

Capítulo 5 - Normalización

Proceso de realización de diseño de BD relacional

- Identificar K (conjunto de todos los **atributos atómicos** del problema actual)
- A partir de K y para el problema actual definir un conjunto de **restricciones de integridad** I (dependencias funcionales)
 - Servirán de guía para hacer un buen diseño
- Aplicar el algoritmo de **normalización**
 - Calcula un esquema de BD relacional a partir de K e I

Atributos atómicos (esquema universal)

- Dado el problema actual, el **esquema universal** consta de todos los atributos atómicos del mismo
 - **Ejemplo:** sistema de bibliotecas
 - Se tiene un *sistema de bibliotecas* de una ciudad, el sistema está formado por *bibliotecas* de las que se proveen su nombre, y su domicilio formado por calle y número; las bibliotecas tienen libros de los que se almacena su ISBN, su título y sus autores; las bibliotecas llevan un número de inventario para distinguir entre las distintas copias de libros; además las bibliotecas tienen *socios* para los que se almacena nombre, DNI y posición (la misma puede ser egresado, docente, estudiante, etc.); a los socios se les *prestan* ejemplares de libros.
 - $SmaBibliotecas = (nombre, DNI, posición, numInv, nombreBib, calle, numero, ISBN, título, autores)$
 - **Significado:** Una tupla de información para una relación de ese esquema para un sistema de bibliotecas consiste de: datos sobre una persona (correspondientes a *nombre, DNI, posición*) datos sobre un libro (correspondientes a *numInv, ISBN, título, autores*), datos sobre una biblioteca (correspondientes a *calle, número, nomBib*), tal que esa persona es socia de esa biblioteca, y se le ha prestado ese libro que es de esa biblioteca.
- Es un esquema de BD relacional pero no conviene usarlo como tal ya que usualmente tiene problemas de calidad como
 - *Redundancia de información*
 - *Manejo de valores nulos*

Descomposición del esquema universal

- Para **solucionar los problemas**, se puede *descomponer* el esquema universal
- **Ejemplo:** Descomponemos el esquema
 - SmaAutomotor = (DNI, nombre, marca, modelo, patente, numSeguro, compañíaSeguro, direcciónCS):
 - en:
 - Persona = (DNI, nombre)
 - Auto = (marca, modelo, patente)
 - CompañíaAseguradora = (compañíaSeguro, direcciónCS)
 - Seguro = (patente, compañíaSeguro, numSeguro)
 - TieneAuto = (patente, DNI)
- Para esto se usan *algoritmos de normalización* que descomponen esquemas universales para evitar los problemas citados
 - No se necesita ser bueno en modelado / diseño, pero sí encontrando restricciones de integridad y el esquema universal
 - Un algoritmo de normalización va descomponiendo un esquema universal hasta obtener un esquema de BD relacional de calidad
 - Sea R un esquema de relación, un conjunto de esquemas de relación $\{R_1, \dots, R_n\}$ es una descomposición de $R \Leftrightarrow R = \bigcup_{i=1}^n R_i$
- **Receta:** arreglar problemas de diseño introduciendo esquemas para conceptos varios que dependen de conocimiento del dominio y que son una descomposición del esquema universal
 - Recordar que modelado ER consideraba la introducción de conceptos varios y, por lo tanto, si se hace bien se obtendrá un buen diseño relacional luego de pasar a tablas
 - La diferencia con modelado ER es que aquí analizamos problemas de diseño y los sacamos uno a uno
- **Meta:** lo mencionado anteriormente pero queremos que la descomposición se haga automáticamente y que no dependa de un conocedor del dominio que hace uso inteligente del mismo
- Para realizar esta descomposición, se *generaliza* la aplicación de los siguientes pasos:
 - El esquema R tiene atributos redundantes J
 - Encontramos propiedad del tipo $K \rightarrow J$
 - Usamos $K \rightarrow J$ para descomponer R en $K \cup J$ y $R_1 = R - J$
 - **A las propiedades del tipo $\alpha \rightarrow \beta$, con α y β conjuntos de atributos, se les llama dependencias funcionales**
 - Los atributos de la derecha de la DF se sacan del esquema que descompongo
 - Se agrega un esquema para los atributos de TODA la DF

