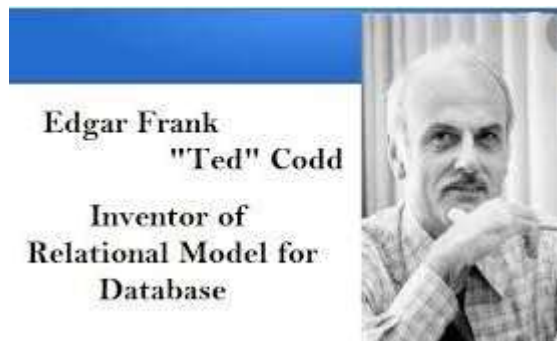


**TEMA 5: MODELO RELACIONAL**

**1- INTRODUCCIÓN**

- El origen del modelo surgió en los años 70.
- Se debe a unas publicaciones previas del señor, Edgar Frank Codd (Alias: "Ted Codd"), científico y matemático, que en revistas especializadas, cuando trabajaba para IBM.



- Codd, se basó en las relaciones de la lógica de predicados, la teoría de conjuntos y el álgebra relacional, para definir un modelo de almacenamiento de datos, el **Modelo Relacional** de datos.
- Propone un modelo de datos basado en la teoría de las relaciones, en donde los datos se estructuran lógicamente en forma de **relaciones**, las *tablas* (filas y columnas), siendo un objetivo fundamental del modelo mantener la independencia de esta estructura lógica respecto al modo de almacenamiento y a otras características de tipo físico.
- El modelo relacional, es un modelo de datos **físico**, desde el punto de vista de cómo se estructura físicamente la información (filas y columnas, en contraposición a los modelos Jerárquicos y en Red), pero se puede considerar también un modelo de datos **lógico**, porque permite representar el nivel de abstracción conceptual, es decir la globalidad de los datos y sus relaciones.

Codd definió las **reglas** que tenía que cumplir cualquier sistema de almacenamiento de la información, con un enfoque matemático, basándose en el **Álgebra Relacional**, en la práctica alguna de ellas no se implementan totalmente en los SGBD Relacionales.

**TEMA 5: MODELO RELACIONAL**

**LAS DOCE REGLAS DE CODD**

**1.- Representación de la información:** la información debe representarse explícitamente a nivel lógico, y de manera única, por medio de valores en tablas. Éste es el principio básico del modelo relacional.

**2.- Acceso garantizado:** Todo dato debe ser accesible mediante una combinación de un nombre de tabla, un valor de su clave y el nombre de una columna. Es una forma de insistir en la obligatoriedad de la clave primaria.

**3.- Tratamiento sistemático de valores nulos:** Los valores nulos, información desconocida o inaplicable, han de ser tratados sistemáticamente por el sistema, el cual ha de ofrecer las facilidades necesarias para su tratamiento.

**4.- Catálogo activo en línea basado en el modelo relacional:** La representación de los metainformación (descripción de la base de datos) debe ser igual a la de los otros datos y su acceso debe poder realizarse por medio del mismo lenguaje relacional que se utiliza para los demás datos.

**5.- Sublenguaje de datos completo:** Debe existir un lenguaje que permita un completo manejo de la BD (definición de datos, vistas, manipulación de datos, restricciones de integridad, autorizaciones y transacciones).

**6.- Actualización de vistas:** Toda vista teóricamente actualizable debe poder ser actualizada por el sistema.

**7.- Inserciones, modificaciones y eliminaciones de alto nivel:** Todas las operaciones de manipulación de datos (consulta, inserción, modificación y borrado) deben operar sobre conjuntos de filas (lenguaje no navegacional). Los sistemas existentes hasta el momento en el que surge el modelo relacional actuaban registro a registro obligando al programador de una base de datos a *navegar* por la misma.

**8.- Independencia física de los datos:** El acceso lógico a los datos debe mantenerse incluso cuando cambien los métodos de acceso o la forma de almacenamiento.

**9.- Independencia lógica de los datos:** Los programas de aplicación no deben verse afectados por cambios realizados en las tablas que estén permitidos teóricamente y que preserven la información.

**10.- Independencia de la integridad:** Las reglas de integridad deben ser definibles por medio del sublenguaje de datos relacional y habrán de almacenarse en el catálogo de la BD (metabase), no en los programas de aplicación.

**11.- Independencia de la distribución:** Debe existir un sublenguaje de datos que pueda soportar BD distribuidas sin alterar los programas de aplicación cuando se distribuyan los datos por primera vez o posteriormente.

**12.- Regla de la no subversión:** Si un SGBD soporta un lenguaje de bajo nivel que permite el acceso fila a fila, éste no puede utilizarse para saltarse las reglas de integridad expresadas por medio del lenguaje de más alto nivel.

