

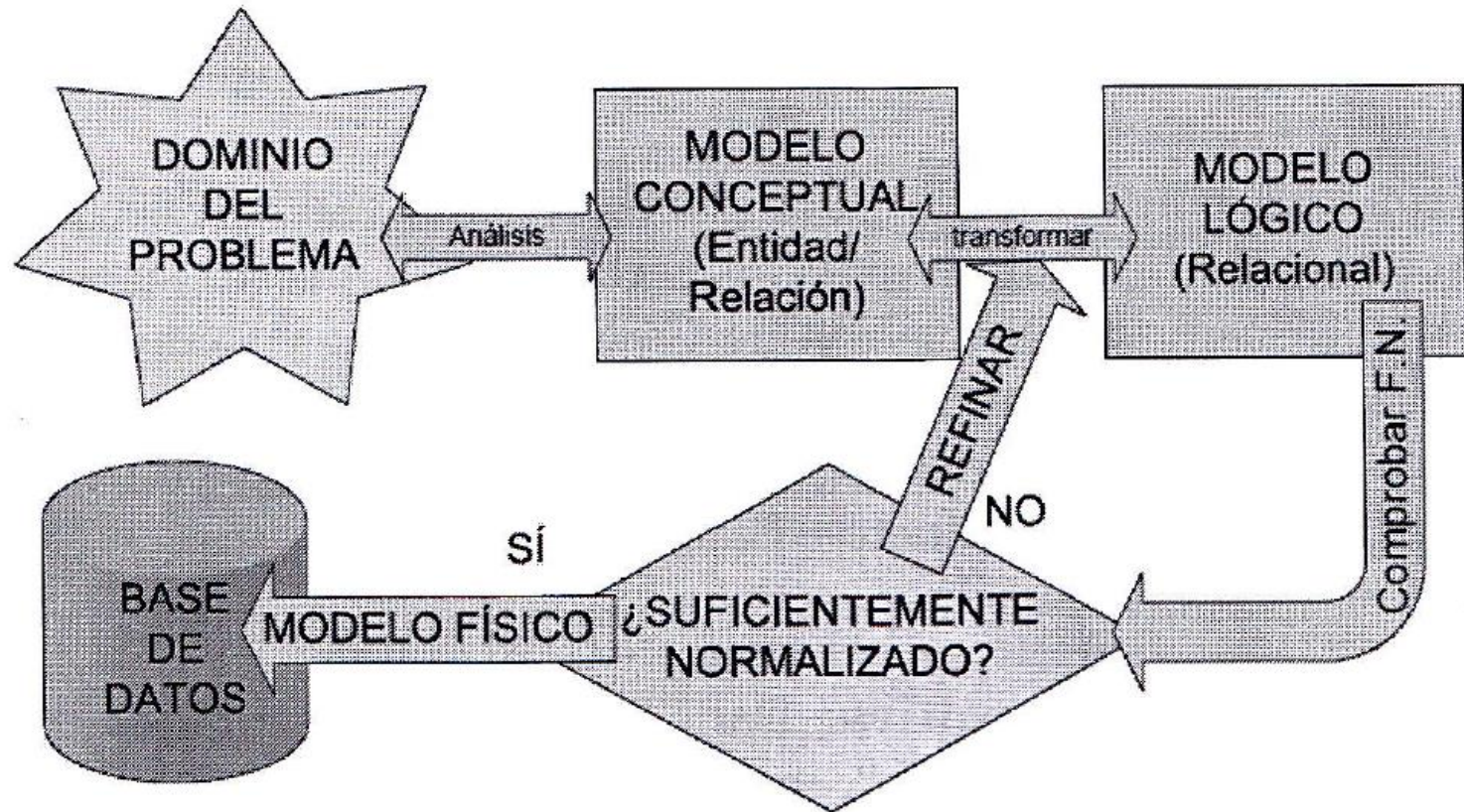
# **UD4**

## **Normalización**

# Normalización

- Habitualmente, el diseño de una base de datos, termina en el paso del modelo entidad-relación al modelo relacional.
- Cuando se diseña un Sistema, no solo una base de datos, sino cualquier tipo de solución informática, se ha de medir la calidad de la misma, y si no cumple determinados criterios, hay que realizar de forma iterativa, sucesivos refinamientos en el diseño.
- Uno de los parámetros que mide la calidad de una base de datos es la **forma normal** en la que se encuentra su diseño.
- El proceso de obligar a los atributos de un diseño a cumplir ciertas formas normales se llama **normalización**.

