

□El objetivo del **diseño lógico** es obtener una representación que use del modo más eficiente posible, los recursos que el modelo de SGBD posee para estructurar los datos y para modelar las restricciones.

#### **Objetivos:**

- >Conocer los aspectos fundamentales del modelo relacional.
- »Saber cómo convertir el modelo conceptual al lógico relacional.
- >Entender y practicar la normalización.

requisitos / enunciado -> DER + Diccionario DATOS-> modelo relacional +normalizar entidades y atributos aplicando 3 FNs -> crear BBDD con SQL



□Mientras que el objetivo fundamental del diseño conceptual es la expresividad de los esquemas conceptuales locales, el objetivo del diseño lógico es **obtener una representación** que use del modo más eficiente posible, los recursos que el modelo de SGBD posee para estructurar los datos y para modelar las restricciones.

□En esta fase, se elige qué modelo de Bases de datos voy a usar para representar el almacenamiento de mi información y en este caso nos centramos en el modelo Relacional, por ser uno de los más representativos del almacenamiento en las empresas.

□En esta fase pensamos sólo en el modelo, no en el Gestor de Base de datos concreto que vamos a utilizar. Este aspecto se deriva en la Fase de Modelo Físico.

□El modelo relacional carece de ciertos rasgos de abstracción que se usan en los modelos conceptuales. Por lo tanto, un primer paso en la fase del diseño lógico consistirá en la conversión de esos mecanismos de representación de alto nivel en términos de las estructuras de bajo nivel disponibles en el modelo relacional.



#### Modelo relacional:

□Los elementos en torno a los cuales se acumula información en el modelo relacional se denominan Relaciones o Tablas y en el modelo relacional siempre lo llamaremos Tabla.

□Una tabla es la forma en que un usuario ve sus datos. Una tabla se divide horizontalmente en filas y verticalmente en columnas. Una fila representa un registro, o un grupo de valores de columnas que contienen toda la información necesaria sobre la entidad de la tabla. Una columna contiene información referente a un único campo o atributo.

□Propiedades generales de las tablas son:

- »Son homogéneas por columna.
- >Todas las filas son diferentes. Implica la existencia de una columna o conjunto de columnas que identifican unívocamente a cada una de las filas de la tabla.
- El orden de las filas y columnas no es significativo.
- »Son independientes del almacenamiento físico de los datos.
- >En las siguientes figuras se muestran ejemplos de tablas que contienen información sobre empleados y departamentos.



# UF-2 MODELO LÓGICO Modelo relacional:

COLUMNAS O ATRIBUTOS CON SU DOMINIO DE EXISTENCIA Y SU GRADO QUE ES EL NÚMERO DE ATRIBUTOS DE LA TABLA

.TUPLAS, REGISTROS, ROWS, FILAS .CARDINALIDAD= NÚMERO FILAS

TABLA DE DEPARTAMENTOS (DEPT)

deptno	dname	loc NEW YORK		
10	ACCOUNTING			
20	RESEARCH	DALLAS		
30	SALES CHICAGO			
40	OPERATIONS	BOSTON		

## TABLA DE EMPLEADOS (EMP)

empno	ename	job	mgr	hiredate	sal	comm	deptno
7839	KING	PRESIDENT		17/11/1981	5000		10
7698	BLAKE	MANAGER	7839	01/05/1981	2850		30
7782	CLARK	MANAGER	7839	09/06/1981	2450		10
7566	JONES	MANAGER	7839	02/04/1981	2975		20
7654	MARTIN	SALESMAN	7698	28/09/1981	1250	1400	30
7499	ALLEN	SALESMAN	7698	20/02/1981	1600	300	30
7844	TÜRNER	SALESMAN	7698	08/09/1981	1500	0	30



- Los elementos en torno a los cuales se acumula información en el modelo relacional se denominan Relaciones o Tablas. Y en el modelo relacional siempre lo llamaremos Tabla.
- •Una tabla es la forma en que un usuario ve sus datos. Una tabla se divide horizontalmente en filas y verticalmente en columnas. Una fila representa un registro, o un grupo de valores de columnas que contienen toda la información necesaria sobre la entidad de la tabla. Una columna contiene información referente a un único campo o atributo.
- Propiedades generales de las tablas son:
- »Son homogéneas por columna.
- >Todas las filas son diferentes. Implica la existencia de una columna o conjunto de columnas que identifican unívocamente a cada una de las filas de la tabla.
- >El orden de las filas y columnas no es significativo.
- »Son independientes del almacenamiento físico de los datos.
- >En la siguientes figuras se muestran ejemplos de tablas que contienen información sobre empleados y departamentos.



