

Normalización en Bases de Datos Relacionales

Luis Valencia Cabrera (lvalencia@us.es)

Research Group on Natural Computing

Departamento de Ciencias de la Computación e Inteligencia Artificial
Universidad de Sevilla

23-11-2022, Bases de Datos

Índice

- 1 Introducción a la normalización
- 2 Dependencias funcionales
- 3 Formas Normales
- 4 Normalización
- 5 Bibliografía

Índice

- 1 Introducción a la normalización
- 2 Dependencias funcionales
- 3 Formas Normales
- 4 Normalización
- 5 Bibliografía

Introducción

Echando la vista atrás...

Mirando un poco el bagaje que llevamos surgen algunas preguntas...

- ¿Hemos aprendido a diseñar y desarrollar **buenas** bases de datos?

Introducción

Echando la vista atrás...

Mirando un poco el bagaje que llevamos surgen algunas preguntas...

- ¿Hemos aprendido a diseñar y desarrollar **buenas** bases de datos?
- ¿Hemos capturado y organizado de forma **adecuada** los datos que responden a los requisitos establecidos?

Introducción

Echando la vista atrás...

Mirando un poco el bagaje que llevamos surgen algunas preguntas...

- ¿Hemos aprendido a diseñar y desarrollar **buenas** bases de datos?
- ¿Hemos capturado y organizado de forma **adecuada** los datos que responden a los requisitos establecidos?
- ¿En qué **sustentamos** nuestra respuesta?

Introducción

Echando la vista atrás...

Mirando un poco el bagaje que llevamos surgen algunas preguntas...

- ¿Hemos aprendido a diseñar y desarrollar **buenas** bases de datos?
- ¿Hemos capturado y organizado de forma **adecuada** los datos que responden a los requisitos establecidos?
- ¿En qué **sustentamos** nuestra respuesta?
- ¿Existe alguna manera de **medir** esto?

Introducción

¿Qué hemos hecho hasta ahora?

Algunos pasos dados...

- Hemos profundizado en el diseño conceptual, empleando...

Introducción

¿Qué hemos hecho hasta ahora?

Algunos pasos dados...

- Hemos profundizado en el diseño conceptual, empleando...
 - Un conjunto de **requisitos de partida**

Introducción

¿Qué hemos hecho hasta ahora?

Algunos pasos dados...

- Hemos profundizado en el diseño conceptual, empleando...
 - Un conjunto de **requisitos de partida**
 - Un cierto **formalismo**, el diagrama Entidad-Interrelación

Introducción

¿Qué hemos hecho hasta ahora?

Algunos pasos dados...

- Hemos profundizado en el diseño conceptual, empleando...
 - Un conjunto de **requisitos de partida**
 - Un cierto **formalismo**, el diagrama Entidad-Interrelación
 - Un proceso de **abstracción** para capturar lo relevante de los requisitos y conjugarlo con el formalismo

Introducción

¿Qué hemos hecho hasta ahora?

Algunos pasos dados...

- Hemos profundizado en el diseño conceptual, empleando...
 - Un conjunto de **requisitos de partida**
 - Un cierto **formalismo**, el diagrama Entidad-Interrelación
 - Un proceso de **abstracción** para capturar lo relevante de los requisitos y conjugarlo con el formalismo
 - El **sentido común**, para terminar de ver la división estructural entre entidades, interrelaciones, jerarquías...

Introducción

¿Qué hemos hecho hasta ahora?

Algunos pasos dados...

- Hemos profundizado en el diseño conceptual, empleando...
 - Un conjunto de **requisitos de partida**
 - Un cierto **formalismo**, el diagrama Entidad-Interrelación
 - Un proceso de **abstracción** para capturar lo relevante de los requisitos y conjugarlo con el formalismo
 - El **sentido común**, para terminar de ver la división estructural entre entidades, interrelaciones, jerarquías...
- Así hemos llegado a un esquema conceptual *razonablemente bueno*...

Introducción

¿Qué hemos hecho hasta ahora?

Bueno, ¿qué más hemos hecho?

- Hemos trasladado el esquema conceptual a un esquema lógico, empleando...

Introducción

¿Qué hemos hecho hasta ahora?

Bueno, ¿qué más hemos hecho?

- Hemos trasladado el esquema conceptual a un esquema lógico, empleando...
 - Un **procedimiento sistemático** para tratar cada elemento del esquema conceptual
 - Un **formalismo**, el modelo relacional
 - Una dosis de *saber hacer* para tomar pequeñas decisiones en función de la semántica de los elementos del esquema conceptual

Introducción

¿Qué hemos hecho hasta ahora?

Bueno, ¿qué más hemos hecho?

- Hemos trasladado el esquema conceptual a un esquema lógico, empleando...
 - Un **procedimiento sistemático** para tratar cada elemento del esquema conceptual
 - Un **formalismo**, el modelo relacional
 - Una dosis de *saber hacer* para tomar pequeñas decisiones en función de la semántica de los elementos del esquema conceptual
- Así hemos llegado a un esquema lógico *coherente* con el diseño conceptual, y por tanto fiel a los requisitos iniciales...

Introducción

¿Qué hemos hecho hasta ahora?

Algún paso adicional se ha llevado a cabo...

Introducción

¿Qué hemos hecho hasta ahora?

Algún paso adicional se ha llevado a cabo...

- Hemos llevado el diseño lógico a una **implementación** empleando distintos **SGBD relacionales** como MySQL y Access...

Introducción

¿Qué hemos hecho hasta ahora?

Algún paso adicional se ha llevado a cabo...

- Hemos llevado el diseño lógico a una **implementación** empleando distintos **SGBD relacionales** como MySQL y Access...
- Se han comenzado a plantear **consultas** a realizar sobre la base de datos, confirmando la *idoneidad de las estructuras diseñadas* y finalmente implementadas, de cara a proporcionar la información requerida
- Se han realizado acciones de **manipulación de datos**, comprobando que permite poblar adecuadamente de datos (*insertar*), *actualizar* o *eliminar* algunos de los mismos

Introducción

¿Qué podemos sacar en claro?

Algunas impresiones iniciales...

- Parece que hemos desarrollado una **capacidad para identificar los elementos esenciales** de nuestros problemas y ver relaciones lógicas, de cara a proporcionar una **organización adecuada**.
- El modelo conceptual en que nos hemos basado dota de unos mimbres idóneos para **agrupar** los **atributos** de acuerdo a entes *tangibles* que intuitivamente se pueden **reconocer**, resulta un agrupamiento *natural*.

Introducción

¿Es suficiente?

Bueno, reflexionemos un poco más...

- Correcto, **estamos en el buen camino**, y todo lo anterior nos ha situado en una posición ventajosa

Introducción

¿Es suficiente?

Bueno, reflexionemos un poco más...

- Correcto, **estamos en el buen camino**, y todo lo anterior nos ha situado en una posición ventajosa
- Esto **no** implica que **podamos garantizar** que hemos dotado del mejor esquema lógico posible y nuestra implementación de la base de datos es la mejor que se puede realizar

Introducción

¿Es suficiente?

Bueno, reflexionemos un poco más...

- Correcto, **estamos en el buen camino**, y todo lo anterior nos ha situado en una posición ventajosa
- Esto **no** implica que **podamos garantizar** que hemos dotado del mejor esquema lógico posible y nuestra implementación de la base de datos es la mejor que se puede realizar
- Es necesario incrementar nuestro sustento, mediante teorías para **analizar objetivamente** el **nivel de organización** de una **base de datos** relacional

Introducción

¿Es suficiente?

Bueno, reflexionemos un poco más...

- Correcto, **estamos en el buen camino**, y todo lo anterior nos ha situado en una posición ventajosa
- Esto **no** implica que **podamos garantizar** que hemos dotado del mejor esquema lógico posible y nuestra implementación de la base de datos es la mejor que se puede realizar
- Es necesario incrementar nuestro sustento, mediante teorías para **analizar objetivamente** el **nivel de organización** de una **base de datos** relacional
- Hará falta **medir** en qué punto se encuentra **nuestra BD**, **identificar efectos no deseados** y establecer *medidas* sistemáticas para *reconducirla* hacia una *organización más adecuada*

Introducción

¿Cómo podemos avanzar?

Primeros pasos...

- Analizaremos posibles problemas derivados de un diseño lógico

Introducción

¿Cómo podemos avanzar?

Primeros pasos...

- Analizaremos posibles problemas derivados de un diseño lógico
- Veremos el concepto de dependencia funcional y describiremos sus tipos e implicaciones

Introducción

¿Cómo podemos avanzar?