**TEMA 5: MODELO RELACIONAL**

**EVOLUCIÓN HISTÓRICA DEL MODELO RELACIONAL**

- 1968-1970 la introducción por Codd de la teoría de las relaciones en el campo de las bases de datos (finales de los años sesenta), con estos primeros desarrollos teóricos surge el Modelo Relacional de datos.
- 1973-78 surgieron los primeros prototipos experimentales de IBM (Ingres, SystemR, SEQUEL) del modelo relacional de datos.
- 1978 surge QBE (Query By Example) de IBM, un primer lenguaje para implementar un modelo de datos relacional.
- 1979 ORACLE comercializa la primera Base de Datos Relacional.
- 1981 surge SQL (Structured Query Language), como lenguaje de consulta de una base de datos relacional. No se estandariza hasta 1986 (ANSI-American National Standard Institute) es el SQL/ANS, y en 1987 (ISO-International Organization for Standardization).
- 1982 surge el DB2 de IBM.
- 1989 SQL Addendum, mejoras del estándar de ANSI
- 1989 Primer manifiesto de los SGBOO
- 1992 SQL 92 – SQL2
- 1999 SQL:1999 – SQL3
- 2003 SQL:2003, revisión y nuevas características sobre el SQL3.
- 2006 SQL:2006, extensión para XML y XQuery.
- 2008 SQL:2008, permite el uso de la cláusula ORDER BY fuera de las definiciones de los cursores. Incluye los disparadores del tipo INSTEAD OF. Añade la sentencia TRUNCATE (ISO/IEC 9075:2008).
- 2011 SQL:2011 (ISO/IEC 9075:2011) soporte mejorado para bases de datos temporales (PERIOD FOR).
- 2016 SQL:2016 (ISO/IEC 9075:2016) Permite búsqueda de patrones, funciones de tabla polimórficas y compatibilidad con los ficheros JSON

## TEMA 5: MODELO RELACIONAL

## OBJETIVOS

El trabajo publicado por Codd (1970), presentaba un nuevo modelo de datos que perseguía una serie de objetivos, que se pueden resumir en los siguientes.

- **Independencia física:** es decir, el modo en el que se almacenan los datos no influya en su manipulación lógica y, por tanto, los usuarios que acceden a esos datos no tienen que modificar sus programas por cambios en el almacenamiento físico.
- **Independencia lógica:** esto es, que el añadir, eliminar o modificar objetos de la base de datos no repercuta en los programas y/o usuarios que están accediendo a subconjuntos parciales de los mismos (vistas).
- **Flexibilidad:** en el sentido de poder presentar a cada usuario los datos de la forma en que éste prefiera.
- **Uniformidad:** las estructuras lógicas de los datos presentan un aspecto uniforme, lo que facilita la concepción y manipulación de la base de datos por parte de los usuarios.
- **Sencillez:** las características anteriores, así como unos lenguajes de usuario muy sencillos, producen como resultado que el modelo de datos relacional sea fácil de comprender y de utilizar por parte del usuario final.

Para conseguir los objetivos citados, Codd introduce el concepto de "relación" (tabla) como una estructura básica del modelo. Todos los datos de la BD se representan en forma de relaciones cuyo contenido varía en el tiempo.

Afrontaremos ahora la componente estática de este modelo de datos, el modelo relacional. Con respecto a la parte dinámica del modelo, se proponen un conjunto de operadores que se aplican a las relaciones, todos ellos conforman el Álgebra Relacional que estudiaremos con posterioridad.

## 2- ESTÁTICA DEL MODELO

- **RELACIÓN (TABLA):** es la estructura básica del modelo relacional. Son las estructuras que almacenan la información, en forma de filas y columnas, se le denomina **relación**. Esta relación representa a una entidad, o bien a una relación, del modelo de datos Entidad/Relación, de manera general. Algunas características fundamentales de las relaciones son:
  - A cada columna de una relación se le denomina **atributo o campo**, representan propiedades de la misma y están caracterizadas por su nombre.
  - A cada fila de una relación se le denomina **tupla**, y compone una ocurrencia de la relación.
  - Al número de atributos o campos (columnas) se le denomina **grado** de la relación.
  - La **cardinalidad** de una relación es el número de tuplas que hay en una relación.

**TEMA 5: MODELO RELACIONAL**

Ejemplo de una relación (tabla):

**Relación\_Personas**

NOMBRE	NACIONALIDAD	...	ATRIBUTO N
LUISA	ESPAÑOLA	...	valor N
MARIO	ESPAÑOLA	...	valor N
...	...	...	...
Valor 1	Valor 2	Valor 3	Valor N

**Tupla 1**

**Tupla 2**

...

**Tupla M**

Matemáticamente, una **relación** se puede definir como un subconjunto del producto cartesiano de una lista de dominios, donde cada elemento de la relación, cada *tupla*, es una serie de *N* valores ordenados.

En esta definición matemática de relación, que es la que aparece en los primeros trabajos de Codd, no se alude a los **atributos** (representan las propiedades de la relación), es decir, al papel que tienen los dominios en la relación y, además, en ella el orden de los valores dentro de una tupla es significativo.

A fin de evitar estos inconvenientes, se puede dar otra definición de relación más adecuada al punto de vista de las bases de datos, para lo cual es preciso distinguir, dos conceptos en la noción de relación:

- **INTENSIÓN o Esquema de relación**, denotado **R (A1:D1, A2:D2, ..., An:Dn)** es un conjunto de *n* pares atributo-dominio subyacente (*Ai:Di*). La *intensión* es la parte definitoria y estática de la relación, que se corresponde con la cabecera cuando la relación se percibe como una tabla. Se denomina Esquema de Relación.
- **EXTENSIÓN u ocurrencia (instancia) de relación** (llamada a veces simplemente relación), denotada por *r(R)* es un conjunto de *m* tuplas {*t1, t2, ... tm*} donde cada tupla es un conjunto de *n* pares atributo-valor. Conjunto de tuplas que en un instante concreto se encuentra en una BBDD. Es la parte dinámica (varia con el tiempo).