# Normalización

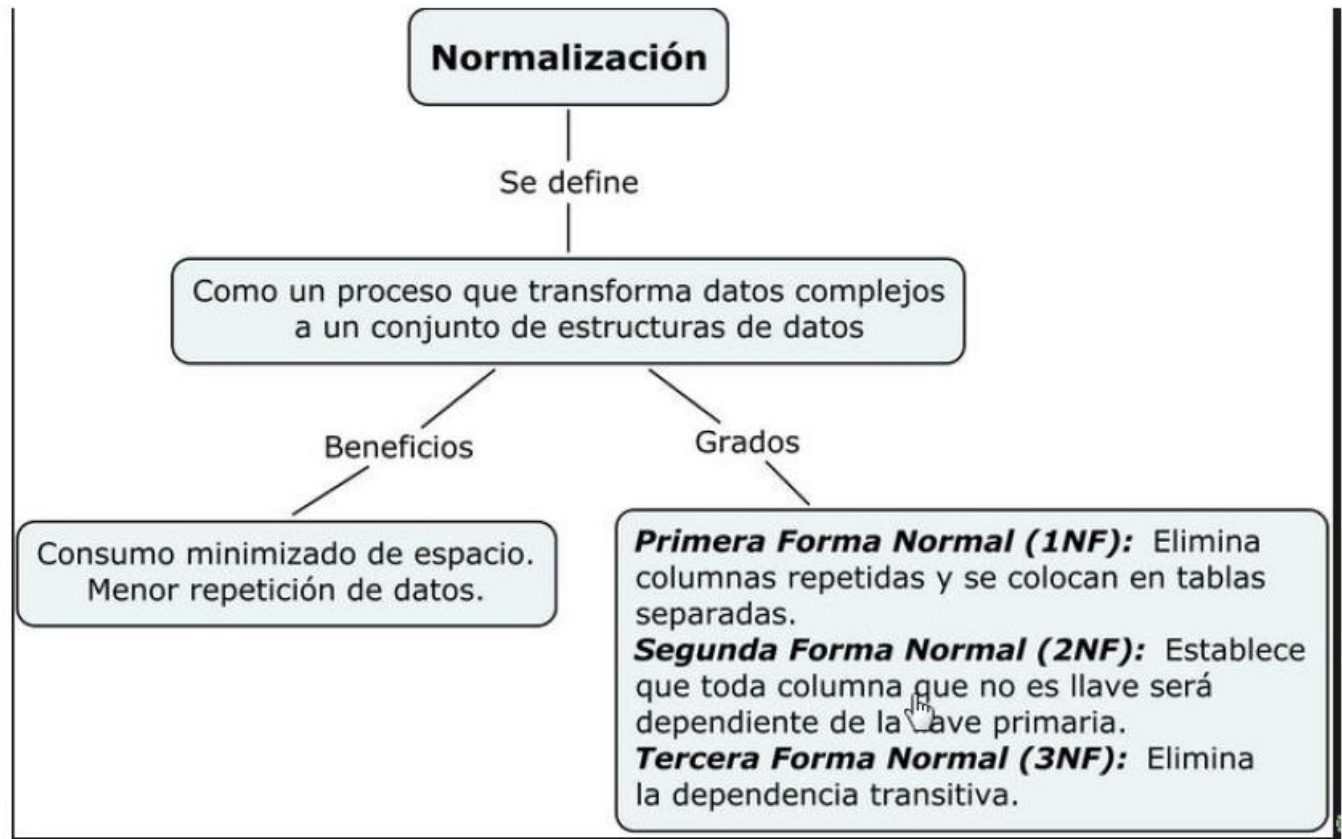
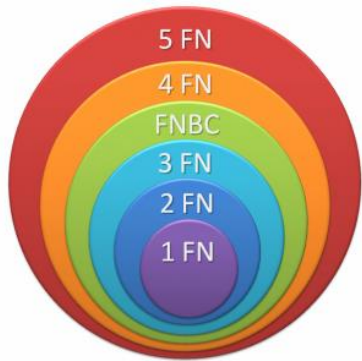


# Normalización

- Las formas normales pretenden alcanzar dos objetivos:
  1. Almacenar en la base de datos cada hecho solo una vez, es decir, evitar la redundancia de datos. De esta manera se reduce el espacio de almacenamiento.
  2. Que los hechos distintos se almacenen en distintos sitios. Esto evita ciertas anomalías a la hora de operar con los datos.
- En la medida que se alcanza una forma normal más avanzada, en mayor medida se cumplen estos objetivos.

# Normalización

- Hay definidas 6 formas normales, cada una agrupa a las anteriores.



# Normalización

La **Teoría de normalización** nos permiten asegurar que un esquema relacional cumple unas ciertas propiedades, **evitando**:

- La **redundancia** de los datos: repetición de datos en un sistema.

Problemas de actualización de los datos en las tablas.

- Proteger la **integridad** de los datos.

- Prevenir **inconsistencias**.

- Evitar campos cuyos valores se repitan frecuentemente.

- Visualizar si se pueden combinar campos poco necesarios para convertirlos en uno solo.

- Estructurar adecuadamente las claves primarias y foráneas.

# Normalización

- En la práctica, si la BD se ha diseñado haciendo uso de modelos semánticos como el modelo E/R no suele ser necesaria la normalización. Por otro lado si nos proporcionan una base de datos creada sin realizar un diseño previo, es muy probable que necesitemos normalizar.
- En la teoría de bases de datos relacionales, las formas normales (FN) proporcionan los criterios para determinar el grado de vulnerabilidad de una tabla a inconsistencias y anomalías lógicas. Cuanto más alta sea la forma normal aplicable a una tabla, menos vulnerable será a inconsistencias y anomalías.

