

Introducción

Echando la vista atrás...

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

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 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*.
- El modelo lógico escogido facilita la identificación de **esquemas de relación** y su conexión mediante **claves ajenas**, dando una estructura razonablemente fácil de manejar para nuestra base de datos.
- Parece por tanto que debemos estar en un **camino prometedor** para el desarrollo de buenas bases de datos, con una organización adecuada y sustentada en cierto modo por los modelos conceptual y lógico escogidos, y las reglas sistemáticas adoptadas...

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**

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(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.
2. Incoherencia de datos.

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.
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.

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**,

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.**

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 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.
- Pasamos de esquema $R(AT)$ a enriquecerlo como $R(AT, DEP)$

Dependencias funcionales

Sea el esquema de relación $R(AT, DEP)$, y sean X e Y dos *subconjuntos* de AT denominados descriptores.

Dependencias funcionales

Sea el esquema de relación $R(AT, DEP)$, y 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 o conjunto de campos implicado

Dependencias funcionales

Sea el esquema de relación $R(AT, DEP)$, y 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 o conjunto de campos 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 el concepto de dependencia funcional con el de campo derivado:
 - Está claro que si *Y* es un campo derivado de *X* (e.g., *edad* derivado de *fecha_nacimiento*), entonces $X \rightarrow Y$.

Dependencias funcionales. Ejemplo

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

- $autor \rightarrow pais$
- $codigo \rightarrow titulo$ $codigo \rightarrow editorial$ $codigo \rightarrow paginas$
- $codigo \rightarrow titulo, editorial, paginas$
- $codigo, autor \rightarrow editorial, pais$

Nota

- No debe confundirse el concepto de dependencia funcional con el 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 (solo que para ese X solo exista un Y).

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 el concepto de dependencia funcional con el 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 (solo que para ese X solo exista un Y).
- 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 = {DNI_profesor, Nombre_profesor}
 - DEP = {DNI_profesor → Nombre_profesor}

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:

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

Dependencias funcionales

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

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

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:

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

Nota: naturalmente, en general esta última condición no se cumple, y el nombre no suele actuar como clave candidata.

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 de ningún subconjunto propio*** de X_1, \dots, X_k .

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 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*
 $\text{alumno, asignatura, conv} \rightarrow \text{nota}$
 $\text{alumno} \not\rightarrow \text{nota}$
 $\text{asignatura} \not\rightarrow \text{nota}$
 $\text{alumno, asignatura} \not\rightarrow \text{nota}$
- El campo *dpto* **NO** tiene dep. 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 .

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 se cumplen estas dos condiciones:

- La dependencia funcional de Y respecto al conjunto X_1, \dots, X_k es:
 - **completa**

Dependencia funcional elemental (I)

El campo Y tiene dependencia funcional elemental del conjunto X_1, \dots, X_k si se cumplen estas dos condiciones:

- La dependencia funcional de Y respecto al conjunto X_1, \dots, X_k es:
 - **completa**
 - **no trivial**

Dependencia funcional elemental (I)

El campo Y tiene dependencia funcional elemental del conjunto X_1, \dots, X_k si se cumplen estas dos condiciones:

- La dependencia funcional de Y respecto al conjunto X_1, \dots, X_k es:
 - **completa**
 - **no trivial**
- El **conjunto implicado** Y es **unitario**.

Dependencia funcional elemental (I)

El campo Y tiene dependencia funcional elemental del conjunto X_1, \dots, X_k si se cumplen estas dos condiciones:

- La dependencia funcional de Y respecto al conjunto X_1, \dots, X_k es:
 - **completa**
 - **no trivial**
- El **conjunto implicado** Y es **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* → *nota*
- *alumno* → *edad*
- *asignatura* → *depto*

Nota:

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

Campos equivalentes

Cuando dos conjuntos de campos 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 $R(AT=\{X,Y,Z,\dots\}, DEP)$, de DEP se derivan las dependencias $X \rightarrow Y$, $Y \rightarrow Z$ e $Y \not\rightarrow X$

- Ejemplo:
 - $DNI_profesor \rightarrow C\acute{o}digo_Postal$;
 - $C\acute{o}digo_Postal \rightarrow Ciudad$;
 - $C\acute{o}digo_Postal \not\rightarrow DNI_profesor$

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:

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

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:

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

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:

- $isbn \rightarrow autor$,
- $autor \rightarrow pais$,
- $autor \rightarrow isbn$

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

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

NO, puesto que:

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

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

Dependencia funcional transitiva y campos equivalentes

Dada la relación y sus dependencias funcionales:

