

A5. Normalización.

Índice.

1.	Reglas de Codd.....	2
2.	Normalización.....	4
2.1.	Forma normal.....	7
2.2.	Conceptos asociados.....	6
2.3.	Formas normales.....	11
2.3.1.	Primera Forma Normal o FN1.....	12
2.3.2.	Segunda Forma Normal o FN2.....	14
2.3.3.	Tercera Forma Normal o FN3.....	16
2.3.4.	Forma Normal Boyce-Codd o FNBC.....	18
2.3.5.	Cuarta Forma Normal o FN4.....	20
2.3.6.	Quinta Forma Normal o FN5.....	20
2.3.7.	Forma Normal de Dominio Clave o FNDK.....	20
2.3.8.	Sexta Forma Normal o FN6.....	20
3.	Desnormalización.....	21

A5. Normalización.

1. Reglas de Codd.

Las reglas de Codd son una serie de reglas propuestas por Edgar F. Codd al modelo relacional:

- **Regla 0 (Regla de fundación)** → todo sistema relacional debe ser capaz de gestionar sus bases de datos con sus capacidades relacionales.
- **Regla 1 (Regla de información)** → toda información se representa unidireccionalmente según valores en posiciones de las columnas de las filas de las tablas, representada explícitamente en el nivel lógico a través de los valores en las tablas.
- **Regla 2 (Regla del acceso garantizado)** → todos los datos deben ser accesibles sin ambigüedad.
- **Regla 3 (Regla del tratamiento sistemático de valores nulos)** → los campos nulos se deben permitir, bajo el significado de información faltante o inaplicable.
- **Regla 4 (Catálogo dinámico en línea basado en el modelo relacional)** → el sistema debe soportar un catálogo en línea (catálogo relacional) que dé acceso a la estructura de la base de datos y sea accesible sólo por usuarios autorizados.
- **Regla 5 (Regla comprensiva del sublenguaje de datos)** → el sistema debe soportar un lenguaje relacional que:
 - Tenga un sintaxis lineal.
 - Pueda ser utilizado de forma interactiva.
 - Tenga soporte de operaciones de definición y de manipulación de datos, de control de seguridad e integridad, y operaciones de administración de las transacciones.
- **Regla 6 (Regla de actualización de vistas)** → todas las vistas que sean teóricamente actualizables deben poder ser actualizadas por el sistema.

A5. Normalización.

1. Reglas de Codd.

- **Regla 7 (Alto nivel de inserción, actualización y borrado)** → el sistema debe permitir la manipulación de alto nivel de los datos, es decir, sobre el conjunto de tuplas.
- **Regla 8 (Independencia física de los datos)** → los programas de aplicación y las actividades del terminal permanecerán inalterados a nivel lógico, aunque se realicen cambios en las representaciones de almacenamiento o en los métodos de acceso.
- **Regla 9 (Independencia lógica de los datos)** → los programas de aplicación y las actividades del terminal permanecerán inalterados a nivel lógico, aunque se realicen cambios en las tablas que preserven la información. La independencia lógica de datos es más difícil de lograr que la independencia física de los datos.
- **Regla 10 (Independencia de la integridad)** → las restricciones de integridad se deben especificar por separado de los programas de aplicación y del almacenamiento en la base de datos. Debe ser posible cambiar esas restricciones sin afectar innecesariamente a las aplicaciones existentes.
- **Regla 11 (Independencia de la distribución)** → la distribución de las porciones de la base de datos en distintas localizaciones debe ser invisible a los usuarios de la base de datos. Los usos existentes deben continuar funcionando con éxito:
 - Cuando una versión distribuida del SGBD se cargue por primera vez.
 - Cuando los datos existentes se redistribuyan en el sistema.
- **Regla 12 (Regla de la no subversión)** → si el sistema proporciona una interfaz de bajo nivel de registro (además de una interfaz racional), dicha interfaz no debe permitir su utilización para subvertir el sistema (pj: sortear las reglas de seguridad racional o las restricciones de integridad).

