

Modelo Relacional: Dependencias Funcionales y Normalización

M. Andrea Rodríguez-Tastets

Universidad de Concepción, Chile
www.inf.udec.cl/~andrea
andrea@udec.cl

II Semestre - 2012

Introducción

Pautas Informales Diseño

Dependencias Funcionales

Formas Normales (1/2)

Formas Normales (2/2)

Algoritmos de Descomposición

Otras Formas Normales

Objetivos de la Unidad

Estudiar y ejercitar estrategias de diseño y normalización

Problemas de Diseño

- ▶ Incapacidad para almacenar ciertos hechos
- ▶ Redundancias y, por tanto, posibilidad de incoherencias
- ▶ Ambigüedades
- ▶ Pérdida de información (aparición de tuplas espúreas)
- ▶ Pérdida de dependencias funcionales, es decir, ciertas restricciones de integridad que dan lugar a interdependencias entre los datos.
- ▶ Aparición en la BD de estados no válidos, es decir, anomalías de inserción, borrado y modificación.

Ejemplos de Problemas

AUTOR	NACIONALIDAD	COD LIBRO	TITULO	EDITORIAL	AÑO
Date, C.	Norteamericana	98987	Database	Addison, W.	1990
Date, C.	Norteamericana	97777	SQL Stan	Addison, W.	1986
Date, C.	Norteamericana	98987	Guide for	Addison, W.	1988
Codd,E.	Norteamericana	7890	Relational	Addison,W.	1990
Gardarin	Francesa	12345	Basi Dati	Paraninfo	1986
Gardarin	Francesa	67890	Comp BD	Eyrolles	1984
Valduriez	Francesa	67890	Comp BD	Eyrolles	1984
Kim,W.	Norteamericana	11223	BD OO	ACM	1989
Lochovsky	Canadiense	11223	BD OO	ACM	1989

Problemas con esta tabla (1/2)

- ▶ Redundancia, ya que la nacionalidad del autor se repite por cada ocurrencia del mismo. Lo mismo sucede cuando un libro tiene más de un autor, se repite la editorial y el año de publicación.
- ▶ Anomalías de modificación, es fácil cambiar el nombre de una editorial en una tupla sin modificar el resto de las que corresponden al mismo libro, lo que da lugar a incoherencias.
- ▶ Anomalías de inserción, ya que si queremos ingresar información de algún autor, del que no hubiera ningún libro en la base datos, no sería posible, ya que `cod_libro` es parte de la clave primaria de la relación (regla de integridad de la entidad). La inserción de un libro, que tiene dos autores obliga a insertar dos tuplas en la relación.
- ▶ Anomalías de borrado, ya que si queremos eliminar un cierto libro, deberíamos perder los datos de su autor y viceversa.

[Introducción](#)[Pautas Informales
Diseño](#)[Dependencias
Funcionales](#)[Formas Normales
\(1/2\)](#)[Formas Normales
\(2/2\)](#)[Algoritmos de
Descomposición](#)[Otras Formas
Normales](#)

Problemas con esta tabla (2/2)

- ▶ Lo anterior sucede pues no se cumple un hecho básico de todo diseño: **hechos distintos, deben almacenarse en objetos distintos.**
- ▶ Una forma de evitar este tipo de problemas consiste en seguir la metodología propuesta en el curso, es decir, un buen diseño conceptual, un traspaso de éste al modelo relacional y un refinamiento con normas de diseño basadas en la normalización de las relaciones.
- ▶ Ante posibles dudas respecto a si un esquema relacional está correcto, aplicaremos a dicho esquema un método formal de análisis, que permita analizar errores y generar esquemas correctos. Esta es la teoría de la normalización.
- ▶ En el ejemplo anterior, el conjunto de las siguientes relaciones no presenta estos problemas:
 - LIBRO(cod_libro, titulo, editorial, año)
 - AUTOR(nombre, nacionalidad)
 - ESCRIBE(cod_libro, nombre)

Pautas Informales Diseño

- ▶ Pauta 1: Diseñe un esquema de relación que sea fácil de explicar su significado
- ▶ Pauta 2: Diseñe los esquema de relaciones de base de modo que no hayan anomalías de inserción, eliminación o modificación en las relaciones.
- ▶ Pauta 3: Hasta lo posible, evite incluir en una relación atributos cuyos valores puedan ser nulos.
- ▶ Pauta 4: Diseñe los esquemas de relación de modo que puedan REUNIRSE (join) mediante condiciones de igualdad sobre atributos que sean claves primarias o claves externas, a fin de garantizar que no se formarán tuplas espurias.

Nociones Generales

- ▶ **Superclave**: es un subconjunto S del conjunto de atributos de una relación R tal que dada dos t_1 y t_2 de cualquier estado de R se debe cumplir que $t_1[S] \neq t_2[S]$
- ▶ Un atributo es **primo** si es miembro de una clave

Dependencias Funcionales (1/3)

- ▶ Es un concepto muy importante en el diseño de base de datos relaciones
- ▶ Una dependencia funcional es una restricción entre dos conjuntos de atributos de la base de datos.

Sea el esquema de relación R definido sobre el conjunto de atributos A y sean X e Y subconjuntos de A llamados descriptores. Se dice que Y depende funcionalmente de X o que X determina o implica a Y , que se representa por $X \rightarrow Y$, si y solo si, cada valor de X tiene asociado en todo momento un único valor de Y .

Dependencias Funcionales (2/3)

Sea el esquema de relación R definido sobre el conjunto de atributos A y sean X e Y subconjuntos de A llamados descriptores. Se dice que Y depende funcionalmente de X o que X determina o implica a Y , que se representa por $X \rightarrow Y$, si y solo si, cada valor de X tiene asociado en todo momento un único valor de Y .

$X \rightarrow Y$ para dos tuplas cualesquiera t_1 y t_2 de un estado relación r de R , si $t_1[X] = t_2[X]$, entonces $t_1[Y] = t_2[Y]$.

Si una restricción de R dice que no puede haber más de una tupla con un valor X dado en cualquier instancia de relación $r(R)$, X es una clave candidata de R y $X \rightarrow Y$ para cualquier subconjunto de atributos de Y de R .

Si $X \rightarrow Y$, no nos dice que $Y \rightarrow X$ o no.

