



DECSAI

Departamento de Ciencias de la Computación e I.A.

Universidad de Granada



Diseño lógico

Diseño de bases de datos relacionales

Diseño lógico de bases de datos relacionales



- El modelo relacional:
 - El concepto de relación: tuplas, atributos y dominios.
 - Restricciones de integridad en el modelo relacional.
- El proceso de diseño lógico en el modelo relacional.
- Del modelo E/R al modelo relacional:
 - Entidades.
 - Entidades débiles.
 - Relaciones.
 - Relaciones de especialización / generalización.
 - Fusión de tablas.
- Normalización.



Diseño lógico de bases de datos relacionales



ETAPA DE DISEÑO LÓGICO

Objetivo: Creación del esquema conceptual y de los esquemas externos de la base de datos en el modelo de datos elegido (p.ej. relacional), independientemente del SGBD que se vaya a utilizar.

Tarea: Transformar los esquemas obtenidos en el diseño conceptual en un conjunto de estructuras propias del modelo de datos elegido.

Resultado: Conjunto de estructuras propias del modelo abstracto de datos (p.ej. relaciones).



El modelo relacional



El modelo de datos relacional organiza y representa los datos en forma de tablas o relaciones:

Una base de datos relacional es una colección de relaciones [tablas], cada una de las cuales tiene un nombre único.

Representación lógica	Representación física	Modelo relacional
 Tabla	Archivo secuencial	Relación
 Fila	Registro	Tupla
 Columna	Campo	Atributo



El modelo relacional



El concepto de relación: Tuplas, atributos y dominios

id_trabajador	nombre	tarifa_hr	tipo_de_oficio	id_supv
1235	F. Aguilera	12,50	Electricista	1311
1412	A. Calvo	13,75	Fontanero	1540
2920	N. Marín	10,00	Carpintero	null
3231	O. Pons	17,40	Albañil	null
1540	J.M. Medina	11,75	Fontanero	null
1311	J.C. Cubero	15,50	Electricista	null
3001	D. Sánchez	8,20	Albañil	3231



El modelo relacional



El concepto de relación: Tuplas, atributos y dominios

- **Atributo (A_i):** Elemento susceptible de tomar valores (cada una de las columnas de la tabla).
- **Dominio (D_i):** Conjunto de valores que puede tomar un atributo (se considera finito).
- **Tupla:** Cada uno de los elementos que contiene una instancia de la relación (filas).



El modelo relacional



El concepto de relación

Relación $R(A_1..A_n)$

Subconjunto del producto cartesiano $D_1 \times \dots \times D_n$
(esto es, una tabla).

En una relación hay que distinguir dos aspectos:

- **Esquema de la relación:** Los atributos $A_1..A_n$
p.ej. Trabajadores (id_trabajador, nombre, tarifa_hr, tipo_de_oficio, id_supv)
- **Instancia de la relación:** El conjunto de tuplas $\{(x_1, x_2, \dots, x_n)\} \subseteq D_1 \times D_2 \times \dots \times D_n$ que la componen en cada momento.



El modelo relacional



El concepto de relación

Relación $R(A_1..A_n)$

Subconjunto del producto cartesiano $D_1 \times \dots \times D_n$
(esto es, una tabla).

Consecuencias de la definición de relación como conjunto de tuplas:

- No existen tuplas duplicadas
(concepto de clave primaria).
- No existe orden en las tuplas
(ni en los atributos).



El modelo relacional



Esquema de la base de datos

Una base de datos relacional es un conjunto finito de relaciones junto con una serie de restricciones o reglas de integridad:

- **Restricción de integridad:** Condición necesaria para preservar la corrección semántica de la base de datos.
- **Esquema de la base de datos:** Colección de esquemas de relaciones junto con las restricciones de integridad que se definen sobre las relaciones.



El modelo relacional



Instancia de la base de datos

- **Instancia (o estado) de la base de datos:** Colección de instancias de relaciones que verifican las restricciones de integridad.