# Normalización

- Proceso consistente en comprobar que las tablas (relaciones) definidas cumplen unas determinadas condiciones.
- La normalización pretende solventar los problemas de las tablas obtenidas en la transformación al modelo relacional, forzando la división de una tabla en dos o mas.
- La normalización permite obtener un conjunto adecuado de tablas de tal forma que:
  - El esquema de la base de datos incluya el mínimo numero de atributos necesarios para dar soporte a los requerimientos del sistema.
  - Resulte mas fácil acceder a la base de datos y mantener los datos de la misma



# Normalización

- Mediante la normalización se pueden solucionar diversos errores en el diseño de la base de datos (anomalías), así como *mejorarlo*.
- Las anomalías de datos son comportamientos que se pueden presentar al insertar, borrar y modificar datos en una base de datos relacional, producidos por un diseño deficiente.
- La aplicación de la teoría de la normalización persigue una disminución de las anomalías.

# Normalización

## Base de datos de un instituto

Nº matrícula	DNI	Nombre	Código asignatura	Nombre asignatura	Calificación
1234	11112222H	José Martínez	INF_01	Java	7
1234	11112222H	José Martínez	INF_02	Lmarcas	5
1235	22223333A	Ana López	INF_01	Java	6
1235	22223333A	Ana López	INF_02	Lmarcas	4
1235	22223333A	Ana López	INF_03	PHP	-

# Normalización

**Anomalía de modificación:** Para cambiar el valor de un atributo, se necesita cambiarlo simultáneamente en varios sitios. Por ejemplo, la modificación del apellido de un estudiante implica modificar todas las tuplas en que aparece:

Nº matrícula	DNI	Nombre	Código asignatura	Nombre asignatura	Calificación
1234	11112222H	José Pérez	INF_01	Java	7
1234	11112222H	José Pérez	INF_02	Lmarcas	5
1235	22223333A	Ana López	INF_01	Java	6
1235	22223333A	Ana López	INF_02	Lmarcas	4
1235	22223333A	Ana López	INF_03	PHP	-

# Normalización

**Anomalía de borrado:** Borrar una tupla puede provocar pérdida de información que no se pretendía eliminar. Por ejemplo, eliminar una matrícula implica la pérdida de información sobre las asignaturas en las que solo esta matriculado un alumno. También se pierde la información del alumno si es su única matrícula:

Nº matrícula	DNI	Nombre	Código asignatura	Nombre asignatura	Calificación
1234	11112222H	José Pérez	INF_01	Java	7
1234	11112222H	José Pérez	INF_02	Lmarcas	5
<del>1235</del>	<del>22223333A</del>	<del>Ana López</del>	<del>INF_01</del>	<del>Java</del>	<del>6</del>
<del>1235</del>	<del>22223333A</del>	<del>Ana López</del>	<del>INF_02</del>	<del>Lmarcas</del>	<del>4</del>
<del>1235</del>	<del>22223333A</del>	<del>Ana López</del>	<del>INF_03</del>	<del>PHP</del>	<del>-</del>

# Normalización

**Anomalía de inserción:** La existencia de un objeto requiere la existencia de otro objeto independiente del primero, pero que pertenece a la misma relación. Por ejemplo, para incluir una asignatura hay que incluir al menos un alumno, es decir, no se puede añadir una asignatura que no tiene alumnos matriculados:

Nº matrícula	DNI	Nombre	Código asignatura	Nombre asignatura	Calificación
1234	11112222H	José Pérez	INF_01	Java	7
1234	11112222H	José Pérez	INF_02	Lmarcas	5
∅	∅	∅	INF_04	Redes	-

# Normalización

- **Edgar F. Codd** originalmente definió las tres primeras formas normales (**1FN, 2FN y 3FN**) en 1970. Estas formas normales se han resumido como requiriendo que todos los atributos sean atómicos, dependan de la clave completa y en forma directa (no transitiva).
- La forma normal de **Boyce-Codd (FNBC)** fue introducida en 1974 por los dos autores que aparecen en su denominación.
- Las cuarta y quinta formas normales (**4FN y 5FN**) se ocupan específicamente de la representación de las relaciones muchos a muchos y uno a muchos entre los atributos y fueron introducidas por **Fagin** en 1977 y 1979 respectivamente.
- Cada forma normal incluye a las anteriores.

# Normalización

- La normalización se realiza mediante un proceso sistemático de aplicación de las denominadas formas normales.
- Se dice que una tabla esta en una determinada forma normal si satisface un cierto número de restricciones impuestas por la correspondiente regla de normalización.
- La aplicación de una de estas reglas a un esquema relacional produce un nuevo esquema relacional en el que no se ha introducido ningún nuevo atributo.
- Si una tabla no satisface una determinada regla de normalización, se procede a descomponerla en otras dos nuevas que sí las satisfagan.
- Esto requiere decidir qué atributos de la tabla original van a residir en una u otra de las nuevas tablas.

# Normalización – Definiciones previas

- Uno de los conceptos fundamentales en la normalización es el de **dependencia funcional**.
- **Una dependencia funcional** es una relación entre atributos de una misma relación.
- Estas dependencias son consecuencia de la estructura de la base de datos y de los objetos del mundo real que describen, y no de los valores actualmente almacenados en cada relación.



# Normalización – Definiciones previas

- **Dependencia funcional:**  $A \rightarrow B$ , representa que B es funcionalmente dependiente de A. Para un valor de A siempre aparece un valor de B.

*Ejemplo: Si A es el D.N.I., y B el Nombre, está claro que para un número de D.N.I, siempre aparece el mismo nombre de titular.*

- **Dependencia funcional completa:**  $A \twoheadrightarrow B$ , si B depende de A en su totalidad. Ejemplo: Tiene sentido plantearse este tipo de dependencia cuando A está compuesto por más de un atributo.