Dependencias Funcionales (3/3)

- ▶ ej: $\text{cod_libro} \rightarrow \text{titulo}$, el código del libro determina el título. El código del libro es el implicante y título es el implicado. Siempre el implicado es un hecho (una información) acerca del implicante.
- ▶ OBS1: la afirmación cod_libro determina titulo NO significa que a partir de cod_libro podamos conocer el título. Es decir, para un esquema R , si tenemos la dependencia funcional $X \rightarrow Y$, dado un valor de X no podemos en general conocer el valor de Y . Solo nos limitaremos a afirmar que para dos tuplas de cualquier extensión de R que tengan el mismo valor de X , el valor de Y también será igual en ambas.
- ▶ OBS2: Las dependencias son predicados o restricciones sobre cualquier extensión válida del esquema de relación, por lo que observar una determinada extensión (datos) no puede llevarnos a afirmar la existencia de una dependencia funcional.

Dependencias Funcionales Completa (1/2)

Si el descriptor X es compuesto, es decir, $X(X_1, X_2)$, se dice que Y tiene dependencia funcional completa de X , si depende funcionalmente de X , pero no depende de ningún subconjunto del mismo, esto es:

- ▶ $X \rightarrow Y$
- ▶ $X_1 \not\rightarrow Y$
- ▶ $X_2 \not\rightarrow Y$.
- ▶ Se representa $X \Rightarrow Y$.
- ▶ $X \Rightarrow Y$ si y solo si NO $\exists X' \subset X/X' \rightarrow Y$.

Dependencias Funcionales Completa (2/2)

La relación PUBLICA(articulo, revista, numero, pagina) que representa la página inicial en la que comienza un artículo en una revista. Un mismo artículo puede aparecer publicado en distintas revistas y en cada una de ellas, en páginas distintas y una revista publica varios artículos, se tiene:

- ▶ articulo, revista, numero $\rightarrow$ pagina
- ▶ articulo $\rightarrow$ | pagina
- ▶ revista $\rightarrow$ | pagina
- ▶ numero $\rightarrow$ | pagina

Dependencias Funcionales Trivial/Elementales

Una dependencia funcional $X \rightarrow Y$ es trivial si Y es un subconjunto de X ($Y \subseteq X$). Ejemplos:

- ▶ $\text{cod_libro} \rightarrow \text{cod_libro}$
- ▶ $\text{articulo, revista} \rightarrow \text{revista}$.

Una dependencia funcional $X \rightarrow Y$ es elemental si Y es un atributo único, no incluido en X y no existe X' incluido en X tal que $X' \rightarrow Y$, es decir, la dependencia funcional elemental es un dependencia funcional completa no trivial, en la que el implicado es un atributo único.

Reglas de Inferencias para DF (1/2)

- ▶ RI1: Reflexiva: Si $Y \subseteq X$, entonces $X \rightarrow Y$. Un conjunto de atributos siempre se determina a sí mismo, una DF trivial.
- ▶ RI2: Aumento: $\{X \rightarrow Y\} \models XZ \rightarrow YZ$.
- ▶ RI3: Transitiva: $\{X \rightarrow Y, Y \rightarrow Z\} \models X \rightarrow Z$
- ▶ RI4: Descomposición o proyección: $\{X \rightarrow YZ\} \models X \rightarrow Y$.
- ▶ RI5: Unión o aditiva: $\{X \rightarrow Y, X \rightarrow Z\} \models X \rightarrow YZ$
- ▶ RI6: Pseudotransitiva: $\{X \rightarrow Y, WY \rightarrow Z\} \models WX \rightarrow Z$

Regla reflexiva: demostración

Supongamos que $X \supseteq Y$ y que existen dos tuplas t_1 y t_2 en alguna instancia de la relación r de R tal que $t_1[X] = t_2[X]$. Entonces, $t_1[Y] = t_2[Y]$ porque $X \supseteq Y$; por lo tanto, debe cumplirse que $X \rightarrow Y$ en r .

Regla de aumento: demostración

Supongamos que $X \rightarrow Y$ se cumple en un empleo de relación r de R , pero que $XZ \rightarrow YZ$ no se cumple. En tal caso, deben existir dos tuplas t_1 y t_2 en r tales que (1) $t_1[X] = t_2[X]$, (2) $t_1[Y] = t_2[Y]$, (3) $t_1[XZ] = t_2[XZ]$ y (4) $t_1[YZ] \neq t_2[YZ]$. Esto no es posible porque a partir de (1) y (3) se deduce (5) $t_1[Z] = t_2[Z]$, y partir de (2) y (5) se deduce (6) $t_1[YZ] = t_2[YZ]$, lo que contradice (4).

Regla transitiva: demostración

Supongamos que se cumple (1) $X \rightarrow Y$ y (2) $Y \rightarrow Z$ en una relación r . Entonces, para dos tuplas cualesquiera t_1 y t_2 en r tales que $t_1[X] = t_2[X]$, debemos tener que (3) $t_1[Y] = t_2[Y]$ por supuesto (1), y por lo tanto, también tenemos (4) $t_1[Z] = t_2[Z]$ por (3) y por la suposición (2); por lo tanto, se debe cumplir que $X \rightarrow Z$ en r .

Con argumentos similares se pueden demostrar las otras reglas RI4 a RI6.

Reglas de Inferencias para DF (cont.)

- ▶ Las reglas RI1, RI2 y RI3 se conocen como las reglas de inferencia de Armstrong. Estas reglas son correctas y completas.
- ▶ Por correcta, cualquiera dependencia que podamos inferir de un conjunto de DF F por RI1, RI2 y RI3 se cumplirá en todos los estados de la relación r en R .
- ▶ Por completa, el empleo repetido de RI1, RI2 y RI3 para inferir dependencias hasta que no sea posible inferir más dependencias producirá el conjunto completo de dependencias F^+ , lo que se denomina cierre o clausura de F .
- ▶ Sea X un conjunto de atributos en el lado izquierdo de alguna dependencia funcional en un conjunto de dependencias funcionales F , X^+ es el conjunto de atributos determinados funcionalmente por X en F .

Clausura de Atributos

Suponga que $\{A_1, A_2, \dots, A_n\}$ es un conjunto de atributos y S es un conjunto de DFs. La clausura de $\{A_1, A_2, \dots, A_n\}$ bajo un conjunto de DF en S es el conjunto de atributos B tal que cada relación que satisface las DF en S también satisface

$A_1 A_2 \dots A_n \rightarrow B$. Esto significa que $A_1 A_2 \dots A_n \rightarrow B$ se deriva de las DFs en S .

Ejemplo: Considere los atributos A, B, C, D, E, F . Suponga las siguientes DF: $AB \rightarrow C$, $BC \rightarrow AD$, $D \rightarrow E$, $CF \rightarrow B$. Determine la clausura de $\{A, B\}^+$.