Primeros pasos...

- Analizaremos posibles problemas derivados de un diseño lógico
- Veremos el concepto de dependencia funcional y describiremos sus tipos e implicaciones
- Estudiaremos los fundamentos de la teoría de la normalización, haciendo uso de los conceptos anteriores

Posibles problemas en un esquema lógico

Sea la relación R(AUTOR,PAÍS,COD-LIBRO,TÍTULO,EDITORIAL)
con el siguiente cuerpo:

AUTOR	PAÍS	COD-LIBRO	TÍTULO	EDITORIAL
Date, C.	USA.	1	DB	Ad
Date, C.	USA	2	SQL(I)	Ad
Gardarin	Chile	3	ModeloER	Verlag
Gardarin	Chile	4	SQL(II)	ACM
Kim, W.	China	4	SQL(II)	ACM

¿Qué problemas observa?

Posibles problemas en un esquema lógico

Sea la relación R(AUTOR,PAÍS,COD-LIBRO,TÍTULO,EDITORIAL) con el siguiente cuerpo:

AUTOR	PAÍS	COD-LIBRO	TÍTULO	EDITORIAL
Date, C.	USA.	1	DB	Ad
Date, C.	USA	2	SQL(I)	Ad
Gardarin	Chile	3	ModeloER	Verlag
Gardarin	Chile	4	SQL(II)	ACM
Kim, W.	China	4	SQL(II)	ACM

¿Qué problemas observa?

1. Redundancia de datos.

Posibles problemas en un esquema lógico

Sea la relación $R(\text{AUTOR}, \text{PAÍS}, \text{COD-LIBRO}, \text{TÍTULO}, \text{EDITORIAL})$ con el siguiente cuerpo:

AUTOR	PAÍS	COD-LIBRO	TÍTULO	EDITORIAL
Date, C.	USA.	1	DB	Ad
Date, C.	USA	2	SQL(I)	Ad
Gardarin	Chile	3	ModeloER	Verlag
Gardarin	Chile	4	SQL(II)	ACM
Kim, W.	China	4	SQL(II)	ACM

¿Qué problemas observa?

1. Redundancia de datos.
2. Incoherencia de datos. Puede ser por...
 - Anomalías de modificación de datos.
 - Anomalías de inserción de datos.
 - Anomalías de borrado de datos.

Medición y soluciones

En este contexto surge la teoría de la normalización en bases de datos, que define una serie de formas normales.

Formas normales (FN)

Conjunto de restricciones sobre tablas relacionales que evitan los problemas de redundancia de datos, y de anomalías de modificación, inserción y borrado de datos.

$$1FN \implies 2FN \implies 3FN \implies FNBC \implies 4FN \implies 5FN$$

Nota: en este curso estudiaremos hasta tercera forma normal, 3FN.

Índice

- 1 Introducción a la normalización
- 2 Dependencias funcionales**
- 3 Formas Normales
- 4 Normalización
- 5 Bibliografía

Dependencias entre atributos

Si las formas normales van a venir a *medir la situación de partida* de nuestro esquema lógico, *identificar problemas* en el mismo y ayudarnos a *mejorarlo*, las dependencias funcionales vendrán a dotarnos de los *elementos básicos* para llevarlo a cabo.

- Entre los atributos de una relación pueden existir **dependencias de varios tipos**.
- Las dependencias son propiedades **inherentes al contenido semántico de los datos**, formando parte de las *restricciones de usuario* del modelo relacional.

Dependencias funcionales

Sea el esquema de relación $R(AT, DEP)$, donde R es la relación, AT su conjunto de atributos y DEP su conjunto de dependencias.

Sean X e Y dos **subconjuntos** de AT denominados descriptores.

Dependencia funcional

Y depende funcionalmente de X si para cada valor de X solo existe un posible valor de Y ; es decir, fijado un valor de los campos que forman X , podemos fijar también el valor que toma cada campo de Y en R . Se escribe $X \rightarrow Y$, leído X *determina* Y , con:

$X \equiv$ determinante o implicante

$Y \equiv$ campo implicado

Nota: $X_1, \dots, X_k \rightarrow Y_1, \dots, Y_n$ significa que cada campo Y_i tiene dependencia funcional respecto del conjunto X_1, \dots, X_k .

Dependencias funcionales. Ejemplo

ESCRIBE(AT={autor,pais,codigo,titulo,editorial,paginas},DEP). DEP:

autor → *pais*

codigo → *titulo* *codigo* → *editorial* *codigo* → *paginas*

codigo → *titulo, editorial, paginas*

codigo, autor → *editorial, pais*

Nota

- No debe confundirse con el concepto de campo derivado. Está claro que si Y es un campo derivado de X (e.g., *edad* derivado de *fecha_nacimiento*), entonces $X \rightarrow Y$.
- Sin embargo, al revés no es cierto. Para que Y tenga dependencia funcional de X , no es necesario que se pueda calcular el valor de Y a partir de X .
- Ejemplo: el campo *fecha_nacimiento* queda fijado cuando sabemos el campo *DNI* (*depende funcionalmente*), pero no podemos calcular la fecha de nacimiento a partir del valor del *DNI* (*no es derivado*).

Dependencias funcionales

Un ejemplo de dependencia funcional es la que existe entre los atributos DNI de profesor y Nombre de Profesor:

- PROFESOR(AT, DEP)
- $AT = \{\text{DNI_profesor}, \text{Nombre_profesor}\}$
- $DEP = \{\text{DNI_profesor} \rightarrow \text{Nombre_profesor}\}$

Si además se cumpliera que no pueden existir dos profesores con el mismo nombre, esto implicaría que el nombre también puede actuar como clave de la relación PROFESOR, y por tanto también determina funcionalmente a su DNI, es decir, se tendría que:

- $\text{Nombre_profesor} \leftrightarrow \text{DNI_profesor}$

Dependencia funcional plena

Diremos que Y tiene dependencia funcional **plena** o **completa** del conjunto de campos X_1, \dots, X_k si **depende funcionalmente** de dicho conjunto, pero **NO depende de ningún subconjunto propio** de X_1, \dots, X_k .

Ejemplos: EVAL(alumno, asignatura, conv, dpto, nota)

- El campo *nota* tiene dependencia plena de *alumno, asignatura, conv*
 $\text{alumno, asignatura, conv} \rightarrow \text{nota}$
 $\text{alumno} \nrightarrow \text{nota}$
 $\text{asignatura} \nrightarrow \text{nota}$
 $\text{alumno, asignatura} \nrightarrow \text{nota}$
- El campo *dpto* **NO** tiene dependencia plena de *alumno, asignatura, conv*
 $\text{alumno, asignatura, conv} \rightarrow \text{dpto}$
 $\text{asignatura} \rightarrow \text{dpto}$

Dependencia funcional trivial

Diremos que Y tiene dependencia funcional **trivial** del conjunto de campos X_1, \dots, X_k si Y es un subconjunto de los campos X_1, \dots, X_k .

Ejemplo: algunas dependencias funcionales triviales

- *autor, codigo* \rightarrow *autor*
- *alumno, asignatura, conv, nota* \rightarrow *nota*
- *libro* \rightarrow *libro*
- *DNI, Nombre_profesor, Depto* \rightarrow *Nombre_profesor*

Dependencia funcional elemental (I)

El campo Y tiene dependencia funcional elemental del conjunto X_1, \dots, X_k si tiene dependencia funcional y es:

- **completa**
- **no trivial**

Nota

En una **dependencia funcional elemental** $X_1, \dots, X_k \rightarrow Y$, el **conjunto implicado** Y siempre ha de ser **unitario**.

Ejemplo:

$$DNI_profesor \rightarrow Nombre_profesor$$

Dependencia funcional elemental (II)

Ejemplo: EVAL(alumno,edad,asignatura,depto,conv,nota)

Un conjunto de dependencias funcionales elementales que describen la relación EVAL es:

- *alumno, asignatura, conv* \rightarrow *nota*
- *alumno* \rightarrow *edad*
- *asignatura* \rightarrow *depto*

Nota

Para la normalización de una tabla relacional, solamente se tienen en cuenta las dependencias funcionales elementales.

Campos equivalentes

Cuando dos o más atributos se implican funcionalmente mutuamente se dice que son equivalentes. Formalizando:

Campos equivalentes

Sea R una tabla relacional y $X_1, \dots, X_k, Y_1, \dots, Y_n$ campos de R . Diremos que X_1, \dots, X_k e Y_1, \dots, Y_n son **conjuntos de campos equivalentes** si:

- X_1, \dots, X_k depende funcionalmente de Y_1, \dots, Y_n , e
- Y_1, \dots, Y_n depende funcionalmente de X_1, \dots, X_k

Lo escribiremos:

$$X_1, \dots, X_k \leftrightarrow Y_1, \dots, Y_n$$

Nota: Habitualmente (no siempre, naturalmente) $k = n = 1$, hablaremos de **campos equivalentes** de una relación (tabla).

