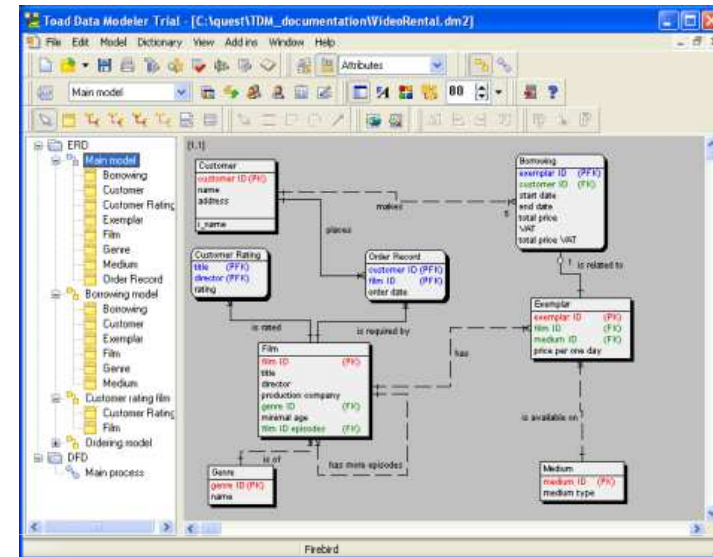


- ALUMNOS (DNIA, Teléfono, Email, Nombre, Apellidos, Dirección)
- MATRICULAN (DNIA, Código\_Curso, Fecha)
- CURSOS (Código\_Curso, Nombre, Descripción)
- TIENEN (Código\_Curso, Código\_Asignatura)
- ASIGNATURAS (Código\_Asignatura, Nombre, Dirección, DNIP)
- PROFESORES (DNIP, Nombre, Apellidos, Teléfono, Email, Dirección, Titulación)
- REALIZAN (DNIA, Código\_Examen, Fecha\_realizacion, Asistencia, Nota)
- EXAMENES (Código\_Examen, Fecha, Código\_Asignatura, DNIP)
- PREGUNTAS (Código\_P, Enunciado, Tipo, Respuesta, Código\_Examen)

## Tema 3.2



# Normalización Algebra relacional



# El Modelo Relacional

## 1.- El Modelo Relacional

### 1.1.- Las Relaciones en el modelo Relacional

### 1.2.- Conceptos necesarios para pasar del Modelo Conceptual (E/R) al Lógico ( Relacional)

## 2.- Transformación de un diagrama E/R al modelo Relacional

## 3.- Normalización

## 4.- MySQL Workbench

# Pérdida de semántica en la transformación al modelo relacional

- Algunas restricciones son necesarias controlarlas con mecanismos externos al modelo relacional.
- Las restricciones de los esquemas E-R que deben contemplarse en la transformación al modelo relacional mediante *checks*, *asepciones* o *disparadores* son:
  - Cardinalidades mínimas de 1 en relaciones N:M y 1:N (excluyendo aquellas que se controlan con la restricción NOT NULL cuando se realiza una propagación de clave).
  - Cardinalidades máximas conocidas en relaciones binarias N:M y 1:N y relaciones ternarias.
  - Exclusividad en las generalizaciones.
  - Inserciones y borrado en las generalizaciones.
  - Restricciones que no figuran en el enunciado original pero que se consideran adecuadas o convenientes (por ejemplo, restricciones que implican operadores de comparación de fechas).

# 3.- Normalización

- La normalización es un proceso que consiste en comprobar que las relaciones definidas cumplen unas determinadas condiciones.
- Su objetivo fundamental es **eliminar la redundancia**.
- Mediante la normalización se pueden solucionar diversos errores en el diseño de la base de datos, así como *mejorarlo*.
- Además, se facilita el trabajo posterior del administrador de la base de datos y de los desarrolladores de aplicaciones.
- El punto de partida del proceso de normalización es el esquema relacional.

# 3.- Normalización

- 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.
- Ejemplo. 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	-

## 3.- 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	-

# 3.- 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 está 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>

## 3.- 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:

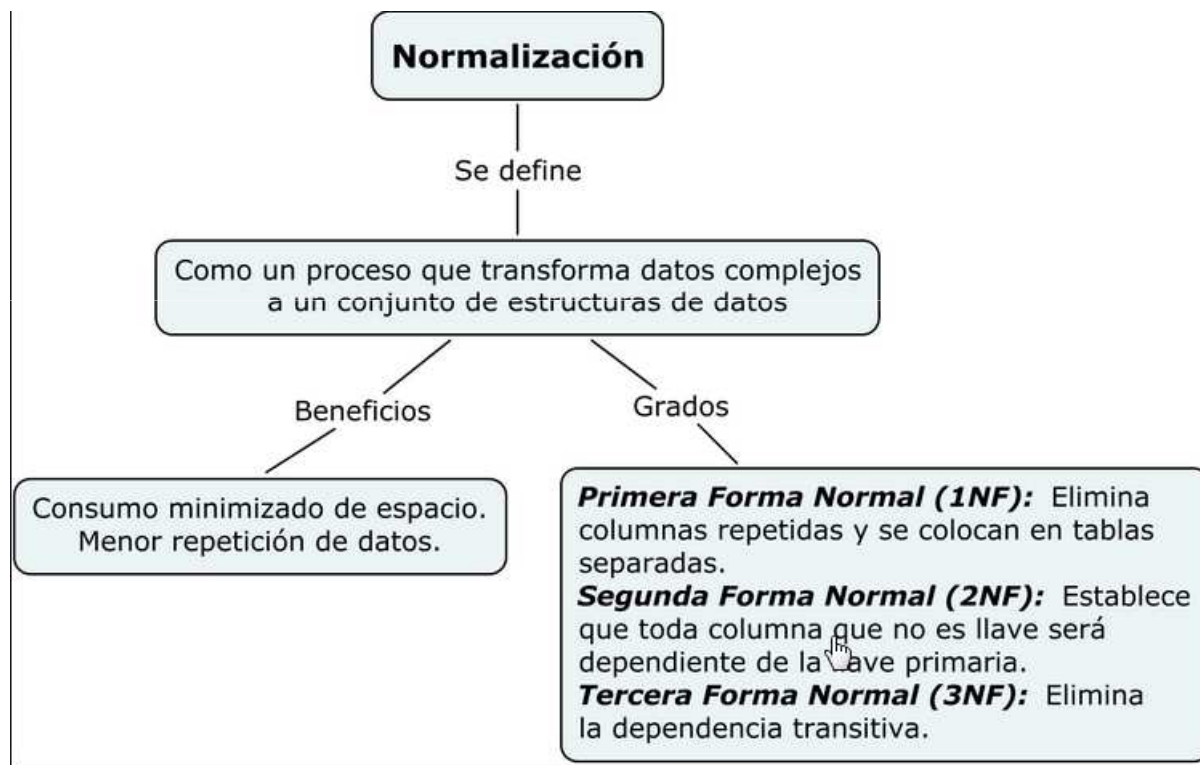
Nº matrícula	DNI	Nombre	Código asignatura	Nombre asignatura	Calificación
1234	11112222H	José Pérez	INF_01	Java	7
1234	11112222H	José Pérez	INF_02	Lmarcas	5
∅	∅	∅	INF_04	Redes	-



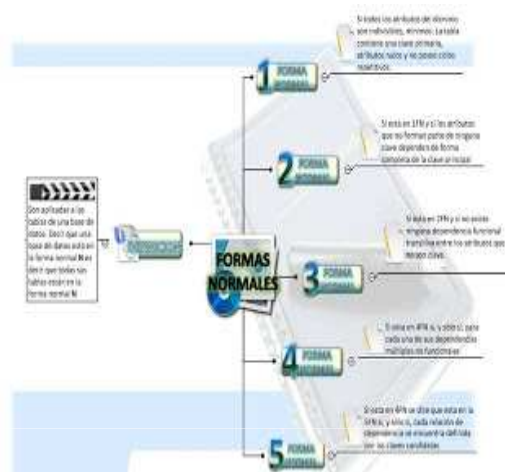
# 3.- Normalización

- La aplicación de la teoría de la normalización persigue una disminución de las anomalías.
- 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.

# 3.- Normalización



# 3.- Normalización



El proceso de normalización de bases de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del modelo entidad-relación al modelo relacional.

Las bases de datos relacionales se normalizan para:

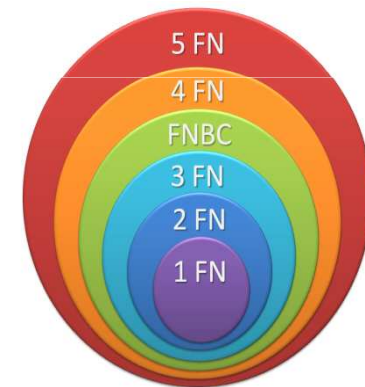
- Evitar la redundancia de los datos.
- Evitar problemas de actualización de los datos en las tablas.
- Proteger la integridad de los datos.

## Formas Normales:

Las formas normales son aplicadas a las tablas de una base de datos. Decir que una base de datos está en la forma normal N es decir que todas sus tablas están en la forma normal.

# 3.- Normalización

- A medida que se avanza en el cumplimiento de reglas y restricciones se alcanza una mayor forma normal.
- Existen seis formas normales, que se deben ir cumpliendo a medida que se realiza el proceso de normalización:



Un esquema relacional que satisface todas

Las restricciones impuestas por la **tercera**

**forma normal** se considera de buena calidad aunque es mejor que satisfaga una interesante propiedad, la forma normal de Boyce-Codd.

La verificación de una forma normal implica el cumplimiento de todas las formas normales anteriores.

# 3.- Normalización

- La normalización es una etapa posterior a la correspondencia entre el esquema conceptual y el esquema lógico, que elimina las dependencias no deseadas entre atributos. Las ventajas de la normalización son las siguientes:
- Evita anomalías en inserciones, modificaciones y borrados.
- Mejora la independencia de datos.
- No establece restricciones artificiales en la estructura de los datos.
- 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.

