

Normalización	
Contenido	
6	Normalización 99
•	Conceptos 103
7	Dependencias Funcionales 105
8	Clausulas 113
9	Normalización de Esquemas 120
10	Formas Normales 131
•	Primera Forma Normal (1FN) 132
•	Segunda Forma Normal (2FN) 137
•	Tercera Forma Normal (3FN) 150
•	Forma Normal Boyce-Codd (FNBC) 168
11	Descomposición 185
•	Descomposición por Join sin Pérdida 187
12	Ejercicios Normalización de Esquemas 199
13	Formas Normales Mutivaluadas 204
•	Cuarta Forma Normal (4FN) 205
•	Dependencias Multivaluadas
•	Cuarta Forma Normal (4FN)
•	Quinta Forma Normal (5FN) 214
•	Dependencias de JOIN
•	Quinta Forma Normal (5FN)
<p>Juan José Ramírez Lama (ULagos) Modelo de Diseño de Datos 2016-2 - @juaramir 99 / 205</p>	

Normalización	
Normalización	
•	Normalización es un proceso que clasifica relaciones, objetos, formas de relación y demás elementos en grupos, en base a las características que cada uno posee.
•	En el proceso de normalización se somete un esquema relación a una serie de pruebas para “certificar” si pertenece o no a una cierta forma normal.
•	Puede considerarse como un proceso durante el cual los esquemas relacionales más pequeños que poseen propiedades deseables.
•	Las formas normales, sin considerar otros factores, no garantizan un buen diseño de base de datos.
<p>Juan José Ramírez Lama (ULagos) Modelo de Diseño de Datos 2016-2 - @juaramir 100 / 205</p>	

Normalización

Después de transformar el MER al MR normalmente tendremos una buena base de datos, pero a veces debido a fallos en el diseño, es preciso normalizar dicho MR aplicando una serie de reglas. El proceso de normalización se hace necesario para:

- Evitar la redundancia de los datos y las inconsistencias.
- Evitar la incapacidad de almacenar ciertos datos.
- Evitar la ambigüedad y pérdida de información.
- Evitar problemas de actualización (anomalías de inserción, borrado y modificación) de los datos en la tabla.
- Proteger la integridad de los datos.

Normalización

- Cuando las reglas de clasificación sean más y más restrictivas, diremos que la relación está en una forma normal más elevada.
- La relación que está en la forma normal más elevada posible es que mejor se adapta a nuestras necesidades debido a que optimiza las condiciones que son de importancia para nosotros:
 - La cantidad de espacio requerido para almacenar los datos es la menor posible;
 - La facilidad para actualizar la relación es la mayor posible;
 - La explicación de la base de datos es la más sencilla posible
- Se dice que una relación está en una determinada forma normal si satisface un cierto conjunto de restricciones.
- El proceso de normalización es reversible y no se pierde información.

Normalización		Conceptos
Contenido		
6	Normalización	99
	• Conceptos	103
7	Dependencias Funcionales	105
8	Clausulas	113
9	Normalización de Esquemas	120
10	Formas Normales	131
	• Primera Forma Normal (1FN)	132
	• Segunda Forma Normal (2FN)	137
	• Tercera Forma Normal (3FN)	150
	• Forma Normal Boyce-Codd (FNBC)	168
11	Descomposición	185
	• Descomposición por Join sin Pérdida	187
12	Ejercicios Normalización de Esquemas	199
13	Formas Normales Mutivaluadas	204
	• Cuarta Forma Normal (4FN)	205
	• Dependencias Multivaluadas	
	• Cuarta Forma Normal (4FN)	
	• Quinta Forma Normal (5FN)	214
	• Dependencias de JOIN	
	• Quinta Forma Normal (5FN)	
<div> <div>Juan José Ramírez Lama (ULagos)</div> <div>Modelo de Diseño de Datos</div> <div>2016-2 - @juaramir</div> <div>103 / 205</div> </div>		

Dependencias Funcionales	
Contenido	
6	Normalización
	• Conceptos
7	Dependencias Funcionales
8	Clausulas
9	Normalización de Esquemas
10	Formas Normales
	• Primera Forma Normal (1FN)
	• Segunda Forma Normal (2FN)
	• Tercera Forma Normal (3FN)
	• Forma Normal Boyce-Codd (FNBC)
11	Descomposición
	• Descomposición por Join sin Pérdida
12	Ejercicios Normalización de Esquemas
13	Formas Normales Mutivaluadas
	• Cuarta Forma Normal (4FN)
	• Dependencias Multivaluadas
	• Cuarta Forma Normal (4FN)
	• Quinta Forma Normal (5FN)
	• Dependencias de JOIN
	• Quinta Forma Normal (5FN)
<div> <div>Juan José Ramírez Lama (ULagos)</div> <div>Modelo de Diseño de Datos</div> <div>2016-2 - @juaramir</div> <div>105 / 205</div> </div>	