Dependencia funcional transitiva

- Existe una **dependencia funcional transitiva** $X \rightarrow Z$, si dado el esquema de la relación $R(AT=\{X,Y,Z\}, DEP)$, en DEP existen las dependencias $X \rightarrow Y$, $Y \rightarrow Z$ y además $Y \nrightarrow X$
- Ejemplo:
 - $DNI_profesor \rightarrow C\acute{o}digo_Postal$;
 - $C\acute{o}digo_Postal \rightarrow Ciudad$;
 - $C\acute{o}digo_Postal \nrightarrow DNI_profesor$

Dependencia funcional transitiva

Conjunto de campos

Dependencia funcional transitiva

El campo Z tiene **dependencia funcional transitiva** respecto de los campos X_1, \dots, X_k a través de los campos Y_1, \dots, Y_n si se cumple que:

- $X_1, \dots, X_k \rightarrow Y_1, \dots, Y_n$ y
- $Y_1, \dots, Y_n \rightarrow Z$, pero
- $Y_1, \dots, Y_n \not\rightarrow X_1, \dots, X_k$

Dependencia funcional transitiva. Ejemplos

Ejemplo 1: LIBRO(isbn,paginas,autor,pais)

Asumiendo que cada libro tenga únicamente un autor,
¿depende *pais* **transitivamente** de *isbn* a través de *autor*?

Dependencia funcional transitiva. Ejemplos

Ejemplo 1: LIBRO(isbn,paginas,autor,pais)

Asumiendo que cada libro tenga únicamente un autor,
¿depende *pais* **transitivamente** de *isbn* a través de *autor*? **SÍ**, pues:

- $isbn \rightarrow autor$ y
- $autor \rightarrow pais$, pero
- $autor \nrightarrow isbn$

Dependencia funcional transitiva. Ejemplos

Ejemplo 1: LIBRO(isbn,paginas,autor,pais)

Asumiendo que cada libro tenga únicamente un autor,
¿depende *pais* **transitivamente** de *isbn* a través de *autor*? **SÍ**, pues:

- $isbn \rightarrow autor$ y
- $autor \rightarrow pais$, pero
- $autor \nrightarrow isbn$

Ejemplo 2: CORREO(username,NIF,nombre,apellidos,edad,pais)

¿Depende *pais* **transitivamente** de *NIF* a través de *username*?

Dependencia funcional transitiva. Ejemplos

Ejemplo 1: LIBRO(isbn,paginas,autor,pais)

Asumiendo que cada libro tenga únicamente un autor,
¿depende *pais* **transitivamente** de *isbn* a través de *autor*? **SÍ**, pues:

- $isbn \rightarrow autor$ y
- $autor \rightarrow pais$, pero
- $autor \nrightarrow isbn$

Ejemplo 2: CORREO(username,NIF,nombre,apellidos,edad,pais)

¿Depende *pais* **transitivamente** de *NIF* a través de *username*?
NO, puesto que:

- $NIF \rightarrow username$ y
- $username \rightarrow pais$, pero
- $username \rightarrow NIF$

Nota: *NIF* y *username* son campos equivalentes.

Dependencia funcional transitiva y campos equivalentes

Dada esta relación y sus dependencias funcionales:

VENTA(codigo,numero,vendedor,articulo,color,precio)

- $codigo \leftrightarrow numero, vendedor$
- $codigo \rightarrow articulo$
- $articulo \rightarrow color, precio$

¿Depende *articulo* **transitivamente** del conjunto de campos *numero, vendedor*?

Dependencia funcional transitiva y campos equivalentes

Dada esta relación y sus dependencias funcionales:

VENTA(codigo,numero,vendedor,articulo,color,precio)

- $codigo \leftrightarrow numero, vendedor$
- $codigo \rightarrow articulo$
- $articulo \rightarrow color, precio$

¿Depende *articulo* **transitivamente** del conjunto de campos *numero, vendedor*? **NO**, ya que:

- $numero, vendedor \rightarrow codigo$
- $codigo \rightarrow articulo$
- $codigo \rightarrow numero, vendedor$

Campos equivalentes. Ejemplo 2

VENTA(codigo,numero,vendedor,articulo,color,precio)

- $codigo \leftrightarrow numero, vendedor$
- $codigo \rightarrow articulo$
- $articulo \rightarrow color, precio$

¿Qué hay de *precio*? ¿Depende **transitivamente** de *numero, vendedor*?

Campos equivalentes. Ejemplo 2

VENTA(codigo,numero,vendedor,articulo,color,precio)

- $codigo \leftrightarrow numero, vendedor$
- $codigo \rightarrow articulo$
- $articulo \rightarrow color, precio$

¿Qué hay de *precio*? ¿Depende **transitivamente** de *numero, vendedor*? **SÍ**, pues:

- $numero, vendedor \rightarrow articulo$
- $articulo \rightarrow precio$
- $articulo \nrightarrow numero, vendedor$

Axiomas de Armstrong

Suponiendo que X , Y y Z son conjuntos de atributos, los Axiomas de Armstrong nos permiten deducir unas dependencias a partir de otras, y son los siguientes:

- **Reflexividad:** Para todo X , tenemos que $X \rightarrow X$.
- **Proyectividad:** Si $X \rightarrow Y$ y $Z \subseteq Y$, entonces $X \rightarrow Z$.
- **Aumentatividad:** Si $X \rightarrow Y$ y $Z \supseteq X$, entonces $Z \rightarrow Y$.
- **Aditividad:** Si $X \rightarrow Y$ y $Z \rightarrow V$, entonces $X \cup Z \rightarrow Y \cup V$.
- **Transitividad:** Si $X \rightarrow Z$ y $Z \rightarrow Y$, entonces $X \rightarrow Y$.
(No confundir con dependencia funcional transitiva.)

Atributos extraños

- Un atributo A perteneciente a X es **extraño** en la dependencia $X \rightarrow Y$ si la dependencia $(X - A) \rightarrow Y$ ya se deduce del resto de las dependencias de la relación (*i.e.*, podemos llegar a la misma dependencia usando los axiomas de Armstrong).
- Por ejemplo, consideremos

$$R(\{A, B, C, D\}, \{A \rightarrow B, B \rightarrow C, AD \rightarrow C\})$$

Puesto que $A \rightarrow C$ se deduce de las dependencias $A \rightarrow B$ y $B \rightarrow C$ por transitividad, entonces D es extraño en la dependencia $AD \rightarrow C$.

Dependencias redundantes

- Una **dependencia** es **redundante** si sus implicados se deducen a partir del resto de las dependencias de la relación (empleando los axiomas de Armstrong).
- Por ejemplo, sea

$$R(\{A, B, C\}, \{A \rightarrow B, B \rightarrow C, A \rightarrow C\})$$

Como el atributo A determina transitivamente al atributo C , la última dependencia $A \rightarrow C$ es una ***dependencia redundante***.

Recubrimiento Minimal

Un **conjunto de dependencias** es **mínimo**, y se denota como DEP^m , cuando se cumplen las siguientes condiciones:

- **Todas** sus **dependencias** son **elementales**, es decir, todas son plenas, no triviales y con un único atributo implicado.
 - **No** existe en ninguna de las dependencias **atributos extraños**.
- **No** existe en la relación ninguna **dependencia redundante**.

Nota: puede ocurrir que existan varios conjuntos de dependencias mínimos válidos, ya que todo depende de la elección de las dependencias redundantes y de los atributos extraños que se eliminan en el proceso del cálculo del conjunto minimal.

Recubrimiento Minimal

Cálculo

Para el cálculo del recubrimiento minimal, partiendo de un conjunto de dependencias dado, hacemos lo siguiente:

1. **Descomponer** las **partes derechas** de las dependencias, de forma que ***todas sean unitarias***.
2. **Eliminar** atributos **extraños**.
3. **Eliminar** dependencias **redundantes**.

