

Bases de Datos 1

Alejandra Lliteras

alejandra.lliteras@lifa.info.unlp.edu.ar



En la clase anterior...

Teoría de diseño de BBDD relacionales

Vimos la clase anterior

- Descomposición
- Análisis y proceso de normalización hasta BCNF o 3FN

Teoría de diseño de BBDD relacionales

► Cómo llevar un esquema R a BCNF

De manera esquemática y simplificada, una vez halladas las dependencias funcionales y las claves candidatas

1-analizar si en el esquema R existe alguna dependencia funcional que no cumple con la definición de BCNF

1.1) si existe tal dependencia funcional, particionar el esquema en dos nuevos esquemas R_i , R_{i+1} , contemplando la dependencia funcional en cuestión. Analizar las 2 particiones generadas

1.1.1) Se pierde información?

1.1.1.1: NO, entonces sigo a 1.1.2

1.1.1.2: SI. La partición es errónea. Reanalizar

1.1.2) Se pierden Dependencias funcionales?

1.1.2.1 NO, entonces sigo a 1.1.3

1.1.2.2 Si. Entonces no es posible llevar a BCNF. Cambia la forma normal analizada.

1.1.3) Determinar en que forma normal esta R_i , R_{i+1} , si no están en BCNF, reiniciar desde 1, sino pasar a 1.2

1.2) Si no existe, el esquema está en BCNF

Teoría de diseño de BBDD relacionales

► Cómo llevar un esquema R a 3FN

De manera esquemática y simplificada, una vez halladas las dependencias funcionales y las claves candidatas y habiendo detectado que no se puede llevar a BCNF

- Se construye una tabla por cada dependencia funcional
- Si la clave de la tabla original, no está incluida en ninguna de las tablas del punto anterior, se construye una tabla con la clave

Teoría de diseño de BBDD relacionales

En síntesis:

Llamamos **normalizar** hasta BCNF o 3FN, al **proceso** que involucra, hasta ahora, los siguientes pasos:

- Encontrar las dependencias funcionales y las claves candidatas
- Llevar a BCNF aplicando el proceso de división sin pérdida de información
 - Comprobar que no se pierden dependencias funcionales en la división
 - Si se pierden dependencias funcionales al dividir, se lleva el esquema correspondiente a 3NF

Teoría de diseño de BBDD relacionales

Alcanza con dejar los esquemas en BCNF o 3FN para quitar la anomalía de redundancia?

Teoría de diseño de BBDD relacionales

Las particiones del FIESTAS, que quedaron en BCNF, son:

F1(#salon, direccion, capacidad)

F3 (nom_contratante, dir_contratante)

F5(#salon, fecha_fiesta, dni_invitado, mesa_invitado)

F7(#salon, fecha_fiesta, cant_invitados, cant_mesas)

F9(dni_invitado, nombre_invitado)

F10(#salon, fecha_fiesta, nom_contratante, servicio_contratado,
dni_invitado)

Teoría de diseño de BBDD relacionales

Como es una instancia de F10?

(#salon, fecha_fiesta, nom_contratante, servicio_contratado, dni_invitado

| #salón | fecha_fiesta | nom_contratante | serv_contratado | dni_invitado |
|--------|--------------|-----------------|-----------------|--------------|
| #1 | 10/03/12 | Pedro Guti | empanadas | 11111111 |
| #1 | 10/03/12 | Pedro Guti | empanadas | 11222222 |
| #1 | 10/03/12 | Pedro Guti | pizza | 11111111 |
| #1 | 10/03/12 | Pedro Guti | pizza | 11222222 |
| #1 | 10/03/12 | María Zeta | empanadas | 11111111 |
| #1 | 10/03/12 | María Zeta | empanadas | 11222222 |
| #1 | 10/03/12 | María Zeta | piza | 11111111 |
| #1 | 10/03/12 | María Zeta | piza | 11222222 |
| #1 | 11/03/12 | Juan Zeballos | Mesa de quesos | 11333333 |
| #1 | 11/03/12 | Juan Zeballos | calentitos | 11333333 |
| ... | ... | ... | ... | ... |

► **Esta clase veremos**

Teoría de diseño para bases de datos relaciones

Dependencia Multivaluada (DM)

DM Trivial

caso especial de DM con \emptyset

4ta. Forma Normal

Teoría de diseño de BBDD relacionales

► Dependencia Multivaluada

- Una dependencia multivaluada, afirma que dos o mas atributos son independientes del resto
- Como consecuencia de la independencia, se tiene redundancia. Esta redundancia no se elimina con las dependencias funcionales

Teoría de diseño de BBDD relacionales

► Dependencia Multivaluada

- se puede decir que: $X \twoheadrightarrow Y$ si dado un valor de X , hay un conjunto de valores de Y asociados y este conjunto de valores de Y **NO** está relacionado (ni funcional ni multifuncionalmente) con los valores de $R - X - Y$ (donde R es el esquema), es decir Y es independiente de los atributos de $R - X - Y$.

