

# BASES DE DATOS

Normalización

# Dependencias funcionales

2

- Una dependencia funcional es una restricción que se establece entre dos conjuntos de atributos de una tabla de la Base de Datos.
- Pensemos en una relación  $R (A_1, A_2, \dots, A_n)$ . Formalmente:
  - Una dependencia funcional, denotada por  $X \rightarrow Y$ , entre dos conjuntos de atributos  $X$  e  $Y$  que son subconjuntos de  $R$ , especifica una **restricción** en las posibles tuplas que pueden formar un estado de relación  $r$  de  $R$ . La restricción dice que dos tuplas  $t_1$  y  $t_2$ , en  $r$  que cumplen que  $t_1[X] = t_2[X]$ , deben cumplir también  $t_1[Y] = t_2[Y]$ .

# Dependencias funcionales

3

- Esto significa que los valores de  $Y$  de una tupla  $r$ 
  - dependen de los valores de  $X$ ,
  - o *están determinados por* los valores de  $X$ ,
  - o bien, que los valores de  $X$  **determinan** de manera única (o **funcionalmente**) los valores de  $Y$ .
  - También decimos que hay una dependencia funcional de  $X$  a  $Y$  o que  $Y$  **depende funcionalmente** de  $X$ .
- La abreviatura de dependencia funcional es **DF**. En una DF  $X \longrightarrow Y$  a  $X$  se lo denomina determinante y a  $Y$  se lo denomina consecuente.

# Dependencias funcionales

4

- Una dependencia funcional es una propiedad de la semántica o significado de los atributos. Los diseñadores de la B. D. utilizarán su comprensión de la semántica de los atributos de  $R$  (esto es, cómo se relacionan unos con otros) para especificar las dependencias funcionales que deben mantenerse en todos los estados de relación  $r$  de  $R$ .

# Dependencias funcionales

5

## Ejemplo 1

Departamentos = (nroDepto, nombreDepto, cantEmpleados)

- nroDepto  $\square$  nombreDepto
- nroDepto  $\square$  cantEmpleados
- nombre  $\square$  cantEmpleados ???
  - Cuando sí?
  - Cuando no?

## Ejemplo 2

Empleados = (nroEmpl, nombre, domicilio, teléfono)

- nroEmpl  $\square$  nombre
- nroEmpl  $\square$  domicilio
- nroEmpl  $\square$  teléfono

# Dependencias funcionales

6

## Ejemplo 3

EmpleadoEnProyecto=(nroEmpl, nroProy,  
horasTrabajadas, nombreEmpleado, nombreProyecto)

- (nroEmpl, nroProy)  $\square$  horasTrabajadas
- nroEmpl  $\square$  nombreEmpleado
- nroProy  $\square$  nombreProyecto

Si continuamos el análisis de la transparencia anterior, por ser CP:

- (nroEmpl, nroProy)  $\square$  nombreEmpleado
- (nroEmpl, nroProy)  $\square$  nombreProyecto

# Dependencias funcionales

7

## Dependencia funcional completa

- Si A y B son atributos de una relación r, B depende funcionalmente de manera completa de A, si B depende de A pero de ningún subconjunto de A.

En el ejemplo anterior

- $(\text{nroEmpl}, \text{nroProy}) \twoheadrightarrow \text{nombreEmpleado}$
  - $\text{nroEmpl} \twoheadrightarrow \text{nombreEmpleado}$
  - Ambas son dependencias funcionales, ¿cuál es completa?
- 
- $(\text{nroEmpl}, \text{nroProy}) \twoheadrightarrow \text{nombreProyecto}$
  - $\text{nroProy} \twoheadrightarrow \text{nombreProyecto}$
  - Ambas son dependencias funcionales, ¿cuál es

# Dependencias funcionales

8

## Dependencia funcional parcial

- $A \twoheadrightarrow B$  es una dependencia funcional parcial si existe algún atributo que puede eliminarse de  $A$  y la dependencia continúa verificándose.