Cierre Transitivo

- Dado un conjunto de dependencias funcionales F , el **cierre transitivo** de F , F^+ , es el conjunto de dependencias funcionales implicadas lógicamente por F .

$$F = \{A \rightarrow B, B \rightarrow C\} \quad F^+ = \{A \rightarrow B, B \rightarrow C, A \rightarrow C\}$$

- Dado un conjunto de atributos X , el **cierre transitivo de X** se define como el conjunto de atributos determinados por X aplicando los axiomas de Armstrong, y se escribe X^+ .
- Por ejemplo, en el esquema

$$R(\{A, B, C, D\}, \{A \rightarrow B, D; B \rightarrow C\})$$

- $A^+ = \{A, B, C, D\}$
- $B^+ = \{B, C\}$
- $C^+ = \{C\}$
- $D^+ = \{D\}$

Claves candidatas

Superclave

Dado un esquema de relación $R(AT, DEP)$, se denomina **superclave** a todo subconjunto de AT tal que determina completamente al conjunto AT , o lo que es lo mismo, cuyo cierre transitivo es AT .

Clave candidata

Denominamos **clave candidata** a toda superclave tal que ningún subconjunto *propio* de la misma es a su vez superclave; dicho de otra forma, es toda superclave tal que ningún subconjunto *propio* determina completamente al conjunto AT .

¿Cómo encontrar todas las claves candidatas?

- Podemos emplear los cierres transitivos de cada subconjunto del conjunto de atributos. Tenemos dos opciones:
 - Por fuerza bruta, ir probando todos los subconjuntos de atributos cuyo cierre transitivo coincida con AT y que no incluyan otros subconjuntos propios.
 - Guiar el proceso de forma más inteligente, a partir de DEP, partiendo de atributos adecuados y descartando irrelevantes. Debemos tener en cuenta que:
 - Todo atributo que no aparece en las dependencias funcionales o solo lo hace como implicante forma parte de toda clave.
 - Descriptores equivalentes conducen a varias claves.
 - Atributos implicados no implicantes no son clave.

Proceso de cálculo de claves candidatas, CCS

1. Seleccionar los atributos que no aparecen en la parte derecha de ninguna dependencia funcional \rightarrow Son parte de toda clave. Podemos denotarlo por ejemplo por I (imprescindibles).
2. Calcular su cierre I^+ , y comprobar si coincide con AT :
 - Si coincide, entonces $CCS = \{I\}$, única clave (cualquier otra superclave debe contener I y ya no sería minimal).
 - Si no, descartamos los atributos que solo aparecen en la parte derecha (D) y analizamos el resto, $R = AT - I - D$. Para obtener las claves candidatas CCS , partiremos del conjunto de “claves parciales” $CPS = \{(I, R)\}$.
 - Guardaremos junto a cada clave parcial los atributos que le falten analizar a la misma.

Algoritmo de cálculo de claves candidatas, CCS

```
if  $I^+ = AT$  then
|    $CCS \leftarrow \{I\}$ 
else
|    $CCS \leftarrow \{\}$ ,  $CPS \leftarrow \{(I, R)\}$ 
|   while  $CPS$  no vacío do
|       |    $(CP, R) \leftarrow$  primero extraído de  $CPS$ 
|       |    $R \leftarrow R \setminus CP^+$ 
|       |   foreach atributo  $A$  en  $R$  do
|           |    $CPA \leftarrow CP \cup \{A\}$ 
|           |    $R \leftarrow R \setminus \{A\}$ 
|           |   if  $CPA^+ = AT$  then
|               |   Añadir  $CPA$  al final de  $CCS$ 
|           |   else if  $R$  no vacío then
|               |   Añadir  $(CPA, R)$  al final de  $CPS$ 
|           end
|       end
|   end
end
```

Campos principales y no principales

Campo principal

Un campo de una tabla relacional R es **principal** si pertenece a alguna de las claves candidatas de R .

Campo no principal

Todo aquel campo que no sea principal.

Campos principales y no principales. Ejemplo

VENTA(codigo,numero,vendedor,articulo,color,precio)

- Dependencias elementales:
 - $codigo \rightarrow numero, \quad codigo \rightarrow vendedor, \quad codigo \rightarrow articulo$
 - $numero, vendedor \rightarrow codigo$
 - $articulo \rightarrow color, \quad color \rightarrow precio$
- Claves candidatas:

Campos principales y no principales. Ejemplo

VENTA(codigo,numero,vendedor,articulo,color,precio)

- Dependencias elementales:
 - $codigo \rightarrow numero, \quad codigo \rightarrow vendedor, \quad codigo \rightarrow articulo$
 - $numero, vendedor \rightarrow codigo$
 - $articulo \rightarrow color, \quad color \rightarrow precio$
- Claves candidatas:
 1. $codigo$
 2. $(numero, vendedor)$
- Campos equivalentes:

Campos principales y no principales. Ejemplo

VENTA(codigo,numero,vendedor,articulo,color,precio)

- Dependencias elementales:
 - $codigo \rightarrow numero, \quad codigo \rightarrow vendedor, \quad codigo \rightarrow articulo$
 - $numero, vendedor \rightarrow codigo$
 - $articulo \rightarrow color, \quad color \rightarrow precio$
- Claves candidatas:
 1. $codigo$
 2. $(numero, vendedor)$
- Campos equivalentes: $codigo \leftrightarrow numero, vendedor$
- Campos principales:

Campos principales y no principales. Ejemplo

VENTA(codigo,numero,vendedor,articulo,color,precio)

- Dependencias elementales:
 - *codigo* → *numero*, *codigo* → *vendedor*,
 codigo → *articulo*
 - *numero*, *vendedor* → *codigo*
 - *articulo* → *color*, *color* → *precio*
- Claves candidatas:
 1. *codigo*
 2. (*numero*, *vendedor*)
- Campos equivalentes: *codigo* ↔ *numero*, *vendedor*
- Campos principales: *codigo*, *numero*, *vendedor*
- Campos no principales:

Campos principales y no principales. Ejemplo

VENTA(codigo,numero,vendedor,articulo,color,precio)

- Dependencias elementales:
 - *codigo* → *numero*, *codigo* → *vendedor*,
 codigo → *articulo*
 - *numero*, *vendedor* → *codigo*
 - *articulo* → *color*, *color* → *precio*
- Claves candidatas:
 1. *codigo*
 2. (*numero*, *vendedor*)
- Campos equivalentes: *codigo* ↔ *numero*, *vendedor*
- Campos principales: *codigo*, *numero*, *vendedor*
- Campos no principales: *articulo*, *color*, *precio*

Conjuntos de Dependencias Equivalentes

Dos **conjuntos de dependencias**, L y M , relativas al mismo universo T de atributos, son **equivalentes** si $L^+ = M^+$, en cuyo caso $R(T, L)$ y $S(T, M)$ son **esquemas equivalentes**, i.e., representaciones alternativas y equivalentes del mismo problema.

Grafo de Dependencias Funcionales (GDF)

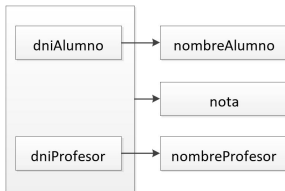
Grafo dirigido que captura las dependencias funcionales de la relación, representando los nodos del grafo tanto los atributos (nodos simples) como los conjuntos de atributos (burbujas) que actúan juntos como implicantes en alguna dependencia funcional, y los arcos las dependencias desde implicantes a implicados. Generalmente se representan dependencias que van de un nodo (sea simple o burbuja) a un atributo (nodo simple).

Ej: sea la relación $R(AT, DEP)$, con

$AT = \{dniProfesor, dniAlumno, nombreProfesor, nombreAlumno, nota\}$ y

$DEP = dniProfesor \rightarrow nombreProfesor; dniAlumno \rightarrow nombreAlumno; dniProfesor, dniAlumno \rightarrow nota,$

el GDF quedaría así:



Índice

- 1 Introducción a la normalización
- 2 Dependencias funcionales
- 3 Formas Normales**
- 4 Normalización
- 5 Bibliografía

Primera Forma Normal: 1FN

Definición (Codd, 1970)

Una tabla relacional R está en **primera forma normal (1FN)** si ***NO contiene campos multivaluados*** (cada atributo 1 valor).

Nota

Por la propia definición del modelo de datos relacional, NO admitimos campos multivaluados.

En consecuencia, TODAS las relaciones del modelo de datos relacional que creamos están automáticamente en 1FN.

Caminando hacia 2FN y 3FN

Para estudiar si una tabla R está en **2FN** o **3FN** se parte de:

1. Conjunto de las dependencias elementales de R
2. Conjunto de todas las claves candidatas para R
3. Conjunto de los campos equivalentes de R
4. Conjunto de los campos principales de R
5. Conjunto de los campos no principales de R

Segunda Forma Normal: 2FN

Definición (Codd, 1970)

Una tabla relacional R está en **segunda forma normal (2FN)** si:

- R está en **1FN**, y
- cada campo **no principal** de R tiene ***dependencia funcional plena*** respecto de cada una de las **claves candidatas** de R .