Ejemplo:

**INTENSIÓN de una relación:**

AUTOR (NOMBRE:Nombres, NACIONALIDAD:Nacionalidades, INSTITUCION:Instituciones)

**EXTENSIÓN de una relación:**

AUTOR

Nombre	Nacionalidad	Institución
Pepe	España	O.N.U.
John	EE.UU.	O.M.S.
Pierre	Francia	N.A.S.A.

**TEMA 5: MODELO RELACIONAL**

- **DOMINIO:** es el conjunto válido de valores que puede tener un atributo (campo) de una relación.

Tipos de dominio:

- **Generales o continuos:** Aquel dominio que viene definido entre 2 valores (un valor mínimo y uno máximo).  
Ej.: edad\_empleado (16..65)
- **Restringidos:** Aquel dominio que se establece su conjunto de valores específicos.  
Ej.: estado\_civil ('s', 'c', 'v', 'd')
- **Compuestos:** Dominio formado por varios dominios simples.  
Ej.: fecha (día/mes/año)  
día (1..12)      mes('E','F',..., 'D')      año(1900..2097)

- **CLAVE:** Atributo o conjunto de atributos o campos que permiten identificar de manera unívoca cada una de las tuplas de una relación, de una tabla.

Tipos de claves:

- **Candidata:** Es el conjunto no vacío de atributos que identifica unívoca y mínimamente cada tupla de una relación.
- **Clave Primaria o principal:** Es la clave candidata que elige el usuario para identificar las tuplas de la relación. Se dice que una clave primaria es compuesta cuando está formada por más de un atributo.
- **Clave Secundaria o alternativa:** Aquella clave candidata que no ha sido elegida como clave primaria o principal.
- **Clave Ajena:** Atributo o conjunto de atributos o campos de una relación cuyos valores coinciden con los valores de la clave principal de otra relación. La clave ajena y la correspondiente clave primaria han de estar definidas sobre los mismos dominios. La clave ajena sirve para relacionar tablas.

Características de las claves:

- Una clave debe ocupar el menor espacio posible en memoria.
- Regla de integridad de entidad: siempre debe tener un valor conocido (no puede tener valor nulo o vacío).
- No deben estar sujetas a cambios (No deben modificarse)
- Se utilizan para crear índices (Estructura de información que ayuda en los procesos de búsqueda de información).

- **INTERRELACIÓN:** Es una relación (tabla) que asocia a otras tablas. Estas se asocian a través de sus claves primarias y ajenas. La tabla que contiene la clave ajena, se le denomina "tabla hija". La tabla que contiene la clave principal, se le denomina "tabla padre".

**TEMA 5: MODELO RELACIONAL**

- **RESTRICCIONES INHERENTES:**

- **Derivadas de la definición de relación:**

- No pueden existir tuplas idénticas (obligatoriedad de la clave primaria).
    - No pueden existir atributos idénticos.
    - El orden de las tuplas y atributos es irrelevante, no significativo.
    - La tabla es **plana**, tiene 2 dimensiones, es decir, la intersección de una fila y una columna alberga un único dato o valor válido del dominio, no admitiéndose por tanto grupos repetitivos. Se dice que la tabla está normalizada, o bien que está en 1ª Forma Normal.
    - Toda la información de una columna debe pertenecer al mismo dominio, o tener el mismo tipo.
    - Toda tupla de información debe tener siempre el mismo tipo de datos.
  - Regla de **Integridad de Entidad** (ningún atributo que forme parte de la clave primaria, puede tomar un valor nulo).

- **RESTRICCIÓN DE INTEGRIDAD REFERENCIAL:**

Es la base fundamental en la integridad y control de los datos de la base de datos. Aunque puede venir controlada por el SGBD, el usuario (administrador) debe controlar el comportamiento de los datos y puede imponer su propia definición. La definición de estas restricciones el modelo de datos la reconoce y las maneja sin problemas.

La integridad referencial es una **restricción de comportamiento**, ya que viene impuesta por el mundo real y es el usuario quien la define al describir el esquema relacional; es también de **tipo implícito**, ya que se define en el esquema y el modelo la reconoce (o así ocurre en algunos productos comerciales) sin necesidad de que se programe ni de que se tenga que escribir ningún procedimiento para obligar a que se cumpla.

**Definición:** “*Si una relación R2, denominada relación que referencia, tiene un descriptor que es la clave primaria de otra relación R1, relación referenciada, se debe cumplir que todo valor de dicho descriptor debe concordar con un valor de la clave primaria de R1 o bien ser un valor nulo, vacío.*”

La definición, dice sencillamente que los valores de una clave ajena deben coincidir con los de la clave primaria asociada a ella o ser nulos. Ejemplo de relación entre tablas (1:N):

**EDITORIAL** (NOMBRE\_E, DIRECCION, CIUDAD, PAIS)  
**LIBRO** (CODIGO, TITULO, IDIOMA, ..., NOMBRE\_E)

En este ejemplo el atributo NOMBRE\_E de la relación LIBRO es clave ajena que referencia a EDITORIAL (relación referenciada), de modo que debe concordar con la clave primaria de la relación EDITORIAL o bien ser nulo, porque los libros de nuestra base de datos deberán pertenecer a una editorial existente, o si se desconoce la editorial, no se tendrá ningún valor para este atributo. Ejemplo de relación entre tablas (N:M):

**AUTOR** (NOMBRE, NACIONALIDAD, INSTITUCION, ..)  
**LIBRO** (CODIGO, TITULO, IDIOMA, EDITORIAL, ...)  
**ESCRIBE** (NOMBRE, CODIGO)

En este ejemplo la relación ESCRIBE posee dos claves ajenas: NOMBRE, que referencia a la relación AUTOR, y COD\_LIBRO, que referencia a la relación LIBRO; en este caso ninguna de las dos claves ajenas puede tomar valores nulos, ya que forman parte de la clave primaria de la relación ESCRIBE.