A5. Normalización.

2. Normalización.

2.1. Forma normal.

Uno de los parámetros que mide la calidad de una base de datos es la **forma normal** en la que se halla su diseño. Dicha forma normal puede alcanzarse cumpliendo con ciertas restricciones que impone cada una de ellas al conjunto de atributos del diseño.

La **normalización** es el proceso de obligar a los atributos de un diseño a cumplir con ciertas formas normales.

Las formas normales pretenden alcanzar dos objetivos:

- Almacenar en la base de datos cada hecho sólo una vez → evitar redundancia y obtener ahorro de espacio.
- Que hechos distintos se almacenen en sitios distintos → evitar anomalías al operar con los datos.

A5. Normalización.

2.2. Conceptos asociados.

Las **claves** del modelo relacional pueden ser de los siguientes tipos:

- **Clave candidata** → conjunto NO vacío de atributos pertenecientes a la relación que satisface las siguientes propiedades:
 - **Unicidad** → NO existencia de varias tuplas en la relación con los mismos valores en los atributos.
 - **Minimalidad** → si el atributo clave es compuesto, entonces NO es posible eliminar ningún componente SIN romper la propiedad de unicidad.
- **Clave primaria** → clave candidata (simple o compuesta) elegida para identificar unívocamente las tuplas de la relación.
- **Clave alternativa** → clave candidata NO elegida como clave primaria.
- **Superclave** → conjunto de uno o más atributos que, tomados colectivamente, pueden identificar unívocamente a cada tupla de la relación.
- **Clave externa o secundaria** → atributo de la relación que es clave principal en otra relación y sirve como enlace entre dichas relaciones, es decir, es un atributo NO primo en una relación y que es PRIMO en otra relación.

A5. Normalización.

2.2. Conceptos asociados.

Los **atributos** pueden ser de los siguientes tipos:

- **Determinante o implicante** → atributo (o conjunto de atributos) del que dependen, funcionalmente, otro conjunto de atributos, al que se llamará determinado (o implicado).
- **Determinado o implicado** → atributo (o conjunto de atributos) que depende funcionalmente de otro conjunto de atributos, al que se llamará determinante (o implicante).
- **Atributo primo** → atributo que está contenido en la clave primaria de la relación.
- **Atributo no primo** → si NO está contenido ni en la clave primaria ni en ninguna de las claves alternativas de la relación.
- **Atributo extraño o ajeno** → si hace que la dependencia funcional no sea plena.

A5. Normalización.

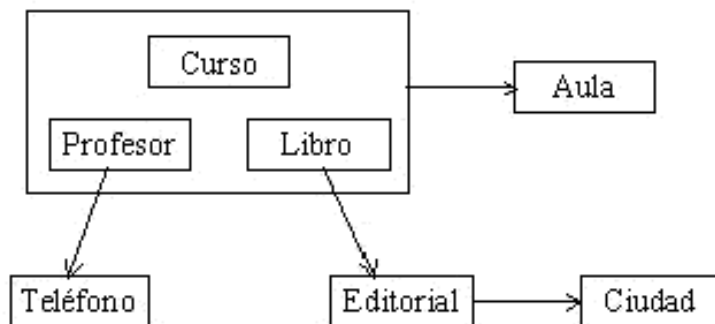
2.2. Conceptos asociados.

Las **dependencias funcionales** pueden ser de los siguientes tipos:

- **Dependencia funcional ($X \rightarrow Y$).**

Se dice que un atributo Y depende funcionalmente de otro atributo X \Leftrightarrow cada valor X tiene asociado, en todo momento, un único valor de Y.

También se dice que X implica Y, y por tanto, X es el implicante.



