
 - $R6 = (A, C, G)$
33. Mayor velocidad de acceso

Ejercicios

8.11

- α tiene que ser una clave primaria para R_1 y α tiene que ser una clave foranea desde R_2 referenciando a R_1 .
- Si la restricción de la clave foranea no se aplica, entonces al hacer la eliminación de algún registro esta eliminación no se propagaría adecuadamente por lo que puede significar que esta siga existiendo en la BD de alguna manera no deseada.
- Para cada esquema r_i ($\alpha\beta$) añadido al esquema principal por alguna regla $\alpha \rightarrow \beta$, α debería convertirse en una llave primaria. Las llaves foraneas se crean así: para cada relación r_i creada por lo anterior, si los atributos de la llave primaria en r_i también ocurren en otra relación r_j , entonces la llave foranea se crea a partir de esos atributos en r_j , haciendo referencia a la llave primaria en r_i .

8.12

Si $U = R_1, R_2, \dots, R_n$ es un esquema de descomposición y $u(U)$ es una relación y además $r_i = \Pi_{R_i}(u)$. Demostrar que:

$$u \subseteq r_1 \bowtie r_2 \bowtie \dots \bowtie r_n$$

Consideremos alguna tupla t en u .

Notemos que $r_i = \Pi_{R_i}(u)$ implica que $t[R_i] \in r_i, 1 \leq i \leq n$. Por lo tanto,

$$t[R_1] \bowtie t[R_2] \bowtie \dots \bowtie t[R_n] \in r_1 \bowtie r_2 \bowtie r_3 \dots \bowtie r_n$$

Por definición de la unión natural tenemos que

$$t[R_1] \bowtie t[R_2] \bowtie \dots \bowtie t[R_n] = \Pi_{\alpha}(\sigma_{\beta}(t[R_1] \times t[R_2] \times \dots \times t[R_n]))$$

Donde la condición β se satisface si los valores de los atributos con el mismo nombre en una tupla son iguales y $\alpha = U$. El producto cartesiano de tuplas individuales genera una tupla. La selección del proceso se satisface por que todos los atributos con el mismo nombre tienen que tener el mismo valor ya que son proyecciones de la misma tupla. Finalmente la proyección elimina nombres de atributos duplicados.

Por la definición de la descomposición, $U = R_1 \cup R_2 \cup \dots \cup R_n$, lo que significa que todos los atributos de t están en $t[R_1] \bowtie t[R_2] \bowtie \dots \bowtie t[R_n]$. Lo que a su vez significa que t es igual al resultado de esta unión. Ya que t es cualquier tupla arbitraria en u ,

$$u \subseteq r_1 \bowtie r_2 \bowtie \dots \bowtie r_n$$

8.19

Del ejercicio 8.6, sabemos que $B \rightarrow D$ no es trivial ya que el lado izquierdo no es una superllave. Por el algoritmo de la descomposición en BCNF. Deducimos las relaciones $\{(A, B, C, E), (B, D)\}$. Y estas se encuentran en BCNF.

8.20

Primero notamos que las dependencias dadas en el ejercicio practica 8.1 forman una cubierta canonica. Por lo que al aplicar el algoritmo de descomposición sin perdidas a 3NF ya no es necesario generar la cubierta canonica y por lo tanto siguiendo ese mismo algoritmo nos queda:

$$R' = \{(A, B, C), (C, D, E), (B, D), (E, A)\}$$

8.21

Normalice el siguiente esquema con las relgas indicadas a 4NF

- *books*(*accessionno*, *isbn*, *title*, *author*, *publisher*)
- *users*(*userid*, *name*, *deptid*, *deptname*)
- *accessionno* → *isbn*
- *isb* → *title*
- *isbn* → *publisher*
- *isbn* →→ *author*
- *userid* → *name*
- *userid* → *deptid*
- *deptid* → *deptname*

Respuesta:

