

# Capítulo 5

## Normalización Introducción

# Diseño de BD Relacionales

- **Meta:** aprender a hacer **diseños de calidad** de esquemas de BD relacionales, idealmente de manera automática.
- **Repaso:**
  - Un esquema relacional es un conjunto de esquemas de relaciones.
  - Un esquema de relación es una lista de nombres de atributos.

# Diseño de BD Relacionales

- **Para tener un diseño de calidad:**
  - Evitar problemas de redundancia de información.
  - Evitar problemas de comprensibilidad.
  - Evitar problemas de incompletitud.
    - Restricciones de integridad incompletas.
    - Relaciones entre atributos no contempladas por esquemas de BD.
  - Evitar problemas de ineficiencia.
    - Chequeo ineficiente de restricciones de integridad.
    - Consultas ineficientes por tener un esquema inadecuado de BD.

# Diseño de BD Relacionales

- **Enfoque ya visto para lograr esta meta**
  - Hacer buen *diseño de esquema-ER*.
  - Mapear esquema-ER a un esquema relacional.
  - **Dificultades:**
    - Es un trabajo **manual**
    - **Corregir** esquemas-ER con problemas de calidad.
    - **Tomar decisiones de diseño:**
      - Considerar alternativas suficientes de diseño
      - Elegir entre las alternativas la mejor basándose en **criterios de calidad**.

# Diseño de BD Relacionales

- **Problema** ¿Cómo evitar tener que diseñar diagrama ER, tener que decidir entre alternativas de diseño ER, tener que corregir problemas de calidad en diseños ER?

# Diseño de BD Relacionales

- **Solución:**
  - Identificar  $K$  el conjunto de todos los atributos (atómicos) del problema actual.
  - Luego a partir de  $K$  y para el problema actual definir un conjunto de **restricciones de integridad  $I$** ,
    - que servirán de guía para hacer un buen diseño.
  - Se aplica un algoritmo llamado de **normalización** que:
    - Calcula un esquema de BD relacional a partir de  $K$  y de  $I$ .

# Diseño de BD Relacionales

Atributos Problema Actual y Restricciones Integridad

Atributos atómicos

Dependencias funcionales

Algoritmo de Normalización

Esquema de BD Relacional de Calidad

# Esquema Universal

- **Situación:** para aplicar un algoritmo de normalización necesitamos conocer primero los **atributos** del problema actual.
- Dado el problema actual, el **esquema universal** consta de todos los atributos atómicos del mismo.

# Esquema Universal

- **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.
  - ¿Cuál es el esquema universal para este problema?

# Esquema Universal

- **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)
  - **¿Cuál es el significado de una tupla de una relación para el esquema universal?**

# Esquema Universal

# Esquema Universal

- Ya que el esquema universal es un esquema de BD relacional,
  - **¿conviene usarlo como diseño de una BD relacional?**

# Esquema Universal

- Ya que el esquema universal es un esquema de BD relacional,
  - **¿conviene usarlo como diseño de una BD relacional?**
  - **Respuesta:** **no**, porque el esquema universal usualmente tiene problemas de calidad, como redundancia de información.

# Esquema Universal

- Sea el siguiente esquema universal:
  - SmaAutomotor = (DNI, nombre, marca, modelo, patente, numSeguro, compañíaSeguro, direcciónCS)
- **Problemas de diseño:**
  - **Redundancia de información:** *direcciónCS* aparece repetido para cada coche asegurado por esa compañía de seguros; *nombre* aparece repetido por cada coche que tiene esa persona.
  - **Manejo de valores nulos:** si una persona no tiene coche, aparecen los demás campos en nulo; si una compañía de seguro no tiene coches asegurados, aparecen los demás campos en nulo.
    - Antes de decidir si agregamos valores nulos, tenemos que hacer consultas.
    - P.ej. si a una persona le sacamos coche asegurado, hay que consultar si tiene otros coches asegurados, sino hay que poner nulos.