Normalización

Conceptos

Clave y SuperClave

- **SuperClave:** Una SuperClave de $R = \{A_1, \dots, A_n\}$ es un conjunto de atributos $S \subseteq R$ tal que no existen 2 tuplas t_1 y t_2 en ningún R talque $t_1[S] = t_2[S]$
- **Clave:** Una clave K es una SuperClave que cumple que si se le quita alguno de sus atributos, deja de ser SuperClave.
- **Clave Candidata, Clave Primaria:** Si una relación tiene más de una clave, cada una es una clave candidata. Una de ellas es arbitrariamente designada como clave primaria. El resto son secundarias.

Juan José Ramírez Lama (ULagos)

Modelo de Diseño de Datos

2016-2 - @juaramir

104 / 205

Dependencias Funcionales

Dependencias Funcionales

Trabajador		
Rut	Nombre	Departamento
11.111.111-1	Jane Doe	Mantenimiento
22.222.222-2	John Doe	Dirección Gestion

- Se dice que un conjunto de atributos Y depende funcionalmente de otro conjunto de atributos X si para cada valor de X hay un unico valor posible para Y . Simbolicamente se denota por $X \rightarrow Y$.
- Por ejemplo el **Nombre** de una persona depende funcionalmente del **Rut**; es decir para un Rut concreto solo hay un nombre posible.
- En la tabla del ejemplo anterior, el departamento no tiene dependencia funcional, ya que para un mismo Rut puede haber mas de un departamento posible. Pero el nombre si que depende del Rut.
- Al conjunto X del que depende funcionalmente el conjunto Y se le llama **determinante**. Al conjunto Y se le llama **implicado**.

Juan José Ramírez Lama (ULagos)

Modelo de Diseño de Datos

2016-2 - @juaramir

106 / 205

Ejemplo: Dependencias Funcionales

La siguiente relación r satisface la DF $AB \rightarrow C$:

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_1	d_2
a_1	b_2	c_2	d_1

- Si se inserta la tupla $\langle a_1, b_1, c_3, d_1 \rangle$ a r , entonces r no satisface la DF.
- Las dos primeras tuplas muestran que las DF no son lo mismo que las restricciones de claves: es evidente que AB no es clave de la relación.

Ejemplo: Dependencias Funcionales

La siguiente relación r satisface la DF $AB \rightarrow C$:

A	B	C	D
a_1	b_1	c_1	d_1
a_1	b_1	c_1	d_2
a_1	b_2	c_2	d_1
a_1	b_1	c_3	d_1

- Si se inserta la tupla $\langle a_1, b_1, c_3, d_1 \rangle$ a r , entonces r no satisface la DF.
- Las dos primeras tuplas muestran que las DF no son lo mismo que las restricciones de claves: es evidente que AB no es clave de la relación.

Definición

- **Atributo Primo**: Un atributo del esquema relacional R es **primo** si es miembro de alguna clave de R .
- **Dependencia Total**: $X \rightarrow Y$ es una dependencia funcional si al eliminación de cualquier atributo A de X hace que la dependencia funcional deje de ser válida (no tiene atributos redundantes a la izquierda).
- **Dependencia Parcial**: $X \rightarrow Y$ es una dependencia funcional **parcial** si es posible eliminar un atributo A de X , y la dependencia funcional sigue siendo válida.
- **Dependencia Transitiva**: Una dependencia funcional $X \rightarrow Y$ en un esquema relacional R es una dependencia funcional transitiva si existe un conjunto de atributos Z que no sea un subconjunto de cualquier clave de R , y se cumplen tanto $X \rightarrow Z$ como $Z \rightarrow Y$.