- **Base de datos relacional:** Instancia de la base de datos junto con su esquema.



El modelo relacional



Restricciones de integridad: Asociadas a las tuplas de una relación

p.ej. $0 \leq \text{edad} \leq 120$
 $\text{impuestos} \leq \text{suelo}$

- En ocasiones, no se conoce el valor de un atributo para una determinada tupla. En esos casos, a ese atributo de esa tupla se le asigna un **valor nulo (null)**, que indica que el valor de ese atributo es desconocido o, simplemente, que ese atributo no es aplicable a esa tupla.



El modelo relacional



Restricciones de integridad: Asociadas a las relaciones de la base de datos

- **Clave primaria:**
Conjunto de atributos seleccionados para identificar unívocamente a las tuplas de una relación.
- **Integridad de entidad:**
Los atributos de la clave primaria no pueden tomar valores nulos, ya que la clave primaria debe permitirnos identificar unívocamente cada tupla de la relación.



El modelo relacional



Restricciones de integridad: Asociadas a las relaciones de la base de datos

- **Clave externa:** Conjunto de atributos de una relación cuyos valores en las tuplas deben coincidir con valores de la clave primaria de las tuplas de otra relación.
- **Integridad referencial:**
Todos los valores no nulos de una clave externa referencian valores reales de la clave referenciada.

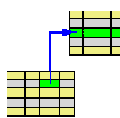


El modelo relacional



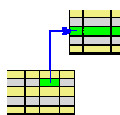
Restricciones de integridad: Asociadas a las relaciones de la base de datos

La integridad referencial mantiene las conexiones en las bases de datos relacionales:



$\text{imparte.NRP} \in \text{profesor.NRP}$

El profesor que imparte una asignatura debe existir en la tabla de profesores.



$\text{cuenta.sucursal} \in \text{sucursal.número}$

Una cuenta tiene que pertenecer a una sucursal existente.



El proceso de diseño lógico en el modelo relacional



Transformación de un diagrama E/R en un esquema relacional:

1. Se transforman en tablas todos los tipos de entidades y relaciones que aparecen en el diagrama E/R.
2. Se seleccionan las claves primarias para cada una de las tablas de nuestro esquema lógico.
3. **Fusión de tablas:** Se pueden combinar aquellas tablas que comparten su clave primaria.
4. **Normalización:** Se normaliza el esquema resultante (al menos, hasta BCNF).
5. Se definen todas las restricciones de integridad que sean aplicables al esquema obtenido.



Del modelo E/R al modelo relacional



Entidades

Cada tipo de entidad da lugar a una tabla en la base de datos.



- **Atributos:**
Los atributos del tipo de entidad.
- **Clave primaria:**
Una de las claves candidatas del conjunto de entidades.



Del modelo E/R al modelo relacional

Entidades

Atributos compuestos

- Se incluyen en la relación todos los atributos simples (atómicos) que forman parte del atributo compuesto.

Atributos derivados

- No se almacenarán en la base de datos, por lo que **no** se incluyen como atributos de las relaciones.



Del modelo E/R al modelo relacional

Entidades

Atributos multivaluados

- Se almacenan en una tabla auxiliar que incluya las columnas necesarias para almacenar la clave primaria del conjunto de entidades más aquéllas que se necesiten para representar un valor del atributo MV.
- Salvo que el atributo MV sea una clave alternativa del conjunto de entidades, todas las columnas de la tabla auxiliar formarán parte de su clave primaria.
- La tabla auxiliar incluirá una clave externa que haga referencia a la tabla correspondiente al conjunto de entidades que incluye el atributo multivaluado.

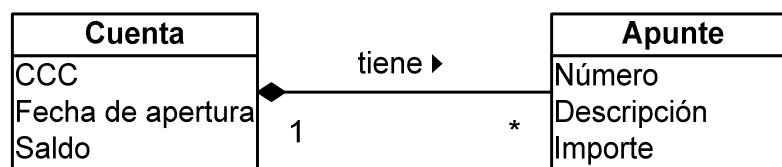


Del modelo E/R al modelo relacional

Entidades débiles

■ Atributos:

Además de los atributos propios de la entidad débil, los atributos pertenecientes a la clave primaria de la entidad fuerte de la que depende existencialmente la entidad débil.