Nota

Si R tiene k claves candidatas y m campos no principales, entonces para asegurar que R está en **2FN** son necesarias $m * k$ comprobaciones.

- ¿ $R1(\{A, B, C, D\}, \{A, B \rightarrow C; A \rightarrow D\})$?

Segunda Forma Normal: 2FN

Definición (Codd, 1970)

Una tabla relacional R está en **segunda forma normal (2FN)** si:

- R está en **1FN**, y
- cada campo **no principal** de R tiene *dependencia funcional plena* respecto de cada una de las **claves candidatas** de R .

Nota

Si R tiene k claves candidatas y m campos no principales, entonces para asegurar que R está en **2FN** son necesarias $m * k$ comprobaciones.

- ¿ $R1(\{A, B, C, D\}, \{A, B \rightarrow C; A \rightarrow D\})$? **Relación no en 2FN**

Segunda Forma Normal: 2FN

Definición (Codd, 1970)

Una tabla relacional R está en **segunda forma normal (2FN)** si:

- R está en **1FN**, y
- cada campo **no principal** de R tiene *dependencia funcional plena* respecto de cada una de las **claves candidatas** de R .

Nota

Si R tiene k claves candidatas y m campos no principales, entonces para asegurar que R está en **2FN** son necesarias $m * k$ comprobaciones.

- ¿ $R1(\{A, B, C, D\}, \{A, B \rightarrow C; A \rightarrow D\})$? **Relación no en 2FN**
- ¿ $R2(\{A, B, C, D\}, \{A, B \rightarrow C; A, B \rightarrow D\})$?

Segunda Forma Normal: 2FN

Definición (Codd, 1970)

Una tabla relacional R está en **segunda forma normal (2FN)** si:

- R está en **1FN**, y
- cada campo **no principal** de R tiene ***dependencia funcional plena*** respecto de cada una de las **claves candidatas** de R .

Nota

Si R tiene k claves candidatas y m campos no principales, entonces para asegurar que R está en **2FN** son necesarias $m * k$ comprobaciones.

- ¿ $R1(\{A, B, C, D\}, \{A, B \rightarrow C; A \rightarrow D\})$? **Relación no en 2FN**
- ¿ $R2(\{A, B, C, D\}, \{A, B \rightarrow C; A, B \rightarrow D\})$? **Relación en 2FN**

Campos principales y no principales. Ejemplo

EVAL(alumno,edad,asignatura,dpto,conv,nota)

- Dependencias funcionales elementales:

Campos principales y no principales. Ejemplo

EVAL(alumno,edad,asignatura,dpto,conv,nota)

- Dependencias funcionales elementales:
 - *alumno* → *edad*

Campos principales y no principales. Ejemplo

EVAL(alumno,edad,asignatura,dpto,conv,nota)

- Dependencias funcionales elementales:
 - *alumno* \rightarrow *edad*
 - *asignatura* \rightarrow *dpto*

Campos principales y no principales. Ejemplo

EVAL(alumno,edad,asignatura,dpto,conv,nota)

- Dependencias funcionales elementales:
 - $alumno \rightarrow edad$
 - $asignatura \rightarrow dpto$
 - $alumno, asignatura, conv \rightarrow nota$

Campos principales y no principales. Ejemplo

EVAL(alumno,edad,asignatura,dpto,conv,nota)

- Dependencias funcionales elementales:
 - $alumno \rightarrow edad$
 - $asignatura \rightarrow dpto$
 - $alumno, asignatura, conv \rightarrow nota$
- Claves candidatas:

Campos principales y no principales. Ejemplo

EVAL(*alumno*,*edad*,*asignatura*,*dpto*,*conv*,*nota*)

- Dependencias funcionales elementales:
 - $alumno \rightarrow edad$
 - $asignatura \rightarrow dpto$
 - $alumno, asignatura, conv \rightarrow nota$
- Claves candidatas:
 - $(alumno, asignatura, conv)$ (i.e. $k = 1$)
- Campos equivalentes:

Campos principales y no principales. Ejemplo

EVAL(alumno,edad,asignatura,dpto,conv,nota)

- Dependencias funcionales elementales:
 - $alumno \rightarrow edad$
 - $asignatura \rightarrow dpto$
 - $alumno, asignatura, conv \rightarrow nota$
- Claves candidatas:
 - $(alumno, asignatura, conv)$ (i.e. $k = 1$)
- Campos equivalentes: \neq

Campos principales y no principales. Ejemplo

EVAL(*alumno*,*edad*,*asignatura*,*dpto*,*conv*,*nota*)

- Dependencias funcionales elementales:
 - $alumno \rightarrow edad$
 - $asignatura \rightarrow dpto$
 - $alumno, asignatura, conv \rightarrow nota$
- Claves candidatas:
 - $(alumno, asignatura, conv)$ (i.e. $k = 1$)
- Campos equivalentes: \neq
- Campos principales:

Campos principales y no principales. Ejemplo

EVAL(*alumno*, *edad*, *asignatura*, *dpto*, *conv*, *nota*)

- Dependencias funcionales elementales:
 - $alumno \rightarrow edad$
 - $asignatura \rightarrow dpto$
 - $alumno, asignatura, conv \rightarrow nota$
- Claves candidatas:
 - $(alumno, asignatura, conv)$ (i.e. $k = 1$)
- Campos equivalentes: \neq
- Campos principales: $alumno, asignatura, conv$

Campos principales y no principales. Ejemplo

EVAL(*alumno*, *edad*, *asignatura*, *dpto*, *conv*, *nota*)

- Dependencias funcionales elementales:
 - $alumno \rightarrow edad$
 - $asignatura \rightarrow dpto$
 - $alumno, asignatura, conv \rightarrow nota$
- Claves candidatas:
 - $(alumno, asignatura, conv)$ (i.e. $k = 1$)
- Campos equivalentes: \neq
- Campos principales: *alumno*, *asignatura*, *conv*
- Campos no principales:

Campos principales y no principales. Ejemplo

EVAL(*alumno*, *edad*, *asignatura*, *dpto*, *conv*, *nota*)

- Dependencias funcionales elementales:
 - $alumno \rightarrow edad$
 - $asignatura \rightarrow dpto$
 - $alumno, asignatura, conv \rightarrow nota$
- Claves candidatas:
 - $(alumno, asignatura, conv)$ (i.e. $k = 1$)
- Campos equivalentes: \neq
- Campos principales: $alumno, asignatura, conv$
- Campos no principales: $edad, dpto, nota$ (i.e. $m = 3$)

Número de comprobaciones:

Campos principales y no principales. Ejemplo

EVAL(*alumno*, *edad*, *asignatura*, *dpto*, *conv*, *nota*)

- Dependencias funcionales elementales:
 - $alumno \rightarrow edad$
 - $asignatura \rightarrow dpto$
 - $alumno, asignatura, conv \rightarrow nota$
 - Claves candidatas:
 - $(alumno, asignatura, conv)$ (i.e. $k = 1$)
 - Campos equivalentes: \neq
 - Campos principales: $alumno, asignatura, conv$
 - Campos no principales: $edad, dpto, nota$ (i.e. $m = 3$)
- Número de comprobaciones: $m * k = 3 * 1 = 3$

Campos principales y no principales. Ejemplo

¿Está en 2FN?

EVAL(alumno,edad,asignatura,dpto,conv,nota)

Comprobaciones:

Campos principales y no principales. Ejemplo

¿Está en 2FN?

Eval(alumno, edad, asignatura, dpto, conv, nota)

Comprobaciones:

1. ¿Es *alumno, asignatura, conv* \rightarrow *edad* completa?

Campos principales y no principales. Ejemplo

¿Está en 2FN?

Eval(alumno, edad, asignatura, dpto, conv, nota)

Comprobaciones:

1. ¿Es *alumno, asignatura, conv* \rightarrow *edad* completa? **NO** (N1)

Campos principales y no principales. Ejemplo

¿Está en 2FN?

EVAL(alumno,edad,asignatura,dpto,conv,nota)

Comprobaciones:

1. ¿Es *alumno, asignatura, conv* \rightarrow *edad* completa? **NO** (N1)
2. ¿Es *alumno, asignatura, conv* \rightarrow *dpto* completa?

Campos principales y no principales. Ejemplo

¿Está en 2FN?

