

# Escuela de Sistemas y Tecnologías

Transparencias de ANALISTA DE SISTEMAS  
*Edición 2013 – Materia:*  
*Diseño e Implementación de BdeD*

TEMA: Modelo Relacional - Normalización

## Modelo Relacional - Definición

- El Modelo Relacional se ha establecido como el principal modelo de datos para las aplicaciones de procesamiento de datos.
- Establece la teoría para las bases de datos relacionales que son las que se emplean habitualmente para almacenamiento de datos.
- El modelo relacional considera una tabla o una relación como conjunto de tuplas (también llamadas registros) y operaciones asociadas. Asimismo, cada tupla se compone de atributos o campos.

## Agenda

- Modelo Relacional
  - Restricciones
  - Dependencias Funcionales
  - Axiomas de Armstrong
  - Estudio de Superclaves
- Normalización
  - Redundancia de Datos
  - Tipos Dependencias Funcionales
  - Formas Normales
  - Descomposición – Join sin Perdida

## Restricciones (1)

- Proporcionan un conjunto de condiciones que deben cumplir los datos que se almacenan en las tablas.
- **Cardinalidad:** relaciona la cantidad de registros entre dos tablas. Establece cuántos registros de una tabla se relacionan con un registro de la otra tabla vinculada.
- **Dominio:** cada atributo o campo tiene asociado un conjunto de valores válidos.
- **Integridad referencial:** establece que no puede agregarse ni borrarse ningún valor de una tabla, sin que esos cambios se reflejen en la tabla con la que esté relacionada.

## Restricciones (2)

- **Reglas de validación:** establece una condición que debe cumplir un determinado dato para que se permita su aceptación.
- **Superclave:** conjunto de atributos que identifican en forma única a cada registro de la tabla.
- **Clave:** conjunto minimal que identifica en forma única a cada registro de la tabla.
- **Clave primaria (primary key):** clave candidata que se elige como identificador de registros de la tabla.
- **Clave externa (foreign key):** conjunto de atributos de una tabla que se relacionan con otra. Los valores posibles de los atributos que conforman una clave externa deben estar en la tabla relacionada.

5

## Dependencias Funcionales (1)

- Restricción que indica una determinada relación entre los valores que pueden tomar los atributos de una tabla.
- En una tabla R se cumple la dependencia funcional  $A \rightarrow C$  si para cada ocurrencia (o valor) de A en R existe un mismo valor de C también en R.
- Se dice que A **determina** C o C **depende funcionalmente** de A.

6

## Dependencias Funcionales (2)

- Por ejemplo sea la tabla R(A, B, C, D) con los atributos A, B, C y D como indica la siguiente figura:

registros →

A	B	C	D
A1	B1	C1	D1
A1	B2	C1	D2
A2	B2	C2	D2
A2	B3	C2	D3
A3	B3	C2	D4

- Para cada valor de A existe un único valor de C.
- Por el contrario, la dependencia funcional  $C \rightarrow A$  **no** se cumple, dado que para el valor de C2 existen dos valores de A (A2 y A3).

7

## Axiomas de Armstrong (1)

- Sea F el conjunto de dependencias funcionales de una tabla. ( $F = \{A \rightarrow C\}$ )
- A la clausura de F la llamaremos  $F^+$ , y es el conjunto de todas las dependencias funcionales que se pueden deducir a partir F, aplicando los siguientes axiomas.
  - **Reflexivo:** Si A es un conjunto de atributos y B está incluido en A entonces se cumple que  $A \rightarrow B$
  - **Aumentativo:** Si se cumple que  $A \rightarrow B$  y existe un atributo o conjunto de atributos X en la tabla, entonces se cumple que  $AX \rightarrow BX$
  - **Transitivo:** Si se cumple que  $A \rightarrow B$  y  $B \rightarrow C$ , entonces se cumple que  $A \rightarrow C$

8

## Axiomas de Armstrong (2)

- A partir de estos axiomas, se puede demostrar que se cumplen los siguientes teoremas:
  - **Unión:** Si  $A \rightarrow B$  y  $A \rightarrow C$  entonces se cumple que  $A \rightarrow BC$
  - **Descomposición:** Si  $A \rightarrow BC$  entonces se cumple que  $A \rightarrow B$  y  $A \rightarrow C$
  - **Pseudotransitivo:** Si  $A \rightarrow B$  y se sabe que también  $YB \rightarrow C$  entonces se cumple que  $YA \rightarrow C$

9

## Estudio de Superclaves (1)

- A partir de las dependencias funcionales se pueden deducir la o las claves de una tabla.
- Decimos que un atributo o conjunto de atributos son claves de una tabla sí y sólo sí esos atributos determinan a **todos** los atributos de la tabla.
- La clave se determina mediante la clausura de atributos.
- El conjunto clausura ( $X^+$ ) en  $F$ , de un determinado conjunto de atributos  $X$ , son todos aquellos atributos que quedan determinados por  $X$  en base a las dependencias funcionales en  $F$ .

