

Diseño de Bases de Datos

Refinamiento de esquemas y
Normalización

El problema de la redundancia

- » se realiza sobre el esquema relacional
- » por *anticipación*, se hace sobre el ERD

- Redundancia: almacenar información en más de un sitio
 - Almacenamiento redundante
 - Posibilidad de inconsistencia
 - Anomalías
 - actualización
 - inserción
 - borrado

■ ■ ■

Empleado

<u>dni</u>	nombre	nreg	cat	horas_ sem	sueldo_ base
2354	García	48	8	40	10
9625	Aragón	22	8	30	10
5557	Pozo	35	5	35	7
2121	Sainz	35	5	40	7



- la combinación 8 -> 10 y 5 -> 7 es redundante
- actualización de base_sueldo: en todas las tuplas
- inserción de empleado: sólo si se sabe la base de su categoría —dejar *null* ?
- borrado de todas las tuplas de una categoría: se pierde la base_sueldo

Descomposición

- » sustituir una relación por otras dos (o más) cada una con un subconjunto de campos y, conjuntamente, incluyendo todos los originales

Empleado(dni, nombre, nreg, cat, horas_sem)

Sueldo(cat, base_sueldo)



■ Requisitos descomposición

- Reunión sin pérdida
- Conservación de las dependencias
- Consultas pueden obligar a la reunión de las relaciones descompuestas
 - penalización en el rendimiento
 - ¿y si la eficiencia resultante no fuera aceptable?

Dependencias funcionales

DF, $X \rightarrow Y$

$\forall t1, t2 : \text{tuplas de } R \mid t1.X = t2.X \Rightarrow t1.Y = t2.Y$

Ej. 1: dos atributos que no son clave

Empleado(dni, nombre, nreg, cat, horas_sem, sueldo_base)

DF cat \rightarrow base_sueldo

Ej. 2: del que depende **es parte** de la clave

Reserva(dni, matr, fecha, tarjeta), dni 1—1 tarjeta

DF dni \rightarrow tarjeta

DF, clave \rightarrow todos los atributos —*definición relacional*

- la dependencia no exige que la clave sea mínima

DF, $X \rightarrow$ todos los atributos

- si X no es mínimo, entonces es una *superclave*

Formas Normales

» si el esquema se encuentra en una de estas formas normales, ciertos tipos de problemas serán eliminados / minimizados

- Basadas en DF
 - 1NF, 2NF, 3NF, BCNF
 - 3NF y BCNF son importantes en el diseño de bases de datos
- Basadas en otros tipos de dependencias
 - multivaluadas (DM): 4NF, reunión (DR): 5NF
- 1NF: no existencia de grupos repetitivos

BCNF

» —*intuitivamente*— permitir solo las dependencias con la clave

- Rel R , DFs F , subconj. atributos X , atrib. A
 R está en BCNF si, para cada DF $X \rightarrow A$,
 - $A \in X$, DF *trivial*, o
 - X contiene una clave de R
- Ej. no BCNF
 - Empleado(dni, nombre, nreg, cat, horas_sem, sueldo_base)
DF $\text{cat} \rightarrow \text{base_sueldo}$
 - Reserva(dni, matr, fecha, tarjeta), dni $1 \text{---} 1$ tarjeta
DF $\text{dni} \rightarrow \text{tarjeta}$
- Solución por descomposición

3NF

- Problema *técnico* con BCNF
 - múltiples claves candidatas solapadas
 - Reserva(dni,matr,fecha,tarjeta), dni ¹—* tarjeta
 - tengo que anotar la tarjeta con la que se pagó, pero
 - DF tarjeta → dni, y esto no cumple BCNF
 - tarjeta,matr,fecha podría ser clave primaria
- Rebajo las condiciones y permito una cierta redundancia: 3NF
 - tercera posibilidad: A es parte de alguna clave de R

Descomposición de una relación

» en dos (o más) que se *reparten* los atributos, garantizando *reunión sin pérdida* y *conservación de dependencias*

- Algoritmo de descomposición en BCNF
 1. R no en BCNF, $X \subset R$, A atributo, $X \rightarrow A$ que provoca el no cumplimiento
 2. Descomponer en R-A y XA
 3. Si R-A o XA no en BCNF, aplicar recursivamente
- Ej. Contratos con atributos CPYDRNV
(idC,idProve,idProy,idDepto,idRepuesto,cantidad,valor)
y DFs $PD \rightarrow R, Y \rightarrow P$

...

1. guiado por $PD \rightarrow R$

- PDR y CPYDNV, ahora aplicar para $Y \rightarrow P$
- PDR, YP y CYDNV

2. guiado por $Y \rightarrow P$

- YP y CYDRNV
- la otra dependencia ya no desnormaliza

- Distintas alternativas, cuya elección se hará respecto a la semántica de la aplicación

ER -> Rel -> Rel Normalizado

- ¿Se puede directamente producir un ER libre de problemas de redundancia?
 - ER es un proceso complejo y subjetivo (piénsese en esquemas con más de 100 tablas)
 - Determinadas restricciones / dependencias no se pueden expresar en ER (al menos fácilmente)
 - Técnica formal para tratar un diseño original que da como resultado un diseño mejor