**TEMA 5: MODELO RELACIONAL**

- **RESTRICCIONES SEMÁNTICAS**

- **Clave primaria (PRIMARY KEY).** Permite declarar un atributo o un conjunto de atributos como *clave primaria* de una relación por lo que sus valores no se podrán repetir ni se admitirán los nulos (o valores “ausentes”).
- **Unicidad (UNIQUE).** Mediante la cual se indica que los valores de un conjunto de atributos (uno o más) no pueden repetirse en una relación. Esta restricción permite la definición de claves alternativas.
- **Obligatoriedad (NOT NULL),** de uno ó más atributos, con lo que se indica que el conjunto de atributos no admite valores nulos.
- **Integridad referencial (FOREIGN KEY).** Si una relación R2 (relación que referencia) tiene un descriptor que es una clave candidata de la relación R1 (relación referenciada), todo valor de dicho descriptor debe, bien concordar con un valor de la clave candidata referenciada de R1, bien ser nulo. El descriptor es, por tanto, una clave ajena de la relación R2. Las relaciones R1 y R2 no son necesariamente distintas. Además, cabe destacar que la clave ajena puede ser también parte (o la totalidad) de la clave primaria de R2.
- **Verificación (CHECK).** Comprueba, en toda operación de actualización, si el predicado es cierto o falso y, en el segundo caso, rechaza la operación. La restricción de verificación se define sobre un único elemento (dominio, relación) y puede o no tener nombre.
- **Aserción (ASSERTION).** Actúa de forma idéntica a la anterior, pero se diferencia de ella en que puede afectar a varios elementos (por ejemplo, a dos relaciones distintas) y su definición, por tanto, no va unida a la de un determinado elemento por lo que siempre ha de tener un nombre, ya que la aserción es un elemento más del esquema que tiene vida por sí mismo.
- **Disparador (“TRIGGER”).** Restricciones en las que el usuario pueda especificar libremente la respuesta (acción) ante una determinada condición. Así como las anteriores reglas de integridad son declarativas, los disparadores son procedimentales, siendo preciso que el usuario escriba el procedimiento que ha de aplicarse en caso de que se cumpla la condición.

**Valor nulo (NULL):** *señal utilizada para representar información desconocida, inaplicable, inexistente, no válida, no proporcionada, indefinida, etc.*

Necesidad de los *valores nulos* en BD:

- Crear tuplas (filas) con ciertos **atributos desconocidos** en ese momento, p.e. el año de edición de un libro.
- **Añadir un nuevo atributo** a una relación existente; atributo que, en el momento de añadirse, no tendría ningún valor para las tuplas de la relación.
- **Atributos inaplicables** a ciertas tuplas, por ejemplo, la editorial para un artículo (ya que un artículo no tiene editorial) o la profesión de un menor.



**TEMA 5: MODELO RELACIONAL**

- **ACCIONES REFERENCIALES (OPCIONES DE BORRADO Y MODIFICACIÓN)**

Además de definir las claves ajenas, hay que determinar las consecuencias que pueden tener ciertas operaciones (borrado y modificación) realizadas sobre tuplas de la relación referenciada. Se pueden distinguir varias opciones de acciones referenciales diferentes que se ejecutarán en estos casos, los modos de operación son:

- **Operación restringida (RESTRICT):** esto es, el borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada; sólo se permite si no existen tuplas con dicha clave en la relación que contiene la clave ajena.

En el ejemplo anterior nos llevaría, a que para poder borrar una editorial de nuestra base de datos no tendría que haber ningún libro que estuviese publicado por dicha editorial, en caso contrario el sistema impediría el borrado y daría un error.

- **Operación en cascada (CASCADE):** esto es, el borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo el borrado o modificación en cascada de las tuplas de la relación que contienen la clave ajena.

En el ejemplo anterior, equivaldría a decir que al modificar el nombre de una editorial en la relación EDITORIAL, se tendría que modificar también dicho nombre en todos los libros de nuestra base de datos publicados por dicha editorial.

- **Operación con puesta a nulos (SET NULL):** esto es, el borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo poner a nulos los valores de las claves ajenas de la relación que referencia.

En el ejemplo anterior, esto nos llevaría a que, cuando se borra una editorial, a los libros que ha publicado dicha editorial y que se encuentran en la relación LIBROS se les coloque el atributo NOMBRE\_E a nulos (NULL). Esta opción, obviamente, sólo es posible cuando el atributo que es clave ajena admite el valor nulo.

- **Operación con puesta a valor por defecto (SET DEFAULT):** esto es, el borrado o la modificación de tuplas de la relación que contiene la clave primaria referenciada lleva consigo poner el valor por defecto a la clave ajena de la relación que referencia.

- **Operación que desencadena un procedimiento de usuario:** en este caso, el borrado o la modificación de tuplas de la tabla referenciada pone en marcha un procedimiento definido por el usuario.

**TEMA 5: MODELO RELACIONAL**

- **VISTA:** Existe otro tipo de tablas, denominadas tablas virtuales o vistas, que se definen sobre una o más tablas base. Las vistas son “ventanas” sobre tablas reales, de las que sólo se almacena su definición, y no tienen, por tanto, representación directa en el almacenamiento; equivalen al esquema externo de la arquitectura ANSI.

