Capítulo 5

Normalización

Metas de la normalización

- Decidir si un esquema *R* está en "buena" forma.
- En el caso en que *R* no está en "buena" forma, decomponerlo en un conjunto de esquemas $\{R_1, R_2, ..., R_n\}$ tal que:
  - Cada esquema está en buena forma
  - La descomposición es de reunión sin pérdida
- La teoría se basa en las dependencias funcionales.

Metas de la normalización

- Cuando descomponemos un esquema R con un conjunto de DFs F en R_1, R_2,.., R_n queremos
  - **Descomposición de reunión sin pérdida**: de otro modo la descomposición va a tener pérdida de información.
  - **No redundancia de información**: Los esquemas R_i preferentemente deben estar en forma normal de Boyce-Codd o en Tercera forma Normal.
  - **Preservación de las dependencias**: Sea F_i el conjunto de DF de F^+ que incluye solo atributos en R_i.
    - Preferentemente la descomposición debe **conservar las dependencias**, esto es, $(F_1 \cup F_2 \cup \ldots \cup F_n)^+ = F^+$
    - De otro modo, chequear actualizaciones para violaciones de las DFs va a requerir computar reuniones naturales lo cual es costoso.

Forma normal de Boyce Codd

- Cuando *R*, *F* tiene redundancia de datos hay DF no trivial tal que su parte izquierda no determina *R*.

Hay DF no trivial tal que su parte izquierda no determina *R.*

$(\Longleftrightarrow)$ $\exists\alpha$: α no es superclave de R $\wedge$ $\alpha^+ \neq \alpha$.

$(\Longleftrightarrow)$ $\neg(\forall\alpha\subseteq R$: α es superclave de R $\vee$ $\alpha^+ = \alpha)$.

$(\Longleftrightarrow)$ $\neg(\forall\ \alpha \subseteq R$: α es superclave de R V
$(\forall\ \beta \subseteq R:\ \alpha \rightarrow \beta \in F^+ \Longrightarrow\ \beta \subseteq \alpha))$.

$(\Longleftrightarrow)$ $\neg(\forall\ \alpha,\beta \subseteq R: \alpha \rightarrow \beta \in F^+ \Longrightarrow$
$(\alpha$ es superclave de R $\vee$ $\beta \subseteq \alpha))$.

Forma normal de Boyce Codd

- **Definición**: Un esquema *R* está en **BCNF** con respecto a un conjunto *F* de DFs si para todas las DFs en $\boldsymbol{F^+}$ de la forma $\alpha \to \beta$, donde $\alpha \subseteq R$ y $\beta \subseteq R$, al menos una de las siguientes propiedades se cumple:
  - $\alpha \to \beta$ es trivial (i.e., $\beta \subseteq \alpha$)
  - α es una superclave de ***R*** (i.e. $\alpha \to R \in F^+$).

Forma normal de Boyce Codd

- **Proposición**: Para comprobar si R_U, F, R_U esquema universal está en FNBC, basta con comprobar las dependencias de F.

Forma normal de Boyce Codd

- **Ejemplo**: Sea $R = (A, B, C)$ esquema con DFs:

  $F = \{A \rightarrow B, B \rightarrow C\}$.

  - $\{A\}$ es clave candidata de R
  - R no está en FNBC
  - Sea la descomposición de R: $R_1 = (A, B)$, $R_2 = (B, C)$
  - Esta descomposición está en FNBC, es de reunión sin pérdida y preserva las dependencias.

Forma normal de Boyce Codd

- Usar sólo *F* es **incorrecto** cuando se prueba un esquema en una descomposición de *R*.
  - Si R_{U} está descompuesto, hay que comprobar dependencias de $\boldsymbol{F^+}$ en los miembros de la descomposición.

Forma normal de Boyce Codd

- **Ejemplo**: Sea el esquema relacional R = (A, B, C, D) con DFs: F = {A→B, B→C}.
  - {A,D} es clave candidata.
  - R no está en FNBC
  - Sea la descomposición de R: R_1 = (A, B), R_2 = (A, C, D)
  - Esta descomposición no está en FNBC, porque no lo está R_2.