Dependencias funcionales (restricciones de integridad)

- Como se mencionó antes, son todas las propiedades del tipo $\alpha \rightarrow \beta$, con α y β conjuntos de atributos
- Hay que definir las restricciones de integridad para el *conjunto de tablas legales*
 - Tablas con las que la empresa quiere poder trabajar
 - Donde las tuplas tienen un cierto significado y cumplen con ciertas propiedades obligatorias

- Las **DF** requieren que para estas tablas, el valor de cierto conjunto de atributos determine *unívocamente* el valor de otro conjunto de atributos
- **Formalización**
 - Sea R un esquema relacional ($\alpha, \beta \subseteq R$), la DF $\alpha \rightarrow \beta$ se cumple en R si y solo si para todas las relaciones legales $r(R)$, cada vez que dos tuplas t_1, t_2 de r coinciden en los atributos α , también lo hacen en β ($t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$)
- **Derivación de DF !!!**
 - *Axiomas de Armstrong*
 - **Reflexividad:** $\beta \subseteq \alpha \Rightarrow \alpha \rightarrow \beta$
 - **Aumentatividad:** $\alpha \rightarrow \beta \Rightarrow \gamma\alpha \rightarrow \gamma\beta$
 - **Transitividad:** $(\alpha \rightarrow \beta \wedge \beta \rightarrow \gamma) \Rightarrow \alpha \rightarrow \gamma$
 - **Dedución:** dado un esquema relacional R , una DF f con atributos en R se deduce de un conjunto de DFs F con atributos en R si existe una lista de DFs f_1, \dots, f_n tales que $f_n = f$ y $\forall 1 \leq i \leq n$
 - $f_i \in F$ ó
 - f_i se obtiene por aplicar la regla de reflexividad ó
 - f_i se obtiene por aplicar aumentatividad o transitividad a pasos anteriores en la lista
 - **Notación:** usaremos $F \vdash f$ para decir que f se deduce de F
Lisa y llanamente: con aplicación de los axiomas se puede llegar desde F a f :)
 - *Reglas adicionales*
 - **Unión:** $(\alpha \rightarrow \beta \wedge \alpha \rightarrow \gamma) \Rightarrow \alpha \rightarrow \beta\gamma$
 - **Descomposición:** $\alpha \rightarrow \beta\gamma \Rightarrow (\alpha \rightarrow \beta \wedge \alpha \rightarrow \gamma)$
 - **Pseudotransitividad:** $(\alpha \rightarrow \beta \wedge \gamma\beta \rightarrow \delta) \Rightarrow \alpha\gamma \rightarrow \delta$
- **Cierre de un conjunto de DF**
 - El cierre de un conjunto de DF F (F^+) son *todas* las dependencias que se deducen de F
 - ¿Cómo saber si vale la pena agregar DF f a F ?
 - En particular, si $f \in F^+$, no vale la pena pero es **muy** costoso calcular F^+ (no es viable)
 - Hay que ver si $F \vdash \alpha \rightarrow \beta$. Por ello, calculemos las DF de F^+ con lado izquierdo α . Luego, solo bastaría con decir que $\alpha \rightarrow \beta \in \{\alpha \rightarrow \phi \mid F \vdash \alpha \rightarrow \phi\}$
 - Otra forma, sin embargo, es usando el **cierre de conjunto de atributos** dado que queremos ver que $\beta \subseteq \{A \in R \mid F \vdash \alpha \rightarrow A\}$ con A atributo (dado que aplicando regla de unión varias veces llegamos a que $F \vdash \alpha \rightarrow \beta$)
- **Cierre de conjunto de atributos**
 - Sea R el esquema universal, F el conjunto de DF del problema, $\alpha \subseteq R$, el *cierre de α bajo F* (α_F^+) se define: $\alpha_F^+ = \{A \in R : F \vdash \alpha \rightarrow A\}$
 - Propiedad: $F \vdash \alpha \rightarrow \alpha_F^+$
 - ¿Cómo saber con esto si vale la pena agregar $\alpha \rightarrow \beta$ a F ?
 - Notemos que $F \vdash \alpha \rightarrow \beta \Leftrightarrow \beta \subseteq \alpha_F^+$
 - Luego, si $B \subseteq \alpha_F^+$ entonces no lo agregamos. Caso contrario, sí porque no se deduce de F .
 - ¿Cómo es el algoritmo para computar α_F^+ ?

```

result := a
while (changes to result)
    for each b -> y in F
        if b subset of result
            result := result union with y