Teoría de diseño de BBDD relacionales

► Dependencia Multivaluada

(otra forma de definirla)

Sea R un esquema de relación

La Dependencia Multivaluada $X \twoheadrightarrow Y$ vale en R si \forall los pares de tuplas t_1 y t_2 en R , tal que

- $t_1[X] = t_2[X]$ existen las tuplas t_3 y t_4 en R tales que:
 - $t_1[X] = t_2[X] = t_3[X] = t_4[X]$
 - $t_3[Y] = t_1[Y]$
 - $t_3[R-X-Y] = t_2[R-X-Y]$
 - $t_4[Y] = t_2[Y]$
 - $t_4[R-X-Y] = t_1[R-X-Y]$

Teoría de diseño de BBDD relacionales

► Dependencia Multivaluada

- Ejemplo:

PERSONA (dni, apellido, dirección)

- Donde
 - El número de dni es único
 - Varias personas pueden tener el mismo apellido y toda persona tiene un único apellido
 - Cada persona puede tener mas de una dirección

| dni | apellido | dirección |
|----------|----------|---------------|
| 22145147 | López | 12 Nro.175 |
| 22145147 | López | 122 nro. 5689 |
| 22145147 | López | 4 nro 321 |
| 4874701 | Torres | 156 nro. 4567 |

PERSONA (dni, apellido, dirección)

- Donde
 - El número de dni es único
 - Varias personas pueden tener el mismo apellido y toda persona tiene un único apellido
 - Cada persona puede tener mas de una dirección

| dni | apellido | dirección |
|----------|----------|---------------|
| 22145147 | López | 12 Nro.175 |
| 22145147 | López | 122 nro. 5689 |
| 22145147 | López | 4 nro 321 |
| 4874701 | Torres | 156 nro. 4567 |

¿Cuáles son las dependencias funcionales de PERSONA?

df1) dni \rightarrow apellido

¿Qué sucede con el atributo dirección?

PERSONA (dni, apellido, dirección)

- Donde
 - El número de dni es único
 - Varias personas pueden tener el mismo apellido y toda persona tiene un único apellido
 - Cada persona puede tener mas de una dirección

| dni | apellido | dirección |
|----------|----------|---------------|
| 22145147 | López | 12 Nro.175 |
| 22145147 | López | 122 nro. 5689 |
| 22145147 | López | 4 nro 321 |
| 4874701 | Torres | 156 nro. 4567 |

df1) dni → apellido

El atributo dirección no está relacionado funcionalmente con el resto de los atributos, hay independencia.

Dirección forma parte de la clave candidata

Cc {dni, dirección}

¿PERSONA está en BCNF?

PERSONA (dni, apellido, dirección)

- Donde
 - El número de dni es único
 - Varias personas pueden tener el mismo apellido y toda persona tiene un único apellido
 - Cada persona puede tener mas de una dirección

| dni | apellido | dirección |
|----------|----------|---------------|
| 22145147 | López | 12 Nro.175 |
| 22145147 | López | 122 nro. 5689 |
| 22145147 | López | 4 nro 321 |
| 4874701 | Torres | 156 nro. 4567 |

df1) dni -> apellido

Cc {dni, dirección}

¿PERSONA está en BCNF?

No. Existe la df1, tal que {dni} no es superclave del esquema PERSONA. Entonces, particiono el esquema PERSONA, contemplando la df1

PERSONA1(dni , apellido)

PERSONA2(dni,dirección)

PERSONA (dni, apellido, dirección)

- Donde
 - El número de dni es único
 - Varias personas pueden tener el mismo apellido y toda persona tiene un único apellido
 - Cada persona puede tener mas de una dirección

| dni | apellido | dirección |
|----------|----------|---------------|
| 22145147 | López | 12 Nro.175 |
| 22145147 | López | 122 nro. 5689 |
| 22145147 | López | 4 nro 321 |
| 4874701 | Torres | 156 nro. 4567 |

df1) dni -> apellido

cc {dni, dirección}

PERSONA1(dni , apellido)

PERSONA2(dni,dirección)

Con el particionamiento propuesto:

¿Se perdió información?

PERSONA1 \cap PERSONA2 es clave en el esquema PERSONA1 {dni}

¿Se perdieron dependencias funcionales?