*Por ejemplo, supongamos que A corresponde al atributo compuesto: D.N.I.\_Empleado + Cod.\_Dpto. y B es Nombre\_Dpto. En este caso B depende del Cod\_Dpto., pero no del D.N.I.\_Empleado. Por tanto no habría dependencia funcional completa.*

# Normalización – Definiciones previas

- **Dependencia transitiva:**  $A \rightarrow B \rightarrow C$ . Si  $A \rightarrow B$  y  $B \rightarrow C$ , Entonces decimos que C depende de forma transitiva de A.

*Ejemplo: Sea A el D.N.I. de un alumno, B la localidad en la que vive y C la provincia. Es un caso de dependencia transitiva  $A \rightarrow B \rightarrow C$ .*

- **Determinante funcional:** todo atributo, o conjunto de ellos, de los que depende algún otro atributo.

*Ejemplo: El D.N.I. es un determinante funcional pues atributos como nombre, dirección, localidad, etc, dependen de él.*

# Normalización – Definiciones previas

- **Dependencia multivaluada:**  $A \twoheadrightarrow B$ . Son un tipo de dependencias en las que un determinante funcional no implica un único valor, sino un conjunto de ellos. Un valor de A siempre implica varios valores de B.

*Ejemplo: CursoBachillerato  $\twoheadrightarrow$  Modalidad. Para primer curso siempre va a aparecer en el campo Modalidad uno de los siguientes valores: Ciencias, Humanidades/Ciencias Sociales o Artes. Igual para segundo curso.*

# Normalización – El proceso de Normalización

- La **normalización** consiste en analizar el conjunto de tablas obtenido al pasar del modelo E/R al modelo relacional, teniendo en cuenta las claves candidatas y las dependencias existentes entre los atributos de cada tabla
- La normalización se descompone en una serie de pasos, cada uno de los cuales se corresponde con una de las llamadas **formas normales**.
- La verificación de una forma normal implica el cumplimiento de todas las formas normales anteriores.

# **Normalización – Primera forma normal:**

## **FN1**

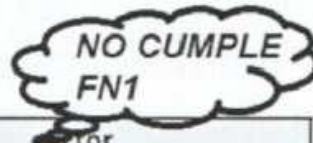
- Se dice que una tabla esta en primera forma normal si impide que un atributo de una tupla pueda tomar mas de un valor. Esto se traduce en que no se admiten atributos multivaluados.
- Todos los atributos son atómicos. Un atributo es atómico si los elementos del dominio son simples e indivisibles.
- La clave primaria no contiene atributos nulos.
- No debe existir variación en el número de columnas.
- Esta forma normal elimina los valores repetidos en una BD.
- Esta forma es inherente al modelo relacional, es decir, todas las tablas construidas basándose en este modelo van a cumplir la FN1.

# Normalización – Primera forma normal: FN1

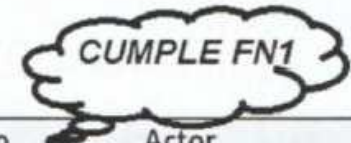
- Para aplicar la primera forma bastará con dividir cada columna no atómica en tantas columnas atómicas como sea necesario.

nombre	teléfono
John Smith	45 35 45 12 35 46 78 98
Carmen Aguilar	55 25 12 45 54 36 11 28

nombre	teléfono
John Smith	45 35 45 12
John Smith	35 46 78 98
Carmen Aguilar	55 25 12 45
Carmen Aguilar	54 36 11 28



Película	Año	Actor
La amenaza Fantasma	1999	Ewan McGregor
		Liam Neeson
		Natalie Portman
Blade Runner	1982	Harrison Ford
		Sean Young
		Rutger Hauer
Avatar	2009	Sam Worthington
		Zoe Saldana
		Sigourney Weaver

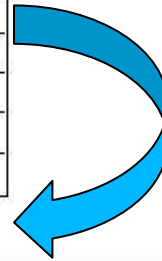


Película	Año	Actor
La amenaza Fantasma	1999	Ewan McGregor
La amenaza Fantasma	1999	Liam Neeson
La amenaza Fantasma	1999	Natalie Portman
Blade Runner	1982	Harrison Ford
Blade Runner	1982	Sean Young
Blade Runner	1982	Rutger Hauer
Avatar	2009	Sam Worthington
Avatar	2009	Zoe Saldana
Avatar	2009	Sigourney Weaver

# Normalización – Primera forma normal: FN1

- Para aplicar la primera forma bastará con dividir cada columna no atómica en tantas columnas atómicas como sea necesario.