#### Lenguajes relacionales:

- »Algebraicos: las operaciones se aplican sobre operandos (relaciones) y el resultado es otra relación=> Álgebra Relacional
- »Predicativos (orientados a tuplas o a dominios): se define el estado sin indicar las operaciones=>Cálculo Relacional.

## Álgebra Relacional:

Operadores Primitivos + Operadores Derivados

#### Operadores primitivos:

# Proyección (π) Selección (σ) Unión (∪) Diferencia (·) Producto Cartesiano (x) O. Unarios O. Binario

## Operadores derivados:

- Combinación o Join (θ)
- Intersección (\(\cappa\))
- División (:)



- □Metodología de diseño lógico: La metodología que se va a seguir para convertir el modelo conceptual al lógico relacional consta de las siguientes fases:
- •Transformar las entidades y/o relaciones en Tablas en función de los grados y cardinalidades de las interrelaciones.
- Asignar los atributos permanentes a las tablas.
- •Expandir las claves.
- •Tratamiento de opcionalidad.
- •Repasar las relaciones de integridad.
- •Normalizar las estructuras de datos.



## MODELO ENTIDAD RELACIÓN

□Grado: llamaremos grado al tamaño de una tabla en base a su número de atributos (columnas); mientras mayor sea el grado, mayor será su complejidad para trabajar con ella.

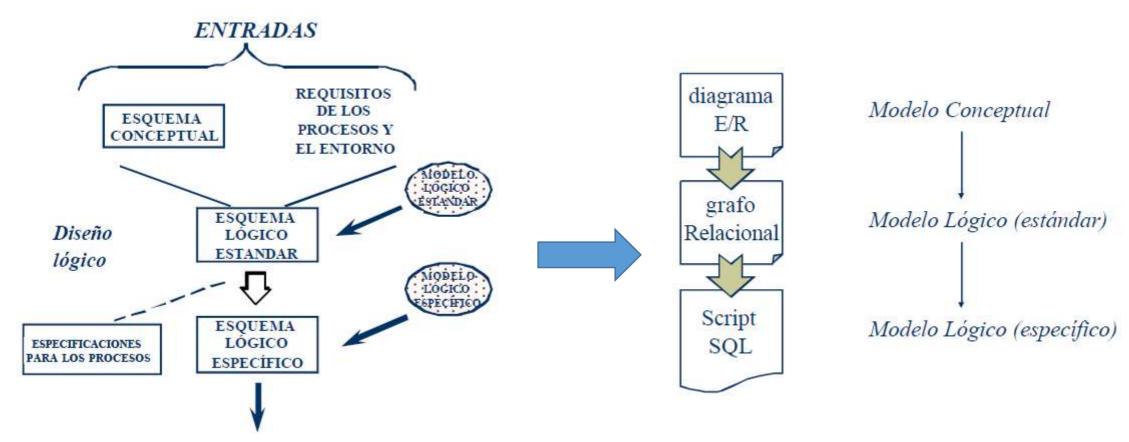
□Cardinalidad: número de tuplas o filas de una relación o tabla.

□Modelo E/R: fue ideado por Peter Chen en los años 1976 y 1977 a través de dos artículos. Se trata de un modelo que sirve para crear esquemas conceptuales de bases de datos. De hecho es prácticamente un estándar para crear esta tarea. 
Se le llama modelo E/R e incluso El (Entidad / Interrelación). Sus siglas más populares son las E/R por que sirven para el inglés y el español. Inicialmente (en la propuesta de Chen) sólo se incluían los conceptos de entidad, relación y atributos. Después se añadieron otras propuestas (atributos compuestos, generalizaciones,...) que forman el llamado modelo entidad relación extendido (se conoce con las siglas ERE)

□Entidad: Se trata de cualquier objeto u elemento (real o abstracto) acerca del cual se pueda almacenar información en la base de datos. Ejemplos de entidades son Pedro, la factura número 32456, el coche matrícula 3452BCW. Una entidad no es un propiedad concreta sino un objeto que puede poseer múltiples propiedades (atributos).



