

Teórico 7

Diseño de Bases de Datos (Normalización)



Objetivos de la Normalización

- Decidir si la relación R está en un “buena” forma.
- En el caso que una relación R no esté en una “buena” forma, descomponer esta en un conjunto de relaciones $\{R1, R2, \dots, Rn\}$ tal que
 - Cada relación está en buena forma.
 - La descomposición es una descomposición sin pérdida (lossless-join decomposition).
- Esta teoría está basada en:
 - Dependencias funcionales.
 - Dependencias multivaluadas.

Retomamos el Ejemplo (clase anterior)

Supongamos que estamos trabajando con información de Personas y vehículos, y dicha información está representada con una sola tabla.

R(DNI, NYApellido, Dir, Npat, Modelo, Marca)

Supongamos la siguiente instancia de la relación R

DNI	NYApellido				
23	Juan Pere				
23	Juan Pere				
24	Carlos Pui				

Consideremos la Siguiete Descomposición de R

Personas= (DNI, NYApellido, Dir, Modelo)

Y

Vehiculos= (Npat, Modelo, Marca)

Las Instancias Serían

$$\text{personas} = \Pi_{\text{Personas}}(r)$$

DNI	NYApellido		
23	Juan Peres		
23	Juan Peres		
24	Carlos Puig		

$$\text{vehículos} = \Pi_{\text{Vehículos}}(r)$$

Npat	Modelo	
NDT 454	2000	
KJI 566	2005	
GTR 654	2000	

Para que una división sea válida y no tenga pérdida de información, el join Natural entre las tablas debe ser igual a la tabla original en cuanto a instancias.

$r = \text{personas} \bowtie \text{vehículos}$

En nuestro caso: $\text{personas} \bowtie \text{vehículos} \neq r$

DNI	NYApellic				
23	Juan Pere				
23	Juan Pere				
23	Juan Pere				

Consideremos la Siguiete Descomposición de R

Personas= (DNI, NYApellido, Dir),

PersonasVehiculos(DNI,Npat)

Y

Vehiculos= (Npat, Modelo, Marca)

Las Instancias Serían

personas = $\Pi_{\text{Personas}}(\mathbf{r})$

DNI	NYApellido	
23	Juan Peres	
24	Carlos Puig	

vehículos = $\Pi_{\text{Vehículos}}(\mathbf{r})$

Npat	Modelo	
NDT 454	2000	
KJI 566	2005	
GTR 654	2000	

personasVehiculos = $\Pi_{\text{PersonasVehiculos}}(\mathbf{r})$

DNI	Npat
23	NDT 454
23	KJI 566
24	GTR 654



Haciendo el Join

$r = \text{personas} \bowtie \text{personasVehiculos} \bowtie \text{vehiculos}$

Resultado:

DNI	NYApeIllic				
23	Juan Pere				
23	Juan Pere				
24	Carlos Pui				

De esta forma se ve que la descomposición realizada no tiene pérdida de información.

Definición

- Sea R un esquema. Un conjunto de esquemas de relación $\{R_1, R_2, \dots, R_n\}$ es una descomposición de R si

$$R = R_1 \cup R_2 \cup \dots \cup R_n$$

- Sea r una instancia de relación del esquema R y sea $r_i = \Pi_{R_i}(r)$ para $i = 1, 2, \dots, n$. Es decir $\{r_1, r_2, \dots, r_n\}$ es la base de datos que resulta de descomponer R en $\{R_1, R_2, \dots, R_n\}$, siempre se cumple que:

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

- Una relación se dice legal si satisface todas las ligaduras impuestas a nuestra base de datos.

Definición

- Sea C un conjunto de ligaduras en la Base de datos. Una descomposición $\{R_1, R_2, \dots, R_n\}$ de un esquema de relación R es una descomposición de reunión sin pérdida para R si para todas las instancias r que son legales bajo C , se cumple:
- $r = \Pi_{R_1}(r) \bowtie \Pi_{R_2}(r) \bowtie \dots \bowtie \Pi_{R_n}(r)$

Normalización Utilizando Dependencias Funcionales

- Un conjunto de dependencias funcionales válidas en una base de datos pueden utilizarse para lograr un buen diseño.