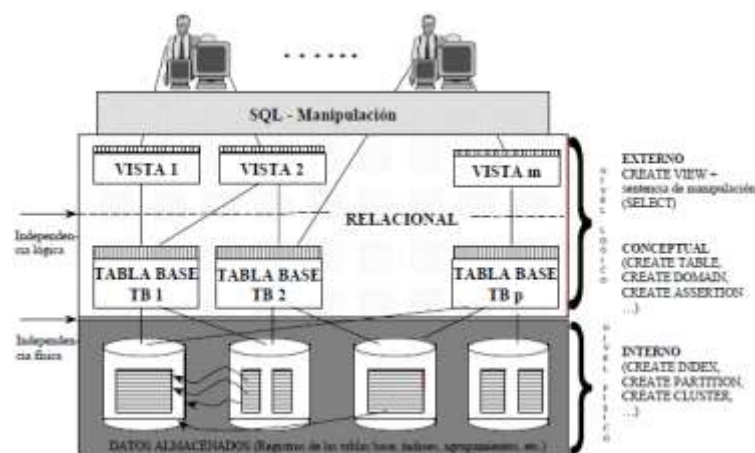
Una VISTA por lo tanto es una relación ficticia en la cual las tuplas que contiene, se obtienen de una o más relaciones base. Es una selección de información de filas y columnas de una o varias tablas. Las vistas son “los datos que se quieren visualizar para un determinado usuario”.

Características de las vistas:

- El usuario no puede distinguir si la información a la que accede es de una vista o de una relación (tabla).
- Los datos físicamente no tienen existencia propia en la vista, sino que existen en una o varias tablas reales, es decir en otra/s relación/es.
- Una vista puede estar basada en otra vista.

Ventajas:

- Nos permite clasificar las distintas visiones de los datos, según el usuario, es decir definir el nivel externo.
- El nivel de seguridad aumenta, dado que se puede decidir, con las vistas, que datos permitimos consultar a los usuarios y cuáles no.
- Las modificaciones de la vista pueden o no afectar a los datos reales de la tabla en la que está basada una vista y viceversa.



- **COMPARACIÓN DE LA TERMINOLOGÍA RELACIÓN, TABLA y FICHERO**

RELACIÓN	~	TABLA	~	FICHERO
TUPLA		FILA		REGISTRO
ATRIBUTO		COLUMNA		CAMPO
GRADO		Nº DE COLUMNAS		Nº DE CAMPOS
CARDINALIDAD		Nº DE FILAS		Nº DE REGISTROS

## **Reglas de transformación de modelo E/R al modelo Relacional**

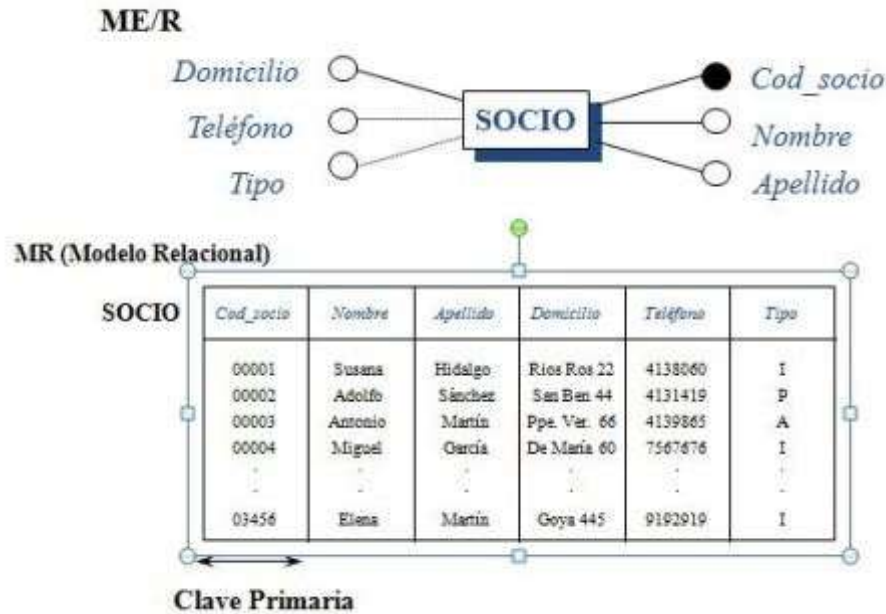
Las siguientes reglas de transformación permiten obtener un Esquema relacional a partir de cualquier modelo gráfico Entidad/Relación.

1. **Transformación de entidades:** cada tipo de entidad se transforma en una relación.
2. **Transformación de atributos:** todo atributo se convierte en una columna de una relación.
  - Identificador principal (AIP): se transforma en la clave primaria de la relación.
  - Identificadores candidatos (AIC): se transforman en claves alternativas en el modelo relacional.
  - Atributos obligatorios: se transforman en una columna de la relación en la cual se ha transformado la entidad, no admitiendo valores nulos.
  - Atributos opcionales: se transforman en una columna de la relación en la cual se ha transformado la entidad, admitiendo valores nulos.
  - Atributos compuestos: se transforma en los atributos simples (campos) que componen el atributo compuesto, desapareciendo este como tal de la relación.
  - Atributos derivados: su cálculo de actualización se realizará posteriormente mediante código, triggers, etc...