En la transparencia anterior

- $(\text{nroEmpl}, \text{nroProy}) \twoheadrightarrow \text{nombreEmpleado}$
- $\text{nroEmpl} \twoheadrightarrow \text{nombreEmpleado}$
- La primera es una dependencia PARCIAL
  
- $(\text{nroEmpl}, \text{nroProy}) \twoheadrightarrow \text{nombreProyecto}$
- $\text{nroProy} \twoheadrightarrow \text{nombreProyecto}$
- La primera es una dependencia PARCIAL



# Dependencias funcionales

9

## Dependencia funcional transitiva

- Dados A, B y C atributos de una relación tales que A  $\rightarrow$  B y B  $\rightarrow$  C entonces decimos que C depende transitivamente de A a través de B.

Retomemos la tabla: **Empleados = (idEmpleado, nombre, salario, fechaIngreso, idDepto, nombreDepto)**

- idEmpleado  $\rightarrow$  nombre, salario, fechaIngreso, idDepto, nombreDepto
- idDepto  $\rightarrow$  nombreDepto

En este ejemplo

- A = idEmpleado
- B = idDepto
- C = nombreDepto

# Dependencias funcionales

10

## Resumen

- Dependencia funcional
- Dependencia parcial
  - Parte\_clave  $\square$  no\_clave
- Dependencia transitiva
  - No\_clave  $\square$  no\_clave

# Normalización

11

## Definición

- Técnica de diseño de BD que comienza examinando las relaciones que existen entre los atributos (dependencias funcionales). La normalización identifica el agrupamiento óptimo de estos atributos, con el fin de identificar un conjunto de relaciones que soporten adecuadamente los requisitos de datos de la organización.

## Propósito

- Producir un conjunto de relaciones (tablas) con una serie de propiedades deseables partiendo de los requisitos de datos de una organización.

# Normalización

12

- La normalización es un mecanismo que permite que un conjunto de tablas, que integran una BD, cumpla una serie de propiedades deseables. Estas propiedades consisten en evitar:
  - Redundancia de datos.
  - Anomalías de actualización.
  - Pérdida de integridad de datos.

# Normalización

13

- **Redundancia y anomalías de actualización.**
  - El objetivo principal es minimizar la redundancia de información.
  - Por ejemplo, cuando generamos una nueva tabla con las CP de las dos entidades que relaciona, hay información repetida. Sin embargo, esta redundancia es necesaria para representar la relación en el modelo físico.
  - Existen otros casos de redundancia «no deseadas», que generan anomalías. Se clasifican en:
    - Anomalías de inserción.
    - Anomalías de borrado.
    - Anomalías de modificación.

# Normalización

14

- **Redundancia y anomalías de actualización.**
  - *Anomalías de inserción:* Supongamos que se dispone de la siguiente tabla  
Empleados = (idEmpleado, nombre, salario, fechaIngreso, idDepto, nombreDepto)
  - Si se agrega un nuevo empleado, se debe indicar toda la información, incluyendo repetir el nombre del depto. donde trabaja, aunque sea un depto. ya existente en la tabla. Se debe tener especial cuidado en escribir el nombre del depto. de la misma forma como fue ingresado antes.

idEmpleado	nombre	salario	fechaIngreso	idDepto	nombreDepto
1	Juan Díaz	15000	01/04/2005	1	Contabilidad
2	Ana Durán	10000	01/06/2007	2	Electrónica y Computación
3	Luis Tas	8000	01/03/2010	1	Contabilidad
4	Ema Dru	12000	01/10/2006	3	Ventas
5	Juan Rus	13500	01/03/2006	2	Computación y Electrónica

# Normalización

15

- **Redundancia y anomalías de actualización.**
  - *Anomalías de borrado:* Para el mismo ejemplo anterior, borremos a Ema Dru. Al borrar esa tupla, en la misma operación se borra información del departamento donde trabaja, y como es la única empleada registrada para ese depto., se pierde a Ventas como departamento de la organización.
  - *Anomalías de modificación:* Continuando con el mismo ejemplo, supongamos que el depto. de Electrónica y Computación pasa a llamarse depto. de Tecnología. Deberían cambiarse en todas las tuplas.