## MODELO ENTIDAD RELACIÓN





## MODELO ENTIDAD RELACIÓN

□El Nivel lógico o conceptual describe la estructura completa de la base de datos a través de lo que llamamos Esquema Conceptual, que se encarga de representar la información de una manera totalmente independiente del Sistema Gestor de Base de Datos. Cuando hemos de desarrollar una base de datos se distinguen claramente dos fases de trabajo: Análisis y Diseño. En la siguiente tabla te describimos las etapas que forman parte de cada fase.

Pasos de las fases de Análisis y de Dsiseño				
Fase de Análisis	Fase de Diseño			
Análisis de entidades: Se trata de localizar y definir las entidades y sus atributos.	Diseño de tablas.			
Análisis de relaciones: Se definirán las relaciones existentes entre entidades.	Normalización.			
Obtención del Esquema Conceptual a través del modelo E-R.	Aplicación de retrodiseño, si fuese necesario.			
Fusión de vistas: Se reúnen en un único esquema todos los esquemas existentes en función de las diferentes vistas de cada perfil de usuario.	Diseño de transacciones: localización del conjunto de operaciones o transacciones que operarán sobre el esquema conceptual.			
Aplicación del enfoque de datos relacional.	Diseño de sendas de acceso: se formalizan los métodos de acceso dentro de la estructura de datos.			

□Para la realización de esquemas que ofrezcan una visión global de los datos, Peter Chen en 1976 y 1977 presenta dos artículos en los que se describe el **modelo Entidad/Relación** (entity/relationship). Con el paso del tiempo, este modelo ha sufrido modificaciones y mejoras. Actualmente, el modelo **entidad/relación extendido (ERE)** es el más aceptado, aunque **existen variaciones** que hacen que este modelo no sea totalmente un estándar

## MODELO ENTIDAD RELACIÓN

□Una **entidad** puede ser un objeto físico, un concepto o cualquier elemento que queramos modelar, que tenga importancia para la organización y del que se desee guardar información. Cada entidad debe poseer alguna característica, o conjunto de ellas, que lo haga único frente al resto de objetos. Por ejemplo, podemos establecer una entidad llamada ALUMNO que tendrá una serie de características. El alumnado podría ser distinguido mediante su número de identificación escolar (NIE), por ejemplo. **Entidad**: objeto real o abstracto, con características diferenciadoras capaces de hacerse distinguir de otros objetos.

□En el modelo Entidad/Relación, la representación gráfica de las entidades se realiza mediante el nombre de la entidad encerrado dentro de un rectángulo. A continuación se muestra la representación de la entidad CLIENTE.





## MODELO ENTIDAD RELACIÓN

#### □Tipos de entidades:

>Entidades Fuertes o Regulares: son aquellas que tienen existencia por sí mismas, es decir, su existencia no depende de la existencia de otras entidades. Por ejemplo, en una base de datos hospitalaria, la existencia de instancias concretas de la entidad DOCTOR no depende de la existencia de instancias u objetos de la entidad PACIENTE. En el modelo E/R las entidades fuertes se representan como hemos indicado anteriormente, con el nombre de la entidad encerrado dentro de un rectángulo. >Entidades débiles: son aquellas cuya existencia depende de la existencia de otras instancias de entidad. Por ejemplo, consideremos las entidades EDIFICIO y AULA. Supongamos que puede haber aulas identificadas con la misma numeración pero en edificios diferentes. La numeración de cada aula no identificará completamente cada una de ellas. Para poder identificar completamente un aula es necesario saber también en qué edificio está localizada. Por tanto, la existencia de una instancia de una entidad débil depende de la existencia de una instancia de la entidad fuerte con la que se relaciona. La entidad Débil: Es un tipo de entidad cuyas propiedades o atributos no la identifican completamente, sino que sólo la identifican de forma parcial. Esta entidad debe participar en una relación que ayude a identificarla. En el modelo E/R una entidad débil se representa con el nombre de la entidad encerrado en un rectángulo doble. En el gráfico se muestra la representación de la entidad AULA.