MATRICULA	CALIFS
331678	CB-001 10, MA-001 9
337890	FS-001 7, HD-002 8
337777	CB-002 7, CS-056 8
446789	MA-031 7, CB-072 8, HD-002 9



MATRICULA	CLAVE	CALIF
331678	CB-001	10
331678	MA-001	9
337890	FS-001	7
337890	HD-002	8
337777	CB-002	7
337777	CS-056	8
446789	MA-031	7
446789	CV-072	8
446789	HD-002	9

# Normalización – Segunda forma normal: FN2

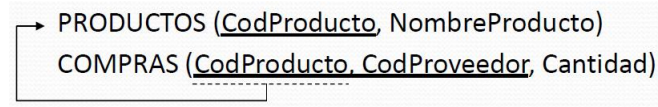
- Se dice que una tabla esta en FN2 si esta en FN1 y además cada atributo que no forma parte de la clave depende funcionalmente de forma completa de la clave primaria (es decir, depende de toda la clave primaria y no de ningún subconjunto de ella), no existen dependencias parciales:

COMPRAS(CodProducto,CodProveedor, NombreProd, Cantidad)



CodProducto -> NombreProd, al no ser dependencia funcional completa no está en 2FN.

- Para pasar a FN2 se separan los grupos de datos relacionados (dependencias parciales de la clave primaria) en tablas separadas y relacionadas entre si mediante uno o varios campos:





# Normalización – Tercera forma normal:

## FN3

- La tercera forma normal consiste en **eliminar las dependencias transitivas**. “Una relación esta en 3FN si y solo si (sii) esta en 2FN y además se cumple que todos los atributos de la relación no dependen transitivamente de la clave primaria.
- Es decir, todas las columnas que no sean claves dependen de la clave completa de forma no transitiva.
- En la practica significa que se debe eliminar cualquier relación que permita llegar a un mismo dato de dos o mas formas diferentes.
- No se da la situación de que un campo dependa de la clave y otro dependa del primero.

# Normalización – Tercera forma normal: FN3

- Se dice que una tabla esta en FN3 cuando esta en FN2 y además, no hay ningún atributo no clave que dependa funcionalmente de forma transitiva de la clave:

ALUMNOS(DNI, Nombre, CodProvincia, Provincia)  
DNI → CodProvincia, CodProvincia → Provincia:  
DNI – → Provincia



- Para obtener una tabla (relación) en FN3 se eliminan las dependencias transitivas trasladándolas a una nueva tabla:

ALUMNOS(DNI, Nombre, CodProvincia)  
PROVINCIAS(CodProvincia, Provincia) ←

# Normalización – Tercera forma normal: FN3

Tomando como ejemplo la tabla siguiente:

NUM_PROV	NOM_PROV	CIUDAD	ESTADO
S1	JUAN	ACAPULCO	2
S2	PEDRO	MERIDA	4
S3	ANA	MERIDA	4
S4	RENE	ACAPULCO	2
S5	PATRICIA	REYNOSA	6
S6	RENATA	REYNOSA	6

Se observa que el atributo ESTADO depende realmente de CIUDAD y no directamente del NUM\_PROV. Para poder normalizar la relación anterior es necesario realizar dos proyecciones y las tablas resultantes serian:

NUM_PROV	NOM_PROV	CIUDAD
S1	JUAN	ACAPULCO
S2	PEDRO	MERIDA
S3	ANA	MERIDA
S4	RENE	ACAPULCO
S5	PATRICIA	REYNOSA
S6	RENATA	REYNOSA

CIUDAD	ESTADO
ACAPULCO	2
MERIDA	4
REYNOSA	6

# Normalización – Tercera forma normal:

## FN3

- Una relación esta en tercera forma normal si, y solo si, esta en 2FN y, además, cada atributo que no esta en la clave primaria no depende transitivamente de la clave primaria. Es decir, los atributos de la relación no dependen unos de otros, dependen únicamente de la clave, este formada por uno o mas atributos. La dependencia  $X \rightarrow Z$  es transitiva si existen las dependencias  $X \rightarrow Y$ ,  $Y \rightarrow Z$ , siendo  $X$ ,  $Y$ , atributos o conjuntos de atributos de una misma relación.
- Para pasar una relación de 2FN a 3FN hay que eliminar las dependencias transitivas.

# Normalización – Tercera forma normal:

## FN3

- 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 del que dependen).
- Teorema de la 3FN: sea una relación formada por los atributos A, B, C con clave primaria formada por el atributo A. Si se cumple que:  $A \rightarrow B$ ,  $B \rightarrow C$ , entonces la relación puede descomponerse en dos relaciones, R1 y R2, con los atributos respectivos: R1 (A, B) y R2 (B, C).

# Normalización – Tercera forma normal: FN3

- Supongamos que tenemos una relación LIBROS en la que representamos los datos de las editoriales de los mismos

<u>COD LIBRO</u>	TÍTULO	EDITORIAL	PAÍS
12345	DISEÑO DE BD RELACIONALES	RAMA	ESPAÑA
34562	INSTALACIÓN y MANTENIMIENTO DE EQUIPOS	MCGRAW-HILL	ESPAÑA
72224	FUNDAMENTOS DE PROGRAMACIÓN	SANTILLANA	ESPAÑA
34522	BASE DE DATOS OO	ADDISON	EEUU

- Las dependencias con respecto a la clave son :
  - TITULO y EDITORIAL dependen directamente del código del libro.
  - El PAIS, aunque en parte también depende del libro, esta mas ligado a la EDITORIAL a la que pertenece el libro. Por esta ultima razón, la relación libros no esta en 3FN.