Apunte (**CCC**, número, descripción, importe)



Del modelo E/R al modelo relacional

Entidades débiles

■ Clave primaria:

La clave primaria de la entidad fuerte más un conjunto de atributos propio de la entidad débil (discriminante).

Clave primaria de la entidad débil
=
Clave primaria de la entidad fuerte
+
Discriminante

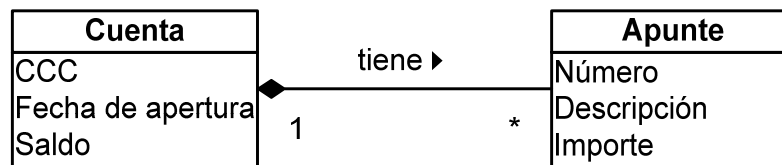
Apunte (**CCC**, **número**, descripción, importe)



Del modelo E/R al modelo relacional

Entidades débiles

- **Clave externa:**
Una, haciendo referencia a la entidad fuerte de la que depende existencialmente la entidad débil.



Apunte (CCC, número, descripción, importe)

└─> Cuenta (CCC, ...)



Del modelo E/R al modelo relacional

Relaciones

Cada tipo de relación da lugar a una tabla en la base de datos.



Atributos:

Los atributos de las claves primarias de las entidades que intervienen en la relación más los atributos propios de la relación.



Del modelo E/R al modelo relacional

Relaciones

Clave primaria:

Si la relación no tiene atributos propios:

- **Relación muchos a muchos:** La unión de las claves de los conjuntos de entidades que intervienen.
- **Relación uno a muchos:** La clave correspondiente al conjunto de entidades que participa en la relación con cardinalidad "muchos".
- **Relación uno a uno:** Una de las claves de las entidades intervinientes en la relación (cualquiera).



Del modelo E/R al modelo relacional

Relaciones

Clave primaria:

Si hay atributos propios de la relación:

- Los atributos correspondientes al tipo de relación, a los que tal vez añadiremos algunos atributos propios de la relación, dependiendo de su semántica.

Claves externas:

- Una por cada una de las claves primarias de los conjuntos de entidades que intervienen en la relación



Del modelo E/R al modelo relacional

Relaciones

SOBRE LA RELACIÓN ENTRE ENTIDADES DÉBILES Y FUERTES...

Las relaciones entre entidades débiles y fuertes no hay que pasarlas a tablas porque la relación se recoge como parte de la clave primaria de la entidad débil (la parte correspondiente a la clave primaria de la entidad fuerte es una clave externa que apunta a la tabla derivada de la entidad fuerte).



Del modelo E/R al modelo relacional

Relaciones n-arias

Atributos: Los atributos de las claves primarias de los conjuntos de entidades que intervienen en la relación más los atributos propios de la relación.

Clave primaria: Estará formada por la unión de las claves primarias correspondientes a todos aquellos conjuntos de entidades que intervengan en la relación con cardinalidad N (más, opcionalmente, alguno[s] de los atributos propios de la relación).

Claves externas: Una por cada uno de los conjuntos de entidades que intervienen en la relación.

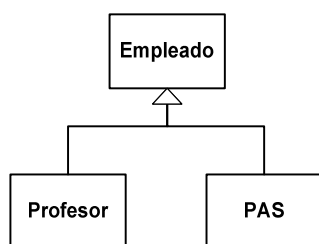


Del modelo E/R al modelo relacional

Relaciones de generalización y especialización

Estrategia A: Una tabla por cada conjunto de entidades

Las particularizaciones heredan la clave primaria del conjunto de entidades de nivel superior (la cual será, en las tablas correspondientes a los subtipos, una clave externa que referencia a la tabla derivada del supertipo).