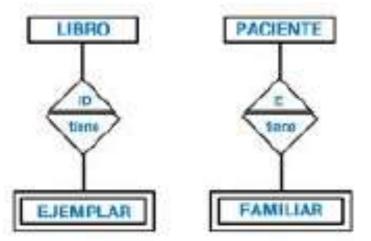




## MODELO ENTIDAD RELACIÓN

Las entidades débiles presentan dos tipos de dependencias:

- Dependencia en existencia: entre entidades, si desaparece una instancia de entidad fuerte desaparecerán las instancias de entidad débiles que dependan de la primera. La representación de este tipo de dependencia incluirá una E en el interior de la relación débil.
- »Dependencia en identificación: debe darse una dependencia en existencia y además, una ocurrencia de la entidad débil no puede identificarse por sí misma, debiendo hacerse mediante la clave de la entidad fuerte asociada. La representación de este tipo de dependencia incluirá una ID en el interior de la relación débil.





## MODELO ENTIDAD RELACIÓN

Las entidades se representan mediante un conjunto de **atributos**. Éstos describen características o propiedades que posee cada miembro de un conjunto de entidades. El mismo atributo establecido para un conjunto de entidades o, lo que es lo mismo, para un tipo de entidad, almacenará información parecida para cada ocurrencia de entidad. Pero, cada ocurrencia de entidad tendrá su propio valor para cada atributo. **Atributo**: Cada una de las propiedades o características que tiene un tipo de entidad o un tipo de relación se denomina atributo; los atributos toman valores de uno o varios dominios.

□Por tanto, un atributo se utilizará para guardar información sobre alguna característica o propiedad de una entidad o relación. Ejemplos de atributos pueden ser: altura, color, peso, DNI, fecha, etc. todo dependerá de la información que sea necesaria almacenar.





## MODELO ENTIDAD RELACIÓN

En el modelo Entidad/Relación los atributos de una entidad son representados mediante el nombre del atributo rodeado por una elipse. La elipse se conecta con la entidad mediante una línea recta. Cada atributo debe tener un nombre único que haga referencia al contenido de dicho atributo. Los nombres de los atributos se deben escribir en letra minúscula. En el gráfico se representan algunos de los atributos para la entidad PACIENTE. Al conjunto de valores permitidos para un atributo se le denomina dominio. Todos los posibles valores que puede tomar un atributo deberán estar dentro del dominio. Varios atributos pueden estar definidos dentro del mismo dominio. Por ejemplo, los atributos nombre, apellido primero y apellido segundo de la entidad PACIENTE, están definidos dentro del dominio de cadenas de caracteres de una determinada longitud.

#### □Tipos de atributos obligatorios y opcionales:

- > Atributo obligatorio: es aquél que ha de estar siempre definido para una entidad o relación. Por ejemplo, para la entidad JUGADOR será necesario tener algún atributo que identifique cada ocurrencia de entidad, podría ser su DNI. Una clave o llave es un atributo obligatorio.
- >Atributo opcional: es aquél que podría ser definido o no para la entidad. Es decir, puede haber ocurrencias de entidad para las que ese atributo no esté definido o no tenga valor.

## MODELO ENTIDAD RELACIÓN

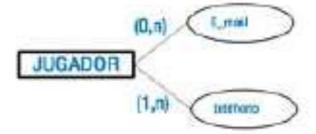
#### □Tipos de atributos simples o atómicos:

- »Atributo simple o atómico: es un atributo que no puede dividirse en otras partes o atributos, presenta un único elemento. No es posible extraer de este atributo partes más pequeñas que puedan tener significado. Un ejemplo de este tipo de atributos podría ser el atributo dni de la entidad JUGADOR del gráfico.
- »Atributo compuesto: son atributos que pueden ser divididos en subpartes, éstas constituirán otros atributos con significado propio. Por ejemplo, la dirección del jugador podría considerarse como un atributo compuesto por la calle, el número y la localidad.

#### □Tipo de atributos monovaluados o multivaluados:

- »Atributo monovaluado: es aquél que tiene un único valor para cada ocurrencia de entidad. Un ejemplo de este tipo de atributos es el dni.
- Atributo multivaluado: es aquél que puede tomar diferentes valores para cada ocurrencia de entidad. Por ejemplo, la dirección de e-mail de un empleado podría tomar varios valores para alguien que posea varias cuentas de correo.