Instancias Legales y Claves Primarias

- Una instancia de una relación R es **legal** si satisface todas las restricciones de integridad sobre R , incluyendo las DFs
- Examinando las instancias de una relación se puede saber si una DF no se cumple
- Las restricciones de clave primaria son un caso especial de DFs.
 - Los atributos en la llave desempeñan el papel de X y el conjunto de todos los atributos de la relación interpretan el de Y
- La definición de DFs no exige que el conjunto X sea mínimo, sin embargo para que X sea clave éste debe ser mínimo

Ejemplo: Clave Primaria

- En el esquema Navegantes(**idn**, nombre, categoria, edad), la DF que define la clave primaria es:

$idn \rightarrow idn, nombre, categoria, edad$

- Por cada valor de *idn* existe un valor único para (*idn*), *nombre*, *categoria* y *edad*.
 - En $idn \rightarrow idn, nombre, categoria, edad$, el atributo *idn* es mínimo.
 - En $idn, nombre \rightarrow categoria, edad$, el conjunto *idn*, *nombre* no es minimal, ya que sabemos que $idn \rightarrow idn, nombre, categoria, edad$

Contenido

6	Normalización	99
	• Conceptos	103
7	Dependencias Funcionales	105
8	Clausulas	113
9	Normalización de Esquemas	120
10	Formas Normales	131
	• Primera Forma Normal (1FN)	132
	• Segunda Forma Normal (2FN)	137
	• Tercera Forma Normal (3FN)	150
	• Forma Normal Boyce-Codd (FNBC)	168
11	Descomposición	185
	• Descomposición por Join sin Pérdida	187
12	Ejercicios Normalización de Esquemas	199
13	Formas Normales Mutivaluadas	204
	• Cuarta Forma Normal (4FN)	205
	• Dependencias Multivaluadas	
	• Cuarta Forma Normal (4FN)	
	• Quinta Forma Normal (5FN)	214
	• Dependencias de JOIN	
	• Quinta Forma Normal (5FN)	

Razonamiento sobre las DFs

- Dado un conjunto de DFs sobre el esquema de una relación *R*, normalmente se cumplen varias DFs adicionales sobre *R* siempre que se cumplan todas las DFs dadas.
- Consideremos el esquema:
Trabajadores (*idt*, nombre, plaza, depto, desde)
- Dado que *idt* es clave, se cumple: $idt \rightarrow depto$, consideremos además que se cumple la DF: $depto \rightarrow plaza$.
 - En cualquier instancia de **Trabajadores** si dos tuplas tienen el mismo valor de *idt*, deben tener el mismo valor de *depto*.
 - Por la segunda DF también deben tener el mismo valor de *plaza*.
 - Por lo tanto, también se cumple la DF $idt \rightarrow plaza$.
- Se dice que una DF *d* está implícita en un conjunto dado *D* de DFs si *d* se cumple en **todas** las instancias de la relación que satisfacen las dependencias en *D*.

Clausulas de un conjunto de DFs

- El conjunto de todas las DFs implícitas en un conjunto dado de DFs *D* se denomina **clausura de *D*** (cierre) y se denota por D^+
- Para obtener D^+ se pueden aplicar reiteradamente los **axiomas de Armstrong**.
- Sean *X*, *Y* y *Z* conjuntos de atributos de un esquema *R*:
 - Reflexividad**: $\forall Y \subseteq X$, entonces $X \rightarrow Y$
 - Aumento**: Si $X \rightarrow Y$, entonces $XZ \rightarrow YZ$ para cualquier *Z*
 - Transitividad**: Si $X \rightarrow Y$ y $Y \rightarrow Z$, entonces $X \rightarrow Z$
- Reglas Adicionales:
 - Unión**: Si $X \rightarrow Y$ y $X \rightarrow Z$, entonces $X \rightarrow YZ$
 - Descomposición**: Si $X \rightarrow YZ$, entonces $X \rightarrow Y$ y $X \rightarrow Z$
 - Pseudotransitividad**: $X \rightarrow Y$ y $WY \rightarrow Z$, entonces $WX \rightarrow Z$

Ejemplo: Clausura de un conjunto de DFs

Consideremos el esquema de relación $R(A, B, C)$ con DFs:

$$A \rightarrow B$$

$$B \rightarrow C$$

aplicando los axiomas de Armstrong obtenemos D^+ :

- $A \rightarrow A$, $B \rightarrow B$, $C \rightarrow C$ por reflexividad
- $A \rightarrow C$, por transitividad usando $B \rightarrow C$
- $AC \rightarrow BC$, por aumento de $A \rightarrow B$ con C
- $AB \rightarrow AC$, por aumento de $B \rightarrow C$ con A
- $AB \rightarrow CB$, por aumento de $A \rightarrow C$ con B

Por lo tanto AB es clave candidata de R .