# Normalización – Tercera forma normal: FN3

<u>COD LIBRO</u>	TÍTULO	EDITORIAL (FK)	<u>EDITORIAL</u>	PAÍS
12345	DISEÑO DE BD RELACIONALES	RAMA	RAMA	ESPAÑA
34562	INSTALACIÓN y MANTENIMIENTO DE EQUIPOS	MCGRAW-HILL	MCGRAW-HILL	ESPAÑA
72224	FUNDAMENTOS DE PROGRAMACIÓN	SANTILLANA	SANTILLANA	ESPAÑA
34522	BASE DE DATOS OO	ADDISON	ADDISON	EEUU

Regla	Descripción
<b>Primera Forma Normal (1FN)</b>	Incluye la eliminación de todos los grupos repetidos
<b>Segunda Forma Normal (2FN)</b>	Asegura que todas las columnas que no son llave sean completamente dependientes de la llave primaria (Primary Key).
<b>Tercera Forma Normal (3FN)</b>	Elimina cualquier dependencia transitiva. Una dependencia transitiva es aquella en la cual las columnas que no son llave son dependientes de de otras columnas que tampoco son llave.

# Normalización

- 1.- Se comprueba que cada tabla tiene un numero fijo de columnas y las variables son sencillas (atómicas)
- 2.- Se identifica la clave primaria
- 3.- Comprueba que todos los atributos (menos la clave primaria) dependen de TODA la clave no de PARTE de ella.
- 4.- Si existe dependencia parcial se rompe la relación en varias subrelaciones.
- 5.- Se comprueba que todos los atributos dependen de la clave y no de otros atributos (dependencias transitivas)
- 6.- Si existen dependencias no relacionadas con la clave primaria

Se subdividen las tablas, y ya tendríamos la base de datos en 3FN



# Forma Normal de Boyce y Codd (FNBC)

- Se define determinante en una relación a un atributo del cual depende funcionalmente de manera completa cualquier otro atributo de la relación. Una relación esta en la Forma Normal de Boyce - Codd (FNBC) si, y solo si, todo determinante de ella es una clave candidata.
- La 2FN y la 3FN eliminan las dependencias parciales y las dependencias transitivas de la clave primaria. Pero este tipo de dependencias todavía pueden existir sobre otras claves candidatas, si las hubiera.
- La BCFN es mas fuerte que la 3FN, por lo tanto, toda relación en FNBC esta en 3FN.

# Forma Normal de Boyce y Codd (FNBC)

- La violación de la FNBC es poco frecuente ya que se da bajo ciertas condiciones que raramente se presentan. Se debe comprobar si una relación no cumple la FNBC si tiene dos o mas claves candidatas compuestas que tienen al menos un atributo en común.
- Teorema de Boyce - Codd: sea una relación R formada por los atributos A, B, C, D con claves candidatas compuestas (A, C) y (B, C) tal que:  $\{AB; A, C \Rightarrow D\}$  entonces la relación puede descomponerse:  
 $R_1 (\{A, B\}, \{AB\})$  y  $R_2 (\{A, C, D\}, \{A, C \Rightarrow D\})$  o bien,  $R_2 (\{B, C, D\}, \{B, C \Rightarrow D\})$

# Forma Normal de Boyce y Codd (FNBC)

- Una forma sencilla de comprobar si una relación se encuentra en FNBC consiste en comprobar, además de que este en 3FN, lo siguiente:
  - (1) Si no existen claves candidatas compuestas (con varios atributos), esta en FNBC.
  - (2) Si existen varias claves candidatas compuestas y estas tienen un elemento común, no esta en FNBC.

# Forma Normal de Boyce y Codd (FNBC)

Supongamos que tenemos una relación EMPLEADOS en la que representamos los datos de los empleados de una fábrica:

DNI	NÚM_SEG_SOC	NOMBRE	APELLIDOS	DEPARTAMENTO	PUESTO	SALARIO
413245-B	28-1234566	JUAN	RAMOS	COMPRAS	GERENTE	2.300
23456-J	28-2345686	PEDRO	PÉREZ	NÓMINAS	AUXILIAR	1.200
123123-C	19-458766	MARÍA	GIL	ALMACÉN	CONSERJE	1.530
1234556-B	45-223344	ANTONIO	SANZ	COMPRAS	GESTIÓN	2.200

Podemos ver que NUM\_SEG\_SOC y el grupo NOMBRE-APELLIDOS son claves candidatas (determinantes). Esta relación se transforma en dos tablas: una contendrá la clave junto con las claves candidatas (EMPLEADOS) y la otra la clave con el resto de campos (EMPLE\_TRABAJO):

<u>DNI</u>	NUM_SEG_SOC	NOMBRE	APELLIDOS
413245-B	28-1234566	JUAN	RAMOS
23456-J	28-2345686	PEDRO	PÉREZ
123123-C	19-458766	MARÍA	GIL
1234556-B	45-223344	ANTONIO	SANZ

<u>DNI (FK)</u>	DEPARTAMENTO	PUESTO	SALARIO
413245-B	COMPRAS	GERENTE	2.300
23456-J	NÓMINAS	AUXILIAR	1.200
123123-C	ALMACÉN	CONSERJE	1.530
1234556-B	COMPRAS	GESTIÓN	2.200

# **Lenguajes Relacionales**

## **Álgebra y Cálculo Relacional**

# Lenguajes Relacionales: Álgebra Relacional

- Base de los lenguajes de consulta de bases de datos utilizados habitualmente, es decir, la base de los lenguajes utilizados para obtener información almacenada en una base de datos.
- Operadores básicos:
  - Selección
  - Proyección
  - Unión
  - Diferencia de conjuntos
  - Producto cartesiano
  - Renombrar