EVAL(alumno,edad,asignatura,dpto,conv,nota)

Comprobaciones:

1. ¿Es *alumno, asignatura, conv* \rightarrow *edad* completa? **NO** (N1)
2. ¿Es *alumno, asignatura, conv* \rightarrow *dpto* completa? **NO** (N2)

Campos principales y no principales. Ejemplo

¿Está en 2FN?

EVAL(alumno,edad,asignatura,dpto,conv,nota)

Comprobaciones:

1. ¿Es *alumno, asignatura, conv* \rightarrow *edad* completa? **NO** (N1)
2. ¿Es *alumno, asignatura, conv* \rightarrow *dpto* completa? **NO** (N2)
3. ¿Es *alumno, asignatura, conv* \rightarrow *nota* completa?

Campos principales y no principales. Ejemplo

¿Está en 2FN?

EVAL(alumno,edad,asignatura,dpto,conv,nota)

Comprobaciones:

1. ¿Es *alumno, asignatura, conv* → *edad* completa? **NO** (N1)
2. ¿Es *alumno, asignatura, conv* → *dpto* completa? **NO** (N2)
3. ¿Es *alumno, asignatura, conv* → *nota* completa? **SÍ**

Campos principales y no principales. Ejemplo

¿Está en 2FN?

EVAL(alumno,edad,asignatura,dpto,conv,nota)

Comprobaciones:

1. ¿Es *alumno, asignatura, conv* → *edad* completa? **NO** (N1)
2. ¿Es *alumno, asignatura, conv* → *dpto* completa? **NO** (N2)
3. ¿Es *alumno, asignatura, conv* → *nota* completa? **SÍ**

¿Conclusión?

Campos principales y no principales. Ejemplo

¿Está en 2FN?

EVAL(alumno, edad, asignatura, dpto, conv, nota)

Comprobaciones:

1. ¿Es *alumno, asignatura, conv* \rightarrow *edad* completa? **NO** (N1)
2. ¿Es *alumno, asignatura, conv* \rightarrow *dpto* completa? **NO** (N2)
3. ¿Es *alumno, asignatura, conv* \rightarrow *nota* completa? **SÍ**

¿Conclusión?

EVAL no está en 2FN porque el campo no principal *edad* no tiene dependencia plena respecto de la clave (*alumno, asignatura, conv*) (lo mismo ocurre con *depto*, pero con encontrar una dependencia no plena ya podemos afirmar que no estamos en 2FN).

Campos principales y no principales. Ejemplo

Normalización

EVAL(alumno,edad,asignatura,dpto,conv,nota)

¿Cómo conseguir un conjunto de tablas en **2FN** equivalente?

Campos principales y no principales. Ejemplo

Normalización

EVAL(alumno,edad,asignatura,dpto,conv,nota)

¿Cómo conseguir un conjunto de tablas en **2FN** equivalente?

- $R1(\textit{alumno}, \textit{edad})$ (por $(N1)$)
 $PK = \textit{alumno}$

Campos principales y no principales. Ejemplo

Normalización

EVAL(*alumno*, *edad*, *asignatura*, *dpto*, *conv*, *nota*)

¿Cómo conseguir un conjunto de tablas en **2FN** equivalente?

- $R1(\textit{alumno}, \textit{edad})$ (por $(N1)$)
 $PK = \textit{alumno}$
- $R2(\textit{asignatura}, \textit{dpto})$ (por $(N2)$)
 $PK = \textit{asignatura}$

Campos principales y no principales. Ejemplo

Normalización

EVAL(*alumno*, *edad*, *asignatura*, *dpto*, *conv*, *nota*)

¿Cómo conseguir un conjunto de tablas en **2FN** equivalente?

- $R1(\textit{alumno}, \textit{edad})$ (por (N1))
 $PK = \textit{alumno}$
- $R2(\textit{asignatura}, \textit{dpto})$ (por (N2))
 $PK = \textit{asignatura}$
- $R3(\textit{alumno}, \textit{asignatura}, \textit{conv}, \textit{nota})$
 $PK = (\textit{alumno}, \textit{asignatura}, \textit{conv})$
 $FK = \textit{alumno}(\rightsquigarrow R1) \quad FK = \textit{asignatura}(\rightsquigarrow R2)$

Segunda Forma Normal. Propiedades

Si el implicante de una dependencia funcional es unitario, entonces la dependencia es automáticamente completa. Por tanto:

Propiedad 1

Si todas las **claves candidatas** de una tabla relacional R en **1FN** son **unitarias**, R está **automáticamente** en **2FN**.

Ejemplo: **ALUMNO**(codigo,nif,nombre,edad,direccion)

- Claves candidatas:

Segunda Forma Normal. Propiedades

Si el implicante de una dependencia funcional es unitario, entonces la dependencia es automáticamente completa. Por tanto:

Propiedad 1

Si todas las **claves candidatas** de una tabla relacional R en **1FN** son **unitarias**, R está **automáticamente** en **2FN**.

Ejemplo: **ALUMNO**(codigo,nif,nombre,edad,direccion)

- Claves candidatas:

1. *codigo*

Segunda Forma Normal. Propiedades

Si el implicante de una dependencia funcional es unitario, entonces la dependencia es automáticamente completa. Por tanto:

Propiedad 1

Si todas las **claves candidatas** de una tabla relacional R en **1FN** son **unitarias**, R está **automáticamente** en **2FN**.

Ejemplo: **ALUMNO**(codigo,nif,nombre,edad,direccion)

- Claves candidatas:

1. *codigo*
2. *nif*

Segunda Forma Normal. Propiedades

Si el implicante de una dependencia funcional es unitario, entonces la dependencia es automáticamente completa. Por tanto:

Propiedad 1

Si todas las **claves candidatas** de una tabla relacional R en **1FN** son **unitarias**, R está **automáticamente** en **2FN**.

Ejemplo: **ALUMNO**(codigo,nif,nombre,edad,direccion)

- Claves candidatas:

1. *codigo*
2. *nif*

¿Conclusión?

Segunda Forma Normal. Propiedades

Si el implicante de una dependencia funcional es unitario, entonces la dependencia es automáticamente completa. Por tanto:

Propiedad 1

Si todas las **claves candidatas** de una tabla relacional R en **1FN** son **unitarias**, R está **automáticamente** en **2FN**.

Ejemplo: **ALUMNO**(codigo,nif,nombre,edad,direccion)

- Claves candidatas:

1. *codigo*
2. *nif*

¿Conclusión? Puesto que no hay atributos multivaluados y ambas claves candidatas son unitarias, la tabla **ALUMNO** está en **2FN**.

Tercera Forma Normal: 3FN

Definición (Cood, 1970)

Una tabla relacional R está en **tercera forma normal (3FN)** si:

- R está en **2FN**, y
- *ningún campo no principal* de R *depende transitivamente* de alguna clave candidata de R .

Nota

Si R está en **2FN** y tiene k claves candidatas y m campos no principales, entonces para asegurar que R también está en **3FN** son necesarias $m * k$ comprobaciones.

- ¿ $R1(\{A, B, C\}, \{A \rightarrow B; A \rightarrow C\})$?

Tercera Forma Normal: 3FN

Definición (Cood, 1970)

Una tabla relacional R está en **tercera forma normal (3FN)** si:

- R está en **2FN**, y
- *ningún campo no principal* de R *depende transitivamente* de alguna clave candidata de R .

Nota

Si R está en **2FN** y tiene k claves candidatas y m campos no principales, entonces para asegurar que R también está en **3FN** son necesarias $m * k$ comprobaciones.

- ¿ $R1(\{A, B, C\}, \{A \rightarrow B; A \rightarrow C\})$? **Relación en 3FN**

Tercera Forma Normal: 3FN

Definición (Cood, 1970)

Una tabla relacional R está en **tercera forma normal (3FN)** si:

- R está en **2FN**, y
- *ningún campo no principal* de R *depende transitivamente* de alguna clave candidata de R .

Nota

Si R está en **2FN** y tiene k claves candidatas y m campos no principales, entonces para asegurar que R también está en **3FN** son necesarias $m * k$ comprobaciones.

- ¿ $R1(\{A, B, C\}, \{A \rightarrow B; A \rightarrow C\})$? **Relación en 3FN**
- ¿ $R2(\{A, B, C\}, \{A \rightarrow B; B \rightarrow C\})$?

Tercera Forma Normal: 3FN

Definición (Cood, 1970)

Una tabla relacional R está en **tercera forma normal (3FN)** si:

- R está en **2FN**, y
- *ningún campo no principal* de R *depende transitivamente* de alguna clave candidata de R .