Forma normal de Boyce Codd

- **Comprobación de FNBC**: Sea R_U universal, con DFs F y sea R_i que forma parte de descomposición de R; para probar que R_i está en FNBC se puede hacer la siguiente comprobación:
  - $\forall\ \alpha \subseteq R_i : \alpha^+ \cap (R_i - \alpha) = \phi \vee R_i \subseteq \alpha^+$
- Supongamos que R_i está en FNBC y $\neg R_i \subseteq \alpha^+$:
  - toda $\alpha \rightarrow \beta$ en F^+ con atributos en R_i es trivial.
  - Esto equivale a $\beta \cap (R_i - \alpha) = \phi$
  - Luego: $\alpha^+ \cap (R_i - \alpha) = \phi$ (tomo $\beta = \alpha^+$)

Forma normal de Boyce Codd

- **Ejercicio**: Sea *F* dado por:
  1. nomBib → calle, numero
  2. calle, numero → nomBib
  3. ISBN → título, editorial, autores, edición
  4. nomBib, numInv → ISBN
- Sea la descomposición:
  - BibLibs = (nomBib, numInv, ISBN)
  - Biblioteca = (nomBib, calle, número)
  - Libro = (ISBN, título, editorial, autores, edición)
- Comprobar que *Biblioteca*, *Libro* están en FNBC.

Forma normal de Boyce Codd

- **Observación**: Si $\alpha \subseteq R_i$ viola la condición:

$$\forall \alpha \subseteq R_i : \alpha^+ \cap (R_i - \alpha) = \phi \vee R_i \subseteq \alpha^+$$

 entonces la siguiente DF es testigo:
 - $\alpha \rightarrow \alpha^+ \cap (R_i - \alpha)$.
 - Notar que por teoría de conjuntos:

 $\alpha^+ \cap (R_i - \alpha) = (\alpha^+ - \alpha) \cap (R_i - \alpha) = (\alpha^+ - \alpha) \cap R_i$
 - Luego $\alpha \rightarrow (\alpha^+ - \alpha) \cap R_i$ es testigo.
- Esta DF muestra que R_i viola la FNBC.

Algoritmo de normalización en FNBC

- *result* := {*R*};
  done := false;
  compute F^+;
  while (not *done)* **do**
    if (there is a schema R_i in *result* that is not in BCNF)
      then begin
        let $\alpha \rightarrow \beta$ be a nontrivial functional dependency in F_i
          such that $\alpha \rightarrow R_i$ is not in F^+, and $\alpha \cap \beta = \varnothing$;
        result := (*result* – R_i) $\cup$ (R_i – β) $\cup$ (α, β);
        end
      else *done* := **true;**

Algoritmo de normalización en FNBC

- Probamos que luego de cada paso de iteración obtenemos una descomposición de reunión sin pérdida.
- Luego del primer paso de iteración obtenemos la descomposición: $\{(R_i - \beta), (\alpha, \beta)\}$
  - Observar que $(R_i - \beta) \cap (\alpha, \beta) = \alpha$
  - Por aumentatividad $\alpha \rightarrow \alpha\,\beta \in F^+$
  - Por lo tanto $\{(R_i - \beta), (\alpha, \beta)\}$ es descomposición de reunión sin pérdida (por proposición del capítulo anterior).

Algoritmo de normalización en FNBC

- Asumimos que al terminar el paso de iteración k tenemos una descomposición $R_1,\ldots,R_{k+1}$ de reunión sin pérdida. O sea, para toda $r(R)$ legal con respecto a F:

  $$r = \Pi_R\,(\,\Pi_{R1}(r) \bowtie \Pi_{R2}(r) \bowtie \ldots \bowtie \Pi_{Rn}(r))\ .$$

- Asumimos que en el paso k+1 para algún j se descompone R_j en $\{R_j - \delta, (\gamma, \delta)\}$.
  - Observamos que $R_j - \delta \cap (\gamma, \delta) = \gamma$
  - Por aumentatividad $\gamma \rightarrow \gamma\ \delta \in F_j^+$
  - Luego $\{R_j - \delta, (\gamma, \delta)\}$ es de reunión sin pérdida
    - $s = \Pi_{Rj - \delta}(s) \bowtie \Pi_{(\gamma,\ \delta)}(s)$ para todo s legal en F_j

Algoritmo de normalización en FNBC

Sea $r(R)$ legal bajo F.

r

= {luego de paso k descomposición de reunión sin pérdida}

$\Pi_R\ (\ \Pi_{R1}(r) \bowtie \ldots \bowtie \Pi_{Rj}(r) \bowtie \ldots \bowtie \Pi_{Rk+1}(r))$