Propiedades Deseables de una Descomposición

- Descomposición de reunión sin pérdidas.
- Conservación de dependencias.

Descomposición sin Pérdida

- Sea R un esquema de relación, y F un conjunto de dependencias funcionales en R . Sea R_1 y R_2 una descomposición de R . Esta descomposición es una descomposición de reunión sin pérdida de R , si al menos una de las siguientes dependencias está en F^+ :

$$R_1 \cap R_2 \rightarrow R_1$$

$$R_1 \cap R_2 \rightarrow R_2$$

Conservación de Dependencias

- Cuando se realiza la descomposición de un esquema de relación R , es deseable que esta descomposición no pierda dependencias funcionales (restricciones a los datos).

Definición

- Sea un esquema R , un conjunto de dependencias funcionales F en R , y una descomposición $\{R_1, R_2, \dots, R_n\}$, considerando a F_i un subconjunto de dependencias de F^+ que sólo incluyen atributos de R_i .
 - Dicha descomposición preserva las dependencias si

$$(F_1 \cup F_2 \cup \dots \cup F_n)^+ = F^+$$

Definiciones

Atributo Primo

– Un atributo del esquema relación R es **primo** si es miembro de cualquier clave(candidata) de R.

Dependencia Total

– $X \rightarrow Y$ es una dep. funcional **total** si la eliminación de cualquier atributo A de X hace que la dep. funcional deje de ser válida. (no tiene atr. redundantes a la izq.)

Dependencia Parcial

– $X \rightarrow Y$ es una dep. funcional **parcial** si es posible eliminar un atributo A de X, y la dep. funcional sigue siendo válida.

Forma Normal de Byce-Codd(BCNF)

- Un esquema de relación R está en BCNF respecto a un conjunto de dependencias F , si para toda dependencia en F^+ de la forma $\alpha \rightarrow \beta$ cumple al menos una de las siguientes condiciones:
 - $\alpha \rightarrow \beta$ es trivial
 - α es superclave del esquema R .

Algoritmo de Descomposición a BCNF

```
resultado := {R };
hecho := false;
computar  $F^+$ ;
while (not hecho) do
  if (hay un esquema  $R_i$  en resultado que no está en BCNF)
  then
    begin
      dado  $\alpha \rightarrow \beta$  una dependencia funcional no trivial que vale en  $R_i$  tal que
       $\alpha \rightarrow R_i$  no está en  $F^+$ , y  $\alpha \cap \beta = \emptyset$ ;

      resultado := (resultado -  $R_i$ )  $\cup$  ( $R_i - \beta$ )  $\cup$  ( $\alpha, \beta$ );
    end
  else
    hecho := true;
```

Algoritmo de Descomposición a BCNF (sigue)

- El algoritmo anterior genera una descomposición en BCNF, además de generar una descomposición sin pérdida.
- El algoritmo no garantiza la conservación de las dependencias.

Tercera Forma Normal (3NF)

Un esquema de relación R está en 3NF respecto a un conjunto de dependencias F , si para toda dependencia en F^+ de la forma $\alpha \rightarrow \beta$ cumple al menos una de las siguientes condiciones:

- $\alpha \rightarrow \beta$ es trivial.
- α es superclave del esquema R .
- Cada atributo A en $\beta - \alpha$ está contenido en alguna clave candidata de R (A es primo)

Algoritmo de Descomposición a 3NF

Dado F_c un conjunto canónico para F ;

$i := 0$;

for each dependencia funcional $\alpha \rightarrow \beta$ en F_c **do**

if ninguno de los esquemas R_j , $1 \leq j \leq i$ contiene $\alpha \beta$

then

begin

$i := i + 1$;

$R_i := \alpha \beta$

end

if ninguno de los esquemas R_j , $1 \leq j \leq i$ contiene una clave candidata de R

then

begin

$i := i + 1$;

$R_i :=$ cualquier clave candidata para R ;

end

return (R_1, R_2, \dots, R_i)

Comparación entre BCNF y 3NF

- Siempre es posible descomponer una relación en un conjunto de relaciones que estén en 3NF, logrando:
 - Una descomposición sin pérdida.
 - Preservando las dependencias funciones.
- Siempre es posible descomponer una relación en un conjunto de relaciones que estén en BCNF, logrando:
 - Una descomposición sin pérdida.
 - No siempre es posible lograr una descomposición que preserve las dependencias.