# Cálculo de las Dependencias

- Entre los atributos de una relación (R) pueden existir dependencias de varios tipos. Las dependencias son propiedades inherentes al contenido semántico de los datos , forman parte de las restricciones de usuario del modelo relacional.
- Existen distintos tipos:
- **1/ DEPENDENCIAS FUNCIONALES:**  
Sea el esquema de relación  $R(AT, DEP)$  ,y X e Y dos subconjuntos de AT llamados descriptores. Y depende funcionalmente de X (X implica o determina Y) si para cada valor de X solo existe un único valor posible para Y. Al conjunto X se le llama **implicante** o **determinante** y al conjunto Y **implicado**.  $X \rightarrow Y$

# Dependencia funcional

- **Definición:** Sean X e Y subconjuntos de atributos de una relación. Diremos que Y tiene una dependencia funcional de X, o que X determina a Y :  
 $X \rightarrow Y$ , si cada valor de X tiene asociado siempre un único valor de Y. El hecho de que X determine a Y no quiere decir que conociendo X se pueda conocer Y, sino que en la relación indicada, cada vez que el atributo X tome un determinado valor, el atributo Y en la misma tupla siempre tendrá el mismo valor.
- Por ejemplo:  
**PRODUCTOS**(CodProducto, Nombre, Precio, Descripción)  
 $\text{CodProducto} \rightarrow \text{Nombre}$  puesto que un CodProducto sólo puede tener asociado un único nombre
- Las dependencias funcionales del sistema se escriben utilizando una flecha, de la siguiente manera:



# Tipos de Dependencias

- **DEPENDENCIAS FUNCIONALES:**
  - ✓ **Dependencia funcional (DF) Trivial:** Si Y es un subconjunto de X .
  - ✓ **Dependencia funcional (DF) Elemental:** Si Y es un único atributo, se trata de una dependencia (no parcial) y no trivial plena (Por ejemplo:  $\text{DNIProfesor} \rightarrow \text{NombreProfesor}$ )
  - ✓ **Dependencia funcional (DF) Transitiva:** Si en el esquema de la relación  $R(\{X,Y,Z\}, \text{DEP})$  existen las dependencias  $X \rightarrow Y$ ;  $Y \rightarrow Z$ ;  $Y \not\rightarrow X$  (por ejemplo:  $\text{NombreProfesor} \rightarrow \text{CódigoPostal}$ ;  $\text{CódigoPostal} \rightarrow \text{Ciudad}$ ;  $\text{CódigoPostal} \not\rightarrow \text{NombreProfesor}$ )
- **2/ DEPENDENCIAS MULTIVALUADAS:** Sean X e Y dos descriptores. **X multidetermina Y** si para cada valor de X existe un conjunto bien definido de valores posibles en Y, con independencia del resto de atributos de la relación.  $X \twoheadrightarrow Y$

DNI	Titulación
11223344L	Magisterio Música
11223333K	Ing. Sup. Informatic
12345678P	Ing. Caminos



# Tipos de Dependencias

- **Dependencia funcional Completa:** Dada una serie de atributos se dice que Y tiene dependencia funcional completa de X, o que  $X \rightarrow Y$ , **si depende funcionalmente de X, pero no depende de ningún otro subconjunto del mismo**. Por ejemplo:
  - COMPRAS (CodProducto, CodProveedor, Cantidad, FechaCompra)  
CodProducto, CodProveedor  $\rightarrow$  FechaCompra, puesto que la fechaCompra es única para la combinación CodProducto, CodProveedor (se puede hacer un pedido al día de cada producto a cada proveedor), y sin embargo, se pueden hacer varios pedidos del mismo producto a diferentes proveedores:  
CodProducto  $\rightarrow$  FechaCompra

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

DNIProfesor  $\rightarrow$  NombreProfesor

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: DNIProfesor  $\Leftrightarrow$  NombreProfesor

Ambos atributos serían equivalentes.

# Tipos de Dependencias

- **Dependencia funcional transitiva:** Dada la tabla T, con atributos (X,Y,Z), donde  $X \rightarrow Y$ ,  $Y \rightarrow Z$  e  $Y \not\rightarrow X$ , se dice que **Z depende transitivamente de X a través de Y**, o que  $X \rightarrow Z$ . Ejemplos:
- **PRODUCTOS**(CodProducto, Nombre, Fabricante, País)  
 $\text{CodProducto} \rightarrow \text{Fabricante}$   
 $\text{Fabricante} \rightarrow \text{País}$   
 $\text{CodProducto} \rightarrow \text{País}$ , es decir, País depende transitivamente de CodProducto.
- **CIUDADES**(ciudad, población, superficie, renta, país, continente)  
Los atributos como *población*, *superficie* o *renta* tienen dependencia funcional de *ciudad*, así que de momento no nos preocupan.  
En esta relación podemos encontrar también las siguientes dependencias:
- $\text{ciudad} \rightarrow \text{país}$ ,  $\text{país} \rightarrow \text{continente}$ . Además,  $\text{país} \rightarrow | \text{ciudad}$ . Es decir, cada *ciudad* pertenece a un *país* y cada *país* a un *continente*, pero en cada *país* puede haber muchas *ciudades*. En este caso *continente* tiene una dependencia funcional transitiva con respecto a *ciudad*, a través de *país*. Es decir, cada *ciudad* está en un *país*, pero también en un *continente*.

# PRIMERA FORMA NORMAL

- Para que una base de datos cumpla la primera forma normal, **cada columna debe ser atómica**, es decir, no está permitido que en una tabla haya atributos que puedan tomar más de un valor, tampoco pueden existir tuplas idénticas o lo que es lo mismo no existen grupos repetitivos.
- 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

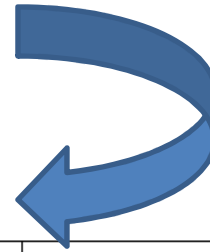
nombre	teléfono
John Smith	45 35 45 12
John Smith	35 46 78 98
Carmen Aguilar	55 25 12 45
Carmen Aguilar	54 36 11 28

# PRIMERA FORMA NORMAL

- Si una relación no está en 1FN, hay que eliminar de ella los grupos repetitivos. Un grupo repetitivo será el atributo o grupo de atributos que tiene múltiples valores para cada tupla de la relación.
- Hay dos formas de eliminar los grupos repetitivos:
  - Repetir los atributos con un solo valor para cada valor del grupo repetitivo. De este modo, se introducen redundancias ya que se duplican valores, pero estas redundancias se eliminarán después mediante las restantes formas normales.
  - La segunda forma de eliminar los grupos repetitivos consiste en poner cada uno de ellos en una relación aparte, heredando la clave primaria de la relación en la que se encontraban.

# PRIMERA FORMA NORMAL

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

# PRIMERA FORMA NORMAL

- La aplicación de esta regla es fácil, por ejemplo, consideramos la tabla ALUMNO, con clave primaria COD\_ALUMNO, en la que el atributo TLF puede tomar varios valores:

<u>COD_ALUMNO</u>	NOMBRE	APELLIDO	TLF	DIRECCION
1111	PEPE	GARCÍA	678-900800 91-2233441 91-1231232	C/Las cañas 45
2222	MARÍA	SUÁREZ	91-7008001	C/Mayor 12
3333	JUAN	GIL	91-7562324 660-111222	C/La plaza
4444	FRANCISCO	MONTAÑA	678-556443	C/La arboleda

Hay dos opciones:

➤ Definir como clave primaria de la tabla COD\_ALUMNO, y el TLF, con el fin de que cada atributo tome un único valor en la tupla correspondiente.

# PRIMERA FORMA NORMAL

<u>COD_ALUMNO</u>	<u>TLF</u>	NOMBRE	APELLIDO	DIRECCIÓN
1111	678-900800	PEPE	GARCÍA	C/Las cañas 45
1111	91-2233441	PEPE	GARCÍA	C/Las cañas 45
1111	91-1231232	PEPE	GARCÍA	C/Las cañas 45
2222	91-7008001	MARÍA	SUÁREZ	C/Mayor 12
3333	91-7562324	JUAN	GIL	C/La plaza
3333	660-111222	JUAN	GIL	C/La plaza
4444	678-556443	FRANCISCO	MONTOYA	C/La arboleda

- O también se eliminan los grupos repetitivos (TLF) y se crea una relación (tabla) junto con la clave inicial:

<u>COD_ALUMNO</u>	NOMBRE	APELLIDO	DIRECCION
1111	PEPE	GARCÍA	C/Las cañas 45
2222	MARÍA	SUÁREZ	C/Mayor 12
3333	JUAN	GIL	C/La plaza
4444	FRANCISCO	MONTOYA	C/La arboleda

<u>COD_ALUMNO(FK)</u>	<u>TLF</u>
1111	678-900800
1111	91-2233441
1111	91-1231232
2222	91-7008001

# PRIMERA FORMA NORMAL



## Concepto de Primera Forma Normal (1FN):



*Se dice que el esquema  $R(T, L)$  **está en 1FN** si en él no existen atributos multivaluados.*



<u>DNI Empleado</u>	Nombre	Puesto Laboral	Nombre-Hijos
14.234.234	Juan	Técnico	Luis, María

~~1FN~~

<u>DNI Empleado</u>	Nombre	Puesto Laboral	<u>Nombre-Hijos</u>
14.234.234	Juan	Técnico	Luis
14.234.234	Juan	Técnico	María

1FN 



- **Interpretación:** Un atributo tiene un único dominio.

Una tabla está en 1FN si sus atributos contienen valores atómicos. En el ejemplo, podemos ver que el atributo nombre\_hijos puede contener más de un valor, por lo que viola 1FN.



# SEGUNDA FORMA NORMAL

- Un diseño se encuentra en 2FN si está en 1FN y además **cada atributo que no forma parte de la clave tiene dependencia completa de la clave principal**, es decir, todas las columnas que formen parte de una *clave candidata* deben aportar información sobre la clave completa. Esta regla significa que **en una relación sólo se debe almacenar información sobre un tipo de entidad**, y se traduce en que los atributos que no aporten información directa sobre la clave principal deben almacenarse en una relación separada. Por ejemplo:
- COMPRAS(CodProducto, CodProveedor, NombreProd, Cantidad, FechaCompra)  
CodProducto -> NombreProd, por tanto al no ser dependencia funcional completa no está en 2FN.
- Lo primero que necesitamos para aplicar esta forma normal es identificar las claves candidatas.

# SEGUNDA FORMA NORMAL

- Se dice que una relación se encuentra en 2FN si y solo si satisface la 1FN, y cada atributo de la relación que no está en la clave depende funcionalmente de forma completa de la clave primaria de la relación. La 2FN se aplica a las relaciones que tienen claves primarias compuestas por dos o más atributos.
- Si una relación está en 1FN y su clave primaria es simple (tiene un solo atributo), entonces también está en 2FN. Las relaciones que no están en 2FN pueden sufrir anomalías cuando se realizan actualizaciones.
- **Teorema de la 2FN:** sea una relación formada por los atributos A, B, C, D con clave primaria compuesta por los atributos A y B. Si se cumple que (D depende funcionalmente de A):  $A \rightarrow D$ , entonces la relación puede descomponerse en dos relaciones R1 y R2 con los atributos respectivos: R1 (A, D) y R2 (A, B, C).
- Para pasar una relación en 1FN a 2FN hay que **eliminar las dependencias parciales de la clave primaria**. Para ello, se eliminan los atributos, que son funcionalmente dependientes, y se ponen en una nueva relación con una copia de su determinante (los atributos de la clave primaria de los que dependen).
- Se crearán dos tablas para eliminar las dependencias funcionales, una de ellas tendrá los atributos que dependen funcionalmente de la clave, y la otra los atributos que forman parte de la clave de la que dependen.