En *books*, vemos que:

$$isbn \rightarrow\rightarrow title, publisher, author$$

sin embargo *isbn* no puede ser una llave primaria, ya que debido a las dependencias multivalor, *isbn* apareceria repetidas veces por lo que debemos dividir esta relación en (*books*) en otras 2 relaciones:

$$books_accnno(accessionno, isbn)$$

$$books_details(isbn, title, publisher, author)$$

despues de esto tenemos todavia que *isbn* →→ *author* y dado que como un solo *isbn* puede tener varios *author* entonces *isbn* apareceria repetidas veces en la relación *book_details*, por lo que *isbn* no puede ser una llave primaria y ninguno de los otros atributos lo son, entonces tenemos que dividir esta relación aun mas.

$$books_details1(isbn, title, publisher)$$

$$books_authors(isbn, author)$$

Por otro lado en la relación usuarios vemos que:

$$deptid \rightarrow deptname$$

y *deptid* no es una superllave por lo que tendremos que extender esa dependencia funcional a otra relación

$$users(userid, name, deptid)$$

$$deparments(deptid, deptname)$$

y ya no quedan mas dependencias funcionales que se puedan derivar ni tampoco dependencias multivalor que causen la violación de la 4NF por lo que el conjunto final de relaciones queda asi:

$$books_accnno(accessionno, isbn)$$

$$books_details1(isbn, title, publisher)$$

$$books_authors(isbn, author)$$

$$users(userid, name, deptid)$$

$$deparments(deptid, deptname)$$

8.22

La repetición de los datos en una BD genera mas uso del espacio en memoria, hace difícil mantener consistencia en la BD. ya que hay que hacer actualizaciones en varias relaciones.

8.23

Algunas dependencias funcionales son llamadas triviales por que estas dependencias funcionales se cumplen para todas las relaciones.

8.24

8.25

Ejemplo contradictorio:

persona(nss, curp, nombre)

Podemos observar que *nss* (numero de seguridad social) determina el *nombre* de una persona, y podemos observar tambien que el *curp* determina el *nombre* de esa misma persona, sin embargo el *nss* NO determina el *curp* ni el *curp* determina el *nss*

8.26

Demostrar por leyes de Armstrong la regla de la descomposición que indica que:

Si $\alpha \rightarrow \beta\gamma$, entonces $\alpha \rightarrow \beta$ y $\alpha \rightarrow \gamma$

$\alpha \rightarrow \beta\gamma$

$\beta\gamma \rightarrow \beta$ regla de reflexividad

$\alpha \rightarrow \beta$ regla de transitividad

$\beta\gamma \rightarrow \gamma$ regla de reflexividad

$\alpha \rightarrow \gamma$ regla de transitividad

8.27

Computar B^+ del ejercicio de practica 8.6:

Comenzamos con $resultado = \{B\}$. Considerando las DFs de la forma $\beta \rightarrow \gamma$, encontramos que las unicas dependencias que satisfacen que $\beta \subseteq resultado$ son $B \rightarrow B$ y $B \rightarrow D$ Por lo tanto $B^+ = resultado = \{B, D\}$.

8.28

Demostrar por contradicción que la descomposición del ejercicio practica 8.1 tiene perdida en la unión.

Definamos r como:

A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_2	b_2	c_1	d_2	e_2

Con $R_1 = (A, B, C)$ y $R_2 = (C, D, E)$

1. $\Pi_{R_1}(r)$ seria:

A	B	C
a_1	b_1	c_1
a_2	b_2	c_1

2. $\Pi_{R_2}(r)$ sería:

C	D	E
c_1	d_1	e_1
c_1	d_2	e_2

3. $\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r)$ sería:

A	B	C	D	E
a_1	b_1	c_1	d_1	e_1
a_1	b_1	c_1	d_2	e_2
a_2	b_2	c_1	d_1	e_1
a_2	b_2	c_1	d_2	e_2

Lo que hace evidente que $\Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) \neq r$, por lo que es una descomposición con pérdida

8.29

8.30

Listar 3 objetivos en el diseño de base de datos relacionales.