Nota

Si R está en **2FN** y tiene k claves candidatas y m campos no principales, entonces para asegurar que R también está en **3FN** son necesarias $m * k$ comprobaciones.

- ¿ $R1(\{A, B, C\}, \{A \rightarrow B; A \rightarrow C\})$? **Relación en 3FN**
- ¿ $R2(\{A, B, C\}, \{A \rightarrow B; B \rightarrow C\})$? **Relación no en 3FN**

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - *codigo* → *isbn*

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
- Claves candidatas:

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
- Claves candidatas:
 1. $codigo$
 2. $isbn$

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
- Claves candidatas:
 1. $codigo$
 2. $isbn$
- Campos equivalentes:

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
- Claves candidatas:
 1. $codigo$
 2. $isbn$
- Campos equivalentes: $codigo \leftrightarrow isbn$

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
- Claves candidatas:
 1. $codigo$
 2. $isbn$
- Campos equivalentes: $codigo \leftrightarrow isbn$
- Campos principales:

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
- Claves candidatas:
 1. $codigo$
 2. $isbn$
- Campos equivalentes: $codigo \leftrightarrow isbn$
- Campos principales: $codigo, isbn$

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
- Claves candidatas:
 1. $codigo$
 2. $isbn$
- Campos equivalentes: $codigo \leftrightarrow isbn$
- Campos principales: $codigo, isbn$
- Campos no principales:

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
- Claves candidatas:
 1. $codigo$
 2. $isbn$
- Campos equivalentes: $codigo \leftrightarrow isbn$
- Campos principales: $codigo, isbn$
- Campos no principales: $titulo, autor, pais_aut$

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
 - Claves candidatas:
 1. $codigo$
 2. $isbn$
 - Campos equivalentes: $codigo \leftrightarrow isbn$
 - Campos principales: $codigo, isbn$
 - Campos no principales: $titulo, autor, pais_aut$
- ¿En 2FN?

Tercera Forma Normal. Ejemplo (I)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

- Dependencias funcionales elementales:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, \quad isbn \rightarrow titulo, \quad isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
 - Claves candidatas:
 1. $codigo$
 2. $isbn$
 - Campos equivalentes: $codigo \leftrightarrow isbn$
 - Campos principales: $codigo, isbn$
 - Campos no principales: $titulo, autor, pais_aut$
- ¿En 2FN? Como todas las claves candidatas son unitarias,
la tabla **LIBRO** está automáticamente en **2FN**.

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N^o de comprobaciones:

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N° de comprobaciones: $3 * 2 = 6$

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N° de comprobaciones: $3 * 2 = 6$

1. ¿*titulo* depende transitivamente de *codigo*?

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N° de comprobaciones: $3 * 2 = 6$

1. ¿*titulo* depende transitivamente de *codigo*? NO
2. ¿*titulo* depende transitivamente de *isbn*?

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N° de comprobaciones: $3 * 2 = 6$

1. ¿*titulo* depende transitivamente de *codigo*? NO
2. ¿*titulo* depende transitivamente de *isbn*? NO
3. ¿*autor* depende transitivamente de *codigo*?

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N° de comprobaciones: $3 * 2 = 6$

1. ¿*titulo* depende transitivamente de *codigo*? NO
2. ¿*titulo* depende transitivamente de *isbn*? NO
3. ¿*autor* depende transitivamente de *codigo*? NO
4. ¿*autor* depende transitivamente de *isbn*?

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N^o de comprobaciones: $3 * 2 = 6$

1. ¿*titulo* depende transitivamente de *codigo*? NO
2. ¿*titulo* depende transitivamente de *isbn*? NO
3. ¿*autor* depende transitivamente de *codigo*? NO
4. ¿*autor* depende transitivamente de *isbn*? NO
5. ¿*pais_aut* depende transitivamente de *codigo*?

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo,isbn,titulo,autor,pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N^o de comprobaciones: $3 * 2 = 6$

1. ¿*titulo* depende transitivamente de *codigo*? NO
2. ¿*titulo* depende transitivamente de *isbn*? NO
3. ¿*autor* depende transitivamente de *codigo*? NO
4. ¿*autor* depende transitivamente de *isbn*? NO
5. ¿*pais_aut* depende transitivamente de *codigo*? SÍ (*autor*) (S1)
6. ¿*pais_aut* depende transitivamente de *isbn*?

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo, isbn, titulo, autor, pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N° de comprobaciones: $3 * 2 = 6$

1. ¿*titulo* depende transitivamente de *codigo*? NO
2. ¿*titulo* depende transitivamente de *isbn*? NO
3. ¿*autor* depende transitivamente de *codigo*? NO
4. ¿*autor* depende transitivamente de *isbn*? NO
5. ¿*pais_aut* depende transitivamente de *codigo*? SÍ (*autor*) (S1)
6. ¿*pais_aut* depende transitivamente de *isbn*? SÍ (*autor*) (S2)

¿Conclusión?

Tercera Forma Normal. Ejemplo (II)

LIBRO(codigo, isbn, titulo, autor, pais_aut)

¿Está la tabla **LIBRO** en **3FN**? N° de comprobaciones: $3 * 2 = 6$

1. ¿*titulo* depende transitivamente de *codigo*? NO
2. ¿*titulo* depende transitivamente de *isbn*? NO
3. ¿*autor* depende transitivamente de *codigo*? NO
4. ¿*autor* depende transitivamente de *isbn*? NO
5. ¿*pais_aut* depende transitivamente de *codigo*? SÍ (*autor*) (S1)
6. ¿*pais_aut* depende transitivamente de *isbn*? SÍ (*autor*) (S2)

¿Conclusión? El campo *pais_aut* tiene dependencia transitiva de *codigo* a través de *autor*. Por tanto, **LIBRO no está en 3FN**.

Tercera Forma Normal. Ejemplo (III)

Normalización

LIBRO(código,isbn,título,autor,*país_aut*)

¿Cómo conseguir un conjunto de tablas en **3FN** equivalente?

- $R1(\text{autor}, \text{pais_aut})$ (por $(S1/S2)$)
 $PK = \text{autor}$
- $R2(\text{codigo}, \text{isbn}, \text{titulo}, \text{autor})$
 $PK = \text{codigo}$
 $FK = \text{autor}(\rightsquigarrow R1)$

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

■ Claves candidatas:

1. nif
2. $codigo$

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:
 1. nif
 2. $codigo$
- Campos equivalentes:

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:
 1. nif
 2. $codigo$
- Campos equivalentes: $nif \leftrightarrow codigo$

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:
 1. nif
 2. $codigo$
- Campos equivalentes: $nif \leftrightarrow codigo$
- Campos principales:

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:
 1. nif
 2. $codigo$
- Campos equivalentes: $nif \leftrightarrow codigo$
- Campos principales: nif , $codigo$

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:

1. nif
2. $codigo$

- Campos equivalentes: $nif \leftrightarrow codigo$

- Campos principales: nif , $codigo$ No principales:

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:
 1. nif
 2. $codigo$
- Campos equivalentes: $nif \leftrightarrow codigo$
- Campos principales: nif , $codigo$ No principales: $edad$

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:

1. nif
2. $codigo$

- Campos equivalentes: $nif \leftrightarrow codigo$

- Campos principales: nif , $codigo$ No principales: $edad$

¿Conclusiones?

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:

1. nif
2. $codigo$

- Campos equivalentes: $nif \leftrightarrow codigo$

- Campos principales: nif , $codigo$ No principales: $edad$

¿Conclusiones? Las claves candidatas son unitarias, luego SOCIO está en **2FN**.

Tercera Forma Normal. Propiedades (I)

Propiedad 2

Si R está en **2FN** y posee un único campo no principal, entonces R está en **3FN**.

Ejemplo: **SOCIO**(nif,codigo,edad)

$nif \rightarrow codigo$, $nif \rightarrow edad$

$codigo \rightarrow nif$

- Claves candidatas:

1. nif
2. $codigo$

- Campos equivalentes: $nif \leftrightarrow codigo$

- Campos principales: nif , $codigo$ No principales: $edad$

¿Conclusiones? Las claves candidatas son unitarias, luego SOCIO está en **2FN**. Tiene 1 campo no principal ($edad$), luego está también en **3FN**.