Clausura de Atributos: Correctitud

Proposición: Si $A_1A_2 \dots A_n \rightarrow B_1B_2 \dots B_m$ es reconocida por la clausura, entonces es verdadero en cualquier relación que satisfaga las DF en el conjunto S .

Prueba: Se prueba por inducción en el número de veces que se aplica la expansión del conjunto de clausura, donde por cada atributo D que se agrega la clausura de atributos, la DF $A_1A_2 \dots A_n \rightarrow D$ se cumple para todos instancia de la relación.

La base de inducción es el caso cuando el número de pasos en cero. Entonces D debe ser uno de los A_i , es decir, la DF trivial.

Para la inducción se supone que D se agrega cuando se usa el DF $B_1B_2 \dots B_m \rightarrow D$. Por la hipótesis de inducción, R satisface $A_1A_2 \dots A_n \rightarrow B_i$, para $i = 1, 2, \dots, m$. Puesto de otra forma, cualquier par de tuplas de R que estén en concordancia con todos los $A_1, A_2, A_3 \dots A_n$ también lo están con los $B_1, B_2, B_3 \dots B_m$. Ya que R satisface $B_1B_2 \dots B_m \rightarrow D$, entonces estas dos tuplas también concuerdan con D . Entonces, R satisface $A_1A_2 \dots A_n \rightarrow D$ y D es reconocida por la clausura.

[Introducción](#)[Pautas Informales
Diseño](#)[Dependencias
Funcionales](#)[Formas Normales
\(1/2\)](#)[Formas Normales
\(2/2\)](#)[Algoritmos de
Descomposición](#)[Otras Formas
Normales](#)

Clausura de Atributos: Completitud (1/2)

Proposition: La clausura de atributos no falla en descubrir una dependencia funcional a partir de una conjunto inicial de DF del esquema S .

Prueba: Se prueba por contradicción. Para ello, se considera la una instancia I de la base de datos con dos tuplas t y s del esquema $S(A_1, A_2, \dots, A_n, B_1, \dots, B_m)$, con A_i en $\{A_1, \dots, A_n\}^+$ y $\{B_1, \dots, B_m\}$ todos los otros atributos fuera de la clausura.

	$\{A_1, A_2, \dots, A_n\}^+$	Otros atributos
t:	1 1 111	0 0 0 ...00
s:	1 1 111	1 1 1 ...11

Clausura de Atributos: Completitud (2/2)

Suponga que $A_1A_2 \dots A_n \rightarrow B$ es una DF y que no se deriva desde el set F por las reglas de inferencia. Esto es, la clausura $\{A_1A_2 \dots A_n\}^+$ usando el set F no incluye B . Así debemos mostrar que hay al menos una instancia de relación que satisface todas las DF en S y que no satisface $A_1A_2 \dots A_n \rightarrow B$, o de lo contrario estaría en la clausura.

Si existe una DF $C_1C_2 \dots C_k \rightarrow D$ en el conjunto S que la instancia I no satisface, todos los C_i deben estar en el conjunto de atributos A_i y D debe estar entre los otros atributos. Pero entonces, la clausura no es correcta. Así la instancia I no puede existir.

Se debe demostrar ahora que I no satisface $A_1A_2 \dots A_n \rightarrow B$. De seguro $A_1 \dots A_n$ están entre los atributos en los cuales t y s concuerdan. Además, B no está entre $\{A_1, A_2, \dots, A_n\}$ o estaría en la clausura, entonces B es uno de los atributos en los cuales s y t no concuerdan. De tal manera I no satisface $A_1A_2A_n \rightarrow B$.

Por lo tanto, la clausura solo incluye las DFs que se derivan de S .

Conjunto Mínimo de DF

- ▶ Un conjunto de dependencias funcionales F es mínimo si:
 - ▶ Toda dependencia en F tiene un solo atributo en su parte derecha.
 - ▶ No podemos reemplazar ninguna dependencia $X \rightarrow A$ en F por una dependencia $Y \rightarrow A$, donde Y es un subconjunto propio de X , y seguir siendo un conjunto de dependencias equivalente en F . Dos conjuntos de DF F y E son equivalentes si todas las DF en F se pueden inferir de E y viceversa.
 - ▶ No podemos quitar ninguna DF de F y seguir siendo un conjunto de dependencias equivalentes a F
- ▶ Un conjunto mínimo está en forma estándar o canónica y sin redundancia.

Equivalencia de Conjuntos DFs

- ▶ Un conjunto de dependencias funcionales E **está cubierto** por un conjunto de dependencias F , si toda DF en E también está en F^+ .
- ▶ Podemos determinar si F **cubre** a E calculando X^+ respecto a F para cada DF $X \rightarrow Y$ en E , y comprobando después que ese X^+ incluya los atributos en Y . Si esto se cumple para todas las DF en E , entonces F cubre a E , y son equivalentes.
- ▶ Dos conjuntos E y F de dependencias funcionales son **equivalentes** si $E^+ = F^+$.

Formas Normales: Noción Intuitivas (1/6)

- ▶ Primera Forma Normal (1FN) fue introducida por Codd, en su primer trabajo. Es una restricción inherente al modelo relacional por lo que su cumplimiento es obligatorio. Consiste en la prohibición de que en una relación existan grupos repetitivos, es decir, un atributo no puede tomar más de un valor del dominio subyacente.
- ▶ Segunda Forma Normal (2FN), fue introducida por Codd. Una relación está en 2FN, si además de estar en 1FN, todos los atributos que no forman parte de ninguna clave candidata suministran información acerca de la clave completa (una dependencia funcional total)

Formas Normales: Noción Intuitivas (2/6)

- ▶ Para la relación PRESTAMO(num_socio, nombre_socio, cod_libro, fec_prest, editorial, pais) las claves candidatas son: (num_socio, cod_libro) y (nombre_socio, cod_libro)
- ▶ Se puede observar que editorial constituye información acerca del libro, pero no, acerca de la clave completa. Luego, la relación préstamo no se encuentra en 2FN.
- ▶ La solución es descomponer esta relación en las siguientes:
PRESTAMO1(num_socio, nombre_socio, cod_libro, fec_prest)
LIBRO(cod_libro, editorial, pais)
- ▶ En la relación PRESTAMO1, el único atributo que no forma parte de las claves candidatas es fec_prest, pero suministra información acerca de la clave completa. Por lo que está en 2FN.
- ▶ La relación LIBRO, la clave es cod_libro, y los dos atributos editorial y pais suministran información de la clave completa.
- ▶ OBS: Una relación que está formada por un único atributo esta en 2 FN.