# Esquema Universal

- **Problema:** *¿Cómo se pueden eliminar los problemas citados?*
- **Solución:** *descomponer* el esquema universal para eliminar los problemas citados.

# 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)

# Problemas de usar esquema universal

- La **teoría de normalización** estudia cómo descomponer esquemas universales para eliminar los problemas citados.
  - **Implicaciones de usar normalización:**
    - no hace falta ser bueno en modelado o diseño;
    - *pero hay que ser bueno encontrando las restricciones de integridad necesarias (i.e. dependencias funcionales) y el esquema universal.*

# Descomposiciones

- **Motivación:** 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  $R$  si

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

# Ejemplo

- Sea el siguiente esquema relacional:
  - SmaBibliotecas = (nombre, DNI, posición, numInv, nombreBib, calle, numero, ISBN, título, autores)
- **Señalar redundancias de información**

# Ejemplo

- Sea el siguiente esquema relacional:
  - SmaBibliotecas = (nombre, DNI, posición, numInv, nombreBib, calle, numero, ISBN, título, autores)
- **Podemos identificar los siguientes casos de redundancia de información:**
  - el mismo usuario saca varios libros: se repite nombre y posición varias veces
  - para cada persona que es socia de biblioteca se repite calle y número
  - el mismo libro se saca varias veces prestado y se repite título y autores
- **Indicar situaciones que obligan a trabajar con valores nulos**

# Ejemplo

- **Situaciones que obligan a trabajar con valores nulos:**
  - Si tengo una *persona* solamente (no es socia, no le prestaron libros, etc.) entonces los campos van a ser nulos;
  - lo mismo si tengo *libro* solamente, los demás campos van a ser nulos.
- **Lo malo de trabajar con valores nulos:**
  - Si borro una tupla de tabla del esquema para eliminar una asociación entre entidades, hay que tener cuidado de no eliminar entidades que deben permanecer.
  - Esto obliga a hacer consulta extra y posible modificación de tupla agregando valores nulos.

# Ejemplo

- **Problema:** ¿Cómo arreglar problemas de redundancia de información?
  - **Idea:** puedo ir descomponiendo gradualmente el esquema universal de acuerdo a conceptos que sirven para sacar atributos que ocasionan redundancia de información afuera.
  - $\text{SmaBibliotecas} = (\text{nombre}, \text{DNI}, \text{posición}, \text{numInv},$   
 $\text{nombreBib}, \text{calle}, \text{numero}, \text{ISBN}, \text{título}, \text{autores})$
  - **¿Cómo eliminar la redundancia: el mismo usuario saca varios libros: se repite nombre y posición varias veces?**

# Ejemplo

- **Problema:** ¿Cómo arreglar problemas de redundancia de información?
- **Idea:** puedo ir descomponiendo gradualmente el esquema universal de acuerdo a conceptos que sirven para sacar atributos que ocasionan redundancia de información afuera.
- Consideremos el esquema anterior:

SmaBibliotecas = (nombre, DNI, posición, numInv,  
                          nombreBib, calle, numero, ISBN, título, autores)

- Persona = (nombre, DNI, posición)
- SmaBibliotecas1 = (DNI, numInv, nombreBib, calle, numero, ISBN, titulo, autores)
- Saqué de SmaBibliotecas: nombre y posición eliminando la primera redundancia de información.

# Ejemplo

- ¿Cómo eliminar la redundancia de información: para cada persona que es socia de biblioteca se repite calle y número?
- SmaBibliotecas1 = (DNI, numInv, nombreBib, calle, numero, ISBN, titulo, autores)

# Ejemplo

- ¿Cómo eliminar la redundancia de información: para cada persona que es socia de biblioteca se repite calle y número?
- SmaBibliotecas1 = (DNI, numInv, nombreBib, calle, numero, ISBN, titulo, autores)
- Sea el esquema:
- Biblioteca = (nombreBib, calle, número)
- SmaBibliotecas2=(DNI, numInv, nombreBib, ISBN, título, autores)
- Saqué de SmaBibliotecas1: calle y número.