# Normalización

16

- El *proceso de normalización*, tal y como fue propuesto en un principio por Codd, hace pasar un esquema de relación por una serie de comprobaciones para “certificar” que satisface una determinada **forma normal**.
- Los esquemas de relación insatisfactorios que no cumplan con las **pruebas de formas normales**, se descomponen en esquemas de relación más pequeños que cumplan esas pruebas y que, por consiguiente, cuentan con las propiedades deseables.



# Normalización

17

- La **forma normal** de una relación hace referencia a la condición de forma más alta que cumple y de este modo, indica el grado al que ha sido normalizada.
- Los diseñadores de Bases de Datos no tienen que normalizar hasta la forma más alta posible. Las relaciones pueden dejarse en formas normales inferiores por razones de rendimiento.

# Normalización

18

## Inicialmente (1972)

- Primera Forma Normal
- Segunda Forma Normal ☐ sobre DF
- Tercera Forma Normal ☐ sobre DF

## Se incorpora luego (1974)

- Forma Normal de Boyce Codd ☐ sobre DF

## Luego 1977 y 1979

- Cuarta forma normal
- Quinta forma normal

# Normalización

19

**Proceso □ incremental □ cada vez más restrictivo**

- Comienza con la B.D. en forma NO normal.
- A medida que se avanza, las relaciones (tablas) tienen un formato cada vez más restringido y son menos vulnerables a anomalías de actualización.
- En general, 1FN se aplica siempre.
- El resto puede ser opcional, las 2FN y 3FN normalmente se aplican también.

# Normalización

20

- Repasemos algunas definiciones:
  - Una **clave** es el conjunto de uno o más atributos que cumple con las propiedades de unicidad e irreducibilidad. Pueden haber más de una y en este caso, cada una de ellas se denomina **clave candidata**. Una de las claves candidatas se designa arbitrariamente como **clave primaria**.
  - Un atributo de una relación  $R$  se denomina **atributo primo** de  $R$  si es miembro de *alguna clave candidata* de  $R$ . Un atributo es **no primo** si no es miembro de ninguna clave candidata.
  - En la relación **EmpleadoEnProyecto** (DNI, númeroProy, horas, nombreEmp, nombreProy, localizaciónProy) tanto DNI como númeroProy son atributos primos y el resto son no primos.

# Normalización

21

## Primera Forma Normal (1FN)

- Establece que el dominio de un atributo debe incluir sólo valores atómicos (simples, indivisibles) y que el valor de cualquier atributo en una tupla debe ser un valor individual proveniente del dominio de ese atributo.
- Un modelo estará en 1FN si para toda relación  $r$  del modelo (tabla) cada uno de los atributos que la forman es si y solo sí monovalente.

# Normalización

22

## Primera Forma Normal (1FN)

- Ejemplo: Personas = (dni, nombre, domicilio, códigosTítulos)

Se observa que el atributo títulos es polivalente

La solución es crear otras dos tablas:

- Personas = (dni, nombre, domicilio)
- Títulos = (código, descripción)
- Posee = (dni, códigoTítulo)

# Normalización

23

## Segunda Forma Normal (2FN)

- Una tabla que tenga atributos que dependan parcialmente de otro no está en 2FN.
- Un modelo está en 2FN sí y solo sí está en 1FN y para toda relación  $r$  (tabla) del mismo no existen dependencias parciales.