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

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

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

^anumero sería el ordinal del número de ventas que lleva el vendedor, mientras que *codigo* sería un identificador único global para la venta

Dependencia funcional transitiva y campos equivalentes

Dada la relación y sus dependencias funcionales:

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

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

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

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

^anumero sería el ordinal del número de ventas que lleva el vendedor, mientras que *codigo* sería un identificador único global para la venta

Campos equivalentes. Ejemplo 2

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

- *codigo* ↔ *numero, vendedor*
- *codigo* → *articulo*
- *articulo* → *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:

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

Resumen: Tipos de Dependencias Funcionales

Tipo	Definición breve	Ejemplo
Plena	Depende del conjunto completo, no de una parte.	alumno, asignatura, conv → nota
Trivial	El conjunto implicado está contenido en el implicante.	$A, B \rightarrow A$
Elemental	Plena + no trivial + un solo atributo implicado.	$\text{DNI} \rightarrow \text{Nombre}$
Transitiva	$X \rightarrow Y, Y \rightarrow Z$, pero $Y \not\rightarrow X$.	$\text{isbn} \rightarrow \text{autor} \rightarrow \text{pais}$
Equivalente	$X \rightarrow Y$ y $Y \rightarrow X$.	$\text{código} \leftrightarrow$ (número, vendedor)

Resumen: Tipos de Dependencias Funcionales

Tipo	Definición breve	Ejemplo
Plena	Depende del conjunto completo, no de una parte.	alumno, asignatura, conv → nota
Trivial	El conjunto implicado está contenido en el implicante.	$A, B \rightarrow A$
Elemental	Plena + no trivial + un solo atributo implicado.	DNI → Nombre
Transitiva	$X \rightarrow Y, Y \rightarrow Z$, pero $Y \not\rightarrow X$.	isbn → autor → país
Equivalente	$X \rightarrow Y$ y $Y \rightarrow X$.	código ↔ (número, vendedor)

Idea clave

Estas dependencias permiten **detectar redundancias y anomalías**, preparando el camino para las **formas normales**.

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.)

Axiomas de Armstrong: Ejemplo Paso a Paso

Ejemplo: Sea $R(\{A, B, C, D\}, \{A \rightarrow B, B \rightarrow C\})$

Axiomas de Armstrong: Ejemplo Paso a Paso

Ejemplo: Sea $R(\{A, B, C, D\}, \{A \rightarrow B, B \rightarrow C\})$

1. Por **Reflexividad**: $A \rightarrow A, B \rightarrow B$
2. Por **Aumentatividad**: de $A \rightarrow B$ se obtiene $AD \rightarrow BD$
3. Por **Transitividad**: de $A \rightarrow B$ y $B \rightarrow C$, deducimos $A \rightarrow C$
4. Por **Aditividad**: de $A \rightarrow B$ y $A \rightarrow C$, deducimos $A \rightarrow BC$

Axiomas de Armstrong: Ejemplo Paso a Paso

Ejemplo: Sea $R(\{A, B, C, D\}, \{A \rightarrow B, B \rightarrow C\})$

1. Por **Reflexividad**: $A \rightarrow A, B \rightarrow B$
2. Por **Aumentatividad**: de $A \rightarrow B$ se obtiene $AD \rightarrow BD$
3. Por **Transitividad**: de $A \rightarrow B$ y $B \rightarrow C$, deducimos $A \rightarrow C$
4. Por **Aditividad**: de $A \rightarrow B$ y $A \rightarrow C$, deducimos $A \rightarrow BC$

Resultado

A partir de R , utilizando estos axiomas, podemos **derivar todas las dependencias posibles** a partir de un conjunto inicial:

$$\{A \rightarrow B, B \rightarrow C, A \rightarrow C, A \rightarrow BC, \dots\}$$

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).
Nota: naturalmente, si existe algún atributo extraño en una dependencia $X \rightarrow Y$, entonces la misma no es plena.
- 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.
 - Por tanto, **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^+ .