# Ejemplo

- **¿Cómo eliminar la redundancia de información: el mismo libro se saca varias veces prestado y se repite título y autores?**
- SmaBibliotecas2=(DNI, numInv, nombreBib, ISBN, título, autores)

# Ejemplo

- ¿Cómo eliminar la redundancia de información: el mismo libro se saca varias veces prestado y se repite título y autores?
- SmaBibliotecas2=(DNI, numInv, nombreBib, ISBN, título, autores)
- Sea el esquema:
- Libro = (ISBN, título, autores)
- SmaBibliotecas3 = (DNI, numInv, nombreBib, ISBN)
- Saqué de SmaBibliotecas2: título y autores.

# Observaciones

- **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 de modelado ER con lo que hicimos aquí es que hoy analizamos problemas de diseño y los fuimos eliminando uno a uno con conceptos que son introducidos.

# Observaciones

- **Meta:** Queremos que la descomposición salga automáticamente y no que dependa de un conocedor del dominio que hace uso inteligente del mismo.
  - Esto se torna muy trabajoso si el esquema universal tiene cientos de atributos, cosa que suele pasar en la vida real.

# Ejemplo

- **Ejemplo:** SmaBibliotecas = (nombre, DNI, posición, numInv,  
nombreBib, calle, numero, ISBN, título, autores)
- Tiene atributos redundantes *nombre* y *posición*
- **El valor de DNI determina únicamente los valores de *nombre* y *posición*** y lo indicamos de la siguiente forma: DNI -> nombre, posición.
- Usamos DNI -> nombre, posición para descomponer SmaBibliotecas en:
  - SmaBibliotecas1 = (DNI, numInv, nombreBib, calle, número, ISBN, título, autores)
  - (DNI, nombre posición)

# Ejemplo

- SmaBibliotecas1 = (DNI, numInv, nombreBib, calle, numero, ISBN, titulo, autores)
- Tiene atributos redundantes *calle* y *número*.
- ¿Qué propiedad podemos usar con esos atributos a la derecha de  $\rightarrow$ ?

# Ejemplo

- $SmaBibliotecas1 = (\text{DNI}, \text{numInv}, \text{nombreBib}, \text{calle}, \text{número}, \text{ISBN}, \text{titulo}, \text{autores})$
- Tiene atributos redundantes *calle* y *número*.
- $\text{nombreBib} \rightarrow \text{calle}, \text{número}$
- Usamos  $\text{nombreBib} \rightarrow \text{calle}, \text{número}$  para descomponer *SmaBibliotecas1* en:
  - (nombreBib, calle, número)
  - $SmaBibliotecas2=(\text{DNI}, \text{numInv}, \text{nombreBib}, \text{ISBN}, \text{título}, \text{autores})$

# Ejemplo

- SmaBibliotecas2=(DNI, numInv, nombreBib, ISBN, título, autores)
- Tiene atributos redundantes título y autores.
- **¿Qué propiedad podemos usar con esos atributos a la derecha de ->?**

# Ejemplo

- SmaBibliotecas2=(DNI, numInv, nombreBib, ISBN, título, autores)
- Tiene atributos redundantes *título* y *autores*.
- ISBN -> título, autores
- Usamos ISBN -> título, autores para descomponer SmaBibliotecas2:
  - (ISBN, título, autores)
  - SmaBibliotecas3 = (DNI, numInv, nombreBib, ISBN)

# Observaciones

- Generalizando, aplicamos 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  $\alpha$  y  $\beta$  conjuntos de atributos se las llama **dependencias funcionales**.

# Observaciones

- **Fijarse que hemos seguido el siguiente procedimiento:**
  - los atributos de la derecha de la DF se sacan del esquema que descompongo y
  - agrego un esquema para los atributos de toda la DF.
- Usualmente puedo ir descomponiendo el esquema universal para eliminar mucha de la redundancia de información en el mismo por medio del uso de dependencias funcionales.
- Más aun, si tenemos las DF del problema, esto es automatizable (debido a la forma repetitiva de descomponer esquemas).