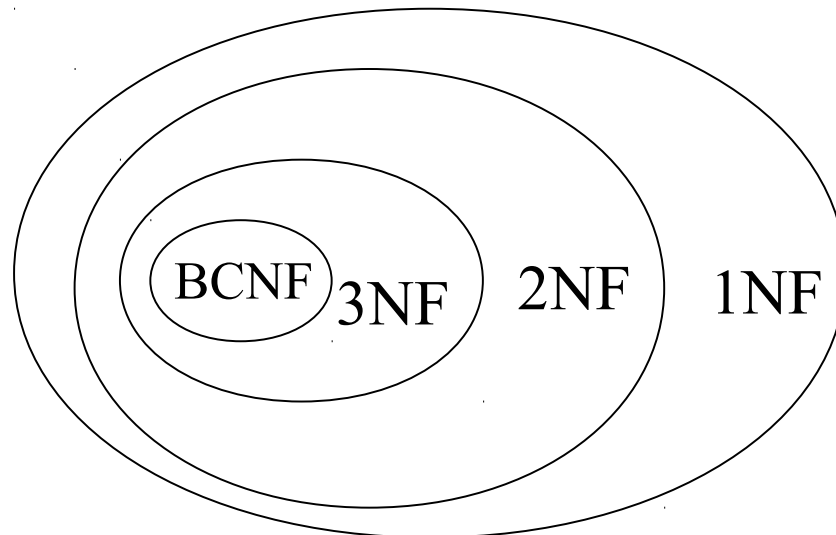
Segunda Forma Normal

- Para todo atributo no primo de R, no depende parcialmente de ninguna clave candidata de R.

Primera Forma Normal

- Un esquema de relación R está 1NF si todos sus atributos son atómicos.

- Las formas normales BCNF, 3NF, 2NF y 1NF son inclusivas es decir, si un esquema R está en BCNF, también lo estará en 3NF, si un esquema R está en 3NF, también lo estará en 2NF, y así...
- Intuitivamente:



Ejemplos de Diferentes Formas Normales en Instancias de Tablas

PersonasVehiculos = (DNI, NYApellido, Dir, Npat, Modelo, Marca)

DNI	NYApellido	Dir	Npat	Modelo	Marca
23	Juan Peres	Bs As 345	NDT 454	2000	Fiat
23	Juan Peres	Bs As 345	KJI 566	2005	V W
24	Carlos Puig	Mendoza 2211	KJI 566	2005	V W
24	Carlos Puig	Mendoza 2211	GFT 654	2001	Honda

Dependencias: DNI → NYApellido, Dir

Npat → Modelo Y Marca

PersonaVehiculos está en 1NF



Ciudades(Nprov,nombreProv,capital,Nciudad,nombreC,habitantesC)

Nprov	NombreProv	capital	Nciudad	nombreC	habitantesC
1	Cordoba	c1	c1	Cordoba	2000000
1	Cordoba	c1	c2	Rio Cuarto	160000
1	Cordoba	c1	c3	Rio Tercero	50000
1	Cordoba	c1	c4	Villa Maria	60000
2	Santa fe	c5	c5	Santa fe	250000
2	Santa fe	c5	c6	Rosario	150000
2	Santa fe	c5	c7	Rufino	80000

Dependencias: Nprov → NombreProv, capital

Nciudad → nombreC,habitantesC, Nprov

Ciudades está en 2NF

Imparte = (estudiante, curso, profesor)

- Restricciones: Un profesor dicta sólo un curso y dado un estudiante y un curso se conoce quien es el profesor.

Estudiante	curso	
Juan	Mate	
Juan	Fisic	
null	Mate	

$F = \{ \text{estudiante, curso} \rightarrow \text{profesor} \}$
 $\text{profesor} \rightarrow \text{curso} \}$

Imparte está en 3NF.