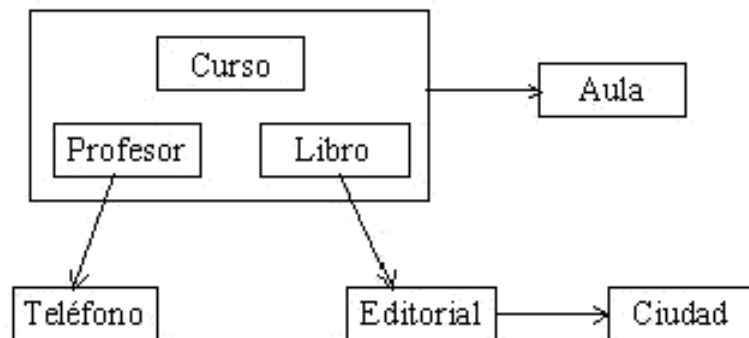
Imparte(Curso, Profesor, Libro, Teléfono, Editorial, Ciudad)

A5. Normalización.

2.2. Conceptos asociados.

- **Dependencia funcional incompleta ($X \not\rightarrow Y$).**

Se dice que un atributo Y tiene dependencia funcional incompleta de otro atributo X \Leftrightarrow depende funcionalmente de X y, también, de algún subconjunto de X.



Imparte(Curso, Profesor, Libro, Teléfono, Editorial, Ciudad)

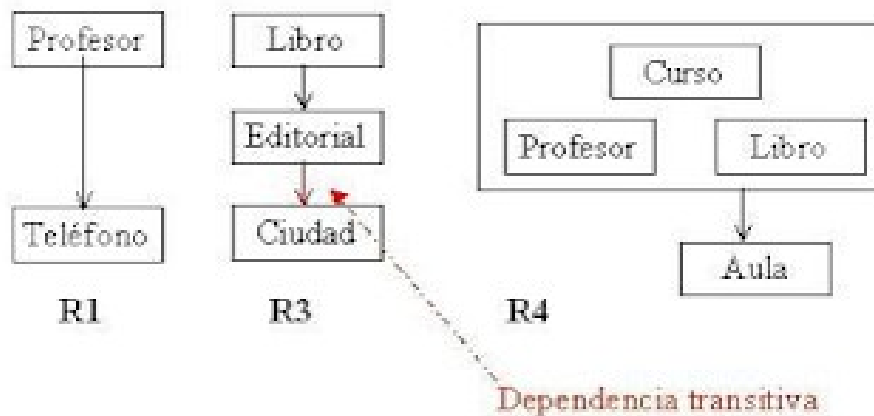
Teléfono y Editorial, Ciudad dependen funcionalmente NO de toda la clave Curso, Profesor, Libro

A5. Normalización.

2.2. Conceptos asociados.

- Dependencia funcional completa ($X \Rightarrow Y$).**

Se dice que un atributo Y tiene dependencia funcional completa con otro atributo X, siendo X una combinación de atributos $X(X_1, X_2, \dots, X_n) \Leftrightarrow$ depende funcionalmente de X pero NO de ningún subconjunto de X.



R1(Profesor, Teléfono)

R3(Libro, Editorial, Ciudad)

R4(Curso, Profesor, Libro, Aula)

A5. Normalización.

2.2. Conceptos asociados.

- **Dependencia funcional transitiva ($X \rightarrow Z$).**

Se dice que los atributos X, Y, Z, donde $X \rightarrow Y$, $Y \rightarrow Z$, se dice que X depende transitivamente de Z $\leftrightarrow Y \not\rightarrow X$



A5. Normalización.

2.3. Formas Normales.

Las formas normales son unas series de reglas con fundamento matemático que ayudan en el proceso de normalización de las bases de datos.

En el ejemplo de la tabla inferior, hay unos datos sobre películas que NO están normalizados porque el campo Habilidad tiene valores que NO son atómicos.

Empleado	Habilidad	Lugar de trabajo
Juan	Mecanografía Taquigrafía Tallado	Oficina 10
María	Escucha	Oficina 6
Elena	Alquimia Malabarismo	Oficina 6
Pedro	Limpieza	Oficina 6

A5. Normalización.

2.3.1. Primera Forma Normal o FN1.

Problema al que se enfrenta.

Problemas relacionados con el tratamiento de tuplas iteradas y ante la posible existencia de tuplas incoherentes (o con sentido contrario).