# Lenguajes Relacionales: Álgebra Relacional

- Otros operadores que simplifican consultas habituales:
  - Intersección de conjuntos
  - Join (reunión) natural
  - División
  - Asignación
- Los operadores se aplican sobre una o mas relaciones (tablas) y dan como resultado una nueva relación

# Operación de Selección (Sigma)

- Notación:
  - $\sigma_p(r)$
- Obtiene las tuplas de la relación,  $r$ , que cumplen la condición especificada,  $p$
- $P$  es una fórmula en cálculo proposicional formada por términos unidos por:
  - $\wedge$  : y
  - $\vee$  : o
  - $\neg$  : no
- Y cada término tiene la forma:
  - Atributo op atributo (o constante)
  - Donde op puede ser:  $=, \neq, >, \geq, <, \leq$



# Operación de Selección (Sigma)

- Tabla1 (A, B, C, D)
- Obtener las tuplas en las que el atributo A sea igual a B y que además cumplan que D sea mayor que 5:
- Solución:  $\sigma_{A=B \wedge D > 5}(\text{Tabla1})$

A	B	C	D
$\alpha$	$\alpha$	1	7
$\alpha$	$\beta$	5	7
$\beta$	$\beta$	12	3
$\beta$	$\beta$	23	10



A	B	C	D
$\alpha$	$\alpha$	1	7
$\beta$	$\beta$	23	10


# Operación de Proyección ( $\Pi$ )

- Notación:
  - $\Pi_{A_1, A_2, \dots, A_k}(r)$
- Obtiene una tabla con las columnas de los atributos listados, eliminando también las filas duplicadas del resultado
- $r$  es el nombre de la relación (tabla)
- $A_1, A_2, \dots, A_k$  son nombre de atributos

# Operación de Proyección ( $\Pi$ )

- Tabla1 (A, B, C, D)
- Eliminar las columnas de los atributos B y D
- Solución:  $\Pi_{A,C}(\text{Tabla1})$

A	B	C	D
$\alpha$	$\alpha$	1	7
$\alpha$	$\beta$	1	7
$\beta$	$\beta$	1	3
$\beta$	$\beta$	2	10



A	C
$\alpha$	1
$\beta$	1
$\beta$	2



# Operación de Unión

- Notación:
  - $r \cup s$
- Obtiene una relación cuyas tuplas son las que hay en  $r$  y las que hay en  $s$
- Para que la unión sea válida:
  - $r$  y  $s$  deben tener el mismo número de atributos
  - Los dominios de los atributos deben ser compatibles

# Operación de Unión

- Tabla1 (A, B, C, D)

A	B	C	D
$\alpha$	$\alpha$	1	7
$\alpha$	$\beta$	1	7
$\beta$	$\beta$	1	3
$\beta$	$\beta$	2	10

- Tabla2 (A, B, C, D)

A	B	C	D
$\alpha$	$\alpha$	2	8
$\beta$	$\beta$	2	4

- Tabla1 U Tabla2

A	B	C	D
$\alpha$	$\alpha$	1	7
$\alpha$	$\beta$	1	7
$\beta$	$\beta$	1	3
$\beta$	$\beta$	2	10
$\alpha$	$\alpha$	2	8
$\beta$	$\beta$	2	4

# Operación Diferencia de Conjuntos

- Notación:
  - $r - s$
- Obtiene una relación con las tuplas existentes en la tabla  $r$  que no están en la tabla  $s$
- Para poder realizar la diferencia:
  - $r$  y  $s$  deben tener el mismo número de atributos
  - Los dominios de los atributos de  $r$  y  $s$  deben ser compatibles

# Operación Diferencia de Conjuntos

- Tabla1 (A, B, C, D)

A	B	C	D
$\alpha$	$\alpha$	1	7
$\alpha$	$\beta$	1	7
$\beta$	$\beta$	1	3
$\beta$	$\beta$	2	10

- Tabla2 (A, B, C, D)

A	B	C	D
$\alpha$	$\alpha$	2	8
$\beta$	$\beta$	1	3

- Tabla1 - Tabla2

A	B	C	D
$\alpha$	$\alpha$	1	7
$\alpha$	$\beta$	1	7
$\beta$	$\beta$	2	10



# Operación Producto Cartesiano

- Notación:
  - $r \times s$
- Obtiene una relación cuyas tuplas están formadas por la concatenación de las tuplas de  $r$  con todas las tuplas de  $s$
- Los atributos de  $r$  y  $s$  han de ser disjuntos (es decir distintos), si esto no fuese así habría que utilizar la operación renombrar



# Operación Producto Cartesiano

- Tabla1 (A, B, C, D)

A	B	C	D
$\alpha$	$\alpha$	1	7
$\alpha$	$\beta$	1	7
$\beta$	$\beta$	1	3

- Tabla2 (E,F)

E	F
2	8
1	3

- Tabla1 x Tabla2

A	B	C	D	E	F
$\alpha$	$\alpha$	1	7	2	8
$\alpha$	$\alpha$	1	7	1	3
$\alpha$	$\beta$	1	7	2	8
$\alpha$	$\beta$	1	7	1	3
$\beta$	$\beta$	1	3	2	8
$\beta$	$\beta$	1	3	1	3