En PERSONA1, vale df1

PERSONA (dni, apellido, dirección)

- Donde
 - El número de dni es único
 - Varias personas pueden tener el mismo apellido y toda persona tiene un único apellido
 - Cada persona puede tener mas de una dirección

| dni | apellido | dirección |
|----------|----------|---------------|
| 22145147 | López | 12 Nro.175 |
| 22145147 | López | 122 nro. 5689 |
| 22145147 | López | 4 nro 321 |
| 4874701 | Torres | 156 nro. 4567 |

df1) dni -> apellido

cc {dni, dirección}

PERSONA1(dni , apellido)

PERSONA2(dni,dirección)

PERSONA1 cumple con la definición de BCNF. Está en BCNF

PERSONA2 cumple con la definición de BCNF. Está en BCNF

En PERSONA2 VALE LA DEPENDENCIA MULTIVALUADA

DM1) dni ->> dirección

Teoría de diseño para bases de datos relaciones

Dependencia Multivaluada trivial



Teoría de diseño de BBDD relacionales

► Dependencia Multivaluada trivial

Sea R un esquema de relación

Una dependencia multivaluada de la forma $X \twoheadrightarrow Y$ que vale en R es trivial si:

- el conjunto de atributos X, Y son todos los atributos del esquema

► Ejemplo

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A4 (codHospital, legajoPaciente, dniMedico)

En A4, vimos que vale la dependencia multivaluada:

DM1) codHospital, legajoPaciente \twoheadrightarrow dniMedico

Por la definición, que vimos recién, DM1, es trivial en A4

Teoría de diseño de BBDD relacionales

Dependencia Multivaluada
Caso especial

\emptyset (vacío)

Teoría de diseño de BBDD relacionales

► Caso especial de dependencia multivaluada

Supongamos la relación:

CasaAlquilerVideo (#video, titulo, #cliente, nyAp_cliente)

Donde

- #Cliente representa a los clientes que alguna vez realizaron un alquiler
- #video son todos los videos que hay en la casa de alquiler

Dep. funcionales:

df1: #video → titulo

df2: #cliente → nyAp_cliente

Clave candidata: {#cliente, #video}

Teoría de diseño de BBDD relacionales

► Caso especial de dependencia multivaluada

Supongamos la relación:

CasaAlquilerVideo (#video, titulo, #cliente, nyAp_cliente)

Dep. funcionales:

df1: #video → titulo

df2: #cliente → nyAp_cliente

Clave candidata: {#cliente, #video}

Notamos que CasaAlquilerVideo no cumple con la definición de BCNF ya que existe, por ejemplo la df1, donde {#video} no es superclave del esquema. Entonces se particiona CasaAlquilerVideo, considerando la df1

Teoría de diseño de BBDD relacionales

► Caso especial de dependencia multivaluada

Supongamos la relación:

CasaAlquilerVideo (#video, titulo, #cliente, nyAp_cliente)

Dep. funcionales:

df1: #video → titulo

df2: #cliente → nyAp_cliente

Clave candidata: {#cliente, #video}

Notamos que CasaAlquilerVideo no cumple con la propiedad de BCNF ya que existe, por ejemplo la df1, donde {#video} no es superclave del esquema. Entonces se particiona CasaAlquilerVideo, considerando la df1:

R1 (#video, titulo)

R2 (#video, #cliente, nyAp_cliente)

Teoría de diseño de BBDD relacionales

► Caso especial de dependencia multivaluada

Supongamos la relación:

CasaAlquilerVideo (#video, titulo, #cliente, nyAp_cliente)

Dep. funcionales:

df1: #video → titulo

df2: #cliente → nyAp_cliente

Clave candidata: {#cliente, #video}

R1 (#video, titulo)

R2 (#video, #cliente, nyAp_cliente)

- R1 cumple con BCNF ya que la única df que vale es df1 y {#video} es superclave en R1
- R2 no cumple con BCNF por la existencia de la df2, cuyo determinante no es superclave en R2, particiono

Teoría de diseño de BBDD relacionales

► Caso especial de dependencia multivaluada

Supongamos la relación:

CasaAlquilerVideo (#video, titulo, #cliente, nyAp_cliente)

Dep. funcionales:

df1: #video → titulo

df2: #cliente → nyAp_cliente

Clave candidata: {#cliente, #video}

R2 (#video, #cliente, nyAp_cliente)

Particionamos el esquema R2 contemplando la df2 (por lo antes descripto)

R3 (#cliente, nyAp_cliente)

R4 (#video, #cliente)

Teoría de diseño de BBDD relacionales

► Caso especial de dependencia multivaluada

Supongamos la relación:

CasaAlquilerVideo (#video, titulo, #cliente, nyAp_cliente)