# SEGUNDA FORMA NORMAL

- Supongamos que tenemos una **relación ALUMNO** en la que representamos los datos de los alumnos y las notas en cada una de las asignaturas en que esta matriculado. La clave es el número de matrícula COD\_ALUMNO y la asignatura ASIGNATURA:

<u>COD_ALUMNO</u>	<u>NOM_ALUM</u>	<u>APE_ALUM</u>	<u>ASIGNATURA</u>	<u>NOTA</u>	<u>CURSO</u>	<u>AULA</u>
1111	PEPE	GARCÍA	LENGUA I	5	1	15
1111	PEPE	GARCÍA	IDIOMA	5	2	16
2222	MARÍA	SUÁREZ	IDIOMA	7	2	16
2222	MARÍA	SUÁREZ	CIENCIAS	7	2	14
3333	JUAN	GIL	PLÁSTICA	6	1	18
3333	JUAN	GIL	MATEMÁTICAS I	6	1	12
4444	FRANCISCO	MONTOYA	LENGUA II	4	2	11
4444	FRANCISCO	MONTOYA	MATEMÁTICAS I	6	1	12
4444	FRANCISCO	MONTOYA	CIENCIAS	8	1	14

Todos los atributos no dependen de la clave completa (COD\_ALUMNO, ASIGNATURA). En primer lugar, hay que ver las dependencias funcionales de cada uno de los atributos con respecto a los atributos de la clave y el resto de atributos:

- NOM\_ALUM y APE\_ALUM, solo dependen de COD\_ALUMNO.
- Los atributos CURSO y AULA están relacionados con la ASIGNATURA, es decir, existe una dependencia entre ASIGNATURA → CURSO, ASIGNATURA → AULA. Una asignatura pertenece a un curso y se imparte en un aula.

# SEGUNDA FORMA NORMAL

- El atributo NOTA depende funcionalmente de la clave, pues para que haya una nota tiene que haber una asignatura y un alumno.

Vistas las dependencias funcionales llegamos a la siguiente conclusión:  
“Para que la relación ALUMNO este en 2FN necesitamos crear tres relaciones: ALUMNO, ASIGNATURAS y NOTAS”:

<u>COD_ALUMNO</u>	NOM_ALUM	APE_ALUM
1111	PEPE	GARCÍA
2222	MARÍA	SUAREZ
3333	JUAN	GIL
4444	FRANCISCO	MONTAÑA

<u>ASIGNATURA</u>	CURSO	AULA
LENGUA I	1	15
IDIOMA	2	16
CIENCIAS	2	14
PLÁSTICA	1	18
MATEMÁTICAS I	1	12
LENGUA II	2	11

<u>COD_ALUMNO (FK)</u>	<u>ASIGNATURA (FK)</u>	NOTA
1111	LENGUA I	5
1111	IDIOMA	5
2222	IDIOMA	7
2222	CIENCIAS	7
3333	PLÁSTICA	6
3333	MATEMÁTICAS I	6
4444	LENGUA II	4
4444	MATEMÁTICAS I	6
4444	CIENCIAS	8

# TERCERA FORMA NORMAL

- La tercera forma normal consiste en **eliminar las dependencias transitivas**. “Una relación esta en 3FN si y solo si (sii) está 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 práctica significa que se debe eliminar cualquier relación que permita llegar a un mismo dato de dos o más formas diferentes.
- No se da la situación de que un campo dependa de la clave y otro dependa del primero.

# TERCERA FORMA NORMAL

- 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

# TERCERA FORMA NORMAL

- Una relación esta en tercera forma normal si, y solo si, está en 2FN y, además, cada atributo que no está 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**, esté 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.
- 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  $R2 (B, C)$ .

# TERCERA FORMA NORMAL

- 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, está mas ligado a la EDITORIAL a la que pertenece el libro. Por esta última razón, la relación libros no está en 3FN.

<u>COD LIBRO</u>	TÍTULO	EDITORIAL (FK)	EDITORIAL	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



# TERCERA FORMA NORMAL

- Tenemos una tabla donde se almacenen datos relativos a ciudades, y una de las columnas es el país y otra el continente al que pertenecen. Por ejemplo:
- **Ciudades**(ID\_ciudad, Nombre, población, superficie, renta, país, continente)

Ciudades						
<u>ID_ciudad</u>	Nombre	población	superficie	renta	país	continente
1	Paris	6000000	15	1800	Francia	Europa
2	Lion	3500000	9	1600	Francia	Europa
3	Berlin	7500000	16	1900	Alemania	Europa
4	Pekin	19000000	36	550	China	Asia
5	Bonn	6000000	12	1900	Alemania	Europa

- Para cada aparición de un determinado país, el continente siempre es el mismo. Es decir, existe una redundancia de datos.
- Existe una relación entre país y continente, y ninguna de ellas es clave candidata. Por lo tanto, si queremos que esta tabla sea 3FN debemos separar esa columna:
- **Ciudades**(ID\_ciudad, Nombre, población, superficie, renta, nombre\_pais)  
**Países**(nombre\_pais, nombre\_continente)

# TERCERA FORMA NORMAL

Ciudades					
<u>ID_ciudad</u>	<u>Nombre</u>	<u>población</u>	<u>superficie</u>	<u>renta</u>	<u>país</u>
1	Paris	6000000	15	1800	Francia
2	Lion	3500000	9	1600	Francia
3	Berlin	7500000	16	1900	Alemania
4	Pekin	19000000	36	550	China
5	Bonn	6000000	12	1900	Alemania
Paises					
<u>país</u>	<u>continente</u>				
Francia	Europa				
Alemania	Europa				
China	Asia				

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.

# El proceso de normalización consiste:

- 1.- Se comprueba que cada tabla tiene un número 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 está 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 BCNF es más fuerte que la 3FN, por lo tanto, toda relación en FNBC está en 3FN.
- 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:  $\{A \Leftrightarrow B; A,C \Rightarrow D\}$  entonces la relación puede descomponerse:
- $R_1 (\{A, B\}, \{A \Leftrightarrow B\})$  y  $R_2 (\{A, C, D\}, \{A,C \Rightarrow D\})$  o bien,  $R_2 (\{B, C, D\}, \{B,C \Rightarrow D\})$ .

# Ejemplo1

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

DNI	NUM_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

# Forma Normal de Boyce y Codd (FNBC)

- La forma normal de Boyce-Codd requiere que no existan dependencias funcionales no triviales de los atributos que no sean un conjunto de la clave candidata. En una tabla en 3FN, todos los atributos dependen de una clave, de la clave completa y de ninguna otra cosa excepto de la clave (excluyendo dependencias triviales). Se dice que una tabla está en FNBC si y solo si está en 3FN y cada dependencia funcional no trivial tiene una clave candidata como determinante. En términos menos formales, una tabla está en FNBC si está en 3FN y los únicos determinantes son claves candidatas.

# 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 esté en 3FN, lo siguiente:
- (1) Si no existen claves candidatas compuestas (con varios atributos), está en FNBC.
- (2) Si existen varias claves candidatas compuestas y éstas tienen un elemento común, no está en FNBC.

# Forma Normal de Boyce y Codd (FNBC)

Referencia cruzada de Tutor/Estudiante		
ID Tutor	Número de seguro social del tutor	ID Estudiante
1078	088-51-0074	31850
1078	088-51-0074	37921
1293	096-77-4146	46224
1480	072-21-2223	31850

El propósito de la tabla es mostrar qué tutores están asignados a qué estudiantes.

Las claves candidatas de la tabla son:

- {Número de seguridad social del tutor, ID Estudiante}
- {ID Tutor, ID Estudiante}

En la tabla de ejemplo anterior existen dos claves candidatas y ambas comparten el atributo ID Estudiante, por lo tanto no está en FNBC.



# Forma Normal de Boyce-Codd (FNBC)

## Ejemplo 2

TUTORÍAS		
<u>DNI</u>	<u>Asignatura</u>	<u>Tutor</u>
12121219A	Lenguaje	Eva
12121219A	Matemáticas	Andrés
3457775G	Lenguaje	Eva
5674378J	Matemáticas	Guillermo
5674378J	Lenguaje	Julia
5634823H	Matemáticas	Guillermo

- Esa tabla está en tercera forma normal (no hay dependencias transitivas), pero no en forma de Boyce - Codd, ya que  $(\text{DNI}, \text{Asignatura}) \rightarrow \text{Tutor}$  y  $\text{Tutor} \rightarrow \text{Asignatura}$ . En este caso la redundancia ocurre por mala selección de clave. La redundancia de la asignatura es completamente evitable. La solución sería:

## Ejemplo 2 (FNBC)

TUTORÍAS	
<u>DNI</u>	<u>Tutor</u>
12121219A	Eva
12121219A	Andrés
3457775G	Eva
5674378J	Guillermo
5674378J	Julia
5634823H	Guillermo

ASIGNATURASTUTOR	
<u>Asignatura</u>	<u>Tutor</u>
Lenguaje	Eva
Matemáticas	Andrés
Matemáticas	Guillermo
Lenguaje	Julia

- “Se dice que una relación se encuentra en FNBC si y sólo si todo determinante es clave candidata”. Una relación está en FNBC si está en 3FN y **cualquier atributo sólo facilita información sobre claves candidatas**, y no sobre atributos que no formen parte de ninguna clave candidata.
- Todos los campos determinantes de la tabla son clave candidata pero se debe escoger solo los campos que no provoquen redundancia.
- En las formas de Boyce-Codd hay que tener cuidado al descomponer ya que se podría perder información por una mala descomposición.

# Cuarta Forma Normal (4FN) Ejemplo1

- Una dependencia multivaluada de una tabla con atributos X, Y, Z de X sobre Z (es decir  $X \twoheadrightarrow Z$ ) ocurre cuando los posibles valores de Y sobre cualquier par de valores X y Z dependen sólo del valor de X y son independientes de Z.

<u>Nº Curso</u>	<u>Profesor</u>	<u>Material</u>
17	Eva	1
17	Eva	2
17	Julia	1
17	Julia	2
25	Eva	1
25	Eva	2
25	Eva	3

La tabla está en FNBC ya que no hay dependencias transitivas y todos los atributos son clave sin dependencia funcional hacia ellos. Sin embargo hay redundancia. Los materiales se van a repetir para cualquier profesor dando cualquier curso, ya que los profesores van a utilizar todos los materiales del curso (de no ser así no habría ninguna redundancia).