Ejemplo:

Empleado (NRP, nombre, dirección...)

Profesor (NRP, departamento, categoría)

PAS (NRP, grupo, nivel)

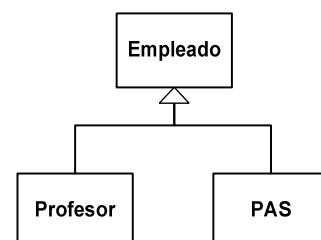


Del modelo E/R al modelo relacional

Relaciones de generalización y especialización

Estrategia B: Una tabla por cada caso particular

Las particularizaciones heredan todos los atributos de la entidad general.



Ejemplo:

Profesor (NRP, nombre, dirección... departamento, categoría)

PAS (NRP, nombre, dirección... grupo, nivel)



Del modelo E/R al modelo relacional

Relaciones de generalización y especialización

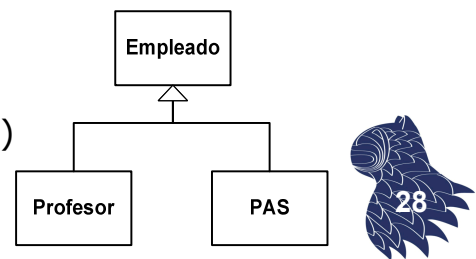
Estrategia C: Una tabla para toda la jerarquía

Adecuada en muy contados casos.

- Se suele añadir una columna artificial (discriminante) que indique el tipo de la entidad representada por cada tupla de la tabla (para permitir el mantenimiento de las restricciones de integridad aplicables).

Ejemplo:

Empleado (NRP, nombre, dirección... **tipo**, departamento, categoría, grupo, nivel)



Del modelo E/R al modelo relacional

Relaciones de generalización y especialización

Formalmente, la primera estrategia es la correcta.

Las otras dos estrategias sólo las emplearemos cuando, por cuestiones de eficiencia, queramos reducir el número de reuniones necesarias para realizar determinadas consultas (motivo por el que la decisión de utilizar un esquema u otro la pospondremos usualmente a la fase de diseño físico de la base de datos).



Del modelo E/R al modelo relacional

Fusión de tablas

Se pueden combinar en una sola todas las tablas que compartan su clave primaria.

p.ej.

Relaciones uno a muchos

Las tablas derivadas de las relaciones muchos a uno se pueden fusionar con las derivadas de las entidades que participan en la relación con cardinalidad N.



Del modelo E/R al modelo relacional

Fusión de tablas

Relaciones uno a uno

Se pueden combinar las tablas derivadas de los dos conjuntos de entidades en una sola o mantener tablas separadas:

- Si la relación es obligatoria en ambos sentidos (las entidades involucradas siempre aparecen conjuntamente), se pueden combinar las tablas derivadas de los dos conjuntos de entidades, manteniendo como clave primaria la clave primaria de uno de los conjuntos de entidades y como clave alternativa la clave primaria del otro conjunto de entidades.
- En cualquier otro caso, siempre se mantendrán tablas separadas para los dos conjuntos de entidades, haciendo que la tabla de una de ellas absorba la tabla que se derivaría de la relación. Si la participación de una de las entidades es obligatoria, se suele elegir su tabla para fusionarla con la tabla derivada de la relación.



Del modelo E/R al modelo relacional

Fusión de tablas

Relaciones de especialización y generalización

A la hora de representar jerarquías de especialización/generalización, a veces fusionaremos las tablas correspondientes a distintos conjuntos de entidades.

Se ha de llegar a un compromiso entre el coste de realizar consultas que involucren reuniones de distintas tablas (cuando tenemos tablas independientes) y el coste que supone desaprovechar espacio de almacenamiento y tener que mantener manualmente determinadas restricciones de integridad (cuando se combinan varias tablas en una sola).