Ejemplo: Clausura de un conjunto de DFs

D^+ contiene las siguientes DFs entre otras:

- De $YR \rightarrow C$ y $C \rightarrow CPYDRNV$ y por transitividad obtenemos: $YR \rightarrow CPYDRNV$
- De $PD \rightarrow R$ y aumento con Y se infiere: $PDY \rightarrow YR$
- De $PDY \rightarrow YR$ y $YR \rightarrow CPYDRNV$ y por transitividad obtenemos: $PDY \rightarrow CPYDRNV$
- De $C \rightarrow CPYDRNV$ y descomposición obtenemos: $C \rightarrow C$ (trivial), $C \rightarrow P$, $C \rightarrow Y$, $C \rightarrow D$, etc.
- Hay otras DFs triviales que se obtienen de la regla de la reflexividad.

Ejemplo: Clausura de un conjunto de DFs

- Supongamos el esquema:

$\text{Contratos}(\text{idc}, \text{idProd}, \text{idProy}, \text{idDepto}, \text{idRepuesto}, \text{cant}, \text{valor})$

- Denotaremos el esquema con $\text{Contratos}(C, P, Y, D, R, N, V)$, donde una tupla en Contratos con idc **C** es un acuerdo por el que el proveedor **P** suministrará **N** ejemplares del repuesto **R** al proyecto **Y** asociado al departamento **D** con valor **V**.
- Se sabe que se cumplen las siguientes restricciones de integridad:
 - El atributo idc es clave: $C \rightarrow CPYDRNV$
 - Cada proyecto compra un repuesto dado mediante un solo contrato: $YR \rightarrow C$
 - Cada departamento compra como máximo un repuesto de cada proveedor: $DP \rightarrow R$

Ejemplo: Clausura de un conjunto de DFs

Considere la relación $R(A, B, C, D, E, F, G, H, I, J)$ con las siguientes DF:

- | | |
|-----------------------|----------------------|
| • $ABC \rightarrow E$ | • $C \rightarrow J$ |
| • $AB \rightarrow G$ | • $CJ \rightarrow I$ |
| • $B \rightarrow F$ | • $G \rightarrow H$ |

Usando los axiomas de Armstrong genere una clave candidata para R .

Ejemplo: Clausura de un conjunto de DFs

Considere la relación $R(A,B,C,D,E,F,G,H,I,J)$ con las siguientes DF:

$$ABC \rightarrow E$$

$$B \rightarrow F$$

$$CJ \rightarrow I$$

$$AB \rightarrow G$$

$$C \rightarrow J$$

$$G \rightarrow H$$

$$AB \rightarrow G$$

$$AB \rightarrow H \text{ (por transitividad } G \rightarrow H)$$

$$ABD \rightarrow HD \text{ (por aumento con } D)$$

$$ABD \rightarrow E \text{ (dada)}$$

$$ABD \rightarrow B \text{ (por reflexividad)}$$

$$ABD \rightarrow F \text{ (por transitividad } B \rightarrow F)$$

$$ABDC \rightarrow FC \text{ (por aumento con } C)$$

$$ABDC \rightarrow C \text{ (por reflexividad)}$$

$$ABDC \rightarrow J \text{ (por transitividad } C \rightarrow J)$$

$$ABDC \rightarrow CJ \text{ (por unión de } ABDC \rightarrow C \text{ y } ABDC \rightarrow J)$$

$$ABDC \rightarrow I \text{ (por transitividad } CJ \rightarrow I)$$

$\therefore ABCD$ es clave candidata para R .

Contenido

6	Normalización	99
	• Conceptos	103
7	Dependencias Funcionales	105
8	Clausulas	113
9	Normalización de Esquemas	120
10	Formas Normales	131
	• Primera Forma Normal (1FN)	132
	• Segunda Forma Normal (2FN)	137
	• Tercera Forma Normal (3FN)	150
	• Forma Normal Boyce-Codd (FNBC)	168
11	Descomposición	185
	• Descomposición por Join sin Pérdida	187
12	Ejercicios Normalización de Esquemas	199
13	Formas Normales Multivaluadas	204
	• Cuarta Forma Normal (4FN)	205
	• Dependencias Multivaluadas	
	• Cuarta Forma Normal (4FN)	
	• Quinta Forma Normal (5FN)	214
	• Dependencias de JOIN	
	• Quinta Forma Normal (5FN)	

¿Por qué normalizar Esquemas?

El universo de datos no normalizado se refiere al conjunto de datos que están reunidos bajo un criterio en común, estos datos son una gran cantidad de información desorganizada y, en algunos casos, compleja para su análisis u otros usos, ya que tiene un albedrío de información, y en ello encontraremos muchas inconsistencias o “defectos”.