# Operación de Renombrado (Ro)

- Notación:
  - $\rho_X(E)$
- Devuelve el resultado de la expresión  $E$  con el nombre  $X$
- Notación:
  - $\rho_{X(A_1, A_2, \dots, A_n)}(E)$
- Devuelve el resultado de la expresión  $E$  con el nombre  $X$  y con los atributos renombrados a  $A_1, A_2, \dots, A_n$

# Operación Intersección de Conjuntos

- Notación:
  - $r \cap s$
- Obtiene una relación que contiene las tuplas de  $r$  que también se encuentran en  $s$
- Para poder realizar la intersección:
  - $r$  y  $s$  deben tener el mismo número de atributos
  - Los dominios de los atributos de  $r$  y  $s$  deben ser compatibles
- $r \cap s = r - (r - s)$



# Operación Intersección de Conjuntos

- Tabla1 (A, B)

A	B
$\alpha$	1
$\alpha$	2
$\beta$	1

- Tabla2 (A,B)

A	B
$\alpha$	2
$\beta$	3

- Tabla1  $\cap$  Tabla2

A	B
$\alpha$	2

# Operación Join Natural (Reunión)

- Notación:
  - $r \bowtie s$
- Obtiene una relación cuyas tuplas son todas las tuplas de  $r$  concatenadas con todas las tuplas de  $s$  que en los atributos comunes (los que se llaman igual) tienen los mismos valores. Los atributos comunes aparecen sólo una vez en el resultado

# Operación Join Natural (Reunión)

- Tabla1 (A, B, C, D)

A	B	C	D
$\alpha$	1	$\alpha$	a
$\beta$	2	$\gamma$	a
$\gamma$	4	$\beta$	b
$\alpha$	1	$\gamma$	a
$\delta$	2	$\beta$	b

- Tabla2 (B, D, E)

B	D	E
1	a	$\alpha$
3	a	$\beta$
1	a	$\gamma$
2	b	$\delta$
3	b	$\epsilon$

- Tabla1  $\bowtie$  Tabla2

A	B	C	D	E
$\alpha$	1	$\alpha$	a	$\alpha$
$\alpha$	1	$\alpha$	a	$\gamma$
$\alpha$	1	$\gamma$	a	$\alpha$
$\alpha$	1	$\gamma$	a	$\gamma$
$\delta$	2	$\beta$	b	$\delta$



# Operación División

- Notación:
  - $r \div s$
- Suponiendo que la cabecera de  $r$  es el conjunto de atributos  $A$  y la cabecera de  $s$  es el conjunto de atributos  $B$ , tales que  $B$  es un subconjunto de  $A$ . Si se considera  $C$  como el subconjunto de los atributos de  $A$  que no están en  $B$ , la división obtiene una relación cuya cabecera es el conjunto de atributos  $C$  y que contiene las tuplas de  $r$  que están acompañadas de todas las tuplas de  $s$

# Operación División

- Tabla1 (A, B, C, D, E)

A	B	C	D	E
$\alpha$	a	$\alpha$	a	1
$\alpha$	a	$\gamma$	a	1
$\alpha$	a	$\gamma$	b	1
$\beta$	a	$\gamma$	a	1
$\beta$	a	$\gamma$	b	3
$\gamma$	a	$\gamma$	a	1
$\gamma$	a	$\gamma$	b	1
$\gamma$	a	$\beta$	b	1

- Tabla2 (D, E)

D	E
a	1
b	1

- Tabla1  $\div$  Tabla2

A	B	C
$\alpha$	a	$\gamma$
$\gamma$	a	$\gamma$



# Operación Asignación

- Notación:
  - Variable  $\leftarrow$  Expresión
- Esta operación facilita un modo conveniente de expresar consultas complejas.
- El resultado de la expresión a la derecha de la flecha se asigna a la variable relación de la izquierda

# Cálculo Relacional de Tuplas

- Lenguaje en el que cada consulta tiene la forma:
  - $\{t \mid P(t)\}$
  - Conjunto de todas las tuplas  $t$  tales que el predicado  $P$  es verdadero para  $t$
- $t$  es una variable tupla,  $t[A]$  denota el valor de la tupla  $t$  en el atributo  $A$
- $t \in r$  denota que la  $t$  está en la relación  $r$
- $P$  es una fórmula similar a las del cálculo de predicados

# Cálculo Relacional. Fórmulas en el Cálculo de Predicados

- Pueden usarse:
  - Conjunto de atributos o constantes
  - Conjunto de operadores de comparación:  $=$ ,  $\neq$ ,  $>$ ,  $\geq$ ,  $<$ ,  $\leq$
  - Conjunto de conectivas: y ( $\wedge$ ), o ( $\vee$ ), no ( $\neg$ )
  - Implicación:  $\Rightarrow$ 
    - $x \Rightarrow y$  : Si  $x$  es verdad, entonces  $y$  es verdad
  - Conjunto de cuantificadores:
    - $\exists t \in r (Q(t))$ : Existe una tupla en la relación  $r$  tal que el predicado  $Q(t)$  es verdad
    - $\forall t \in r (Q(t))$ : El predicado  $Q$  es verdad para todas las tuplas de la relación  $r$