= {{$R_j - \delta$, (γ, δ)}} de reunión sin pérdida}

$\Pi_R\ (\ \Pi_{R1}(r) \bowtie \ldots \bowtie \Pi_{Rj\,-\,\delta}\ (\Pi_{Rj}(r)\ ) \bowtie \Pi_{(\gamma,\,\delta)}(\Pi_{Rj}(r)\ ) \bowtie \ldots \bowtie \Pi_{Rk+1}(r))$

= {Π_A ($\Pi_B(s) = \Pi_A(s)$ cuando $A \subseteq B$}

$\Pi_R\ (\ \Pi_{R1}(r) \bowtie \ldots \bowtie \Pi_{Rj\,-\,\delta}(r) \bowtie \Pi_{(\gamma,\,\delta)}(r) \bowtie \ldots \bowtie \Pi_{Rk+1}(r))$

Por lo tanto, luego del paso de iteración $k+1$ se obtiene una descomposición de reunión sin pérdida.

Algoritmo de normalización en FNBC

- **Ejercicio**: Aplicar el algoritmo de normalización en FNBC a:
  - R = (A, B, C, D, E, F)
  - F = {A → CB, E → FA}

Algoritmo de normalización en FNBC

- **Ejercicio**: Sea el esquema universal:

  BibLibs = (nomBib, calle, número, numInv, ISBN, título, editorial, autores, edición)

  Sea F dado por:

  – nomBib → calle, número
  – calle, número → nomBib
  – ISBN → título, editorial, autores, edición
  – nomBib, numInv → ISBN

  Aplicar el algoritmo de normalización en FNBC.

FNBC y preservación de dependencias

- No es siempre posible obtener una descomposición en FNBC que preserva las dependencias.
- **Ejemplo**:
  - $R = (J, K, L)$
    $F = \{JK \rightarrow L, L \rightarrow K\}$
    Hay dos claves candidatas: JK y JL
  - R no está en FNBC.
  - Toda descomposición de R falla en preservar:

$$JK \rightarrow L$$

Tercera forma normal

- **Problema**: Hay algunas situaciones donde
  - usar FNBC no preserva las dependencias, y
  - el chequeo eficiente de violaciones de DFs en actualizaciones es importante.
- **Solución**: definir una forma normal más débil, llamada tercera forma normal.
  - Permite alguna redundancia de información.
  - Pero las DFs pueden ser chequeadas en relaciones individuales sin computar reuniones naturales.
  - Hay siempre una descomposición en 3FN que es de reunión sin pérdida y que preserva las dependencias.

Tercera forma normal

- **Definición**: Un esquema *R* está en **tercera forma normal** (3FN) si para todas las DF:

  $$\alpha \rightarrow \beta \in F^+$$

  al menos una de las siguientes condiciones se cumple:
  - $\alpha \rightarrow \beta$ es trivial (i.e., $\beta \subseteq \alpha$)
  - α es superclave para *R* (i.e. $\alpha \rightarrow R \in F^+$).
  - Cada **atributo** *A* en $\beta - \alpha$ está contenido en una clave candidata de ***R***.

Tercera forma normal

- **Nota***:* cada atributo puede estar en una clave candidata diferente
- **Nota**: γ clave candidata de R si y solo si
  - $\gamma \rightarrow R \in F^+$, y
  - $\forall\ C \in \gamma : \neg\ (\gamma - \{C\}) \rightarrow R \in F^+$
- Si un esquema está en FNBC, entonces está en 3FN.
- La tercera condición es una relajación mínima de FNBC para garantizar preservación de las dependencias.

Tercera forma normal

- **Ejercicio**: Sea el esquema relacional *R* = (J,K,L) con DFs:

  $F = \{JK \rightarrow L, L \rightarrow K\}$.

  Probar que *R* está en 3FN.

- Se permite redundancia de información:

J	K	L
j_1	l_1	k_1
j_2	l_1	k_1
j_3	l_1	k_1
null	l_2	k_2

Tercera forma normal

- Repetición de la información
  - Ejemplo: relación $l_{1,}$ k_1
- Necesidad de usar valores nulos
  - Ejemplo: para representar la relación $l_{2,}$ $k_{2,}$ donde no hay un valor correspondiente para j).

Tercera forma normal

- **Ejercicio**: Sea R = (I, S, C, D, A, O), con las DFs:

$$F = \{S \rightarrow D;\ I \rightarrow A;\ IS \rightarrow C;\ A \rightarrow O\}$$

  - Sea R_1 = (I, S, C, D),
  - ¿está R_1 en 3FN? Justifique su respuesta.