Formas Normales: Noción Intuitivas (3/6)

- ▶ Tercera Forma Normal (3FN), propuesta por Codd. Una relación está en 3FN, si además de estar en 2FN, los atributos que no forman parte de ninguna clave candidata facilitan información sólo acerca de la(s) clave(s) y no acerca de otros atributos.
- ▶ En forma general: Si $X \rightarrow A$ es una dependencia no trivial se cumple en R , o bien (a) X es una superclave de R o (b) A es un atributo primo.
- ▶ En la relación PRESTAMO1, el atributo fec_prest facilita información acerca de las claves, ya que no existen más atributos. Por lo que está en 3FN.
- ▶ En la relación LIBRO, el atributo país entrega información acerca de la editorial que publica el libro, por lo que no está en 3FN.
- ▶ La solución es descomponerla en:
LIBRO1(cod_libro, editorial) EDITORIAL(editorial, país) ,
que están en 3FN, ya que todo atributo no clave facilita información acerca de la clave

Formas Normales: Noción Intuitivas (4/6)

- Forma Normal de Boyce y Codd (FNBC). La relación PRESTAMO1, que está en 3FN, todavía presenta anomalías, ya que num_socio y nombre_socio, se repiten innecesariamente por cada cod_libro. Una relación está en FNBC si y solo si, el conocimiento de las claves candidatas permite averiguar todas las interrelaciones existentes entre los datos de la relación, o lo que es igual, las claves candidatas son los únicos descriptores sobre los que se facilita información por cualquier otro atributo. Formalmente, siempre que una dependencia funcional no trivial $X \rightarrow A$ es válida en R , entonces, X es una superclave de R .

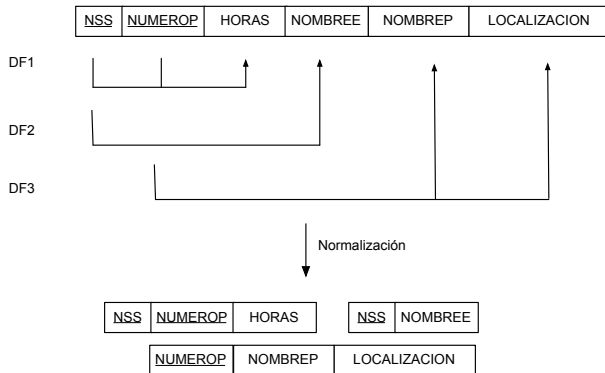
Formas Normales: Noción Intuitivas (5/6)

- ▶ En la relación PRESTAMO1, num_socio es información acerca de nombre_socio y viceversa. Ninguno de estos atributos son clave (aunque formen parte de la clave). Para solucionarlo la descomponemos:
SOCIO(num_socio, nombre_socio)
PRESTAMO2(num_socio, cod_libro, fec_prest), que están en FNBC.
- ▶ Hasta ahora nuestro esquema relacional está compuesto por las siguientes relaciones en FNBC:
LIBRO1(cod_libro, editorial)
EDITORIAL(editorial, país)
SOCIO(num_socio, nombre_socio)
PRESTAMO2(num_socio, cod_libro, fec_prest)

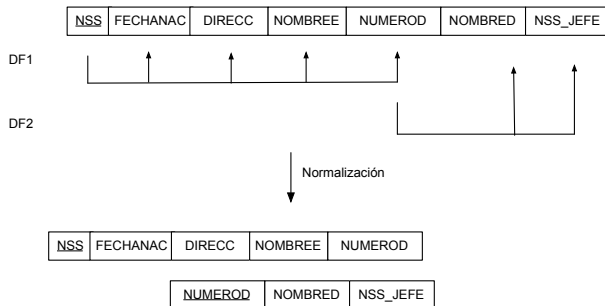
Resumen Formas Normales Basadas en Claves

FN	Comprobación	Solución
1	Una relación no debe tener ningún atributo no atómico ni relaciones anidadas	Formar relaciones nuevas por cada atributo no atómico o relación anidada
2	Para las relaciones en las que la clave primaria contiene múltiples atributos, ningún atributo no clave debería depender funcionalmente de una parte de la clave primaria	Descomponer y crear una nueva relación para cada clave parcial con su atributo(s) dependientes. Asegurarse de mantener una relación con la clave primaria original y todos los atributos que dependen funcionalmente en forma total de ella
3	Una relación no debería tener un atributo no clave determinado funcionalmente por otro atributo no clave (o por un conjunto de atributos no clave). Esto es, no debería existir una dependencia transitiva por parte de un atributo no clave de una clave primaria.	Descomponer y crear una relación que incluya el atributo o atributos no clave que determinen funcionalmente a otro o a otros no clave.

Ejemplo 2FN



Ejemplo 3FN



Definiciones Generales: 2FN, 3FN

- ▶ 2FN: Un esquema de relación R está en 2FN si ningún atributo A de R depende parcialmente de **alguna clave** de R . O dicho de otra forma, un esquema de relación R está en 2FN si todo atributo **no primo** A de R depende funcionalmente de manera total de **toda** clave de R .
- ▶ 3FN: Un esquema de relación R está en 3FN si, siempre que una dependencia funcional no trivial $X \rightarrow A$ se cumple en R , o bien (a) x es una **superclave** de R , o (b) A es un atributo **primo** de R .

Formal Normal de Boyce-Codd (FNBC)

- FNBC: Un esquema de relación R está en FNBC si, siempre que una dependencia no trivial $X \rightarrow A$ es válida en R , entonces X es una superclave de R . A diferencia de la 3FN, no se permite que si A es primo.

Ejemplo (1)

Asuma la siguiente relación IMPARTE con tres atributos (ESTUDIANTE,CURSO,PROFESOR) con las siguientes dependencias funcionales (ejemplo irreal):

ESTUDIANTE	CURSO	PROFESOR
A	DB	X
B	DB	Y
B	SO	Z
B	TE	W
C	DB	X
C	SO	Z
D	DB	V
E	DB	Y

$DF = \{ESTUDIANTE, CURSO\} \rightarrow PROFESOR; PROFESOR \rightarrow CURSO$

Para esta relación, las claves candidatas son:

$DF = \{ \{ESTUDIANTE, CURSO\}, \{ESTUDIANTE, PROFESOR\} \}$

Entonces, esta relación está en 3FN pero no en FNBC

Ejemplo (1)

La descomposición no es simple y existen tres posibilidades:

$\{\text{ESTUDIANTE}, \text{PROFESOR}\}$ y $\{\text{ESTUDIANTE}, \text{CURSO}\}$
 $\{\text{CURSO}, \text{PROFESOR}\}$ y $\{\text{ESTUDIANTE}, \text{CURSO}\}$
 $\{\text{CURSO}, \text{PROFESOR}\}$ y $\{\text{ESTUDIANTE}, \text{PROFESOR}\}$

Estas tres descomposiciones pierden la primera dependencia funcional $DF = \{\text{ESTUDIANTE}, \text{CURSO}\} \rightarrow \text{PROFESOR}$.