Dep. funcionales:

df1: #video → titulo

df2: #cliente → nyAp_cliente

Clave candidata: {#cliente, #video}

R3 (#cliente, nyAp_cliente)

R4 (#video, #cliente)

- R3 cumple con BCNF ya que la única df que vale es df2 y {#cliente} es superclave en R3
- R4 cumple con BCNF, todos sus atributos forman parte de la clave

Teoría de diseño de BBDD relacionales

- ▶ Caso especial de dependencia multivaluada

R4 (#video, #cliente)

Donde

- #Cliente representa a los clientes que alguna vez realizaron un alquiler
- #video son todos los videos que hay en la casa de alquiler

Supongamos un ejemplo con datos particulares en la partición R4.
Que forma tendrían que tener?

| <u>#cliente</u> | <u>#video</u> |
|-----------------|---------------|
| c1 | v1 |
| c1 | v2 |
| c1 | v3 |
| c2 | v1 |
| c2 | v2 |
| c2 | v3 |

Teoría de diseño de BBDD relacionales

► Caso especial de dependencia multivaluada

R4 (#video, #cliente)

Donde

- #Cliente representa a los clientes que alguna vez realizaron un alquiler
- #video son todos los videos que hay en la casa de alquiler

► Se dice en este caso, que en R4 valen las siguientes dependencias multivaluadas:

- DM1) $\emptyset \twoheadrightarrow \#cliente$
- DM2) $\emptyset \twoheadrightarrow \#video$

Teoría de diseño de BBDD relacionales

¿ Cómo proceder cuando se hallan dependencias multivaluadas?

Teoría de diseño de BBDD relacionales

Formas normales
4FN

Teoría de diseño de BBDD relacionales

- ▶ 4FN (Cuarta Forma Normal)
 - Un esquema R está en 4NF con respecto a un conjunto de dependencias multivaluadas D , si \forall dependencia multivaluada de la forma $X \twoheadrightarrow Y$ se cumple que:
 - $X \twoheadrightarrow Y$ es una dependencia multivaluada trivial

Teoría de diseño de BBDD relacionales

► 4FN (Cuarta Forma Normal)

En otras palabras:

- Un esquema está en 4FN cuando:
 - No tiene dependencias multivaluadas
- O bien,
- Las dependencias multivaluadas que en él valen, son triviales.

Teoría de diseño de BBDD relacionales

Habíamos estado trabajando sobre el esquema FIESTAS

FIESTAS (#salon, dirección, capacidad, fecha_fiesta, nom_contratante, cant_invitados, nombre_invitado, cant_mesas, mesa_invitado, servicio_contratado, dir_contratante, dni_invitado)

- En cada salón se realiza una sola fiesta por día, en un día puede haber varias fiestas en diferentes salones.
- Para cada fiesta puede figurar más de un contratante
- Cada invitado a una fiesta tiene asociado un número de mesa
- La cantidad de mesas del salón varia para cada fiesta
- Servicio contratado es una lista que describe los tipos de comida contratados para una fiesta (Esto quiere decir que para cada comida de la fiesta, aparece una tupla diferente en el esquema)
- Una persona puede ir a más de una fiesta en el mismo salón en diferentes fechas
- Una persona puede ir a mas de una fiesta el mismo día

Teoría de diseño de BBDD relacionales

FIESTAS (#salon, dirección, capacidad, fecha_fiesta, nom_contratante, cant_invitados, nombre_invitado, cant_mesas, mesa_invitado, servicio_contratado, dir_contratante, dni_invitado)

Y habíamos hallado las dependencias funcionales y la clave candidata válidas en el esquema FIESTAS:

Dependencias Funcionales

1. #salon → dirección, capacidad
2. nom_contratante → dir_contratante
3. #salon, fecha_fiesta, dni_invitado → mesa_invitado
4. #salon, fecha_fiesta → cant_invitados, cant_mesas
5. dni_invitado → nombre_invitado

Clave candidata: (#salon, fecha_fiesta, dni_invitado, nom_contratante, servicio_contratado)

Teoría de diseño de BBDD relacionales

FIESTAS (#salon, dirección, capacidad, fecha_fiesta, nom_contratante, cant_invitados, nombre_invitado, cant_mesas, mesa_invitado, servicio_contratado, dir_contratante, dni_invitado)

Y luego del proceso de particionamiento para llevar a BCNF, llegamos a las siguientes particiones:

F1(#salon, direccion, capacidad)

F3 (nom_contratante, dir_contratante)

F5(#salon, fecha_fiesta,dni_invitado, mesa_invitado)

F7(#salon, fecha_fiesta, cant_invitados, cant_mesas)

F9(dni_invitado, nombre_invitado)

F10(#salon, fecha_fiesta, nom_contratante, servicio_contratado, dni_invitado)

Teoría de diseño de BBDD relacionales

F10(#salon, fecha_fiesta, nom_contratante, servicio_contratado,
dni_invitado)

¿Tiene dependencias multivaluadas F10?