- 1 **Almacenamiento redundante**: alguna información se almacena de manera repetida.
- 2 **Anomalías de actualización**: se generan inconsistencias si algunas copias no se mantienen actualizadas.
- 3 **Anomalías de inserción**: puede que no sea posible almacenar determinada información a menos que otra información, sin relación alguna con ella, también se almacene.
- 4 **Anomalía de eliminación**: podría no ser posible eliminar cierta información sin perder otra.

¿Por Qué Normalizar Esquemas?

En relación a tablas no normalizadas (cuando almacenamos información no normalizada) puede suceder:

- Repetición de nombres de cada tabla.
- Presencia de dos filas iguales.
- Los datos de una misma columna de un mismo tipo.
- De inserción: imposibilidad de adicionar datos en la BD por la ausencia de otros.
- De borrado: pérdida no intencionada de datos debido a la eliminación de otros.

Ejemplo Anomalías

Supongamos la siguiente relación Horas_Emp:

Horas_Emp					
id	nombre	depto	categoria	horas_extras	horas_trab
11	juan	48	8	10	40
12	pedro	22	8	10	30
13	enrique	35	5	7	30
14	maria	35	6	8	32
15	carolina	35	8	10	40

- Supongamos además que el atributo horas_extras depende del atributo categoría, p.e. por cada valor de categoría existe un valor único para horas_extras.
- **Almacenamiento redundante**: para la categoría 8 corresponden 10 horas extras, lo cual está almacenado tres veces.

Ejemplo Anomalías

Supongamos la siguiente relación Horas_Emp:

id	nombre	depto	categoria	horas_extras	horas_trab
11	juan	48	8	10	40
12	pedro	22	8	10	30
13	enrique	35	5	7	30
14	maria	35	6	8	32
15	carolina	35	8	10	40

- **Anomalías de actualización**: si se modifica el valor de horas_extras en la primera tupla y no se modifica en el resto de las tuplas con categoría 8 se producen inconsistencias.
- **Anomalías de inserción**: no podemos insertar un nuevo empleado a menos que sepamos el número de horas extras para la categoría del empleado.
- **Anomalías de eliminación**: si borramos todas las tuplas para una categoría determinada, perdemos la información de la categoría y sus horas extras (ejemplo, si borramos el empleado con id=14).

Origen de Redundancia en Esquemas de BDs

- La redundancia surge cuando un esquema relacional fuerza una asociación entre atributos que no es natural.
 - Ejemplo: en la relación Horas Emp el atributo horas extras no tiene relación directa con el atributo depto.
- Se pueden utilizar las dependencias funcionales (DFs) para identificar estas situaciones y sugerir maneras de refinar (cambiar) los esquemas.
- La idea fundamental es que muchos problemas que surgen de la redundancia se pueden abordar sustituyendo una relación dada por un conjunto de relaciones "mas pequeñas".

Descomposición de Relaciones

- La descomposición del esquema de una relación R consiste en sustituir el esquema de la relación por dos (o más) esquemas que contienen un subconjunto de los atributos de R y, conjuntamente, incluyen todos los atributos de R
- Ejemplo: La relación Horas_Emp se puede descomponer en dos relaciones:
 - Horas_Emp(id,nombre,depto,categoria,horas_extras,horas_trab)
 - Horas_Emp(id, nombre, depto, categoria, horas_trab)
 - Categoria_Horas(categoria, horas_extras)

Horas_Emp				
id	nombre	depto	categoria	horas_trab
11	juan	48	8	40
12	pedro	22	8	30
13	enrique	35	5	30
14	maria	35	6	32
15	carolina	35	8	40

Categorias_Horas	
categoria	horas_extras
8	10
5	7
6	8

Problemas con la Descomposición

- La descomposición del esquema de una relación puede crear más problemas de los que soluciona
- Se deben formular repetidamente dos preguntas importantes:
 - 1 ¿Hace falta descomponer la relación?
 - 2 ¿Qué problemas provoca una descomposición dada?

Problemas con la Descomposición

- Para ayudar con la primer pregunta se han propuesto varias **formas normales (FNs)** para las relaciones.
- Si el esquema de una relación está en alguna FN, se sabe que no pueden surgir determinadas clases de problemas
- Con respecto a la segunda pregunta resultan de especial interés dos propiedades de las descomposiciones:
 - 1 **Reunión sin pérdida:** la descomposición permite recuperar cualquier instancia de la relación descompuesta a partir de las instancias de las relaciones de menor tamaño
 - 2 **Conservación de las dependencias:** la descomposición permite hacer que se cumpla cualquier restricción de la relación original con sólo hacer que se cumplan algunas restricciones en cada una de las relaciones de menor tamaño

Etapas de la Normalización