Sin embargo, la tercera de las descomposiciones no genera tuplas espúrias en la reunión.

Ejercicios

- ▶ Considere la siguiente relación R con 5 atributos $R(A, B, C, D, E)$. Se tiene además las siguientes dependencias: $A \rightarrow B$, $BC \rightarrow E$, $ED \rightarrow A$. Entonces:
 - ▶ Liste todas las claves candidatas de R
 - ▶ ¿Está R en 3FN?
 - ▶ ¿Está R en FNBC?
- ▶ Considere el esquema de relación $R(A, B, C)$, la cual tiene la DF $B \rightarrow C$. Si A es una clave candidata para R , ¿Es posible para R estar en FNBC? Si es así, ¿bajo qué condición?
- ▶ Asuma un esquema de relación $R(A, B, C, D)$, entonces
 - ▶ Asume que no hay registros con valores NULOS. Escriba una consulta SQL que evalúe si la siguiente DF se satisface : $A \rightarrow B$.
 - ▶ Asuma ahora que pueden haber valores NULOS. Escriba nuevamente la anterior en SQL.

[Introducción](#)[Pautas Informales
Diseño](#)[Dependencias
Funcionales](#)[Formas Normales
\(1/2\)](#)[Formas Normales
\(2/2\)](#)[Algoritmos de
Descomposición](#)[Otras Formas
Normales](#)

Propiedades adicionales de un buen diseño

- ▶ Descomposición y conservación de las dependencias
- ▶ Descomposición y reunión sin pérdidas (no aditivas)

- ▶ Sea $D = \{R_1, \dots, R_m\}$ una descomposición de R , el objetivo es que toda dependencia funcional $X \rightarrow Y$ especificada en F apareciera directamente en uno de los esquemas de relación R_i de la descomposición D , o bien, que se pudiera inferir de las dependencias que aparecen en alguna R_i .
- ▶ La unión de las dependencias en las relaciones individuales en D deben ser equivalente a las DF F de R .
- ▶ Dado un conjunto de dependencias F sobre R , la **proyección** de F sobre R_i , denotada por $\pi_{R_i}(F)$ donde R_i es un subconjunto de R , es el conjunto de dependencias $X \rightarrow Y$ en $F+$ tal que atributos $X \rightarrow Y$ están todos contenidos en R_i . Así pues, la proyección de F sobre cada esquema de la relación R_i de la descomposición D es el conjunto de dependencias funcionales en $F+$, cierre de F , tales que todos sus atributos de la parte izquierda y la parte derecha estén en R_i . Decimos que una descomposición D de R es conservadores de las dependencias si:
$$((\pi_{R_1}(F) \cup \dots \cup (\pi_{R_m}(F)))^+ = F+$$

Ejemplo (1)

Asuma la siguiente relación IMPARTE con tres atributos (ESTUDIANTE,CURSO,PROFESOR) con las siguientes dependencias funcionales:

ESTUDIANTE	CURSO	PROFESOR
A	DB	X
B	DB	Y
B	SO	Z
B	TE	W
C	DB	X
C	SO	Z
D	DB	V
E	DB	Y

DF={ESTUDIANTE,CURSO} → PROFESOR; PROFESOR → CURSO;
{PROFESOR,ESTUDIANTE} → CURSO

Algoritmo 1: Encontrar una clave K para el esquema de relación R basado en dependencias funcionales F

1. Asignar $K := R$;
2. Para cada atributo A en K {
 - ▶ calcula $(K - A)^+$ respecto de F ;
 - ▶ si $(K - A)^+$ contiene todos los atributos de R , hacer $K := K - \{A\}$;}

Continuación Ejemplo 1:

Para esta relación, las claves candidatas son:

$DF = \{\{ESTUDIANTE, CURSO\}, \{ESTUDIANTE, PROFESOR\}\}$

Entonces, esta relación esta en 3FN pero no en FNBC

Algoritmo 2: Cobertura Mínima

Una cobertura mínima de un conjunto DF F es un conjunto mínimo de dependencias funcionales en F que es equivalente a F :

1. Asignar $G := F$
2. Reemplazar cada DF $X \rightarrow \{A_1, A_2, \dots, A_n\}$ en G por las n DF $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_n$.
3. Para cada DF $X \rightarrow A$ en G
 - ▶ Para cada atributo B que sea un elemento de X
 - ▶ Si $((G - \{X \rightarrow A\}) \rightarrow \{(X - \{B\}) \rightarrow A\})$ es equivalente a G Entonces reemplazar $X \rightarrow A$ con $(X - \{B\}) \rightarrow A$ en G .
4. Para cada DF restante $X \rightarrow A$ en G
 - ▶ Si $(G - \{X \rightarrow A\})$ es equivalente a G , Entonces eliminar $X \rightarrow A$ de G

Algoritmo 3: Conservación de Dependencias

1. Encontrar la cobertura mínima G para F
2. Para cada parte izquierda X de una dependencia funcional que aparezca en G , crear un esquema de relación D con los atributos $\{X \cup \{A_1\} \cup \{A_2\} \cdots \cup \{A_k\}\}$, donde $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_k$ sean las únicas dependencias en G con X como parte izquierda (X es la clave de esta relación)
3. Colocar cualquier atributo restante (no colocado) en un esquema de relación para asegurar la propiedad de conservación de atributos.

Descomposición y Reunión sin Pérdida (no aditivas)

- ▶ Esta propiedad garantiza que no se generán tuplas espurias cuando se aplique una operación de JOIN NATURAL a las relaciones de descomposición.
- ▶ Una descomposición $D = \{R_1, R_2, \dots, R_m\}$ de R tiene la propiedad de reunión sin pérdida respecto al conjunto de dependencias funciones F sobre R si, por cada estado de la relación r de R que satisfaga F , se cumple lo siguiente, donde $*$ es el JOIN NATURAL de todas las relaciones de D :
$$*((\pi_{R_1}(r)) \cup \dots \cup (\pi_{R_m}(r))) = r$$

Algoritmo 4: Comprobar la reunión sin pérdida (no aditiva)

Sea D una instancia de bases de datos.