- En cada salón se realiza una sola fiesta por día, en un día puede haber varias fiestas en diferentes salones.
- Para cada fiesta puede figurar más de un contratante
- Cada invitado a una fiesta tiene asociado un número de mesa
- La cantidad de mesas del salón varia para cada fiesta
- Servicio contratado es una lista que describe los tipos de comida contratados para una fiesta (Esto quiere decir que para cada comida de la fiesta, aparece una tupla diferente en el esquema)
- Una persona puede ir a más de una fiesta en el mismo salón en diferentes fechas
- Una persona puede ir a mas de una fiesta el mismo día

Teoría de diseño de BBDD relacionales

F10(#salon, fecha_fiesta, nom_contratante, servicio_contratado, dni_invitado)

¿Las siguientes propuestas son válidas de acuerdo a la semántica de los atributos?

#salon -->> DNI_Invitado **X** El invitado depende de la fecha de la fiesta

#salon, fecha_fiesta -->> DNI_invitado ? **✓**

| #salón | Fecha_fiesta | Nom_contr | Serv_contr | Dni_invitado |
|--------|--------------|---------------|----------------|--------------|
| #1 | 10/03/12 | Pedro Guti | empanadas | 11111111 |
| #1 | 10/03/12 | Pedro Guti | empanadas | 11222222 |
| #1 | 10/03/12 | Pedro Guti | pizza | 11111111 |
| #1 | 10/03/12 | Pedro Guti | pizza | 11222222 |
| #1 | 10/03/12 | María Zeta | empanadas | 11111111 |
| #1 | 10/03/12 | María Zeta | empanadas | 11222222 |
| #1 | 10/03/12 | María Zeta | pizza | 11111111 |
| #1 | 10/03/12 | María Zeta | pizza | 11222222 |
| #1 | 11/03/12 | Juan Zeballos | Mesa de quesos | 11333333 |
| #1 | 11/03/12 | Juan Zeballos | calentitos | 11333333 |
| ... | ... | ... | ... | ... |

Teoría de diseño de BBDD relacionales

F10(#salon, fecha_fiesta, nom_contratante, servicio_contratado,
dni_invitado)

- En cada salón se realiza una sola fiesta por día, en un día puede haber varias fiestas en diferentes salones.
- Para cada fiesta puede figurar más de un contratante
- Cada invitado a una fiesta tiene asociado un número de mesa
- La cantidad de mesas del salón varia para cada fiesta
- Servicio contratado es una lista que describe los tipos de comida contratados para una fiesta (Esto quiere decir que para cada comida de la fiesta, aparece una tupla diferente en el esquema)
- Una persona puede ir a más de una fiesta en el mismo salón en diferentes fechas
- Una persona puede ir a mas de una fiesta el mismo día

En F10, dijimos que valía la DM: #salon, fecha_fiesta -->> DNI_invitado

¿Qué otras dependencias multivaluadas valen en F10?

Teoría de diseño de BBDD relacionales

F10(#salon, fecha_fiesta, nom_contratante, servicio_contratado,
dni_invitado)

- En cada salón se realiza una sola fiesta por día, en un día puede haber varias fiestas en diferentes salones.
- Para cada fiesta puede figurar más de un contratante
- Cada invitado a una fiesta tiene asociado un número de mesa
- La cantidad de mesas del salón varia para cada fiesta
- Servicio contratado es una lista que describe los tipos de comida contratados para una fiesta (Esto quiere decir que para cada comida de la fiesta, aparece una tupla diferente en el esquema)
- Una persona puede ir a más de una fiesta en el mismo salón en diferentes fechas
- Una persona puede ir a mas de una fiesta el mismo día

#salon, fecha_fiesta ->> servicio_contratado

#salon, fecha_fiesta ->> nom_contratante

Teoría de diseño de BBDD relacionales

Análisis para dejar una partición en 4FN



Teoría de diseño de BBDD relacionales

F10(#salon fecha_fiesta, nom_contratante, servicio_contratado, dni_invitado)

Dependencias Multivaluadas válidas sobre F10:

DM1) #salon, fecha_fiesta \twoheadrightarrow servicio_contratado

DM2) #salon, fecha_fiesta \twoheadrightarrow nom_contratante

DM3) #salon, fecha_fiesta \twoheadrightarrow dni_invitado

El esquema F10 no está en 4NF porque existen dependencias multivaluadas 1-3 y no son triviales en F10. Entonces se particiona F10 utilizando una de las dependencias multivaluadas, por ejemplo DM1