Normalización

La normalización permite obtener un conjunto adecuado de relaciones de tal forma que:

- El esquema de la base de datos incluya el mínimo número de atributos necesarios para dar soporte a los requerimientos del sistema.
- Resulte más fácil acceder a la base de datos y, sobre todo, mantener los datos de la base de datos (**redundancia mínima**: salvo los atributos que forman parte de claves externas, los demás se representarán una única vez en la base de datos).



Normalización



En una base de datos normalizada:

- Las actualizaciones se consiguen realizar con un número mínimo de operaciones (mejorando la eficiencia de la BD y reduciendo la posibilidad de que aparezcan inconsistencias).
- Se reduce al mínimo el espacio de almacenamiento necesario para almacenar los datos de la BD (reduciendo los costes de operación de la BD).



Normalización



Las relaciones que almacenan datos redundantes presentan **anomalías de actualización** (la inserción, eliminación o modificación de los datos puede provocar la aparición de inconsistencias), por lo que resulta adecuado descomponerlas:

- **Sin pérdidas** (de forma que la relación original se pueda reconstruir a partir de las relaciones en las que la hayamos descompuesto).
- **Preservando las dependencias** (para que podamos mantener las restricciones de integridad de la relación original introduciendo restricciones en las relaciones provenientes de la descomposición de la relación original).



Normalización



La descomposición sin pérdidas es indispensable, la descomposición que preserva las dependencias no siempre es posible.

A veces, el diseñador tiene que elegir entre

- no normalizar, o bien,
- perder dependencias.



Normalización



Dependencias funcionales

Describen relaciones entre los atributos de una relación:

$$A \rightarrow B$$

B depende funcionalmente de A ($A \rightarrow B$) cuando cada valor de A en una relación R aparece siempre asociado al mismo valor de B en R.





Dependencias funcionales

Formalmente:

Sea un esquema R , sean α y β subconjuntos de atributos, $\alpha \subseteq R$ y $\beta \subseteq R$. Decimos que α determina funcionalmente a β , o que β depende funcionalmente de α , o que $\alpha \rightarrow \beta$, si y sólo si se verifica, que para toda relación r instancia de ese esquema:

$$\forall t_1, t_2 \in r ; t_1[\alpha] = t_2[\alpha] \Rightarrow t_1[\beta] = t_2[\beta]$$



Identificación de dependencias funcionales

- La identificación de las dependencias funcionales existentes es relativamente fácil si se conoce el significado de cada atributo y las relaciones existentes entre ellos.
- Toda la información necesaria debería figurar en el documento de especificación de requerimientos, bien en la parte correspondiente a los requerimientos funcionales o bien en el diccionario de datos que ha de acompañar al modelo semántico de la base de datos.



Normalización



La identificación de dependencias funcionales sirve para:

- Especificar las **restricciones de integridad** asociadas a una relación (claves candidatas: clave primaria y claves alternativas).
- Detectar posibles **anomalías de actualización** y evitarlas, ya sea reorganizando el esquema de la base de datos (recomendado) o tomando las medidas oportunas al implementar las aplicaciones que funcionen sobre la base de datos (trabajo adicional que habrá que justificar razonadamente).



Normalización



El proceso de normalización

- La normalización consiste en analizar el conjunto de relaciones obtenido a partir del diagrama E/R teniendo en cuenta las claves candidatas y las dependencias existentes entre los atributos de cada relación.
- La normalización se suele descomponer en una serie de pasos, cada uno de los cuales corresponde a una **forma normal** específica de propiedades conocidas.



Normalización: 1NF



1NF: Primera Forma Normal

Todos los atributos tienen dominios atómicos.

Para obtener una relación en 1NF:

Se eliminan los atributos compuestos y multivaluados.

There multiple items for some of the orders all stored in one record and is difficult to work with and retrieve information.