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, del esquema

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

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

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^+ .
- Y del esquema

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

- $A^+ = \{A, B, C, D\}$ $E^+ = \{E\}$
- $B^+ = \{B, C\}$ $F^+ = \{F\}$
- $C^+ = \{C\}$ $AE^+ = \{A, B, C, D, E, F\} = AT$
- $D^+ = \{D\}$

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 → 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 ← {I}
else
| CCS ← {}, CPS ← {(I, R)}
| while CPS no vacío do
| | (CP, R) ← primero extraído de CPS
| | R ← R \ CP+
| | foreach atributo A en R do
| | | CPA ← CP ∪ {A}
| | | R ← R \ {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
```

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* → *numero*, *codigo* → *vendedor*,
codigo → *articulo*
 - *numero*, *vendedor* → *codigo*
 - *articulo* → *color*, *articulo* → *precio*
- Claves candidatas:

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*, *articulo* → *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* → *numero*, *codigo* → *vendedor*,
 codigo → *articulo*
 - *numero*, *vendedor* → *codigo*
 - *articulo* → *color*, *articulo* → *precio*
- Claves candidatas:
 1. *codigo*
 2. (*numero*, *vendedor*)
- Campos equivalentes: *codigo* ↔ *numero*, *vendedor*
- Campos principales:

Campos principales y no principales. Ejemplo

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

- Dependencias elementales:
 - $codigo \rightarrow numero, \quad codigo \rightarrow vendedor, \quad codigo \rightarrow articulo$
 - $numero, vendedor \rightarrow codigo$
 - $articulo \rightarrow color, \quad articulo \rightarrow precio$
- Claves candidatas:
 1. $codigo$
 2. $(numero, vendedor)$
- Campos equivalentes: $codigo \leftrightarrow numero, vendedor$
- Campos principales: $codigo, numero, vendedor$
- Campos no principales:

Campos principales y no principales. Ejemplo

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

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

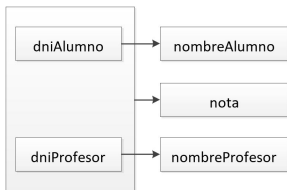
Grafo dirigido que captura las dependencias funcionales de la relación, representando los nodos del grafo tanto los atributos (nodos simples) como los conjuntos de atributos (burbujas) que actúan juntos como implicantes en alguna dependencia funcional, y los arcos las dependencias desde implicantes a 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

Hacia las Formas Normales

Resumiendo

Hasta ahora hemos visto:

- Qué son las dependencias funcionales y cómo se clasifican.
- Cómo identificarlas y utilizarlas para analizar un esquema.
- Cómo derivar claves y dependencias mediante axiomas.

Hacia las Formas Normales

Rescapitulando

Hasta ahora hemos visto:

- Qué son las dependencias funcionales y cómo se clasifican.
- Cómo identificarlas y utilizarlas para analizar un esquema.
- Cómo derivar claves y dependencias mediante axiomas.

A continuación

Vamos a estudiar cómo **medir y mejorar la estructura** de una base de datos a partir de estas dependencias.

**Problemas estructurales ⇒ Dependencias
funcionales ⇒ Formas normales ⇒ Normalización**

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:
 - $alumno \rightarrow edad$

Campos principales y no principales. Ejemplo

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

- Dependencias funcionales elementales:
 - *alumno* → *edad*
 - *asignatura* → *dpto*
 - *alumno, asignatura, conv* → *nota*
- Claves candidatas:

Campos principales y no principales. Ejemplo

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

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

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?

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)

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?

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?

EVAL no está en 2FN porque el campo no principal *edad* no tiene dependencia plena respecto de la clave (*alumno, asignatura, conv*), ya que sabemos que *alumno* → *edad* (lo mismo ocurre con *dpto* respecto a *asignatura*, 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?

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

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:
 - $codigo \rightarrow isbn$

Tercera Forma Normal. Ejemplo (I)

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

- Dependencias funcionales elementales:
 - *codigo* → *isbn*
 - *isbn* → *codigo*, *isbn* → *titulo*, *isbn* → *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:
 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:
 - $codigo \rightarrow isbn$
 - $isbn \rightarrow codigo, isbn \rightarrow titulo, isbn \rightarrow autor$
 - $autor \rightarrow pais_aut$
- 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* → *isbn*
 - *isbn* → *codigo*, *isbn* → *titulo*, *isbn* → *autor*
 - *autor* → *pais_aut*
 - Claves candidatas:
 1. *codigo*
 2. *isbn*
 - Campos equivalentes: *codigo* ↔ *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: $3 * 2 = 6$

- 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^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*?

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*?

Tercera Forma Normal. Ejemplo (III)

Normalización

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

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

- $R1(\text{autor}, \text{país_aut})$ (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:
 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: $nif \leftrightarrow codigo$
- 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 → *codigo*, *nif* → *edad*
codigo → *nif*

- Claves candidatas:
 1. *nif*
 2. *codigo*
- Campos equivalentes: *nif* ↔ *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 → *cod_alum*, *cod_alum* → *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*)
 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: \nexists

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: $\#$

¿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.

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