Solución.

Aplicar el concepto de dependencia funcional.

Siempre está en FN1.

Una relación con tuplas NO iteradas NI incoherentes.

FN1.

- Prohibición de que los atributos tengan más de un valor.
- Todos los dominios subyacentes de la relación tienen valores atómicos.
- Las tuplas no se iteran.
- Todos los atributos NO primos dependen de la clave o llave.

A5. Normalización.

2.3.1. Primera Forma Normal o FN1.

Empleado	Habilidad	Lugar de trabajo
Juan	Mecanografía Taquigrafía Tallado	Oficina 10
María	Escucha	Oficina 6
Elena	Alquimia Malabarismo	Oficina 6
Pedro	Limpieza	Oficina 6



Empleado	Habilidad	Lugar de trabajo
Juan	Mecanografía	Oficina 10
Juan	Taquigrafía	Oficina 10
Juan	Tallado	Oficina 10
María	Escucha	Oficina 6
Elena	Alquimia	Oficina 6
Elena	Malabarismo	Oficina 6
Pedro	Limpieza	Oficina 6
Clave primaria		

A5. Normalización.

2.3.2. Segunda Forma Normal o FN2.

Problema al que se enfrenta.

Problemas relacionados con la iteración de datos, debido a la existencia de dependencias funcionales incompletas.

Solución.

Aplicar el concepto de dependencia funcional completa.

Siempre está en FN2.

- Una relación binaria.
- Una relación con todos los atributos primos.
- Una relación con claves o llaves simples.

FN2.

Un diseño está en FN2 si, está en FN1 y, además, cada atributo que NO forma parte de la clave tiene dependencia funcional completa de la clave principal.

Un diseño está en FN2 si, está en FN1 y, además, todos los atributos NO primos tienen dependencia funcional completa de los atributos primos.

A5. Normalización.

2.3.2. Segunda Forma Normal o FN2.

Empleado	Habilidad	Lugar de trabajo
Juan	Mecanografía	Oficina 10
Juan	Taquigrafía	Oficina 10
Juan	Tallado	Oficina 10
María	Escucha	Oficina 6
Elena	Alquimia	Oficina 6
Elena	Malabarismo	Oficina 6
Pedro	Limpieza	Oficina 6
Clave primaria		



Empleado	Habilidad
Juan	Mecanografía
Juan	Taquigrafía
Juan	Tallado
María	Escucha
Elena	Alquimia
Elena	Malabarismo
Pedro	Limpieza
Clave primaria	

Empleado	Lugar de trabajo
Juan	Oficina 10
María	Oficina 6
Elena	Oficina 6
Pedro	Oficina 6
Clave primaria	

A5. Normalización.

2.3.3. Tercera Forma Normal o FN3.

Problema al que se enfrenta.

Problemas relacionados con la existencia de varias formas de acceso a los atributos NO primos a través de atributos NO primos, es decir, por medio de la transitividad.

Solución.

Aplicar el concepto de dependencia funcional transitiva entre los atributos NO primos.

Siempre está en FN3.

- Una relación binaria.
- Una relación con todos los atributos primos.
- Una relación con un único atributo NO primo.

FN3.

Un diseño está en FN3 si, está en FN2 y, además, NO hay ningún atributo NO primo que dependa de forma transitiva de la clave.

Un diseño está en FN3 si, está en FN2 y, además, cualquier atributo tiene dependencia funcional completa de la clave y NO hay dependencias transitivas con respecto a atributos que NO sean primos.

A5. Normalización.