1. Crear una matriz inicial S con una fila i por cada relación R_i en D y una columna j por cada atributo A_j en R .
2. Asignar $S(i, j) := b_{ij}$ para todas las entradas de la matriz; (* cada b_{ij} es un símbolo distinto asociado a los índices (i, j) *)
3. Por cada fila i que represente el esquema de relación R_i { para cada columna j que represente al atributo A_j { si $(R_i$ incluye al atributo $A_j)$, entonces asigne $S(i, j) := a_j$; }; } (* cada a_j es un símbolo distinto asociado al índice (j) *)
4. Repetir el siguiente bucle hasta que una ejecución de bucle completa no modifique S
 - { para cada dependencia funcional $X \rightarrow Y$ en F
 - { para todas las filas en S que tiene los mismo símbolos en las columnas correspondientes a los atributos en X
 - { hacer que los símbolos de cada columna correspondan a un atributo en Y sean iguales en todas las filas de la siguiente manera: si cualquiera de las filas tiene un símbolo 'a' en la columna, asignar el mismo símbolo 'a' a las otras filas en esa columna; si no hay símbolo 'a' para el atributo en ninguna de las filas, escoger uno de los símbolos 'b' que aparecen en una de las filas del atributo y asignar ese símbolo 'b' a las otras filas en esa columna; }; }; }
5. Si una fila consta exclusivamente de símbolos 'a', la descomposición posee la propiedad de reunión sin pérdidas; en caso contrario, no la posee;

Introducción

Pautas Informales
DiseñoDependencias
FuncionalesFormas Normales
(1/2)Formas Normales
(2/2)Algoritmos de
DescomposiciónOtras Formas
Normales

Ejemplo 1

$R = \{NSS, NOMBREE, NUMEROP, NOMBREP, LOCALIZADORP, HORAS\}$
 $D = \{R1, R2, R3\}$
 $R_1 = EMP = \{NSS, NOMBREE\}$
 $R_2 = PROYECTO = \{NUMEROP, NOMBREP, LOCALIZADORP\}$
 $R_3 = TRABAJA_EN = \{NSS, NUMEROP, HORAS\}$
 $F = \{NSS \rightarrow \{NOMBREE\}; NUMEROP \rightarrow \{NOMBREP, LOCALIZADORP\};$
 $\{NSS, NUMEROP\} \rightarrow HORAS\}$

Matriz Inicial antes del ciclo de DFs:

	NSS	NOMBREE	NUMEROP	NOMBREP	LOCALIZADORP	HORAS
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	b_{32}	a_3	b_{34}	b_{35}	a_6

Ejemplo 1

Matriz Inicial antes del ciclo de DFs:

	NSS	NOMBREE	NUMEROP	NOMBREP	LOCALIZADORP	HORAS
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	b_{32}	a_3	b_{34}	b_{35}	a_6

Matriz Final: (después de las 2 primeras DF)

	NSS	NOMBREE	NUMEROP	NOMBREP	LOCALIZADORP	HORAS
R_1	a_1	a_2	b_{13}	b_{14}	b_{15}	b_{16}
R_2	b_{21}	b_{22}	a_3	a_4	a_5	b_{26}
R_3	a_1	a_2	a_3	a_4	a_5	a_6

Algoritmo 5: Descomposición relacional con FNBC y Propiedad de Reunión sin Pérdidas

Entrada: Una relación R y dependencias funcionales F

1. Asignar $D := \{R\}$;
2. Mientras haya un esquema de relación Q en D que no esté en FNBC hace {
 - ▶ escoger un esquema de relación Q en D que no esté en FNBC;
 - ▶ encontrar una dependencia $X \rightarrow Y$ en Q que viole FNBC
 - ▶ Reemplazar Q en D por dos esquema $(Q - Y)$ y $(X \cup Y)$}

Algoritmo 6: Algoritmo que conserva las dependencias funcionales y con propiedad de reunión sin pérdida

Entrada: Una relación R y dependencias funcionales F

1. Encontrar una cobertura mínima G para F ;
2. Para cada parte izquierda X de una dependencia funcional que aparezca en G , crear un esquema de relación D con los atributos $\{X\} \cup \{A_1\} \cup \{A_2\} \cup \dots \cup \{A_k\}$, donde $X \rightarrow A_1, X \rightarrow A_2, \dots, X \rightarrow A_k$ sean las únicas dependencias en G con X como parte izquierda (X es la clave de esta relación);
3. Si ninguno de los esquemas de relación en D contiene una clave de R , crear un esquema de relación adicional en D que contenga atributos que formen una clave de R .

Problemas con Valores Nulos

- ▶ Nulos en atributos que servirán para reunir relaciones individuales en la descomposición
- ▶ Tuplas colgantes, las que aparecen únicamente en una de las relaciones de descomposición de una relación, de modo que se pierden si se realiza una reunión o join

Comentarios Finales de Algoritmos

- ▶ Uno de los problemas con estos algoritmos es que el diseñador debe especificar TODAS las dependencias funcionales importantes.
- ▶ Los algoritmos no son deterministas: (1) Muchas coberturas mínimas pueden existir, (2) Orden en que se consideran las DF

Dependencias Multivaluadas

- ▶ Las dependencias multivaluadas son una generalización de las dependencias funcionales. En éstas un conjunto de valores del implicado, son determinados por un implicante. Esta situación aparece cuando existen grupos repetitivos.
- ▶ La dependencia multivaluada se denota $X \twoheadrightarrow Y$, y se lee X multidetermina a Y , y significa que X implica un conjunto de valores de Y con independencia de los demás atributos de la relación.
- ▶ Las dependencias multivaluadas dependen del contexto, es decir influye el resto de los atributos de la relación.