Teoría de diseño de BBDD relacionales

F10(#salon fecha_fiesta, nom_contratante, servicio_contratado,
dni_invitado)

Dependencias Multivaluadas válidas sobre F10:

DM1) #salon, fecha_fiesta ->> servicio_contratado

DM2) #salon, fecha_fiesta ->> nom_contratante

DM3) #salon, fecha_fiesta ->> dni_invitado

F11(#salon, fecha_fiesta, servicio_contratado)

F12(#salon, fecha_fiesta, nom_contratante,
dni_invitado)

Teoría de diseño de BBDD relacionales

F11(#salon, fecha_fiesta, servicio_contratado)

F12(#salon, fecha_fiesta, nom_contratante, dni_invitado)

La partición F11 esta en 4NF ya que sólo vale la dependencia multivaluada 1 que es trivial en ella.

La particion F12 no esta en 4NF porque existen las dependencias multivaluadas:

DM2) #salon, fecha_fiesta $\rightarrow\rightarrow$ nom_contratante

DM3) #salon, fecha_fiesta $\rightarrow\rightarrow$ dni_invitado

y éstas no son triviales en F12.

Entonces divido la partición F12 utilizando la dependencia multivaluada 2 quedando lo siguiente:

F13(#salon, fecha_fiesta, nom_contratante)

F14(#salon, fecha_fiesta, dni_invitado)

Teoría de diseño de BBDD relacionales

F13(#salon, fecha_fiesta, nom_contratante)

F14(#salon, fecha_fiesta, dni_invitado)

La partición F13 está en 4NF ya que sólo vale la dependencia multivaluada 2 que es trivial en ella.

La partición F14 está en 4NF porque existe solamente la dependencia multivaluada DM3 que es trivial en ella.

Teoría de diseño de BBDD relacionales

En síntesis



Resumiendo el proceso de normalización hasta 4FN

1. Encontrar las dependencias funcionales
2. Encontrar las claves candidatas
3. Verificar si el esquema cumple con la definición de BCNF, si no se cumple descomponer la relación sin perder información ni dependencias funcionales
 - Si se pierden dependencias funcionales, llevar a 3NF
 - Esto hasta dejar las particiones en BCNF o 3FN
4. Expresar en este punto, cual es la clave primaria y que particiones quedaron en BCNF o 3FN (según corresponda)
5. Encontrar las dependencias multivaluadas sobre la última partición realizada (aquella que tiene la clave primaria del esquema) y verificar 4NF, si no se cumple dividir la relación

Resumiendo el proceso de normalización hasta 4FN

1. Encontrar las dependencias funcionales
2. Encontrar las claves candidatas
3. Verificar si el esquema cumple con la definición de BCNF, si no se cumple descomponer la relación sin perder información ni dependencias funcionales
 - Si se pierden dependencias funcionales, llevar a 3NF
 - Esto hasta dejar las particiones en BCNF o 3FN
4. Expresar en este punto, cual es la clave primaria y que particiones quedaron en BCNF o 3FN (según corresponda)
5. Encontrar las dependencias multivaluadas sobre la última partición realizada (aquella que tiene la clave primaria del esquema) y verificar 4NF, si no se cumple dividir la relación

Las particiones del FIESTAS, que quedaron en BCNF, son:

F1(#salon, direccion, capacidad)

F3 (nom_contratante, dir_contratante)

F5(#salon, fecha_fiesta, dni_invitado, mesa_invitado)

F7(#salon, fecha_fiesta, cant_invitados, cant_mesas)

F9(dni_invitado, nombre_invitado)

F10(#salon, fecha_fiesta, nom_contratante, servicio_contratado, dni_invitado)

Resumiendo el proceso de normalización hasta 4FN

1. Encontrar las dependencias funcionales
2. Encontrar las claves candidatas
3. Verificar si el esquema cumple con la definición de BCNF, si no se cumple descomponer la relación sin perder información ni dependencias funcionales
 - Si se pierden dependencias funcionales, llevar a 3NF
 - Esto hasta dejar las particiones en BCNF o 3FN
4. Expresar en este punto, cual es la clave primaria y que particiones quedaron en BCNF o 3FN (según corresponda)
5. Encontrar las dependencias multivaluadas sobre la última partición realizada (aquella que tiene la clave primaria del esquema) y verificar 4NF, si no se cumple dividir la relación
 - Esto se hace hasta dejar las particiones del esquema en 4FN
6. Expresar las particiones resultantes que quedaron en 4FN
 - Explicar porque las particiones descritas en el ítem 4 (excepto la analizada en el punto 5) quedaron en 4FN
7. Indicar que particiones en 4FN quedan en el esquema final (que no sean proyecciones de atributos claves de otras particiones en 4FN)