# 4FN: Ejemplo1

- Los materiales del curso dependen del curso y no del profesor en una dependencia multivaluada. Para el par N° de curso y profesor podemos saber los materiales, pero por el curso y no por el profesor.
- Ocurre esta forma normal cuando una tabla está en forma normal de Boyce Codd y toda dependencia multivaluada es una dependencia funcional. Para la tabla anterior la solución serían dos tablas:

<u>Nº Curso</u>	<u>Material</u>
17	1
17	2
25	1
25	2
25	3

<u>Nº Curso</u>	<u>Profesor</u>
17	Eva
17	Julia
25	Eva

El teorema de Fagin indica cuando hay tres pares de conjuntos de atributos X, Y y Z si ocurre  $X \twoheadrightarrow Y | Z$  (Y y Z tienen dependencia multivaluada sobre X), entonces las tablas X,Y y X,Z reproducen sin perder información lo que poseía la tabla original

# Otras formas normales

- Existen más formas normales (4FN, 5FN, FNDK, 6FN). Las formas normales 4 y 5 se ocupan de las dependencias entre atributos multivaluados, (múltiples valores). Están relacionadas con claves compuestas. Estas formas normales procuran minimizar el número de campos involucrados en una clave compuesta.
- La forma normal Dominio Clave (FNDK) trata las restricciones y los dominios de los atributos y la 6FN está relacionada con las BD temporales.

# EJEMPLO I (1)

- Supongamos que estamos trabajando sobre un sistema que registra las plantas situadas en determinadas ubicaciones y las características del suelo asociadas a ellas.

## **La ubicación:**

Código Ubicación: 11

Nombre Ubicación: Getafe

## **Contiene las siguientes tres plantas:**

Código de Planta: 431

Nombre de Planta: Leuca

Categoría del suelo: A

Descripción del suelo: Arenisca

Código de Planta: 446

Nombre de Planta: Protea

Categoría del suelo: B

Descripción del suelo: Arenisca/Caliza

Código de Planta: 482

Nombre de Planta: Erica

Categoría del suelo: C

Descripción del suelo: Caliza

## **La ubicación:**

Código Ubicación: 12

Nombre Ubicación: Leganés

## **Contiene las siguientes dos plantas:**

Código de Planta: 431

Nombre de Planta: Leuca

Categoría del suelo: A

Descripción del suelo: Arenisca

Código de Planta: 449

Nombre de Planta: Restio

Categoría del suelo: B

Descripción del suelo: Arenisca/Caliza

## EJEMPLO I (2)

- Primero vamos a intentar reorganizar estos datos en forma tabular:

CÓDIGO DE UBICACIÓN	NOMBRE DE UBICACIÓN	CÓDIGO DE PLANTA	NOMBRE DE PLANTA	CATEGORÍA DEL SUELO	DESCRIPCIÓN DEL SUELO
11	Getafe	431 446 482	Leuca Protea Erica	A B C	Arenisca Arenisca/Caliza Caliza
12	Leganés	431 449	Leuca Restio	A B	Arenisca Arenisca/Caliza

A continuación , crearemos una tabla, los campos nulos representarán a los campos que no tienen información en la tabla anterior.

## EJEMPLO I (3)

CÓDIGO DE UBICACIÓN	NOMBRE DE UBICACIÓN	CÓDIGO DE PLANTA	NOMBRE DE PLANTA	CATEGORÍA DEL SUELO	DESCRIPCIÓN DEL SUELO
11	Getafe	431	Leuca	A	Arenisca
NULL	NULL	446	Protea	B	Arenisca/Caliza
NULL	NULL	482	Erica	C	Caliza
12	Leganés	431	Leuca	A	Arenisca
NULL	NULL	449	Restio	B	Arenisca/Caliza

Esta tabla no tiene mucho sentido, Las primeras tres filas forman un grupo, ya que todas pertenecen a la misma ubicación. Por otro lado, los datos no están completos ya que si tomamos la tercera fila, no se puede saber dónde se encuentra Erica. Tampoco se puede utilizar el campo código de Ubicación como clave primaria ya que contiene NULL. **No tiene sentido una tabla en la que no se pueden identificar sus registros de forma exclusiva.** La solución consiste en asegurarse en que cada fila de la tabla sea única y que no forma parte de un grupo o conjunto, para ello eliminaremos los grupos y convertiremos cada fila en un registro completo y único.



# EJEMPLO I (4)

<u>CÓDIGO DE UBICACIÓN</u>	NOMBRE DE UBICACIÓN	<u>CÓDIGO DE PLANTA</u>	NOMBRE DE PLANTA	CATEGORÍA DEL SUELO	DESCRIPCIÓN DEL SUELO
11	Getafe	431	Leuca	A	Arenisca
11	Getafe	446	Protea	B	Arenisca/Caliza
11	Getafe	482	Erica	C	Caliza
12	Leganés	431	Leuca	A	Arenisca
12	Leganés	449	Restio	B	Arenisca/Caliza

La clave primaria debe ser una combinación de Código de Ubicación y Código de planta. Esta tabla cumple con la 1FN ya que no incluye grupos repetitivos ( cada intersección de fila y columna contiene un valor y sólo uno) y tiene una clave primaria.

**TABLAUBICACION**(CODUBICACION, NOMUBICACION, CODPLANTA, NOMPLANTA, CATEGORIASUELO, DESCSUELO)

# EJEMPLO I (5)

- Problemas: El código 11 y su nombre se registra tres veces, lo que, además del gasto de espacio en memoria que supone, puede dar lugar a errores si se comete un fallo al introducir los datos, por ejemplo en algún registro se escribe “Getafw” en lugar de Getafe.
- El siguiente paso consiste en buscar dependencias parciales, e.d, campos que dependan de una parte de la clave y no de la clave entera. En nuestro ejemplo, tenemos:
- CODUBICACION → NOMUBICACION
- CODPLANTA → NOMPLANTA, CATEGORIASUELO, DESC SueLO
- Hay que extraer todos estos datos y colocarlos en una nueva tabla, compuesta de campos que tengan dependencia parcial y de los campos de los que dependan:

<u>CÓDIGO DE UBICACIÓN</u>	<u>CÓDIGO DE PLANTA</u>
11	431
11	446
11	482
12	431
12	449

# EJEMPLO I (6)

<u>CÓDIGO DE PLANTA</u>	NOMBRE DE PLANTA	CATEGORÍA DEL SUELO	DESCRIPCIÓN DEL SUELO
431	Leuca	A	Arenisca
446	Protea	B	Arenisca/Caliza
482	Erica	C	Caliza
449	Restio	B	Arenisca/Caliza

<u>CÓDIGO DE UBICACIÓN</u>	NOMBRE DE UBICACIÓN
11	Getafe
12	Leganés

- Ahora, si está en 2FN ya que está en 1FN y no incluye dependencias parciales:
- **TABLA PLANTA UBICACION**(COD UBICACION, COD PLANTA)
- **TABLA UBICACION**(COD UBICACION, NOM UBICACION)
- **TABLA PLANTA** (COD PLANTA, NOM PLANTA, CATEGORIASUELO, DESC SUELO)

# EJEMPLO I (7)

- Todavía se puede detectar una dependencia indirecta, y es que la categoría del suelo determina la descripción del mismo, por lo que estamos almacenando datos de forma redundante. La solución consiste en extraer los datos sobrantes y colocarlos en su propia tabla. En esta fase estamos buscando dependencias transitivas o relaciones entre dos campos que no sean claves.

<u>CÓDIGO DE PLANTA</u>	NOMBRE DE PLANTA	CATEGORÍA DEL SUELO
431	Leuca	A
446	Protea	B
482	Erica	C
449	Restio	B

CATEGORÍA DEL SUELO	DESCRIPCIÓN DEL SUELO
A	Arenisca
B	Arenisca/Caliza
C	Caliza

De esta forma también estamos reduciendo la posibilidad de que se cometan errores en la escritura de un nombre de suelo, ya que sólo se escribirá una vez.

## EJEMPLO I (8)

- Ahora, si está en 3FN ya que está en 2FN y no tiene dependencias transitivas (un atributo que no es clave depende de una clave primaria a través de otro atributo que no es clave):
- **TABLAPLANTAUBICACION**(CODUBICACION, CODPLANTA)
- **TABLAUBICACION**(CODUBICACION, NOMUBICACION)
- **TABLAPLANTA** (CODPLANTA, NOMPLANTA, CATEGORIASUELO)
- **TABLASUELO** (CATEGORIASUELO, DESC SueLO)

# EJEMPLO II (1)

- Ejemplo con anomalías de datos presentes en la 3FN y que se resuelven con su transformación en la FNBC.

**ESTUDIANTECURSOPROFESOR(ESTUDIANTE,CURSO,PROFESOR)**

Supongamos las siguientes premisas:

- Cada profesor imparte un solo curso
- Cada curso puede llevar asignado uno o varios profesores
- Cada estudiante solo tiene un profesor por curso
- Cada estudiante puede seguir uno o varios cursos

¿Cuál sería la clave? Ninguno de estos campos sería suficiente por sí mismo para identificar de forma exclusiva un registro, por lo que habrá que utilizar dos, pero ¿qué dos seleccionar? Si elegimos estudiante y profesor podemos obtener el curso, **si elegimos estudiante y curso** también obtenemos el profesor, por lo que ambas elecciones serían correctas. Vamos a elegir esta última:

**ESTUDIANTECURSOPROFESOR(ESTUDIANTE,CURSO,PROFESOR)**

## EJEMPLO II (2)

- La tabla obtenida está en **1FN** ya que tiene una clave y no consta de grupos repetidos. También está en **2FN**, ya que el profesor depende de los otros dos campos (los estudiantes tienen varios cursos y, por tanto, varios profesores y los cursos tienen varios profesores). También está en **3FN** ya que sólo tiene un campo que no sea clave.