Dependencias Multivaluadas

Definición: Una dependencia multivaluada (DMV) $X \twoheadrightarrow Y$ especificada sobre el esquema de relación R , donde X e Y son subconjuntos de R , especifica la siguiente restricción sobre cualquier estado de relación r de R : Si existen dos tuplas t_1 y t_2 en r tales que $t_1[X] = t_2[X]$, entonces deberán existir también dos tuplas t_3 y t_4 en r con las siguientes propiedades, en las que emplearemos Z para denotar $(R - (X \cup Y))$:

- ▶ $t_3[X] = t_4[X] = t_1[X] = t_2[X]$
- ▶ $t_3[Y] = t_1[Y]$ y $t_4[Y] = t_2[Y]$
- ▶ $t_3[Z] = t_2[Z]$ y $t_4[Z] = t_1[Z]$

Debido a la simetría de la relación si $X \twoheadrightarrow Y$, entonces $X \twoheadrightarrow Z$

Ejemplo

<u>NOMBREE</u>	<u>NOMBREP</u>	<u>NOMBRED</u>
Rojas	X	Ana
Rojas	Y	Pedro
Rojas	X	Pedro
Rojas	Y	Ana

El empleado Rojas trabaja en 2 proyectos y tiene 2 dependientes.
 $NOMBREE \twoheadrightarrow NOMBREP$ y $NOMBREE \twoheadrightarrow NOMBRED$

Dependencias Multivaluadas

La definición formal especifica que, dado un cierto valor de X , el conjunto de valores de Y determinado por este valor de X está completamente determinado sólo por X , y no depende de los valores de los atributos restantes Z de R . Así pues, siempre que existan dos tuplas con distintos valores de Y pero el mismo valor de X , estos valores de Y deberán repetirse en cada valor distinto de Z que ocurra con este mismo valor de X . De manera informal, esto equivale a que Y sea un atributo multivaluado de las entidades representadas por las tuplas de R .

Dependencias multivaluadas (DMV): razonamiento

► Trivial:

$A_1 A_2 \dots A_n \twoheadrightarrow B_1 B_2 \dots B_m$, donde $A_1 A_2 \dots A_n$ se cumple en cualquier relación si $\{B_1, B_2, \dots, B_m\} \subseteq \{A_1, A_2, \dots, A_n\}$.

► Transitividad:

Si $A_1 A_2 \dots A_n \twoheadrightarrow B_1 B_2 \dots B_m$ y $B_1 B_2 \dots B_m \twoheadrightarrow C_1 C_2 \dots C_k$ se cumplen, entonces $A_1 A_2 \dots A_n \twoheadrightarrow C_1 C_2 \dots C_k$ también se cumple.

► Promoción:

Si $A_1 A_2 \dots A_n \rightarrow B_1 B_2 \dots B_m$ se cumple, entonces se cumple $A_1 A_2 \dots A_n \twoheadrightarrow B_1 B_2 \dots B_m$.

► Complementación:

Si $A_1 A_2 \dots A_n \twoheadrightarrow B_1 B_2 \dots B_m$ es un DMV para una relación R , entonces $A_1 A_2 \dots A_n \twoheadrightarrow C_1 C_2 \dots C_k$ también se cumple en R cuando C_i son todos atributos en R que no son ni A_i ni B_j .

► Trivial adicional:

Si todos los atributos de R son $\{A_1, A_2, \dots, A_n, B_1, B_2, \dots, B_m\}$, entonces $A_1 A_2 \dots A_n \twoheadrightarrow B_1 B_2 \dots B_m$ se cumple en R .

4FN

- ▶ Una relación está en cuarta forma normal (4FN) si está en 3FN y todas las dependencias multivaluadas en ella son de hecho dependencias funcionales.
- ▶ La cuarta forma normal es transgredida cuando una relación contiene dos o más atributos multivaluados independientes.
- ▶ Un esquema de relación R está en 4FN respecto a un conjunto de dependencias funcionales F (que incluye dependencias funcionales y dependencias multivaluadas) si, para cada dependencia multivaluada no trivial $X \twoheadrightarrow Y$ en $F+$, X es una superclave

Ejemplo

<u>NOMBREE</u>	<u>NOMBREP</u>	<u>NOMBRED</u>
Rojas	X	Ana
Rojas	Y	Pedro
Rojas	X	Pedro
Rojas	Y	Ana

El empleado Rojas trabaja en 2 proyectos y tiene 2 dependientes.

$NOMBREE \twoheadrightarrow NOMBREP$ y $NOMBREE \twoheadrightarrow NOMBRED$

No está en 4FN porque NOMBREE no es una superclave

Ejemplo: Descomposición

<u>NOMBREE</u>	<u>NOMBREP</u>	<u>NOMBRED</u>
----------------	----------------	----------------

Se descompone en:

<u>NOMBREE</u>	<u>NOMBREP</u>
----------------	----------------

<u>NOMBREE</u>	<u>NOMBRED</u>
----------------	----------------

Ejemplo: Importancia

<u>NOMBREE</u>	<u>NOMBREP</u>	<u>NOMBRED</u>
Perez	X	Pedro
Perez	Y	Pedro
Perez	W	Pedro
Perez	Z	Pedro
Perez	X	Jose
Perez	Y	Jose
Perez	W	Jose
Perez	Z	Jose
Perez	X	Ana
Perez	Y	Ana
Perez	W	Ana
Perez	Z	Ana

<u>NOMBREE</u>	<u>NOMBREP</u>
Perez	X
Perez	Y
Perez	W
Perez	Z

<u>NOMBREE</u>	<u>NOMBRED</u>
Perez	Pedro
Perez	Jose
Perez	Ana

4FN: descomposición

- ▶ Encontrar una violación de 4FN en R , sea $A_1A_2 \dots A_n \twoheadrightarrow B_1B_2 \dots B_m$, donde $A_1A_2 \dots A_n$ no es una superclave.
- ▶ Si hay tal violación, descomponga el esquema R que viola la DMV en dos esquemas: (i) R_1 , con atributos A_i y B_j y (ii) R_2 con todos los atributos en R que no estén en A o B .
- ▶ Si $R_1 \neq R$ and $R_2 \neq R$, analice las dependencias funcionales en R_1 y R_2 y descomponga si es necesario.

Nota: Este algoritmo no siempre permite descomponer y generar esquemas que están en 4FN. Esto se da cuando la descomposición produce la misma relación original.