Personas(DNI,NYApellido,Dir)

DNI	NYApellido	
23	Juan Peres	
24	Carlos Puig	

Dependencia: DNI \rightarrow NYApellido, Dir

Personas está en BCNF.

Ejemplos de Normalización

Ejemplo nro 1:

•Vivero (#sector, dni_empleado, nombre_empleado, tipo_planta, nombre_especie, hora, dia, cant_luz)

Que representa la información de un vivero y teniendo en cuenta lo siguiente:

Los números de sector no se repiten.

En un sector pueden trabajar más de un empleado y puede haber más de un tipo de plantas (por ejemplo, margaritas y rosas).

Un empleado puede trabajar en varios sectores pero en un sector se encarga de cuidar un solo tipo de planta.

Los tipos de plantas pertenecen a una única especie.

Todos los días se registra a que hora se empezó a regar cada sector y quién fue el empleado que lo hizo.

Un sector se riega una sola vez por día.

En cada hora del día y por cada sector, se verifica con un sensor, la cantidad de luz que entra.

Primero: Definir las dep. funcionales

- 1) dni_empleado, #sector → tipo_planta
- 2) tipo_planta → nombre_especie
- 3) #sector, dia → hora, dni_empleado
- 4) dia, hora, #sector → cant_luz
- 5) dni_empleado → nombre_empleado

Segundo: Definir las claves candidatas

(#sector, dia)

- Tercer paso llevar a BCNF o 3NF.

Comienzo dividiendo por 2)

R1 (tipo_planta, nombre_especie)

FR1 {2)}

R1 con respecto a FR1 está en BCNF.

R2(#sector, dni_empleado, nombre_empleado, nombre_planta, hora, dia, cant_luz)

FR2 { 1), 3), 4), 5)}

R2 con respecto a FR2 está en 1NF, por lo tanto debo dividirla.

Ahora Divido por 1)

R3(dni_empleado, #sector, tipo_planta)

FR2 {1))

R3 con respecto a FR3 está en BCNF.

R4(#sector, dni_empleado, nombre_empleado, hora, dia, cant_luz)

FR4 {3), 4), 5)}

R4 con respecto a FR4 está en 2NF, por lo tanto debo dividirla.

Div R4 por 3)?

NOOOOOOO!!!

porque R4 por 3) está en BCNF.

Div R4 por 5)

R5(dni_ empleado, nombre_ empleado)

FR5 {5)}

R5 con respecto a FR5 está en BCNF.

R6(#sector, dni_ empleado, hora, dia, cant_ luz)

FR6 {3), 4)}

R6 con respecto a FR6 está en BCNF.

Esquema Final Resultante está en BCNF

El esquema final está compuesto por las siguientes relaciones:

R1, R3, R5 Y R6

Ejemplo nro 2:

Torneo_de_tenis (#cancha, dniJugador, dirJugador, dniArbitro, dirArbitro, hora_partido, fechaPartido, resultado, #partido)

Este esquema representa la información de un torneo de tenis ya realizado y teniendo en cuenta lo siguiente:

El #partido representan el orden de cada partido en el campeonato, los partidos se juegan en una cancha, en una fecha y en un determinado horario. Los jugadores pueden jugar un solo partido por día, por cada partido se conoce su resultado (por ej. 6/3, 6/2) y el arbitro, un arbitro puede dirigir más de un partido por día, pero uno sólo a la vez.

Primero: Definir las dep. funcionales

- 1) #cancha, hora_partido, fechaPartido → #partido
- 2) #partido → #cancha, hora_partido, fechaPartido, resultado, dniArbitro
- 3) dniJugador, fechaPartido → #partido
- 4) dirArbitro, hora_partido, fechaPartido → #partido
- 5) dniJugador → dirJugador
- 6) dniArbitro → dirArbitro

Segundo: Definir las claves candidatas

(dniJugador, fechaPartido) (dniJugador, #partido)

- Tercer paso llevar a BCNF o 3NF.