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**(nif,cod_alum,asignatura)

$nif \rightarrow cod_alum$, $cod_alum \rightarrow nif$

- Claves candidatas:

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**(nif,cod_alum,asignatura)

$nif \rightarrow cod_alum$, $cod_alum \rightarrow nif$

- Claves candidatas:

1. $(nif, asignatura)$

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**(*nif*,*cod_alum*,*asignatura*)

$nif \rightarrow cod_alum$, $cod_alum \rightarrow nif$

- Claves candidatas:
 1. (*nif*, *asignatura*)
 2. (*cod_alum*, *asignatura*)
- Campos equivalentes:

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**(nif,cod_alum,asignatura)

$nif \rightarrow cod_alum$, $cod_alum \rightarrow nif$

- Claves candidatas:
 1. $(nif, asignatura)$
 2. $(cod_alum, asignatura)$
- Campos equivalentes: $nif \leftrightarrow cod_alum$

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**(*nif*,*cod_alum*,*asignatura*)

$nif \rightarrow cod_alum$, $cod_alum \rightarrow nif$

- Claves candidatas:
 1. (*nif*, *asignatura*)
 2. (*cod_alum*, *asignatura*)
- Campos equivalentes: $nif \leftrightarrow cod_alum$
- Campos principales:

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**(*nif*,*cod_alum*,*asignatura*)

$nif \rightarrow cod_alum$, $cod_alum \rightarrow nif$

- Claves candidatas:
 1. (*nif*, *asignatura*)
 2. (*cod_alum*, *asignatura*)
- Campos equivalentes: $nif \leftrightarrow cod_alum$
- Campos principales: *nif*, *cod_alum*, *asignatura*

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**($nif, cod_alum, asignatura$)

$nif \rightarrow cod_alum, cod_alum \rightarrow nif$

- Claves candidatas:
 1. ($nif, asignatura$)
 2. ($cod_alum, asignatura$)
- Campos equivalentes: $nif \leftrightarrow cod_alum$
- Campos principales: $nif, cod_alum, asignatura$
- Campos no principales:

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**($nif, cod_alum, asignatura$)

$nif \rightarrow cod_alum, cod_alum \rightarrow nif$

- Claves candidatas:
 1. ($nif, asignatura$)
 2. ($cod_alum, asignatura$)
- Campos equivalentes: $nif \leftrightarrow cod_alum$
- Campos principales: $nif, cod_alum, asignatura$
- Campos no principales: \emptyset

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**(nif,cod_alum,asignatura)

$nif \rightarrow cod_alum$, $cod_alum \rightarrow nif$

- Claves candidatas:
 1. $(nif, asignatura)$
 2. $(cod_alum, asignatura)$
- Campos equivalentes: $nif \leftrightarrow cod_alum$
- Campos principales: $nif, cod_alum, asignatura$
- Campos no principales: \emptyset

¿Conclusión?

Tercera Forma Normal. Propiedades (II)

Propiedad 3

Si todos los campos de R son principales, entonces R está automáticamente en **3FN**.

Ejemplo: **CURSA**(*nif*,*cod_alum*,*asignatura*)

$nif \rightarrow cod_alum$, $cod_alum \rightarrow nif$

- Claves candidatas:
 1. (*nif*, *asignatura*)
 2. (*cod_alum*, *asignatura*)
- Campos equivalentes: $nif \leftrightarrow cod_alum$
- Campos principales: *nif*, *cod_alum*, *asignatura*
- Campos no principales: \emptyset

¿Conclusión? CURSA está automáticamente en **3FN**, ya que sólo posee campos principales.

Tercera Forma Normal

Resumen

- En ausencia de campos multivaluados, siempre están en 3FN las:
 - Relaciones binarias (2 campos)
 - Relaciones con todos sus atributos principales
 - Relaciones en 2FN con un único atributo no principal
- La 3FN no se cumple cuando hay atributos multivaluados (no 1FN), dependencias no plenas (1FN pero no 2FN) o atributos no principales que dependen funcionalmente de otros atributos no principales (2FN pero no 3FN)

Índice

- 1 Introducción a la normalización
- 2 Dependencias funcionales
- 3 Formas Normales
- 4 Normalización**
- 5 Bibliografía

Dependencias entre atributos

La **normalización** trata las dependencias que hay entre los atributos de una relación. Los pasos que se realizan son los siguientes:

1. Cálculo de dependencias funcionales entre los atributos.
2. Cálculo del recubrimiento minimal.
3. Cálculo de claves candidatas, atributos principales y no ppales.
4. Cálculo de la forma normal en la que se encuentra la relación.
5. **Aplicar** métodos de **análisis** o **síntesis** para obtener la forma normal deseada, realizando en ambos casos:
 - **Descomposición** de la relación en dos o más relaciones que mantengan las propiedades deseadas.
 - **Conexión** de las tablas mediante claves ajenas.

Normalización por Método de Análisis

Paso a la Segunda Forma Normal

- Para pasar una tabla en 1FN a 2FN hay que ***eliminar las dependencias parciales*** de las claves candidatas. Para ello:
 - Se eliminan de la relación original los atributos que son funcionalmente dependientes de otros de forma parcial (dependen de un subconjunto de alguna clave).
 - Dichos atributos se ponen en una nueva tabla con una copia de su determinante (formado por los atributos de la clave de los que depende de forma plena).

Normalización por Método de Análisis

Paso a la Segunda Forma Normal

Consideremos la relación

INSCRIPCIÓN(Estudiante,Actividad,Precio)

Un estudiante puede inscribirse en varias actividades y una actividad tiene muchos estudiantes, por lo que Estudiante y Actividad son la clave primaria. Si suponemos que Precio sólo depende de Actividad, esta relación no está en 2FN porque tenemos la dependencia funcional

Actividad \rightarrow Precio

Normalización por Método de Análisis

Paso a la Segunda Forma Normal

Podemos descomponer la relación

INSCRIPCIÓN(Estudiante,Actividad,Precio)

en dos relaciones

R1(Estudiante,Actividad)

R2(Actividad,Precio)

R1.Actividad es clave ajena a R2.

De este modo, R1 y R2 mantienen la información de la relación original INSCRIPCIÓN, y ambas están en 2FN (y 3FN).

Normalización por Método de Análisis

Paso a la Tercera Forma Normal

- Para pasar una relación en 2FN a 3FN hay que eliminar las dependencias transitivas. Para ello, se eliminan los atributos que dependen transitivamente y se ponen en una nueva relación con una copia de su determinante (el atributo o atributos no clave de los que depende).

Ej.: ESCRITOR(Nombre, Pais, Poblacion_pais)

- Dependencia funcional transitiva

Nombre \rightarrow *Poblacion_pais*

- Podemos descomponer la relación en dos relaciones en 3FN
R1(Nombre, Pais)
R2(Pais, Poblacion_pais)
Nótese que R1.Pais es clave ajena hacia R2.Pais.

Normalización por Método de Síntesis

En lugar de ir descomponiendo la relación original, partimos del recubrimiento minimal y las claves, y hacemos lo siguiente:

1. Por cada dependencia funcional del recubrimiento, consideramos un esquema de relación
2. Unificamos los esquemas que provengan de dependencias funcionales con exactamente el mismo implicante
3. Si ninguno de los esquemas contiene a ninguna clave candidata de la relación original, añadimos un esquema adicional con los campos de al menos una de las claves candidatas, considerada la principal
4. Eliminamos esquemas redundantes (un esquema será redundante si sus atributos ya se encuentran contenidos completamente en otro esquema)

Índice

- 1 Introducción a la normalización
- 2 Dependencias funcionales
- 3 Formas Normales
- 4 Normalización
- 5 Bibliografía**

Bibliografía I



Ramez Elmasri, Shamkant B. Navathe

Fundamentos de Sistemas de Bases de Datos.

Pearson Educación S.A. - Addison Wesley Quinta Ed. (2007)

ISBN: 978-84-7829-085-7



Adoración de Miguel, Mario Piattini, Esperanza Marcos

Diseño de Bases de Datos Relacionales.

RA-MA Editorial (1999)

ISBN: 978-84-7897-385-9



Mercedes Marqués

Apuntes de Bases de Datos.

Universidad Jaume I en Castellón (2011)

ISBN: 978-84-693-0146-3

Bibliografía II



Luis Grau Fernandez, Ignacio López Rodríguez

Problemas de Bases de Datos.

Sanz Y Torres, S.L. - 3ª Edición (2006)

ISBN: 978-84-960-9469-7



Dolores Cuadra, Elena Castro, Ana Mª Iglesias, Paloma
Martínez, Fco. Javier Calle, César de Pablo, Harith Al-Jumaily,
Lourdes Moreno

*Desarrollo De Bases De Datos: Casos Prácticos Desde El
Análisis A La Implementación.*

RA-MA Editorial - 1ª Edición (2007)

ISBN: 978-84-7897-835-9