**Transformación del dominio del atributo:** si el atributo estaba definido sobre un dominio, en el modelo relacional queda también definido sobre el mismo dominio. Se realizará posteriormente mediante código, restricciones, etc...

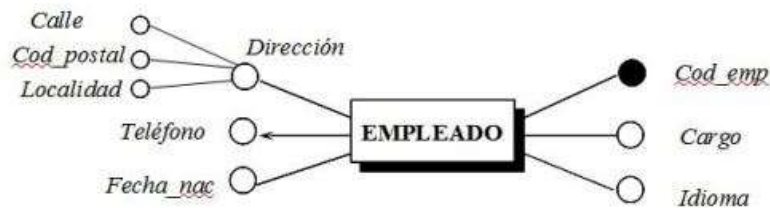
**TEMA 5: MODELO RELACIONAL**

*Ejemplo de transformación de una entidad*



***SOCIO* (*Cod\_socio*, *Nombre*, *Apellido*, *Domicilio*, *Teléfono*, *Tipo*).**

*Ejemplo de transformación de una entidad con atributos opcionales, compuestos y multivaluados.*



***EMPLEADO* (*Cod\_emp*, *Calle*, *Cod\_postal*, *Localidad*, *Cargo*, *Idioma*, *Fecha\_nac*)**

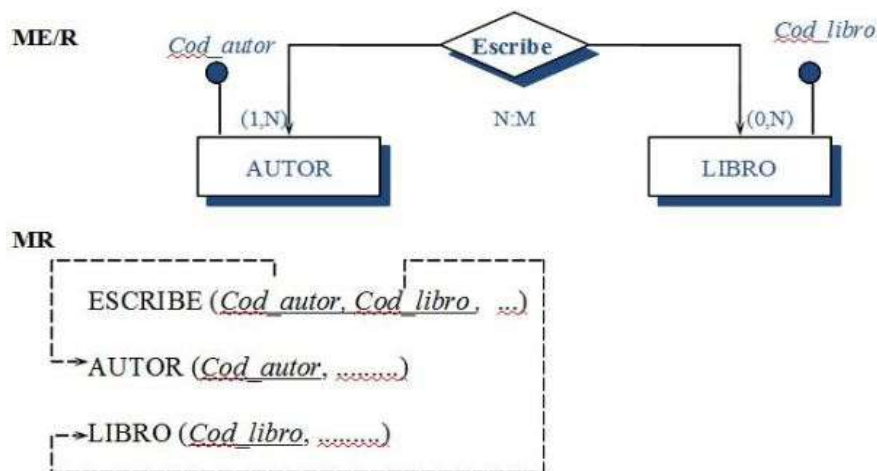
***TELEFONOS* (*Cod\_emp*, *Teléfono*)**

admite valores nulos

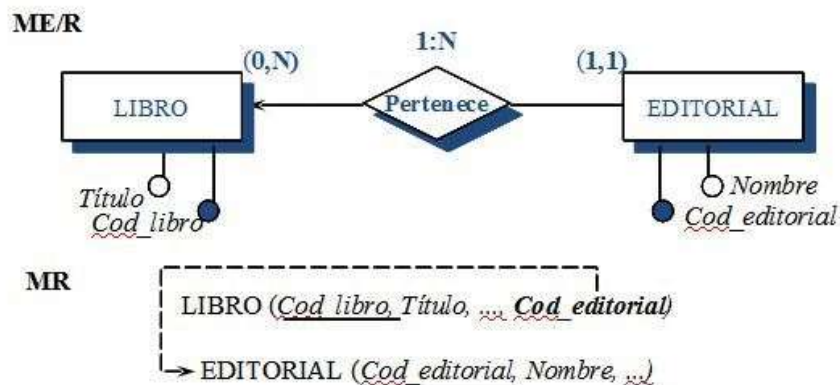
**TEMA 5: MODELO RELACIONAL**

3. **Transformación relaciones:** toda relación se transformará al Modelo Relacional, según sea su tipo de correspondencia.

A. **Cuando el tipo de correspondencia es N:M (varios a varios), se genera una nueva relación (tabla).** Se transforma en una relación que tendrá como clave primaria la concatenación de los AIP's de los tipos de entidad que asocia. Normalmente, los atributos que forman la clave primaria de esta relación serán clave ajena (de manera individual), respecto a cada una de las relaciones donde ese mismo atributo es clave primaria.



B. **Cuando el tipo de correspondencia es 1:N, se aplica el mecanismo de propagación de clave (M.P.C.),** que consiste en generar una nueva columna, en la relación (tabla) que representa la entidad con cardinalidad máxima N, normalmente esta nueva columna (que se suele convertir en clave ajena), tendrá el mismo nombre y dominio que el AIP (propagado) de la entidad con cardinalidad máxima 1 de la relación 1:N que se está transformando. Sobre esta nueva columna habrá que realizar siempre lo que se llama el "Control de Cardinalidad Mínima" (C.C.M.), e indicar si admitiría valores nulos o no (SI NULOS/NO NULOS).

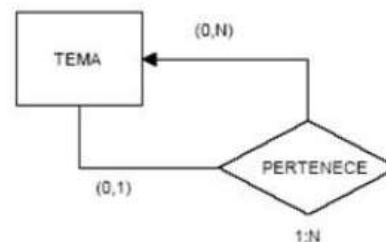


**TEMA 5: MODELO RELACIONAL**

**Excepciones** a esta regla de transformación de las relaciones con tipo de correspondencia **1:N**, se genera una nueva relación (tabla) cuando:

- El número de ocurrencias que se interrelacionan de la entidad que propaga su clave es muy pequeño, lo que supondría la existencia de una columna con muchos valores nulos.