10

## Estudio de Superclaves (2)

- Ejemplo:
  - $R = (A, B, C, G, H, I)$
  - $F = \{A \rightarrow B, A \rightarrow C, CG \rightarrow H, CG \rightarrow I, B \rightarrow H\}$
  - $A^+ = \{A, B, C, H\}$
- Nótese que el atributo **A** no determina a todos los atributos de la tabla en consecuencia **A no** es clave.
- $AG^+ = \{A, B, C, G, H, I\}$
- Esta pareja de atributos, **si** determinan a todos los atributos de la tabla, por lo cual se lo puede considerar una Superclave de  $R$ .

11

## Estudio de Superclaves (3)

- Métodos para determinar una superclave:
- **Método Analítico:**
  - Todo atributo que no aparece a la derecha de ninguna DF, debe pertenecer a toda superclave.
  - Si un conjunto de atributos forman una superclave, entonces todos los conjuntos que los incluyan son superclaves también.
  - Si un atributo que pertenece a una superclave, se puede derivar de otro conjunto de atributos, una nueva superclave será la que resulta de quitar el atributo derivado y agregar los atributos que lo derivan.

12

## Estudio de Superclaves (4)

- **Método Mecánico:**
  - Determinar la clausura de F de todos los posibles subconjuntos de atributos de la tabla.
- **Método Combinado:**
  - Resulta de combinar las dos técnicas anteriores. Es el más corto puesto que comienza buscando un conjunto de atributos que pertenezcan a la clave y luego descarta o confirma otras claves por el método mecánico.

13

## Estudio de Superclaves (5)

- **Atributo Primo:**
  - Un atributo es primo si forma parte de alguna clave de la tabla.
- **Clave:**
  - Es la superclave mínima. Es decir no se puede formar un subconjunto de la superclave, sin dejar de determinar a todos los atributos de la tabla.

14

## Redundancia de Datos

- Surge cuando se repiten datos en diferentes tablas, ocupando recursos innecesarios.
- Provocan anomalías de inserción, modificación o eliminación de datos.
  - Cuando se necesita agregar un registro puede que se deban repetir algunos datos, que están en otras tablas relacionadas.
  - Cuando se modifica o elimina un dato de una tabla, puede que se tenga que actualizar los datos relacionados en otras tablas.
- Introduce valores nulos, con lo cual se desconoce el valor del atributo en ese momento.
- Puede generar registros erróneos. Por lo general surgen de relacionar tablas mediante atributos inadecuados.

15

## Tipos Dependencias Funcionales

- **Total:**
  - $X \rightarrow Y$  es **total** si no tiene atributos redundantes a la izquierda. Es decir, si se quita algún atributo de X, la dependencia deje de ser válida.
- **Parcial:**
  - $X \rightarrow Y$  es **parcial** si es posible sacar atributos de X y la dependencia sigue siendo válida. Si se cumple que  $X \rightarrow Y$  entonces se cumple  $(X-A) \rightarrow Y$
  - Las dependencias funcionales con un único atributo determinante (lado izquierdo) no pueden ser parciales.
- **Transitiva:**
  - $X \rightarrow Y$  es **transitiva** si X no es un subconjunto de ninguna clave.
- **Observaciones:** Una DF puede ser transitiva y parcial o total y transitiva a la vez.

16

## Normalización - Definición

- Uno de los problemas más frecuentes en un sistema de base de datos relacional es la redundancia de datos.
- El proceso consiste básicamente, en dividir una tabla repartiendo sus atributos en tablas más pequeñas. Esta descomposición debe cumplir ciertas especificaciones.
- El proceso de normalización se basa en una serie de reglas para determinar una forma normal.
- Las formas normales establecen la calidad del diseño del sistema relacional de bases de datos. Se aplican a cada tabla

17

## Formas Normales (1)

- **Primera Forma Normal (1NF):**
  - Los dominios de los atributos deben ser valores atómicos. No pueden existir atributos multivalorados o compuestos.
  - *Método* para determinar la forma normal: los atributos son atómicos.
- **Segunda Forma Normal (2NF):**
  - Debe estar en 1NF
  - Todos los atributos *no primos* dependen totalmente de toda clave.
  - *Método* para determinar la forma normal: todas las dependencias funcionales parciales deben tener el atributo determinado primo. También acepta dependencias funcionales transitivas o totales.