<u>ESTUDIANTE</u>	<u>CURSO</u>	PROFESOR
María	BD	Luis
José	PROG	Maite
Ana	IAW	Pedro
David	BD	Luis
María	IAW	Carmen
Alicia	IAW	Carmen
Pepe	PROG	Maite

## EJEMPLO II (3)

- El hecho de que Carmen imparta el curso de IAW está almacenado de forma redundante, como ocurre con Maite y PROG. El problema es que el profesor determina el curso o el curso viene determinado por el profesor. La tabla se ajusta a la 3FN pero, un atributo clave depende de un atributo que no es clave, para evitar esto eliminamos el atributo y lo colocamos en otra tabla con su clave:

<u>ESTUDIANTE</u>	<u>PROFESOR</u>		
María	Luis		
José	Maite	<u>PROFESOR</u>	<u>CURSO</u>
Ana	Pedro	Luis	BD
David	Luis	Maite	PROG
María	Carmen	Pedro	IAW
Alicia	Carmen	Carmen	IAW
Pepe	Maite		



## EJEMPLO II (4)

- El campo Profesor no es una clave candidata (ya que por sí solo no identifica un registro), pero determina el curso, por lo que la tabla no estaba en la FNBC.
- Aunque seleccionamos el campo Curso como parte de la clave principal en la tabla original, el profesor determina el curso, que es la razón por la que lo convertimos en clave primaria de esa tabla. El problema de redundancia queda resuelto. Por lo tanto está en la **FNBC**, ya que cumple las siguientes condiciones:
- Está en 3FN
- Cada determinante es una clave candidata.
- **ESTUDIANTEPROFESOR**(ESTUDIANTE, PROFESOR)
- **PROFESORCURSO**(PROFESOR, CURSO)

## EJEMPLO II (5)

- Si elegimos el campo **Estudiante y Profesor** como clave. La tabla estaría en 1FN porque tiene una clave y no incluye grupos repetidos, pero no está en 2FN porque el curso solo viene determinado por el profesor, por lo que si eliminamos el curso y creamos otra tabla con él y su clave (profesor) obtendremos el mismo resultado anterior:

<u>ESTUDIANTE</u>	CURSO	<u>PROFESOR</u>	<u>ESTUDI ANTE</u>	<u>PROFES OR</u>		
María	BD	Luis	María	Luis		
José	PROG	Maite	José	Maite		
Ana	IAW	Pedro	Ana	Pedro		
David	BD	Luis	David	Luis		
María	IAW	Carmen	María	Carmen		
Alicia	IAW	Carmen	Alicia	Carmen		
Pepe	PROG	Maite	Pepe	Maite		
					<u>PROFESOR</u>	CURSO
					Luis	BD
					Maite	PROG
					Pedro	IAW
					Carmen	IAW

# EJEMPLO III (1)

## ESTUDIANTECURSOPROFESOR(ESTUDIANTE,CURSO,PROFESOR)

Supongamos las siguientes premisas:

- Cada profesor imparte un solo curso
- Cada curso puede llevar asignado uno o varios profesores
- Cada estudiante puede tener varios profesores por curso
- Cada estudiante puede seguir uno o varios cursos

<u>ESTUDIANTE</u>	CURSO	<u>PROFESOR</u>
María	BD	Luis
José	PROG	Maite
Ana	IAW	Pedro
David	BD	Luis
María	IAW	Carmen
Alicia	IAW	Carmen
Pepe	PROG	Maite
José	PROG	Pedro

## EJEMPLO III (2)

- Se sigue repitiendo información: Maite (PROG), Carmen (IAW), Alumno: José (PROG)
- El problema es que la tabla almacena más de un tipo de circunstancias: una R estudiante-curso y una R estudiante-profesor. Para evitar esto separamos los datos en tablas:
- **ESTUDIANTEPROFESOR**(ESTUDIANTE, PROFESOR)
- **ESTUDIANTECURSO**(ESTUDIANTE, CURSO)
- Esta situación se da cuando tenemos varias dependencias multivalor.

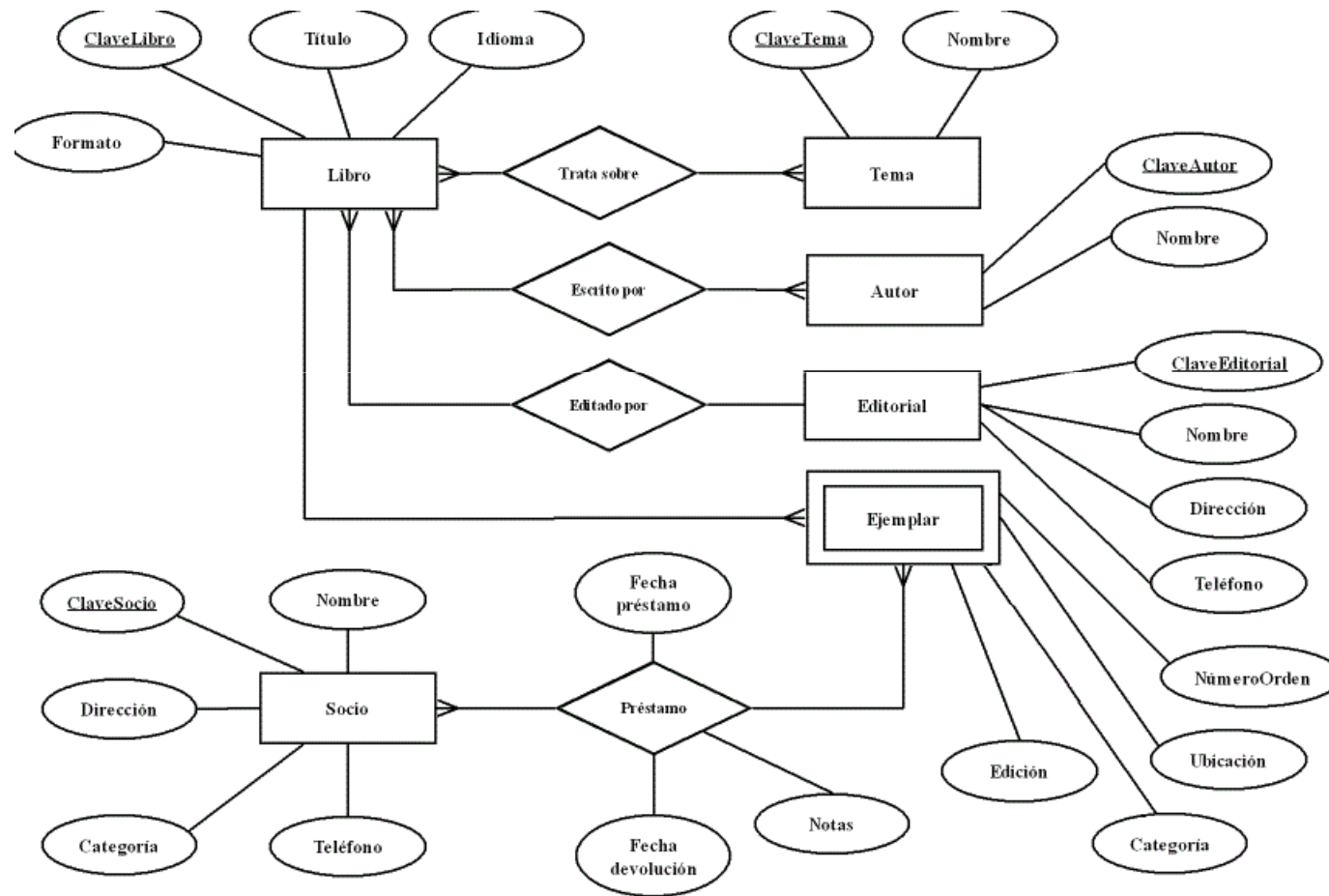
## EJEMPLO III (3)

- Una dependencia multivalor se establece entre dos atributos cuando, para cada valor del primer atributo, existe uno o varios valores asociados al segundo atributo. Para cada valor de estudiantes, existen varios valores de curso. Esta es la primera dependencia multivalor. Después para cada valor de estudiante, existe uno o varios valores de profesor, esta es la segunda.
- Por lo tanto, una tabla está en **4FN** si se cumplen los siguientes criterios:
- Si está en FNBC
- Si no contiene más de una dependencia multivalor.

## 5FN y OTRAS ...

- Básicamente una tabla está en 5FN, si no se puede dividir en tablas más pequeñas con diferentes claves (la mayor parte de las tablas se pueden dividir en tablas más pequeñas con la misma clave).

# Ejemplo



# Ejemplo

- **Libro**(ClaveLibro, Título, Idioma, Formato, ClaveEditorial(FK))
- **Tema**(ClaveTema, Nombre)
- **Autor**(ClaveAutor, Nombre)
- **Editorial**(ClaveEditorial, Nombre, Dirección, Teléfono)
- **Ejemplar**(ClaveEjemplar, ClaveLibro(FK), NúmeroOrden, Edición, Ubicación, Categoría)
- **Socio**(ClaveSocio, Nombre, Dirección, Teléfono, Categoría)
- **Préstamo**(ClaveSocio(FK), ClaveEjemplar(FK), Fecha\_préstamo, Fecha\_devolución, Notas)
- **Trata\_sobre**(ClaveLibro(FK), ClaveTema(FK))
- **Escrito\_por**(ClaveLibro(FK), ClaveAutor(FK))



# Ejemplo

## Primera forma normal

Todos los atributos son atómicos.

## Segunda forma normal

- En el caso de *Libro*, la única clave candidata es ClaveLibro. Todos los demás valores son repetibles, pueden existir libros con el mismo título y de la misma editorial editados en el mismo formato e idioma. Es decir, no existe ningún otro atributo o conjunto de atributos que puedan identificar un libro de forma unívoca.
- Se pueden dar casos especiales, como el del mismo libro escrito en diferentes idiomas. En ese caso la clave será diferente, de modo que los consideraremos como libros distintos. Lo mismo pasa si el mismo libro aparece en varios formatos, o ha sido editado por distintas editoriales. Es decir, todos los atributos son dependencias funcionales de ClaveLibro.

# Ejemplo

- Con *Tema* y *Autor* no hay dudas.
- Los tres atributos de *Editorial* también tienen dependencia funcional de *ClaveEditorial*.
- Y lo mismo cabe decir para las entidades *Ejemplar*, *Socio* y *Préstamo*.

## **Tercera forma normal**

- En *Libro* no hay ningún atributo que tenga dependencia funcional de otro atributo que no sea la clave principal. Todos los atributos definen a la entidad *Libro* y a ninguna otra.
- Las entidades con sólo dos atributos no pueden tener dependencias transitivas, como *Tema* o *Autor*.
- Con *Editorial* tampoco existen, todos los atributos dependen exclusivamente de la clave principal.
- En el caso del *Ejemplar* tampoco hay una correspondencia entre ubicación y edición. Y para *Préstamo* los tres atributos que no forman parte de la clave candidata se refieren sólo a la entidad *Préstamo*.

## **Forma normal de Boyce/Codd**

Tampoco existen atributos que den información sobre otros atributos que no sean o formen parte de claves candidatas.

## **Cuarta forma normal**

No hay atributos multivaluados.

# Dinámica del modelo relacional: álgebra relacional

- La dinámica del modelo relacional actúa sobre conjuntos de tuplas y se expresa mediante lenguajes de manipulación relacionales que asocian una sintaxis a las operaciones. La dinámica la constituye una serie de operadores que aplicados a las relaciones dan como resultado nuevas relaciones.