*En el Modelo ER:*



*Modelo Relacional MR:*

*solución 1.-*

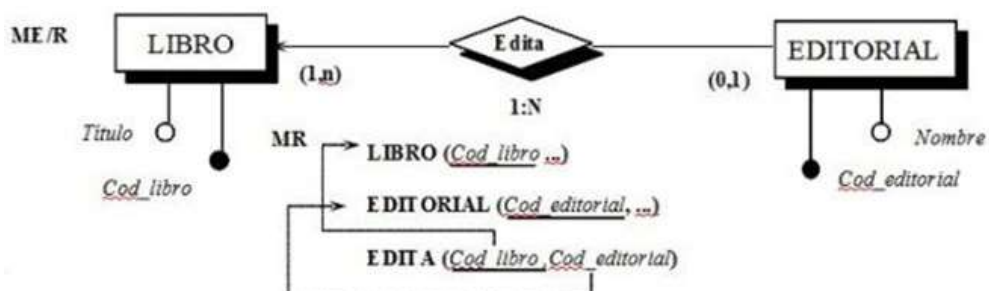
TEMA (Cod\_tema, ....., Cod\_tema\_sup)

*solución 2.-*

PERTENECE (Cod\_tema, cod\_tema\_sup)

TEMA (Cod\_tema)

- En un futuro se prevea que esa relación se pueda convertir en una con un tipo de correspondencia N:M.



- Habiendo *muchos* atributos propios en la relación, se quiere evitar la acumulación de atributos que pertenecen a ambas entidades, no solo a una de ellas.

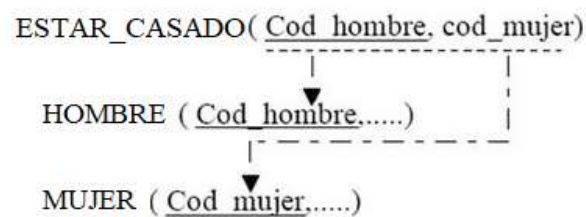
**TEMA 5: MODELO RELACIONAL**

C. **Cuando el tipo de correspondencia es 1:1:** el método de transformación depende de las dependencias de las cardinales mínimas de ambas entidades asociadas en dicha relación.

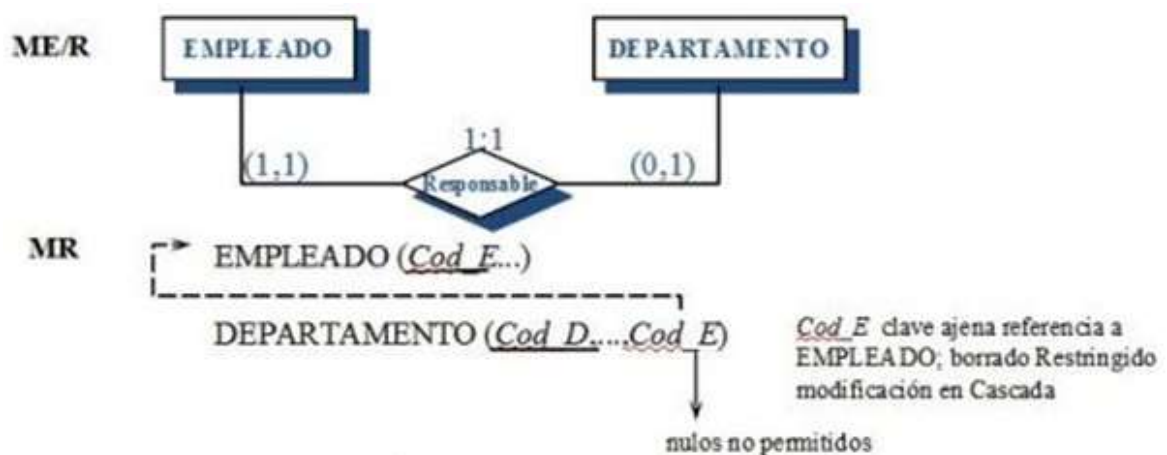
- (0,1) – (0,1): Se genera tabla. ¡Ojo! La clave de la relación (tabla), ¡Sólo uno de los atributos!



*Modelo relacional:*



- (0,1) – (1,1): Se aplica el M.P.C., se pasa la clave de la tabla que tiene como cardinalidad mínima 1 hacia la que tiene Cardinalidad mínima 0.



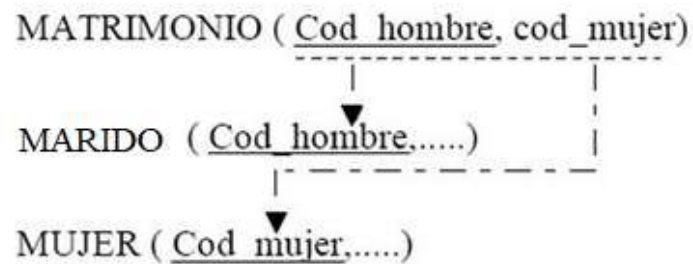


**TEMA 5: MODELO RELACIONAL**

- (1,1) – (1,1): De manera general, se debe aplicar el M.P.C., se pasa la clave a cualquiera de las 2 entidades, pero no a ambas. Excepcionalmente, y si hay atributos en la relación, puede ser conveniente generar una tabla, como en el siguiente ejemplo.