# Dependencias Funcionales

- Hay que definir las restricciones de integridad para el *conjunto de tablas legales*.
  - **Tablas legales** son las tablas con las que la empresa quiere poder trabajar.
  - Son tablas donde las tuplas tienen un cierto significado y cumplen con ciertas propiedades obligatorias.
- Las **dependencias funcionales (DF)** requieren que para las tablas legales,
  - el valor de un cierto conjunto de atributos determine únicamente el valor de otro conjunto de atributos.

# Dependencias Funcionales

- **Formalización:**

- Sea  $R$  un esquema relacional

$$\alpha \subseteq R \text{ y } \beta \subseteq R$$

- La dependencia funcional

$$\alpha \rightarrow \beta$$

**se cumple en**  $R$  si y solo si para todas las relaciones legales  $r(R)$ , cada vez que dos tuplas  $t_1$  y  $t_2$  de  $r$  coinciden en los atributos  $\alpha$ , también coinciden en los atributos  $\beta$ . Formalmente:

$$t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$$

# Dependencias Funcionales

- ¿Las DF solo pueden ser usadas para eliminar redundancia de información?
- Ahora veremos ejemplos que la respuesta es no.

# Dependencias Funcionales

- **Ejemplo:** Anteriormente luego de eliminar redundancia de información en *SmaBibliotecas* llegamos al esquema:

$\text{SmaBibliotecas3} = (\text{DNI}, \text{numInv}, \text{nombreBib}, \text{ISBN})$

- Pero este esquema tiene problema de nulos porque puede haber una persona que no tenga libros prestados,
  - En ese caso vamos a necesitar tener nulos en *numInv* e *ISBN* para esa persona.
- Pero tenemos la dependencia funcional:  $\text{numInv}, \text{nombreBib} \rightarrow \text{ISBN}$ .
- Descomponemos SmaBibliotecas3 en:
- $(\text{DNI}, \text{numInv}, \text{nombreBib})$  y  $(\text{numInv}, \text{nombreBib}, \text{ISBN})$
- Desaparece usar nulos en *ISBN*.
- Usamos una dependencia funcional para reducir el problema de usar valores nulos (Aunque no lo pudimos eliminar completamente).

# Dependencias Funcionales

- **Ejemplo:** Sea el esquema universal de un banco.

*Préstamo = (numSucursal, ciudad, activo, numCliente,  
numPréstamo, importe)*

- Lo descomponemos en:

*SucursalCliente = (numSucursal, ciudad, activo, numCliente)*

*ClientePréstamo = (numCliente, numPréstamo, importe)*

- **Evaluación de la descomposición:** Si tenemos un cliente con varios préstamos en distintas sucursales:

- no se puede decir el *préstamo* que pertenece a cada sucursal en la descomposición que tenemos.
- Luego en la descomposición que tenemos se perdió información con relación al esquema universal.

# Dependencias Funcionales

- Aquí se pierde una relación entre atributos del problema al descomponer.
  - Esto es algo grave y hay que evitarlo.
- El problema del ejemplo anterior se resuelve así:
- numSucursal  $\rightarrow$  ciudad, activo
- Uso esta dependencia funcional para descomponer *préstamo* en:
- Sucursal = (numSucursal, ciudad, activo)
- Prest = (numSucursal, numCliente, numPrestamo, importe)
- Así resolvimos el problema usando una DF para descomponer el esquema *préstamo*.

# Dependencias Funcionales

- **Observaciones:**
  - Dado esquema  $R$  universal, suele haber dependencias funcionales con atributos en  $R$  donde los atributos a la derecha son de información redundante.
    - Cada una de esas dependencias se puede usar para hacer un paso automático de descomposición de  $R$  que elimina la redundancia de información.
    - No lo dijimos antes, pero en cada una de esas dependencias los atributos a la izquierda no determinan todos los atributos del esquema  $R$ .
  - Recordar que las DF se usan también para tratar con otros problemas como uso de valores nulos y preservar relaciones entre atributos del problema en la descomposición.