OPERACIONES BÁSICAS	Operaciones unarias	Selección Proyección
	Operaciones binarias	Unión Diferencia Producto cartesiano
OPERACIONES DERIVADAS	Intersección Cociente Combinación	

# Operaciones básicas unarias

- **Selección.** Esta operación obtiene un subconjunto de filas de una tabla con todas sus columnas. Se pueden seleccionar determinadas filas incluyendo en la operación una condición. En ésta se pueden utilizar los operadores booleanos: Y (AND), O (OR) , NO (NOT) para expresar varios criterios. El grado del resultado es igual al de la tabla tras la operación y la cardinalidad del resultado puede ser igual o menor. Se representa de la siguiente manera:  **$\sigma_{\text{condición}}(\text{Tabla})$ .**

# Selección

- A partir de la tabla EMPLEADOS, seleccionamos aquellas filas cuyo departamento es el 20.

Nº. EMPLE	APELLIDO	SALARIO	COMISIÓN	Nº. DEPART	JEFE
7369	SÁNCHEZ	1.040		20	7902
7499	ARROYO	2.080	390	30	7698
7521	SALA	1.625	650	30	7698
7566	JIMÉNEZ	3.867		20	7839
7654	MARTÍN	1.625	1820	30	7698
7698	NEGRO	3.705		30	7839
7782	CEREZO	3.185		10	7839
7788	GIL	3.900		20	7566
7839	REY	6.500		10	
7876	ALONSO	1.430		20	7788

$\sigma_{\text{departamento}=20}(\text{Empleados})$ .

Nº. EMPLE	APELLIDO	SALARIO	COMISIÓN	Nº. DEPART	JEFE
7369	SÁNCHEZ	1.040		20	7902
7566	JIMÉNEZ	3.867		20	7839
7788	GIL	3.900		20	7566
7876	ALONSO	1.430		20	7788

# Operación Selección ( $\sigma$ )

- Selección
- Selecciona las tuplas que satisfacen un predicado
- Denotada por  $\sigma$
- Sintaxis:  $\sigma_{predicado}(relación)$
- Ejemplo: Sucursales de la ciudad de Madrid

nombreSuc	ciudadSuc	Activo
Castellana	Madrid	90000
Ganivet	Granada	21000
Paseo	Almería	17000
Zapillo	Almería	4000
Ronda	Almería	80000
Aduana	Roquetas	3000
Los Pinos	Huercal	37000
Sol	Madrid	71000

*sucursales*

nombreSuc	ciudadSuc	Activo
Castellana	Madrid	90000
Sol	Madrid	71000

$\sigma_{ciudadSuc="Madrid"}(sucursales)$

Ej. Sucursales de Madrid con activo mayor de 800000

Ej. Sucursales de Madrid con activo mayor de 800000

nombreSuc	ciudadSuc	Activo
Castellana	Madrid	90000
Ganivet	Granada	21000
Paseo	Almería	17000
Zapillo	Almería	4000
Ronda	Almería	80000
Aduana	Roquetas	3000
Los Pinos	Huercal	37000
Sol	Madrid	71000

*sucursales*

nombreSuc	ciudadSuc	Activo
Castellana	Madrid	90000

$\sigma_{ciudadSuc="Madrid" \text{ AND } activo > 800000}(sucursales)$

# Proyección ( $\pi$ )

- **Proyección.** Esta operación da como resultado una nueva tabla a partir de otra con el subconjunto de columnas indicado. Las filas duplicadas solo aparecerán una vez. La proyección se representa de la siguiente manera:  $\pi_{col1,col2,..}(Tabla)$ .

**Ejemplo:**  $\pi_{APELLIDO,SALARIO}(EMPLEADOS)$

Nº. EMPL	APELLIDO	SALARIO	COMISIÓN	Nº. DEPART	JEFE
7369	SÁNCHEZ	1.040		20	7902
7499	ARROYO	2.080	390	30	7698
7521	SALA	1.625	650	30	7698
7566	JIMÉNEZ	3.867		20	7839
7654	MARTÍN	1.625	1820	30	7698
7698	NEGRO	3.705		30	7839
7782	CEREZO	3.185		10	7839
7788	GIL	3.900		20	7566
7839	REY	6.500		10	
7876	ALONSO	1.430		20	7788

APELLIDO	SALARIO
SÁNCHEZ	1.040
ARROYO	2.080
SALA	1.625
JIMÉNEZ	3.867
MARTÍN	1.625
NEGRO	3.705
CEREZO	3.185
GIL	3.900
REY	6.500
ALONSO	1.430

# Operación Proyección ( $\Pi$ )

- Devuelve una relación con ciertas columnas omitidas
- Denotada por  $\Pi$
- **Sintaxis:**  $\Pi$  *columnas(relación)*
- **Ejemplo:** Nombre y teléfono de los empleados

nombreEmp	dniEmp	telefono	NombreSuc
García	10	101010	Castellana
Torres	11	111111	Castellana
López	12	121212	Paseo
Villegas	13	131313	Paseo
Fernández	14	141414	Zapillo
Umutia	15	151515	Sol

*empleados*

nombreEmp	telefono
García	101010
Torres	111111
López	121212
Villegas	131313
Fernández	141414
Umutia	151515

$\Pi$  *nombreEmp,telefono* (*empleados*)



# COMBINACIÓN DE SELECCIÓN CON PROYECCIÓN

- Esta operación permite seleccionar algunas columnas de una selección de registros de una tabla en función de unas condiciones.
- El grado del resultado es igual o menor al de la tabla tras la operación y la cardinalidad del resultado también.
- La operación SELECCIÓN combinada con la operación PROYECCIÓN, puede tener el formato:
- $\Pi$  campos\_proyectados ( $\sigma$ (condicion) (TABLA))

# Secuencias de operaciones

- Aplicar operaciones sobre resultados de otras operaciones. A veces, el resultado de una operación es una relación útil para selección-proyección.
- **Ejemplo: Nombre y teléfono de empleados que trabajan en la sucursal Castellana**

nombreEmp	dniEmp	telefono	NombreSuc
García	10	101010	Castellana
Torres	11	111111	Castellana
López	12	121212	Paseo
Villegas	13	131313	Paseo
Fernández	14	141414	Zapillo
Uruñia	15	151515	Sol

*empleados*

nombreEmp	telefono
García	101010
Torres	111111

$\Pi_{\text{nombreEmp, telefono}} (\sigma_{\text{nombreSuc} = \text{"Castellana"}}(\text{empleados}))$

# Selección, Proyección:

- Obtenemos las columnas APELLIDO y SALARIO de aquellas filas de la tabla EMPLEADOS cuyo departamento es 20. Se representa:  
 $\pi_{\text{APELLIDO, SALARIO}}(\sigma_{\text{NºDEPART}=20}(\text{EMPLEADOS}))$ .

Nº. EMPL	APELLIDO	SALARIO	COMISIÓN	Nº. DEPART	JEFE
7369	SÁNCHEZ	1.040		20	7902
7499	ARROYO	2.080	390	30	7698
7521	SALA	1.625	650	30	7698
7566	JIMÉNEZ	3.867		20	7839
7654	MARTÍN	1.625	1820	30	7698
7698	NEGRO	3.705		30	7839
7782	CEREZO	3.185		10	7839
7788	GIL	3.900		20	7566
7839	REY	6.500		10	
7876	ALONSO	1.430		20	7788

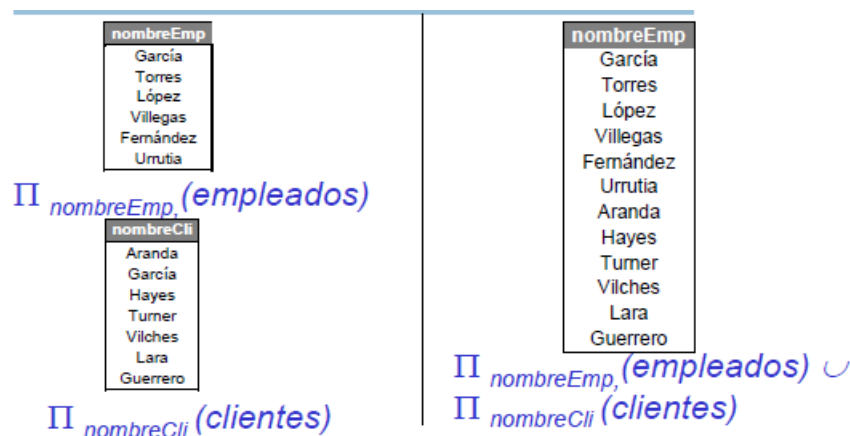
APELLIDO	SALARIO
SÁNCHEZ	1.040
JIMÉNEZ	3.867
GIL	3.900
ALONSO	1.430

# La operación unión **U**

Da lugar a una nueva relación con todas y cada una de las tuplas de las relaciones que participan en la unión.

- $R \cup S$  incluye las tuplas que están en  $R$ , en  $S$  o en ambas
- Las tuplas repetidas son eliminadas.
- Las relaciones a unir han de ser compatibles:
  - Igual número de atributos
  - Dominios iguales dos a dos

**Ejemplo: Nombres que aparecen en Empleados o Clientes**



# UNIÓN

JUGA\_ASI

COD	Nombre	Apellido_1	Apellido_2	Cod_EQUIPO
123232	Pedro	Climon	Cereza	ASI2T
665543	Elena	Raiz	Serrano	ASI2T
827654	Gloria	Tierra	Pelaez	ASI1M

JUGA\_DAI

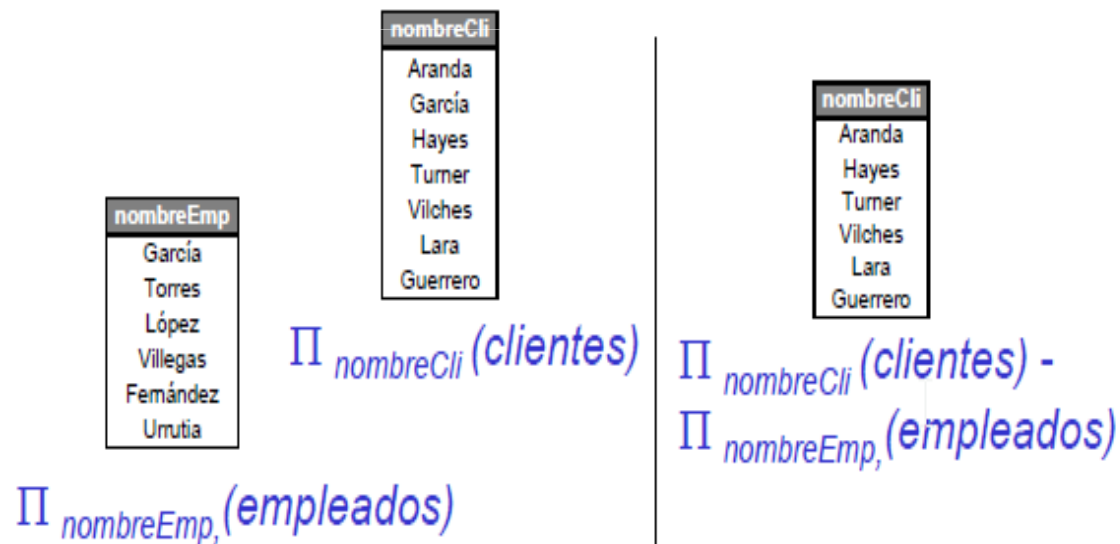
COD	Nombre	Apellido_1	Apellido_2	Cod_EQUIPO
889272	Luis	Garcia	Prat	DAI2T
827654	Ernesto	Villa	Grande	DAI2M
827655	Eva	Tierra	Pelaez	DAI1M
789009	Carlos	Arias	Arias	DAI2M