Dependencia de Unión y 5FN

- ▶ Una relación está en quinta forma normal (5FN) si y sólo si está en 4FN y el contenido de su información no puede ser descompuesto con varias relaciones menores.
- ▶ Si una relación sólo puede ser descompuesta en relaciones menores, todas ellas con la misma clave, entonces ya está en quinta forma normal.
- ▶ Esta forma normal tiene que ver con los casos en que la información total puede ser reconstruida desde pequeñas unidades de información, las que se pueden mantener con menos redundancia. La 2FN, 3FN y 4FN también sirven a este propósito, pero la 5FN generaliza los casos que no han sido cubiertos por las formas normales anteriores.
- ▶ Una dependencia de reunión, denotada por $DR(R_1, R_2, \dots, R_n)$, especificada sobre el esquema de relación R , especifica una restricción sobre los estados r de R . La restricción establece que todo estado permitido r de R debe tener una descomposición de reunión sin pérdidas para dar $R_1, R_2, \dots, yR_n$
- ▶ Un esquema R está en quinta forma normal (o forma normal de proyección-reunión) (FNDR) respecto a un conjunto F de dependencias funcionales, multivaluadas y de reunión si, para cada dependencia de reunión no trivial $DR(R_1, R_2, \dots, R_n)$, en $F+$ (esto es implicada por F), toda R_i es una superclave de R .

Ejemplo 5FN

- Considérese el caso de vendedores que venden ciertos productos de distintas compañía. Los vendedores representan compañía, las compañía fabrican productos y los vendedores venden productos.
- Supongamos que existe la siguiente restricción: si un vendedor vende un determinado producto, y el vendedor representa a una compañía que produce dicho producto, entonces el vendedor vende el producto de la compañía.

<u>VENDEDOR</u>	<u>COMPAÑIA</u>	<u>PRODUCTO</u>
V1	C1	P1
V1	C2	P1
V2	C1	P1
V2	C1	P2
V2	C2	P1
V2	C2	P2

Ejemplo 5FN:Descompisición

Ahora, en este caso, se puede reconstruir toda la información a partir de las siguientes relaciones:

<u>VENDEDOR</u>	<u>PRODUCTO</u>
V1	P1
V2	P1
V2	P2

<u>VENDEDOR</u>	<u>COMPañIA</u>
V1	C1
V1	C2
V2	C1
V2	C2

<u>COMPañIA</u>	<u>PRODUCTO</u>
C1	P1
C1	P2
C2	P1
C2	P2

5FN:Comentarios

- ▶ La 5FN no difiere de la 4FN a menos de que exista una restricción de simetría, tal como la restricción del caso anterior. Si no existe una restricción de este tipo, entonces una relación en cuarta forma normal también está en quinta forma normal.
- ▶ Una ventaja de la quinta forma normal es la de que ciertas redundancias pueden ser eliminadas.
- ▶ Debe observarse que aunque la forma normalizada involucra más relaciones, hay un número menor de ocurrencias (esto no se aprecia con claridad cuando hablamos de pocos hechos).
- ▶ La ventaja se obtiene con más hechos registrados, puesto que el tamaño de las relaciones (cardinalidad) normalizadas crece en forma aditiva, mientras que cuando no lo están, su tamaño crece en forma multiplicativa.
- ▶ La 4FN y la 5FN tienen que ver con combinaciones de hechos multivaluados.
- ▶ Una de las diferencias entre ambas radica en que los hechos considerados bajo la 5FN no son independientes. Otra de las diferencias es que la 4FN, aunque puede considerar más de dos hechos multivaluados, sólo los considera en parejas.
- ▶ Esto se explica por el proceso de normalización implicado por la 4FN, ya que se realiza una descomposición en 2 relaciones, cada una con menos atributos que la relación original.
- ▶ Cualquiera de estas relaciones que infrinja la 4FN será nuevamente descompuesta en 2 relaciones, y así sucesivamente, hasta que todas las relaciones resultantes estén en 4FN.

Ejercicios

- Considere la siguiente relación:
RELACION(NUM_PEDIDO,FECHA_PED,NUMCLIENTE,
NUM_ARTICULO,NOMBRE_ARTICULO,PRECIO_UNI,CANTIDAD)

Para esta relación, estos son las DFs:

- NUM_PEDIDO $\rightarrow$ FECHA_PED,NUMCLIENTE
- NUM_PEDIDO,NUM_ARTICULO $\rightarrow$ CANTIDAD
- NUM_ARTICULO $\rightarrow$
NOMBRE_ARTICULO,PRECIO_UNI

Determine las claves candidatas. Normalice (hasta donde pueda) la relación manteniendo las dependencias funcionales y sin generación de tuplas espurias.

- Asuma un esquema de relación $R(A,B,C)$ y dos descomposiciones: $R_1(A, B)$ y $R_2(B, C)$. Asuma las siguientes instancias $R_1 = \{(5, 1), (6, 1)\}$ y $R_2 = \{(1, 8), (1, 9)\}$. Indique las instancias posibles que contienen el mínimo y máximo número de tuplas en R . Según eso, ¿ Puede B ser una clave de R?

Ejercicios

- Considere la siguiente instancia de corredores:

Rut	Nombre	Edad	Rank
11111111	Ulloa	24	1
22222222	Salas	27	1
33333333	Roa	30	2
44444444	Fuentes	21	null

Considere las siguientes consultas:

```
(1) SELECT C.nombre  
FROM Corredores C  
WHERE NOT EXISTS ( SELECT *  
FROM Corredores C2  
WHERE C2.edad < 22 AND C.rank <= S2.rank )
```

```
(2) SELECT *  
FROM Corredores C.nombre  
WHERE C.rank > ANY ( SELECT C2.rank  
FROM Corredores C2  
WHERE C2.edad < 22 )
```

Indique la respuesta de ambas consultas. ¿Ambas entregan el mismo resultado siempre?

[Introducción](#)[Pautas Informales
Diseño](#)[Dependencias
Funcionales](#)[Formas Normales
\(1/2\)](#)[Formas Normales
\(2/2\)](#)[Algoritmos de
Descomposición](#)[Otras Formas
Normales](#)

Ejercicios

- ▶ Para la siguiente relación $R(A, C, B, D, E)$ y dependencias funcionales $A \rightarrow B$, $C \rightarrow D$. Indique la forma normal en que se encuentra y normalice si es necesario.
- ▶ Considere una relación $R(A, B, C)$. Se descompone en relaciones $R1(A, B)$ y $R2(B, C)$.
 1. Exprese una definición de una descomposición sin pérdida (no genera tuplas espúrias) respecto a este ejemplo. Responda indicando en SQL o en álgebra relacional lo que significa esta descomposición.
 2. Suponga que $B \rightarrow C$. ¿Es la descomposición de R en $R1$ y $R2$ sin pérdida?
- ▶ Dada una relación $R(A, B, C, D)$ y la dependencia multivaluada $A \twoheadrightarrow C$. Entregue una instancia con el menos número de tuplas (pero mayor que cero) que cumpla esta dependencia multivaluada.