```

- Usos de este conjunto para superclave y clave candidata
 - α es **superclave** de R si y solo si $\alpha \rightarrow R \in F^+$
 - Para ello, se puede computar α_F^+ y chequear si este contiene todos los atributos de R
 - α es **clave candidata** de R si y solo si es *superclave* de R y $\forall A \in \alpha, \alpha - \{A\}$ no es superclave de R (i.e., es minimal)

Normalización

- **Repaso de lo anterior**
 - Si R es un esquema con redundancia de información en un conjunto de atributos β y la DF $\alpha \rightarrow \beta$ (α, β disjuntos) se cumple en R :
 - Si α no determina todos los atributos de R , entonces la DF puede ser usada para eliminar redundancia de información por medio de la descomposición de R
 - Se saca β de R y se crea un esquema con los atributos de $\alpha \cup \beta$
 - Si hay 2 tuplas distintas de R que coinciden en α y vale $\alpha \rightarrow \beta$ y α no determina todos los atributos de R , entonces para esas tuplas se van a repetir los valores de β , por lo que tenemos redundancia
 - Sea R y F (conjunto de DF), que α no determine todos los atributos de R es lo mismo que decir alguna:
 - $F \not\models \alpha \rightarrow R$
 - α no es superclave de R
- **Forma normal de Boyce Codd (FNBC)**
 - Esquema R que no tiene redundancia de información proveniente de DF por medio de una propiedad
 - **Comprobación de si un esquema está o no en FNBC**
 - Un esquema está en FNBC con respecto a un conjunto F de DFs si para todas las DFs en F^+ de la forma $\alpha \rightarrow \beta$, donde $\alpha, \beta \subseteq R$, al menos una de las siguientes se cumple
 - $\alpha \rightarrow \beta$ es trivial (i.e., $\beta \subseteq \alpha$)
 - α es una superclave de R (i.e., $\alpha \rightarrow R \in F^+$)
 - !!! *Comprobar que no está en FNBC*: se busca una DF de F^+ que no cumple la condición (se dice *violación* o *DF testigo*) \rightarrow a veces, pero no siempre, está en F
 - Sea R esquema universal, F conjunto de DFs; entonces una descomposición $\{R_1, \dots, R_n\}$ de R está en FNBC con respecto F si y solo si cada R_i está en FNBC con respecto a F
 - **Propiedades**
 - Para comprobar si R, F está en FNBC, donde los atributos de F están contenidos en R (!!! importante esta propiedad), basta con chequear la condición de FNBC para las DF de

F (hay que ver que ninguna sea testigo)

- *Consecuencia*: para descomponer el esquema universal, basta con buscar testigos en F
- Sea R_u universal con DFs F y sea R_i que forma parte de descomposición de R_u ; para probar que R_i está en FNBC se puede hacer la siguiente comprobación:
$$\forall \alpha \subseteq R_i, (a^+ \cap (R_i - \alpha) = \phi) \vee R_i \subseteq a^+$$

(la primera condición del \vee implica que todas las DF con lado izquierdo α son triviales)
 - En caso que no se cumpla, y sea $\alpha \subseteq R_i$ el que viola la condición, entonces el DF testigo es $\alpha \rightarrow \alpha^+ \cap (R_i - \alpha)$
 - Usamos esta DF para descomponer R_i

Algoritmo de normalización en FNBC

```
result := {R};

while (hay un esquema R_i en result que no está en FNBC) {
    a -> b := DF testigo de R_i y a ∩ b = ∅
    result := (result - R_i) ∪ (R_i - b) ∪ (a, b)
}
```

- **Aclaraciones importantes**

- Para encontrar la DF testigo, usar el algoritmo de comprobación de que un esquema está en FNBC (si no se encuentra, entonces se termina el algoritmo de normalización)
 - Usar la condición $\forall \alpha \subseteq R_i, (a^+ \cap (R_i - \alpha) = \phi) \vee R_i \subseteq a^+$