JUGADORES0607= JUGA\_ASI UNION JUGA\_DAI

COD	Nombre	Apellido_1	Apellido_2	Cod_EQUIPO
123232	Pedro	Climon	Cereza	ASI2T
665543	Elena	Raiz	Serrano	ASI2T
827654	Gloria	Tierra	Pelaez	ASI1M
889272	Luis	Garcia	Prat	DAI2T
827654	Ernesto	Villa	Grande	DAI2M
827655	Eva	Tierra	Pelaez	DAI1M
789009	Carlos	Arias	Arias	DAI2M

# La operación diferencia de conjuntos

- Incluye tuplas que están en una relación pero no en la otra.  $R - S$  incluye las tuplas que están en  $R$  y no están en  $S$ .
- $R$  y  $S$  deben ser compatibles
- **Ejemplo: Clientes que no son empleados**



# La operación diferencia de conjuntos

EQUIPOS06

EQUIPO	CICLO	CURSO	GRUPO	TURNO
ASI1M	ASI	1	NULL	MAÑANA
ASI2M	ASI	2	NULL	MAÑANA
ASI1T	ASI	1	NULL	TARDE
ESO2A	ESO	2	A	NULL

EQUIPOS07

EQUIPO	CICLO	CURSO	GRUPO	TURNO
ASI1M	ASI	1	NULL	MAÑANA
ASI2M	ASI	2	NULL	MAÑANA
ASI1T	ASI	1	NULL	TARDE
ASI2T	ASI	2	NULL	TARDE
GS1	GARANTIA SOCIAL	NULL	1	NULL

EQUIPOS NUEVOS = EQUIPOS07-EQUIPOS06

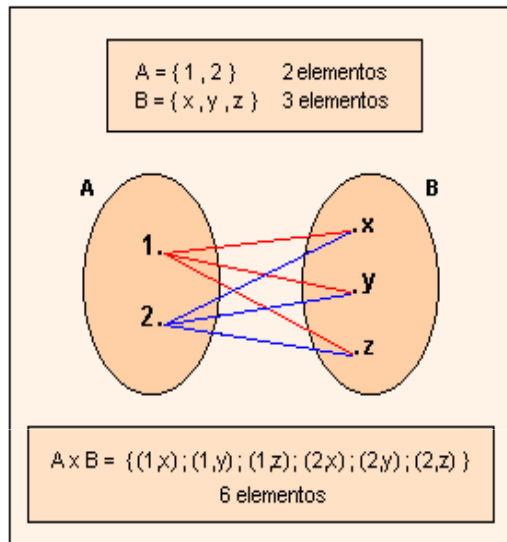
EQUIPO	CICLO	CURSO	GRUPO	TURNO
ASI2T	ASI	2	NULL	TARDE
GS1	GARANTIA SOCIAL	NULL	1	NULL

# La operación producto cartesiano(x)

- Operación para combinar relaciones
- $R \times S$  contiene todas las tuplas de  $R$  combinadas con todas las tuplas de  $S$
- Número de columnas de  $R \times S = \text{Cols. de } R + \text{Cols. de } S$
- Número de tuplas de  $R \times S = \text{Tuplas de } R * \text{Tuplas de } S$
- Para evitar ambigüedades: Usar sintaxis *tabla.columna*
- El producto cartesiano de dos tablas  $R$  y  $S$  es el conjunto de combinar todas las filas de  $R$  con todas las filas de  $S$ , y sus atributos corresponden a los de  $R$  seguidos por los de  $S$ .
- No es necesario que  $R$  y  $S$  sean compatibles
- En ocasiones, que de una consulta resulte un producto cartesiano se debe a no especificar las relaciones entre las tablas de forma adecuada



# La operación producto cartesiano(x)



- El **producto cartesiano** de dos conjuntos **A x B** es el conjunto de todos los **pares ordenados** que se pueden formar con un elemento perteneciente al conjunto A y un elemento del conjunto B.
- Los elementos de  $A \times B$  son pares ordenados. Cada par que se forma con un elemento del conjunto A y uno del conjunto B, **en ese orden** y recibe el nombre de par ordenado. Sus elementos se colocan entre paréntesis, separados por coma.

# La operación producto cartesiano(x)

R	A	B
	a	1
	a	3

S	D	E	F
	a	3	1
	b	4	2
	c	5	4

$R \times S$	A	B	D	E	F
	a	1	a	3	1
	a	1	b	4	2
	a	1	c	5	4
	a	3	a	3	1
	a	3	b	4	2
	a	3	c	5	4

- El producto cartesiano es una relación que contiene todas las tuplas que resultan de combinar cada tupla de R con cada tupla de S

# La operación producto cartesiano(x)

JUGA\_ASI

COD	Nombre	Apellido_1	Apellido_2	Cod_EQUIPO
123232	Pedro	Climon	Cereza	ASI2T
665543	Elena	Raiz	Serrano	ASI2T
827654	Gloria	Tierra	Pelaez	ASI1M

EQUIPOS\_NUEVOS

Cod_EQUIPO	CICLO	CURSO	GRUPO	TURNO
ASI2T	ASI	2	NULL	TARDE
GS1	GARANTIA SOCIAL	NULL	1	NULL

PRODUCTO = JUGA\_ASI X EQUIPOS\_NUEVOS

COD	Nombre	Apellido_1	Apellido_2	Juega_ASI	Cod_EQUIPO	Equipos_Nuevos.Cod_EQUIPO	CICLO	CURSO	GRUPO	TURNO
123232	Pedro	Climon	Cereza	ASI2T		ASI2T	ASI	2	NULL	TARDE
123232	Pedro	Climon	Cereza	ASI2T		GS1	GARANTIA SOCIAL	NULL	1	NULL
665543	Elena	Raiz	Serrano	ASI2T		ASI2T	ASI	2	NULL	TARDE
665543	Elena	Raiz	Serrano	ASI2T		GS1	GARANTIA SOCIAL	NULL	1	NULL
827654	Gloria	Tierra	Pelaez	ASI1M		ASI2T	ASI	2	NULL	TARDE
827654	Gloria	Tierra	Pelaez	ASI1M		GS1	GARANTIA SOCIAL	NULL	1	NULL

# La operación producto cartesiano(x)

nombreSuc	ciudadSuc	Activo
Castellana	Madrid	90000
Ganivet	Granada	21000
Paseo	Almería	17000
Zapillo	Almería	4000
Ronda	Almería	80000
Aduana	Roquetas	3000
Los Pinos	Huercal	37000
Sol	Madrid	71000

Sucursales

nombreEmp	dniEmp	telefono	NombreSuc
García	10	101010	Castellana
Torres	11	111111	Castellana
López	12	121212	Paseo
Villegas	13	131313	Paseo
Fernández	14	141414	Zapillo
Urrutia	15	151515	Sol

Empleados

Selección de las tuplas relacionadas:

$\sigma_{empleados.nombreSuc = sucursales.nombreSuc}(sucursales \times empleados)$

Empleados. nombreEmp	Empleados. dniEmp	Empleados. telefono	Empleados. nombreSuc	Sucursales. nombreSuc	Sucursales. ciudadSuc	Sucursales. activo
García	10	101010	Castellana	Castellana	Madrid	90000
Torres	11	111111	Castellana	Castellana	Madrid	90000
López	12	121212	Paseo	Castellana	Madrid	90000
Villegas	13	131313	Paseo	Castellana	Madrid	90000
Fernández	14	141414	Zapillo	Castellana	Madrid	90000
Urrutia	15	151515	Sol	Castellana	Madrid	90000
García	10	101010	Castellana	Ganivet	Granada	21000
Torres	11	111111	Castellana	Zapillo	Almería	4000
López	12	121212	Paseo	Zapillo	Almería	4000
Villegas	13	131313	Paseo	Zapillo	Almería	4000
Fernández	14	141414	Zapillo	Zapillo	Almería	4000
Villegas	13	131313	Paseo	Sol	Madrid	71000
Fernández	14	141414	Zapillo	Sol	Madrid	71000
Urrutia	15	151515	Sol	Sol	Madrid	71000

# La operación producto cartesiano(x)

nombreSuc	ciudadSuc	Activo
Castellana	Madrid	90000
Ganivet	Granada	21000
Paseo	Almería	17000
Zapillo	Almería	4000
Ronda	Almería	80000
Aduana	Roquetas	3000
Los Pinos	Huercal	37000
Sol	Madrid	71000

Sucursales

nombreEmp	dniEmp	telefono	NombreSuc
García	10	101010	Castellana
Torres	11	111111	Castellana
López	12	121212	Paseo
Villegas	13	131313	Paseo
Fernández	14	141414	Zapillo
Urrutia	15	151515	Sol

Empleados

**Ejemplo: Nombres de empleados que trabajan en la ciudad de Madrid:**

$\Pi_{empleados.nombreEmp} (\sigma_{sucursales.nombreSuc = empleados.nombreSuc} (\sigma_{sucursales.ciudadSuc = "Madrid"} (sucursales \times empleados)))$

Empleados.nombreEmp
García
Torres
Urrutia

# OPERACIONES DERIVADAS

- La **operación unión natural** en el álgebra relacional es la que permite reconstruir las tablas originales previas al proceso de normalización. Consiste en combinar las proyección, selección y producto cartesiano en una sola operación, donde la condición es la igualdad Clave Primaria = Clave Externa (o Foránea), y la proyección elimina la columna duplicada (clave externa).
- Expresada en las operaciones básicas, queda:

# Combinación o *join*

- **Combinación o *join*.** *Con esta operación se obtiene el producto cartesiano de dos tablas cuyas filas cumplan una condición determinada. La combinación de dos tablas con respecto a una cierta condición de combinación es otra tabla constituida por pares de filas de ambas tablas concatenadas tales que en cada par las correspondientes filas satisfacen la condición. Se representa del siguiente modo: (Tabla1\*Tabla2)condición.*
- **Ejemplo:** Combinación de las tablas VENTAS y ARTICULOS La combinación de ambas tablas da lugar a una nueva tabla en la que las filas de cada venta contienen los datos del artículo correspondiente. El criterio de combinación es la igualdad en el código de artículo de ambas tablas. Los códigos de artículo están representados por las columnas CODI y CODIGO.