Tercera forma normal

- **Proposición**: Para comprobar si R_U, F, R_U esquema universal está en 3FN, basta con comprobar las DFs de F.
  - Además se pueden descomponer las DFs de F de modo que sus lados derechos consistan solo de atributos sencillos y utilizar el conjunto resultante en lugar de F.

Tercera forma normal

- Usar clausuras de atributos para chequear para cada DF $\alpha \rightarrow \beta$, si α es una superclave.
- Si α no es una superclave, tenemos que verificar si cada atributo en β está contenido en una clave candidata de *R*
  - Esta prueba es bastante más cara, porque involucra encontrar claves candidatas.
  - Probar 3FN ha sido probado que es **NP-hard**.
  - Sin embargo, la descomposición en 3FN puede ser hecha en tiempo polinomial.

Algoritmo de normalización en 3FN

- Let F_c be a canonical cover for F;
  $i := 0$;
  for each functional dependency $\alpha \to \beta$ in F_c **do**
  if none of the schemas R_j, $1 \leq j \leq i$ contains $\alpha\ \beta$
  then begin
  $i := i + 1$;
  $R_i := \alpha\ \beta$
  end
  if none of the schemas R_j, $1 \leq j \leq i$ contains a candidate key for R
  then begin
  $i := i + 1$;
  $R_i :=$ any candidate key for R;
  end
  return $(R_1, R_2, \ldots, R_i)$

Algoritmo de normalización en 3FN

- **Ejercicio**: Sea el esquema universal:
  - BibLibs = (nomBib, calle, número, numInv, ISBN, título, editorial, autores, edición)

  Sea *F* dado por:

  1. nomBib → calle, numero
  2. calle, numero → nomBib
  3. ISBN → título, editorial, autores, edición
  4. nomBib, numInv → ISBN

  Descomponer BibLibs en 3FN.

Corrección del algoritmo de descomposición en 3FN

- El algoritmo de descomposición en 3FN garantiza la preservación de las dependencias,
  - debido a que hay un esquema para cada DF en F_c.
- La descomposición obtenida es de reunión sin pérdida.
  - Una clave candidata está en uno de los esquemas R_i de la descomposición.
  - Ejercicio de la práctica.

Corrección del algoritmo de descomposición en 3FN

- Si un esquema R_i está en la descomposición generada por el algoritmo anterior, entonces R_i satisface 3FN.
  - Sea R_i generado por la DF $\alpha \to \beta$
  - Sea $\gamma \to$ B una DF no trivial en R_i. (Necesitamos solo considerar DFs cuya partes derechas tienen un solo atributo)
  - *B* puede estar en β o α pero no en ambos.
    - Consideramos cada caso por separado.

Corrección del algoritmo de descomposición en 3FN

- **Caso 1**: Si $B \in \beta$:
  - If γ is a superkey, the 2nd condition of 3NF is satisfied
  - Otherwise α must contain some attribute not in γ
  - Since $\gamma \rightarrow$ B is in F^+ it must be derivable from F_c, by using attribute closure on γ.
  - Attribute closure not have used $\alpha \rightarrow \beta$ - if it had been used, α must be contained in the attribute closure of γ, which is not possible, since we assumed γ is not a superkey.
  - Now, using $\alpha \rightarrow$ (β- {B}) and $\gamma \rightarrow$ B, we can derive $\alpha \rightarrow B$

    (since $\gamma \subseteq \alpha\ \beta$, and B $\notin \gamma$ since $\gamma \rightarrow B$ is non-trivial)
  - Then, *B* is extraneous in the right-hand side of $\alpha \rightarrow \beta$; which is not possible since $\alpha \rightarrow \beta$ is in F_c.
  - Thus, if *B* is in β then γ must be a superkey, and the second condition of 3NF must be satisfied.

Corrección del algoritmo de descomposición en 3FN

- **Caso 2**: $B \in \alpha$.
  - Debido a que α es una clave candidata, se satisface trivialmente la tercera alternativa en la definición de 3FN.
  - De hecho, no podemos probar que γ es una superclave.
  - Esto muestra exactamente porqué la tercera alternativa está presente en la definición de la 3FN.

Q.E.D.

Comparison of BCNF and 3NF

- Es siempre posible descomponer un esquema en esquemas en 3FN y
  - La descomposición es de reunión sin pérdida.
  - Las dependencias son preservadas.
- Es siempre posible descomponer un esquema en FNBC y
  - La descomposición es de reunión sin pérdida.
  - Puede no ser posible preservar las dependencias.