Ejercicio Grupal

Teoría de diseño de BBDD relacionales

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Donde:

- Un paciente tiene asignado para cada hospital un número de legajo
- Un legajo en un hospital se asigna a una única persona
- En un hospital trabajan muchos médicos y un médico puede trabajar en diversos hospitales
- Un médico atiende a muchos pacientes
- Cada hospital posee un nombre y el mismo nombre se puede repetir para diferentes hospitales
- Un paciente se atiende en muchos hospitales y de cada hospital que se atiende se registran los médicos que lo atienden

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajopaciente, dniMedico)

Donde:

- Un paciente tiene asignado para cada hospital un número de legajo
- Un legajo en un hospital se asigna a una única persona
- En un hospital trabajan muchos médicos y un médico puede trabajar en diversos hospitales
- Un médico atiende a muchos pacientes
- Cada hospital posee un nombre y el mismo nombre se puede repetir para diferentes hospitales
- Un paciente se atiende en muchos hospitales y de cada hospital que se atiende se registran los médicos que lo atienden

Dependencias Funcionales

df1) codHospital → nombreHospital

df2) legajoPaciente, codHospital → dniPaciente

df3) dniPaciente, codHospital → legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

ATENCIONES cumple con la definición de BCNF?

Para toda dependencia funcional se cumple que:

X es superclave de R

o bien

$X \rightarrow A$ es una dependencia funcional trivial

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital → nombreHospital

df2) legajoPaciente, codHospital → dniPaciente

df3) dniPaciente, codHospital → legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

Cómo el esquema **ATENCIONES** no cumple con la definición de BCNF, ya que al menos encontramos a la df1 donde {codHospital} no es superclave del esquema **ATENCIONES** y sabemos que se puede particionar para eliminar anomalías, procedemos a particionar **ATENCIONES** contemplando la df1

1. codHospital → nombreHospital

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital → nombreHospital

df2) legajoPaciente, codHospital → dniPaciente

df3) dniPaciente, codHospital → legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

Cómo el esquema **ATENCIONES** no cumple con la definición de BCNF, ya que al menos encontramos a la df1 donde {codHospital} no es superclave del esquema **ATENCIONES** y sabemos que se puede particionar para eliminar anomalías, procedemos a particionar **ATENCIONES** contemplando la df1.

Al particionar, tenemos:

A1(codHospital, nombreHospital)

A2 (codHospital, dniPaciente, legajoPaciente, dniMedico)

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A1 (codHospital, nombreHospital)

A2 (codHospital, dniPaciente, legajoPaciente, dniMedico)

Con el particionamiento propuesto:

¿Se perdió información?

¿Se perdieron dependencias funcionales?

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A1 (codHospital, nombreHospital)

A2 (codHospital, dniPaciente, legajoPaciente, dniMedico)

Con el particionamiento propuesto:

¿Se perdió información?

A1 \cap A2 es clave en el esquema ATENCIONES , {codHospital}
Entonces no se perdió información

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A1 (codHospital, nombreHospital)

A2 (codHospital, dniPaciente, legajoPaciente, dniMedico)

**Con el particionamiento propuesto:
¿Se perdieron dependencias funcionales?**

En A1, vale df1

En A2 valen las dfs 2 y 3

Entonces, no se perdieron dependencias funcionales.

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A1(codHospital, nombreHospital)

A2 (codHospital, dniPaciente, legajoPaciente, dniMedico)

Ambos esquemas quedaron en BCNF?

- En A1, vale **df1**. Donde {codHospital} es superclave del esquema A1. A1 cumple BCNF.

- En A2 valen las dfs 2 y 3. En particular, existe la df2, donde {legajoPaciente, codHospital} no es superclave de A2. Entonces, podemos afirmar que no cumple BCNF

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A2 (codHospital, dniPaciente, legajoPaciente, dniMedico)

- En A2 valen las dfs 2 y 3. En particular, existe la **df2**, donde {legajoPaciente, codHospital} no es superclave de A2. Entonces, podemos afirmar que no cumple BCNF.
- Por lo antes mencionado, particionamos A2, para llevar a BCNF el esquema

A3 (codHospital, legajoPaciente, dniPaciente)

A4 (codHospital, legajoPaciente, dniMedico)

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A3 (codHospital, legajoPaciente, dniPaciente)

A4 (codHospital, legajoPaciente, dniMedico)

¿Ambos esquemas quedaron en BCNF?