# Combinación o *join*

VENTAS			ARTICULOS			
Codi	Fecha	Cantidad	Código	Denominación	Existencias	PVP
5100	18/11/05	100	5100	Patatas	500	3,2
5200	19/11/05	120	5200	Cebollas	250	4,1
5100	19/11/05	45				

(Cuentas x Artículos) <small>CODI=CÓDIGO</small>					
Codi	Fecha	Cantidad	Denominación	Existencias	PVP
5100	18/11/05	100	Patatas	500	3,2
5200	19/11/05	120	Cebollas	25	4,1
5100	19/11/05	45	Patatas	500	3,2

- La combinación ( $\Theta$ ) es un producto cartesiano seguido de una restricción. La combinación natural ( $*$ ) es un producto cartesiano seguido de una restricción por igualdad y proyección.



# OPERACIONES DERIVADAS

- **JOIN: Unión Natural (Natural Join)**
- El resultado del join es una relación que puede tener atributos de ambas tablas y se obtiene combinando las tuplas de ambas que tengan el mismo valor en uno o varios atributos comunes (esa igualdad entre campos debe especificarse en una condición). Si no se especifica la igualdad entre uno o más campos se produce un producto cartesiano.
- Normalmente la operación de join se realiza entre los atributos comunes de dos tablas que corresponden a la clave primaria de una tabla y la clave foránea correspondiente de la otra tabla.
- También es muy común un join con una clave de una tabla intermedia que me permita llegar a la información de otra tabla.
- Se expresa como la combinación de una proyección y una selección en la que la condición principal es la igualdad entre los campos comunes.

# JOIN

JUGADORES

COD	Nombre	Apellido_1	Apellido_2	Cod_EQUIPO
123232	Pedro	Climon	Cereza	ASI2T
665543	Elena	Raiz	Serrano	ASI2T
827654	Gloria	Tierra	Pelaez	ASI1M
889272	Luis	Garcia	Prat	DAI2T
827654	Ernesto	Villa	Grande	DAI2M
827655	Eva	Tierra	Pelaez	DAI1M
789009	Carlos	Arias	Arias	DAI2M

EQUIPOS

EQUIPO	CICLO	CURSO	GRUPO	TURNO
ASI1M	ASI	1	NULL	MAÑANA
ASI2M	ASI	2	NULL	MAÑANA
ASI1T	ASI	1	NULL	TARDE
ASI2T	ASI	2	NULL	TARDE
GS1	GARANTIA SOCIAL	NULL	1	NULL

Mostrar con un join los jugadores del turno de tarde  
(nombre, apellido1, apellido2, equipo)

# JOIN

- $\pi$  (jugadores.nombre, jugadores.apellido\_1, jugadores.apellido\_2, equipo.equipo)  $\sigma$  (jugadores.cod\_equipo = equipo.equipo) AND (equipo.turno='t') (Jugadores, equipos);

Nombre	Apellido_1	Apellido_2	Cod_EQUIPO
Pedro	Climon	Cereza	ASI2T
Elena	Raiz	Serrano	ASI2T

# JOIN: Unión Natural

I

R1			R2	
E#	Nombre	D#	D#	Descrip
320	José	D1	D1	Central
322	Rosa	D3	D3	I+D
*	Maria	D3	D4	Ventas
*	José	D5		

**R1 ⋈ R2**

E#	Nombre	D#	Descrip
320	José	D1	Central
322	Rosa	D3	I+D
*	Maria	D3	I+D

## combinación natural ( natural join)

- Es una combinación que no indica condición alguna y que automáticamente obtiene las tuplas combinadas cuyos atributos comunes a ambas tablas sean del mismo valor. Es decir, si la tabla R1 y la tabla R2 tienen en común el atributo D#:
- $R1 \bowtie R2$  Obtiene los datos combinados de R1 y R2 cuyos D# coincidan.
- $R \bowtie S = \sigma_{R.A1=S.A1 \wedge R.A2=S.A2 \wedge \dots \wedge R.An=S.An}(R \times S)$
- En ese esquema los atributos A1, A2,...An son atributos comunes a ambas relaciones.

# La operación intersección de conjuntos

- Incluye tuplas que están en todas las relaciones.
- $R \cap S$  incluye las tuplas que están en  $R$  y en  $S$
- **Ejemplo: Clientes que también son empleados:**

nombreCli	dniCli	Domicilio
Aranda	1	La Reina nº7
García	2	Fragata azul nº8
Hayes	3	Gibraltar español nº14
Turner	4	Gibraltar español nº17
Vilches	5	Diamante S/N
Lara	6	Gato negro nº13
Guerrero	7	Perro nº1

Clientes

nombreEmp	dniEmp	telefono	NombreSuc
García	10	101010	Castellana
Torres	11	111111	Castellana
López	12	121212	Paseo
Villegas	13	131313	Paseo
Fernández	14	141414	Zapillo
Urrutia	15	151515	Sol

Empleados

$\Pi \text{ nombreCli}(\text{clientes}) \cap \Pi \text{ nombreEmp}(\text{empleados})$

nombreCli
García

# La operación intersección de conjuntos

- La intersección de dos tablas, R y S da como resultado el conjunto de todas las filas que se encuentran en las dos, tanto en la relación R como en S. (las comunes)
- R y S deben ser compatibles
- Equivale a la operación  $R - (R - S)$

EQUIPOS06

EQUIPO	CICLO	CURSO	GRUPO	TURNO
ASI1M	ASI	1	NULL	MAÑANA
ASI2M	ASI	2	NULL	MAÑANA
ASI1T	ASI	1	NULL	TARDE
ESO2A	ESO	2	A	NULL

EQUIPOS07

EQUIPO	CICLO	CURSO	GRUPO	TURNO
ASI1M	ASI	1	NULL	MAÑANA
ASI2M	ASI	2	NULL	MAÑANA
ASI1T	ASI	1	NULL	TARDE
ASI2T	ASI	2	NULL	TARDE
GS1	GARANTIA SOCIAL	NULL	1	NULL

# La operación intersección de conjuntos

COMUNES= EQUIPOS06  $\cap$  EQUIPOS07

EQUIPO	CICLO	CURSO	GRUPO	TURNO
ASI1M	ASI	1	NULL	MAÑANA
ASI2M	ASI	2	NULL	MAÑANA
ASI1T	ASI	1	NULL	TARDE

# DIVISION /

- **Cociente.** El cociente se realiza entre dos tablas, Tabla1 y Tabla2, con diferentes columnas que cumplen las siguientes condiciones: Tabla1 debe tener columnas de Tabla2 y el numero de columnas ha de ser mayor que el de Tabla2. Tabla2 debe tener, al menos, una fila. Se puede expresar en función de la proyección, del producto cartesiano y de la diferencia de la siguiente forma:
- **$\text{Tabla1:Tabla2} = \pi_C(\text{Tabla1}) - \pi_C(\text{Tabla2} \times \pi_C(\text{Tabla1}) - \text{Tabla1})$**
- Donde C representa el conjunto de atributos de la Tabla1 menos los atributos de la Tabla2.



# DIVISION /

COCHES

MODELO	COLOR
Seat	Rojo
Renault	Verde
Ford	Azul
Ford	Negro
Seat	Azul

COLORES

COLOR
Rojo
Azul

COCHES : COLORES

MODELO
Seat

## DIVISION :

- La división entre R y S da como resultado una relación con los campos que están en R pero no en S y que contiene las tuplas que están combinadas con todas y cada una de las tuplas de S.
- Condiciones:
  - $\text{Grado}(R) > \text{grado}(S)$
  - Conjunto atributos de S incluido en el conjunto de atributos de R

# DIVISION /

EQUIPOS07

EQUIPO	CICLO	CURSO	GRUPO	TURNO
ASI1M	ASI	1	NULL	MAÑANA
ASI2M	ASI	2	NULL	MAÑANA
ASI1T	ASI	1	NULL	TARDE
ASI2T	ASI	2	NULL	TARDE
GS1	GARANTIA SOCIAL	NULL	1	NULL

VISTA\_CICLO\_TURN0

CICLO	TURNO
ASI	MAÑANA

DIVISION = EQUIPOS07 / VISTA\_CICLO\_TURN0

EQUIPO	CURSO	GRUPO
ASI1M	1	NULL
ASI2M	2	NULL

^

# DIVISION /

- Si R y S son relaciones de grado G1 y G2 respectivamente y A es el conjunto de atributos comunes a ambas relaciones;  $R : S$  obtiene una relación de grado  $G1-G2$  en la que las tuplas resultantes son las tuplas formadas por los atributos distintos de A que poseen todos los valores posibles de A en la tabla S. Es decir, se obtienen las tuplas cuyos contenidos en los atributos comunes poseen todas las combinaciones almacenadas en S.

R	D	E	F
	a	34	uno
	a	34	dos
	c	23	uno
	d	11	dos

S	F
	uno
	dos

$R : S$	D	E
	a	34

# Ejemplo1

**EJEMPLO 1.-** Para la siguiente Base de Datos Relacional:

FEDERACION (NOMBRE#, DIRECCION, TELEFONO)

MIEMBRO (DNI#, NOMBRE\_M, TITULACION)

COMPOSICION (NOMBRE#, DNI#, CARGO, FECHA\_INICIO)

Se pide dar respuesta algebraica a las siguientes consultas:

1. Obtener el nombre de los presidentes de federación.
2. Obtener la dirección de aquellas federaciones que tienen gerente.
3. Obtener las federaciones que no tienen asesor técnico.
4. Obtener las federaciones que tienen todos los cargos.
5. Obtener las federaciones que tienen asesor técnico y psicólogo.

# Enlaces

- <http://uazuay.edu.ec/analisis/El%20modelo%20relacional.pdf>
- <http://www.ub.edu.ar/catedras/ingenieria/Datos/capitulo3/ejerciciosar.htm>
- <http://ocw.uc3m.es/ingenieria-informatica/disenio-de-bases-de-datos/teoria/Tema3%28TransformacionRelacional%29.pdf>
- <http://www.unirioja.es/cu/arjaime/Temas/02.Modelo E R.pdf>
- <http://www-oei.eui.upm.es/Asignaturas/BD/BD/docbd/tema/tema2.pdf>
- <http://mysql.conclase.net/curso/?cap=004a>
- <http://es.slideshare.net/dante1665/modelos-relacionales-de-bases-de-datos>
- <http://www.youtube.com/watch?v=rCOEp an IQ>
- <http://www.juntadeandalucia.es/averroes/iesbajoguadalquivir/inf/sghbd/PrincipiosBasesDatosRelacionales JorgeSanchez.pdf>
- <http://www.monografias.com/trabajos98/forma-normal-boyce-codd/forma-normal-boyce-codd.shtml#ixzz3tC2dhCFk>