*Modelo relacional:*



**TEMA 5: MODELO RELACIONAL**

D. **Relaciones n-arias** (Aquellas en las que participan más de dos entidades).

Si la relación es:

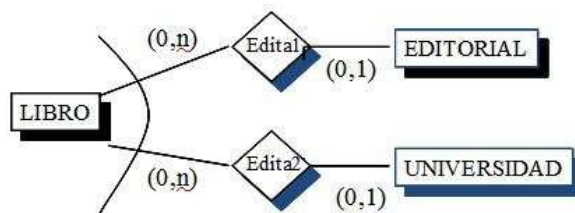
- a) **N:M:O**, se debe generar una tabla, todas las entidades participan con una cardinalidad de muchos, la clave de la tabla que queda como resultado es la unión de las claves de las entidades que la relaciona.
- b) **1:N:M**, se debe generar una tabla, como hay una de las entidades que participa con 1, la clave primaria será la combinación de las claves de las dos entidades que participan con N o M.
- c) **1:1:1 ó 1:1:N**, se aplica el mecanismo de propagación de clave (MPC) de las claves principales hacia la tabla que representa la entidad con cardinalidad máxima de varios (N). Es decir, se propaga la clave principal de las entidades que participan con 1 en la que participa con N, siempre que la participación sea obligatoria (cardinalidad mínima=1). Algunos autores recomiendan siempre crear una nueva tabla.

4. **Transformación de restricciones:** Mediante observaciones o código

5. **Transformación de relaciones débiles en existencia:** Se pasa la clave de la entidad fuerte a la entidad débil

6. **Transformación de relaciones débiles en identificación:** Se pasa la clave de la entidad fuerte a la entidad débil, siendo esta clave en la entidad débil.

7. **Transformación de relaciones exclusivas:** Se tratan según su correspondencia. Además hay que definir mediante código como controlar dicha exclusividad.



**EDITORIAL** (Cod E.....)

**UNIVERSIDAD** (Cod U.....)

**LIBRO** (Cod L....., Cod E, Cod U)

**TEMA 5: MODELO RELACIONAL**

**8. Transformación de jerarquías:**

- A. Consiste en transformar la relación ES\_UN en **una única tabla** donde se mezclan los atributos del supertipo y los de los subtipos, además se debe recoger un campo más, que represente el atributo discriminante (tipo de la relación) que nos indica como es la la jerarquía respecto a la UNIÓN de sup tipos (indicando si admite -Parcial-, o no admite -Total-, valores nulos).

Se debe aplicar este método de transformación cuando, los atributos de los subtipos son similares y las relaciones de los subtipos son idénticas.

- Ventaja: Disminuye el tiempo de acceso.

- B. Se genera una tabla para el supertipo y otra para cada uno de los subtipos, en la tabla del supertipo se debe incluir un atributo tipo para representar la unión.

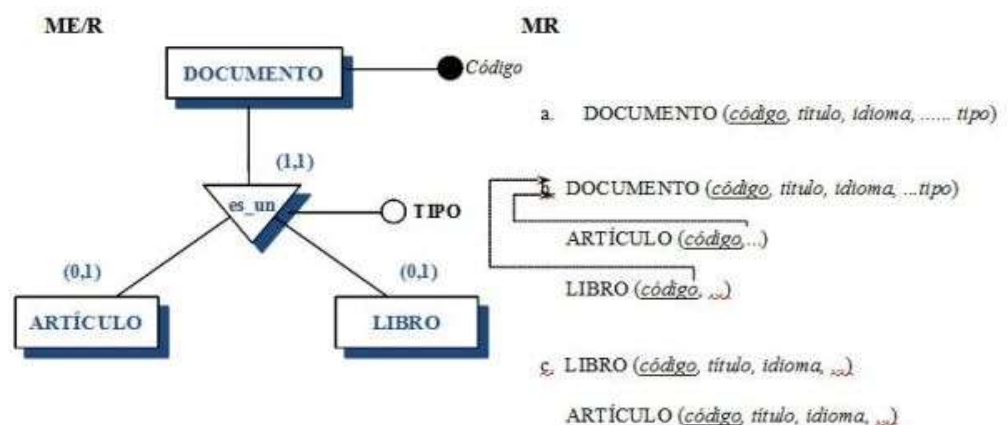
Se utiliza cuando los subtipos tienen muchos atributos diferentes y cuando las relaciones con otras entidades son distintas.

- Ventaja: Aumenta la capacidad/organización de la información.

- C. Se genera una tabla para cada uno de los subtipos que contendrá atributos comunes y propios del subtipo, y no se genera tabla para el supertipo.

Se utiliza cuando interese tener los atributos comunes y no comunes en cada subtipo.

**Transformación de Tipos y Subtipos**



**TEMA 5: MODELO RELACIONAL**

**VENTAJAS E INCONVENIENTES DE ESTOS TRES MÉTODOS:**

- **Eficiencia-Velocidad de acceso a la información:** Desde este punto de vista la mejor opción es la A (único acceso) y la peor la B. La C también es rápida, pero no dispone del atributo discriminante.
- **Contenido semántico de la información-Cantidad de información:** La mejor es la B (información precisa y no mezclada) y la peor es la A.

**9. Transformación de la dimensión temporal:**

- Si es una entidad genera tabla.
- Si es un atributo genera una columna en la tabla que represente la entidad/relación donde aparezca. Hay que tener en cuenta que puede influir en la formación de la clave de la relación.

AMPLIAR CON: <https://slideplayer.es/slide/3393696/>