• En A3, vale df2 y df3.

• ¿A3 está en BCNF?

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

A3 (codHospital, legajoPaciente, dniPaciente)

A4 (codHospital, legajoPaciente, dniMedico)

- En A3, vale df2 y df3.

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

- ¿Cuáles son las claves candidatas de A3?

cc1{legajoPaciente, codHospital}

cc2{dniPaciente, codHospital}

A3 cumple con la definición de BCNF?

Para toda dependencia funcional , de la forma $X \rightarrow A$, válida en A3 se cumple que:

X es superclave de R

o bien

$X \rightarrow A$ es una dependencia funcional trivial

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

A3 (codHospital, legajoPaciente, dniPaciente)

A4 (codHospital, legajoPaciente, dniMedico)

• En A3, vale df2 y df3.

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

• ¿Cuáles son las claves candidatas de A3?

cc1{legajoPaciente, codHospital}

cc2{dniPaciente, codHospital}

A3 cumple con la definición de BCNF?

Si, cumple con la definición, los determinantes de ambas dependencias funcionales, son clave (un caso particular de superclave) en A3

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A3 (codHospital, legajoPaciente, dniPaciente)

A4 (codHospital, legajoPaciente, dniMedico)

A4 cumple con la definición de BCNF?

• En A4, todos los atributos forman parte de la clave del esquema. Cualquier dependencia que se halle va a ser trivial. A4 está en BCNF.

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital → nombreHospital

df2) legajoPaciente, codHospital → dniPaciente

df3) dniPaciente, codHospital → legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A3 (codHospital, legajoPaciente, dniPaciente)

A4 (codHospital, legajoPaciente, dniMedico)

¿Que pasó con la cc2?

- Al decidir particionar A2 contemplando la df2, se define en ese punto del proceso a la cc1 como clave del esquema.

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajopaciente, dniMedico)

Donde:

- Un paciente tiene asignado para cada hospital un número de legajo
- Un legajo en un hospital se asigna a una única persona
- En un hospital trabajan muchos médicos y un médico puede trabajar en diversos hospitales
- Un médico atiende a muchos pacientes
- Cada hospital posee un nombre y el mismo nombre se puede repetir para diferentes hospitales
- Un paciente se atiende en muchos hospitales y de cada hospital que se atiende se registran los médicos que lo atienden

A4 (codHospital,legajoPaciente, dniMedico)

¿Hay redundancia?
¿Hay dependencias multivaluadas?

DM1) codHospital,legajoPaciente->> dniMedico

► Ejercicio

ATENCIONES(codHospital, nombreHospital, dniPaciente, legajoPaciente, dniMedico)

Dependencias Funcionales

df1) codHospital \rightarrow nombreHospital

df2) legajoPaciente, codHospital \rightarrow dniPaciente

df3) dniPaciente, codHospital \rightarrow legajoPaciente

Claves Candidatas

cc1) {codHospital, legajoPaciente, dniMedico}

cc2) {codHospital, dniPaciente, dniMedico}

A4 (codHospital, legajoPaciente, dniMedico)

**¿Hay redundancia?
¿Hay dependencias multivaluadas?**

DM1) codHospital, legajoPaciente \rightarrow dniMedico

Dado que la única DM que vale en A4 es trivial,
podemos decir que la partición cumple con la
definición de la 4FN

Ejercicio para analizar
individualmente para la próxima
clase



Teoría de diseño de BBDD relacionales – EJERCICIO–

RECITALES(idRecital, idGrupo, idIntegranteGrupo, marcaInstrumento, vocalista, idOrganizador, nombreGrupo, nombreOrganizador)

Donde

- ▶ en un recital se presentan diversos grupos y un grupo se presenta en diversos recitales
- ▶ cada grupo tiene diversos integrantes. Los integrantes del grupo pueden variar para diferentes recitales
- ▶ cada grupo tiene solamente un vocalista, el vocalista del grupo no varía para los diferentes recitales en los que el grupo se presenta
- ▶ de un grupo se conoce el nombre, pero puede suceder que dos grupos se llamen de igual manera
- ▶ de cada integrante de un grupo y para cada recital en los que el grupo se presenta, se sabe que marca de instrumento uso el integrante
- ▶ un integrante de un grupo puede pertenecer a más de un grupo
- ▶ un vocalista de un grupo puede ser vocalista de otros grupos
- ▶ cada recital tiene diversos organizadores y un organizador puede organizar diversos recitales
- ▶ el nombre del organizador se puede repetir para diferentes idOrganizador. El idOrganizador es único

Realizar el proceso de normalización para llevar RECITALES a 4FN