2.3.3. Tercera Forma Normal o FN3.

Torneo	Año	Ganador	Fecha nacimiento
<u>Chess Teis</u>	2018	<u>Guillermino</u>	21 de julio de 1975
<u>Chess Cangas</u>	2018	<u>Guillermino</u>	21 de julio de 1975
<u>Chess Teis</u>	2019	Napoleón	28 de septiembre de 1980
<u>Chess Cangas</u>	2019	Pancracio	14 de marzo de 1977
Clave primaria			



Torneo	Año	Ganador	Ganador	Fecha nacimiento
<u>Chess Teis</u>	2018	<u>Guillermino</u>	<u>Guillermino</u>	21 de julio de 1975
<u>Chess Cangas</u>	2018	<u>Guillermino</u>	Napoleón	28 de septiembre de 1980
<u>Chess Teis</u>	2019	Napoleón	Pancracio	14 de marzo de 1977
<u>Chess Cangas</u>	2019	Pancracio	Clave primaria	
Clave primaria				

A5. Normalización.

2.3.4. Forma Normal de Boyce-Codd o FNBC.

Problema al que se enfrenta.

Problemas relacionados con la existencia de varias claves candidatas que ocasionan solapamiento.

Solución.

Aplicar el concepto de dependencia funcional completa con atributos NO primos.

Siempre está en FNBC.

- Una relación binaria.
- Una relación con claves no solapadas.

FNBC.

Un diseño está en FNBC si, está en FN3 y, además, todo determinante de la relación es clave candidata.

Un diseño está en FNBC si, está en FN3 y, además, todos los atributos (primos y no primos) dependen funcionalmente de la clave principal, y SÓLO de ella.

A5. Normalización.

2.3.4. Forma Normal de Boyce-Codd o FNBC.

Proyecto	Tarea	Empleado
P01	Análisis	Pancracio
P01	Calidad	<u>Guillermino</u>
P02	Diseño	<u>Fulgencio</u>
P02	Calidad	<u>Guillermino</u>
P15	Diseño	<u>Fulgencio</u>
P20	Análisis	Pancracio
P30	Análisis	<u>Pericles</u>



Proyecto	Empleado
P01	Pancracio
P01	<u>Guillermino</u>
P02	<u>Fulgencio</u>
P02	<u>Guillermino</u>
P15	<u>Fulgencio</u>
P20	Pancracio
P30	<u>Pericles</u>
Clave primaria	

Empleado	Tarea
Pancracio	Análisis
<u>Guillermino</u>	Calidad
<u>Fulgencio</u>	Diseño
<u>Pericles</u>	Análisis
Clave primaria	

A5. Normalización.

2.3.5. Cuarta Forma Normal o FN4.

Problema al que se enfrenta.

Problemas relacionados con la existencia de dependencias entre los atributos multivaluados.

2.3.6. Quinta Forma Normal o FN5.

Problema al que se enfrenta.

Problemas relacionados con la existencia de dependencias entre atributos multivaluados independientes.

2.3.7. Forma Normal de dominio clave o FNDK.

Problema al que se enfrenta.

Problemas relacionados con restricciones y dominios de los atributos.

2.3.8. Sexta Forma Normal o FN6.

Problema al que se enfrenta.


Problemas relacionados con algunas consideraciones con las bases de datos temporales.

A5. Normalización.


3. Desnormalización.

Habitualmente el diseño de una base de datos concluye con el paso del **modelo entidad/relación** al **modelo relacional** y aplicando las **formas normales** para depurar los ficheros resultantes, pero:


- Se puede unir varias relaciones en una si se usan frecuentemente juntas → **desnormalización**.

Cliente(#Cliente, nombre, CP)
CódigoPostal(CP, población)  Cliente(#Cliente, nombre, CP, población)

- Se puede dividir algunas relaciones para reorganizar tuplas (**partición horizontal**) o atributos (**partición vertical**) si se realiza con frecuencia accesos simultáneos.

Empleado(#Empleado, nombre, tf, fechaEvaluación, aspecto1, aspecto2)
 
Empleado(#Empleado, nombre, tf) Evaluación(#Empleado, fechaEvaluación, aspecto1, aspecto2)

- Se puede introducir otros cambios para conseguir un acceso más eficiente (**optimización**).

Cliente(#Cliente, nombre, apellido, teléfono)
 
Cliente(#Cliente, nombre, primerApellido, segundoApellido, tf1, tf2, tf3, tf4)