Order ID	Customer ID	Order Date	Items
1	ALWAO	01/06/2009	3 White Chocolate, 2 Bean Curd, 1 Thin Bread
2	ANDRC	01/06/2009	2 Thin Bread
3	ANDRC	02/06/2009	5 Pickled Herring, 2 Teatime Chocolate Biscuits
4	BERGS	03/06/2009	1 White Chocolate, 4 Teatime Chocolate Biscuits, 5 Red Caviar

Better approach is to split into individual records (rows)

Order ID	Order Item ID	Customer ID	Quantity	Item
1	1	1 ALWAO	3	White Chocolate
1	2	1 ALWAO	2	Bean Curd
1	3	1 ALWAO	1	Thin Bread
2	1	1 ANDRC	2	Thin Bread
3	1	1 ANDRC	5	Pickled Herring
3	2	1 ANDRC	2	Teatime Chocolate Biscuits
4	1	1 BERGS	1	White Chocolate
4	2	1 BERGS	4	Teatime Chocolate Biscuits
4	3	1 BERGS	5	Red Caviar

For each item now appears in a separate row as a record and is linked by the common 'Order ID' field being duplicated.



Normalización: 2NF



2NF: Segunda Forma Normal

Todos los atributos no primos (los que no forman parte de claves candidatas) dependen funcionalmente de las claves candidatas de forma completa.

- Una dependencia funcional es completa cuando el determinante no se puede simplificar.

Para obtener una relación en 2NF:

Si existe una dependencia funcional incompleta $CK \rightarrow \beta$ (esto es, $\alpha \rightarrow \beta$ con $\alpha \in CK$, siendo CK una clave candidata de la relación), β se traslada a una nueva relación junto con el determinante α y eliminamos β de la relación original.



Normalización: 3NF



3NF: Tercera Forma Normal

Ningún atributo no primo depende transitivamente, a través de un atributo no primo, de una clave candidata.

- Si $A \rightarrow B$ y $B \rightarrow C$, entonces C depende transitivamente de A a través de B (esto es, $A \rightarrow C$).
- Esta dependencia transitiva puede causar anomalías de actualización cuando B no es una clave candidata de la relación.

Para obtener una relación en 3NF:

Se eliminan las dependencias transitivas problemáticas trasladándolas a una nueva relación.



Normalización: BCNF



La definición original de Codd para la 3NF puede no producir diseños satisfactorios si hay varias claves candidatas y éstas se solapan:

BCNF: Forma Normal de Boyce y Codd

Todo determinante es una clave candidata.

- Toda relación en BCNF está en 3NF.

Diferencia entre 3NF y BCNF:

Dada una dependencia funcional $A \rightarrow B$, 3NF la permite en una relación si B es un atributo primo y A no es una clave candidata, mientras que BCNF requiere que A sea una clave candidata.



Normalización: 4NF



Otros tipos de dependencias, distintas a las dependencias funcionales, también pueden introducir redundancia en los datos almacenados en una relación:

4NF: Cuarta Forma Normal

Como consecuencia de la 1NF, pueden aparecer dependencias multivaluadas que habrá que eliminar.

Para que una relación esté en 4NF, todo determinante de una dependencia multivaluada debe ser una clave candidata (y, por tanto, una dependencia funcional).



Normalización: 5NF



5NF: Quinta Forma Normal

Hay relaciones que no se pueden descomponer sin pérdidas en dos proyecciones, pero sí en tres o más.

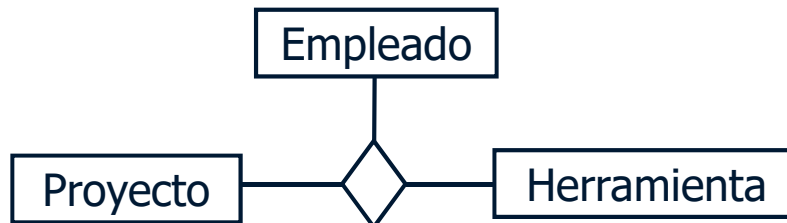
Una relación está en 5NF si, y sólo si,
**toda dependencia de reunión
es consecuencia de las claves candidatas.**



Normalización: Ejemplo



“Un empleado asignado a un proyecto debe saber emplear **todas** las herramientas necesarias para llevar a cabo el proyecto.”