# Normalización

24

## Segunda Forma Normal (2FN)

### ▪ Ejemplo 1

*Alquileres* = (*idCliente*, *idPropiedad*, *nombreCliente*,  
*nombrePropietario*, *montoAlquiler*, *fechaInicio*,  
*duración*)

Dependencias:

- *idCliente*, *idPropiedad*  $\square$  *nombreCliente*, *nombrePropietario*, *montoAlquiler*, *fechaInicio*, *duración* (DF)
- *idCliente*  $\square$  *nombreCliente* (Dependencia Parcial)
- *idPropiedad*  $\square$  *nombrePropietario* (Dependencia Parcial)

**Solución:**

- *Clientes* = (*idCliente*, *nombreCliente*)
- *Propiedades* = (*idPropiedad*, *nombrePropietario*)



# Normalización

25

## Segunda Forma Normal (2FN)

### ▪ Ejemplo 2

*empleadoProyecto = (dniEmpleado, idProyecto, horasTrabajadas, nombreEmpleado, nombreProyecto, fechaInicioProyecto, fechaInicioEmpleadoProyecto)*

Dependencias:

- dniEmpleado, idProyecto  $\square$  horasTrabajadas, nombreEmpleado, nombreProyecto, fechaInicioProyecto, fechaInicioEmpleadoProyecto (DF)
- dniEmpleado  $\square$  nombreEmpleado (DP)
- idProyecto  $\square$  nombreProyecto, fechaInicioProyecto (DP)

**Solución:** ?

# Normalización

26

## Tercera Forma Normal (3FN)

- Una tabla que tenga atributos que dependan transitivamente de otro no está en 3FN.
- Un modelo está en 3FN sí y solo sí está en 2FN y para toda relación  $r$  (tabla) del mismo no existen dependencias transitivas.

# Normalización

27

## Tercera Forma Normal (3FN)

### ▪ Ejemplo 1

*empleados* = (*dniEmpleado*, *nombreEmpleado*,  
*nroDepto*, *nombreDepto* )

Dependencias:

- *dniEmpleado* □ *nombreEmpleado*, *nroDepto*, *nombreDepto* (DF)
- *nroDepto* □ *nombreDepto* (DT)

**Solución:**

- *empleados* = (*dniEmpleado*, *nombreEmpleado*, *nroDepto*)
- *departamentos* = (*nroDepto*, *nombreDepto*)

# Normalización

28

## Tercera Forma Normal (3FN)

### ▪ Ejemplo 2

*Propiedades = (idPropiedad, municipio, númeroParcela, área, precio, tasa fiscal)*

Dependencias:

- idPropiedad  $\square$  municipio, numeroParcela, área, precio, tasa fiscal (DF)
- municipio  $\square$  tasa fiscal (DT)
- área  $\square$  precio (DT)

**Solución:** ?

# Normalización

29

- Ejercicio 1: Llevar la relación propuesta a la tercera forma normal.

**ítemsFactura** (númeroFactura, idProducto, descripciónProducto, fecha, idCliente, domicilioCliente, telCliente, nomCliente, cantidad, precioUnitario)

Sus dependencias son:

númeroFactura, idProducto □ descripciónProducto, fecha, idCliente, domicilioCliente, telCliente, nomCliente, cantidad, precioUnitario

idProducto □ descripciónProducto, precioUnitario

númeroFactura □ fecha, idCliente

idCliente □ domicilioCliente, telCliente, nomCliente

# Normalización

30

- Ejercicio 2: Llevar la relación propuesta a la tercera forma normal.

CURSADAS (MateriaCódigo, MateriaNombre, DeptoCódigo, DeptoNombre, AlumLegajo, AlumApellido, AlumNombre, AlumDomicilio, AlumNroDoc, Teléfonos, Cuatrimestre, AñoAcadémico, CursadaResultado)

Sus dependencias son, además de la impuesta por la clave primaria:

MateriaCódigo → MateriaNombre, DeptoCódigo, DeptoNombre

DeptoCódigo → DeptoNombre

AlumLegajo → AlumApellido, AlumNombre, AlumDomicilio, AlumNroDoc, teléfonos