1. Descomposición sin pérdidas
2. Preservación de las dependencias
3. Minimización de la repetición de los datos.

Preguntas:

Número	Pregunta / Tópico	Metadato
1	Considere el esquema <i>employee (ID, name, street, city, salary)</i> , ahora descompóngalo, en los siguientes subesquemas <i>employee1 (ID, name)</i> <i>employee2 (name, street, city, salary)</i> , incorporeal datos de prueba, asumiendo que dos empleados se llaman igual, por ejemplo “Kim”. Genere el JOIN Natural y muestre si con este esquema se obtuvo descomposición sin pérdida es decir si fue una buena o mala descomposición.	Pag. 327 Korth & Silberschartz
2	Un modelo de datos es un conjunto de entidades y sus relaciones en el mundo real. Muestre las restricciones que se mantienen en la base de datos Universidad, acerca de que los estudiantes e instructores se identifiquen por una llave única, que los estudiantes e instructores tengan un solo nombre. Que los instructores estén asociados en un departamento y que cada departamento tenga asignado un presupuesto..	Pag. 330 Korth & Silberschartz
3	Que características se pueden considerar al comparar una superllave, y una dependencia funcional.	Pag. 331 Korth & Silberschartz
4	Una dependencia funcional permite expresar restricciones que no pueden expresarse con suuperllaves. Considere el esquema <i>inst_dept</i> y la dependencia	Pag. 331 Korth & Silberschartz
5	Que es la descomposición utilizando las dependencias funcionales	Pag. 329 Korth & Silverschartz
6		
7	Cúal es el propósito de la normalización.	
8	Qué tan buena es la forma normal de Boyce and Codd BCNF.	
9	Cuál es el propósito de la teoría de dependencias funcionales.	
10	Qué la cláusura de un conjunto de dependencias funcionales.	Diapositiva 8.26
11	Determine el conjunto de dependencias funcionales por medio de un ejemplo.	Diapositiva 8.28
12	Cuál es el algoritmo o procedimiento para calcular el conjunto de F^* .	Diapositiva 8.29
13	Cuáles son las reglas adicionales para determinar el conjunto de dependencias funcionales.	Diapositiva 8.30
14	Cual es el algoritmo para el calculo de la cerradura del conjunto de atributos.	Diapositiva 8.31
15	Cuáles son los diferentes usos del algoritmo del cerradura del conjunto de atributos.	Diapositiva 8.33
17	Qué es la cobertura canónica F del conjunto de dependencias funcionales.	Diapositiva 8.34
18	Como se determinan los atributos extraños de un conjunto de dependencias funcionales.	Diapositiva 8.36
19	Como se prueba si un atributo es extraño.	Diapositiva 8.36

20	Cuál es la cobertura conónica para un conjunto de dependencias funcionales.	
21	Qué es la descomposición de la unión sin perdida.	Diapositiva 8.39
22	Cual es el algoritmo de prueba de preservación de dependencias.	
23	Cual es la prueba para BCNF la forma normal	Diapositiva 8.44
24	Cuál es el algoritmo de BCNF en Pseudo Código.	Diapositiva 8.46.
25	Muestre un ejemplo de descomposición de BCNF.	Diapositiva 8.47
26	Muestre con un ejemplo la descomposición en tercera forma normal	Diapositiva 8.57 y 8.58
27	Cuál es el propósito de diseño de bases de datos según Korth	Diapositiva 8.60
28	Qué son las dependencias multivaluadas muestre un ejemplo	Diapositiva 8.60
29	Cuáles las dos formas o usos en que se pueden utilizar las dependencias funcionales	Diapositiva 8.61
30	Cuáles son las características de las teoría de dependencias funcionales en cuanto a cerradura.	Diapositiva 8.67
31	Cómo define Korth la cuarta forma normal.	Diapositiva 8.68
32	Muestre con un ejemplo las dependencias funcionales multivaluadas.	Diapositiva 8.71
33	Qué ventajas pueden obtenerse de la desnormalización de un esquema de bases de datos	Diapositiva 8.75
	Problemas	
	Resolver los ejercicios prácticos 8.1 1 8.12 . Resolver los ejercicios 8.19 a 8.30.	Pag. 368 a 374 korth & Silberschartz.