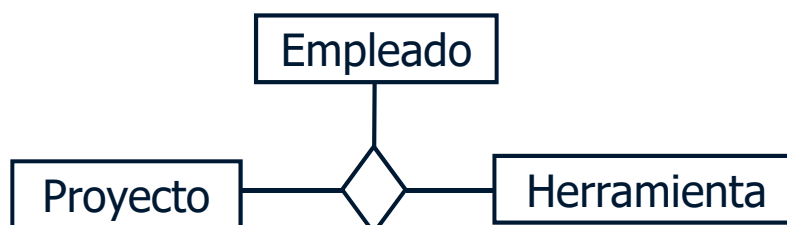
- BCNF.
- Dependencia multivaluada:
proyecto ->> herramienta | empleado
- Descomposición sin pérdidas con dos proyecciones.



Normalización: Ejemplo



“Un empleado asignado a un proyecto debe saber emplear **alguna** de las herramientas necesarias para llevar a cabo el proyecto.”



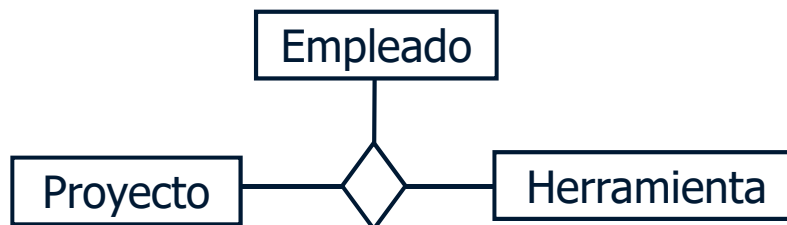
- 4NF.
- Sin dependencias multivaluadas no triviales.
- Descomposición sin pérdidas con tres proyecciones:
{(#E,#P), (#E,#H), (#P,#H)}



Normalización: Ejemplo



“Registro de las herramientas que utilizan los distintos empleados durante la realización de cada proyecto.”



- 5NF.
- Se trata realmente de una relación ternaria:
No existe ninguna descomposición sin pérdidas.



Normalización: Ejemplo



Resumen

Una relación ternaria muchos-a-muchos-a-muchos está en

- BCNF si se puede reemplazar por 2 relaciones binarias.
- 4NF si se puede reemplazar por 3 relaciones binarias.
- 5NF si no se puede descomponer en relaciones binarias.

Cuando la relación ternaria es 1-1-1, 1-1-N o 1-N-N, ésta no se puede reemplazar por relaciones binarias sin perder dependencias funcionales (las que tienen una entidad con cardinalidad 1 en su consecuente y las otras dos en su antecedente).



Normalización



Cuando se decida no normalizar tras haber encontrado una dependencia entre los atributos de una relación, **se ha de justificar el porqué.**

p.ej.

CP → *Municipio* en una dirección, pero tal vez no nos interese tener que mantener una tabla aparte con todos los municipios de España y sus códigos postales.



Bibliografía



- C.J. Date:
"Introducción a los sistemas de bases de datos".
Prentice Hall, 2001 [7ª edición]. ISBN 968-444-419-2.
- Ramez A. Elmasri & Shamkant B. Navathe:
"Fundamentos de Sistemas de Bases de Datos".
Addison-Wesley, 2007 [5ª edición]. ISBN 84-782-9085-0.
- Thomas M. Connolly & Carolyn E. Begg:
"Sistemas de Bases de Datos"
Addison-Wesley, 2005 [4ª edición]. ISBN 84-782-9075-3.
- Henry F. Korth, Abraham Silberschatz & S. Sudarshan:
"Fundamentos de Bases de Datos".
McGraw-Hill, 2006 [5ª edición]. ISBN 84-481-4644-1.
- Olga Pons, Nicolás Marín, Juan Miguel Medina, Silvia Acid & Mª Amparo Vila: "Introducción a las Bases de Datos: El modelo relacional". Paraninfo, 2005. ISBN 8497323963