18

## Formas Normales (2)

- **Tercera Forma Normal (3NF):**
  - Debe estar en 2NF
  - No tiene dependencias funcionales transitivas con atributos derivados no primos.
  - *Método* para determinar la forma normal: todas las dependencias funcionales son totales o transitivas pero con el atributo determinado primo.
- **Boyce Codd Forma Normal (BCNF):**
  - Debe estar en 3NF
  - Todas las dependencias funcionales son de la forma superclave → atributo
  - *Método* para determinar la forma normal: todas las dependencias funcionales son totales.

19

## Formas Normales (3)

- Para que una base de datos se encuentre en una determinada forma normal, *todas* sus tablas deben estar por lo menos en esa forma normal.
- Para admitir un buen diseño, el esquema debe estar por lo menos en 3NF.
- Cuando una tabla no se encuentra en la forma normal deseada, para pasarla a una forma normal superior, se la debe descomponer en tablas más pequeñas (particiones) relacionándolas por uno o más atributos.
- La descomposición en tablas consiste en determinar cuál es la dependencia funcional que infringe la forma normal deseada; y descomponer la tabla siguiendo el siguiente método:

20



## Descomposición (1)

- Sea  $R$  un esquema y  $F$  el conjunto de dependencias funcionales. Dado que  $X \rightarrow Y$  no cumple con la forma normal a la que se aspira llegar, entonces:
  - Dividir  $R$  en dos tablas:
    - $R1 = (R - Y)$
    - $R2 = (X, Y)$
  - Volver a realizar el análisis de la forma normal para cada una de las particiones resultantes.
  - Si no se logra la forma normal deseable, se repite el procedimiento únicamente para aquella partición que no cumple la forma normal deseada.

21

## Descomposición (2)

- Para que la descomposición sea válida deben cumplirse las siguientes premisas:
  1. **No perder dependencias funcionales:** todas las dependencias funcionales que tenía la tabla antes de su descomposición, deben estar en algunos de los conjuntos  $F$  de las particiones resultantes (o poder deducirse). A los conjuntos  $F$  de las particiones se le llama proyecciones de  $F$ .
  2. **Cumplir con el teorema de Join sin Pérdidas:** la misma información que tenía  $R$  la deben tener  $R1$  y  $R2$ . Sino, cuando se vuelven a unir las particiones pueden aparecer registros que no estaban en la tabla original. Para ello, se debe demostrar que se cumple el teorema del Join sin Pérdidas.

22

## Join Sin Perdida

- Dado  $R$  y las particiones  $R1$  y  $R2$ , la descomposición es correcta si en  $F^+$  están algunas de estas dos dependencias funcionales :
 
$$R1 \cap R2 \rightarrow R1-R2$$

$$\text{ó}$$

$$R2 \cap R1 \rightarrow R2-R1$$
- Para que una descomposición sea válida no pueden perderse dependencias funcionales (se debe cumplir con el teorema del Join sin Pérdidas). Si alguno de estos postulados no se cumpliera la descomposición no es posible independientemente de que se lograra la forma normal requerida.

23

## Resumen (1)

Pasos básicos que se deben seguir para el análisis y diseño de un modelo relacional de base de datos:

1. Se parte de una determinada realidad y a través de un proceso de abstracción. se modela utilizando el DER.
2. Una vez obtenido el DER, se pasa a tablas y se obtienen los esquemas.
3. Para cada tabla se estudia la relación entre sus atributos y se deduce el conjunto  $F^+$  de dependencias funcionales.
4. Se estudian las superclaves y los atributos primos de cada tabla
5. Definir el tipo de dependencia funcional, para cada una en el conjunto  $F^+$  de cada tabla.

24

## Resumen (2)

6. Definir la forma normal de cada tabla.
7. Si existe alguna tabla (basta con una) que no esté por lo menos en 3NF, se la descompone con el propósito de que las particiones si estén en 3NF
8. Finalmente se está en condiciones de implementar el modelo relacional en un motor de base de datos.

FIN

Modelo Relacional - Normalización