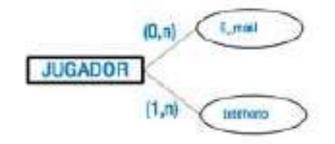


## MODELO ENTIDAD RELACIÓN

#### □Tipo de atributos monovaluados o multivaluados:

□Atributo multivaluado: En este tipo de atributos hay que tener en cuenta los siguientes conceptos:

- La cardinalidad de un atributo indica el número mínimo y el número máximo de valores que puede tomar para cada ejemplar de la entidad o relación a la que pertenece.
- La cardinalidad mínima indica la cantidad de valores del atributo que debe existir para que la entidad sea válida. Este número casi siempre es 0 o 1. Si es 0, el atributo podría no contener ningún valor y si es 1, el atributo debe tener un valor.
- ▶La cardinalidad máxima indica la cantidad máxima de valores del atributo que puede tener la entidad. Por lo general es 1 o n. Si es 1, el atributo no puede tener más que un valor, si es n, el atributo puede tener múltiples valores y no se especifica la cantidad absoluta; El atributo E\_mail de la figura, puede ser opcional y no contener ningún valor, o bien, almacenar varias cuentas de correo electrónico de un jugador. Como ves, la cardinalidad representada en la imagen es (0,n).





## MODELO ENTIDAD RELACIÓN

- □**Atributos derivados o almacenados**: el valor de este tipo de atributos puede ser obtenido del valor o valores de otros atributos relacionados. Un ejemplo clásico de atributo derivado es la edad. Si se ha almacenado en algún atributo la fecha de nacimiento, la edad es un valor calculable a partir de dicha fecha.
- □Claves entidad: Por tanto, los valores de los atributos de una entidad deben ser tales que permitan identificar unívocamente a la entidad. En otras palabras, no se permite que ningún par de entidades tengan exactamente los mismos valores de sus atributos. Teniendo en cuenta esto, presta atención a los siguientes conceptos:
- >Superclave (Superllave): Es cualquier conjunto de atributos que permite identificar de forma única a una ocurrencia de entidad. Una superclave puede tener atributos no obligatorios, es decir, que no identificarían por si solos una ocurrencia de entidad.
- >Clave candidata: Si de una superclave no es posible obtener ningún subconjunto que sea a su vez superclave, decimos que dicha superclave es clave candidata.
- >Clave primaria (Primary Key): También llamada llave primaria o clave principal. De todas las claves candidatas, el diseñador de la base de datos ha de escoger una, que se denominará clave principal o clave primaria. La clave primaria es un atributo o conjunto de ellos, que toman valores únicos y distintos para cada ocurrencia de entidad, identificándola unívocamente. No puede contener valores nulos.
- >Claves alternativas: son el resto de claves candidatas que no han sido escogidas como clave primaria.

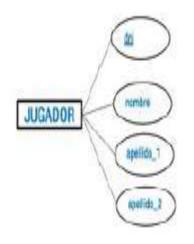
## MODELO ENTIDAD RELACIÓN

- □**Atributos derivados o almacenados**: el valor de este tipo de atributos puede ser obtenido del valor o valores de otros atributos relacionados. Un ejemplo clásico de atributo derivado es la edad. Si se ha almacenado en algún atributo la fecha de nacimiento, la edad es un valor calculable a partir de dicha fecha.
- □Claves entidad: Por tanto, los valores de los atributos de una entidad deben ser tales que permitan identificar unívocamente a la entidad. En otras palabras, no se permite que ningún par de entidades tengan exactamente los mismos valores de sus atributos. Teniendo en cuenta esto, presta atención a los siguientes conceptos:
- >Superclave (Superllave): Es cualquier conjunto de atributos que permite identificar de forma única a una ocurrencia de entidad. Una superclave puede tener atributos no obligatorios, es decir, que no identificarían por si solos una ocurrencia de entidad.
- >Clave candidata: Si de una superclave no es posible obtener ningún subconjunto que sea a su vez superclave, decimos que dicha superclave es clave candidata.
- >Clave primaria (Primary Key): También llamada llave primaria o clave principal. De todas las claves candidatas, el diseñador de la base de datos ha de escoger una, que se denominará clave principal o clave primaria. La clave primaria es un atributo o conjunto de ellos, que toman valores únicos y distintos para cada ocurrencia de entidad, identificándola unívocamente. No puede contener valores nulos.
- >Claves alternativas: son el resto de claves candidatas que no han sido escogidas como clave primaria.

## MODELO ENTIDAD RELACIÓN

□La representación en el modelo Entidad/Relación de las claves primarias puede realizarse de dos formas:

- »Si se utilizan elipses para representar los atributos, se subrayarán aquellos que formen la clave primaria.
- »Si se utilizan círculos para representar los atributos, se utilizará un círculo negro en aquellos que formen la clave primaria.





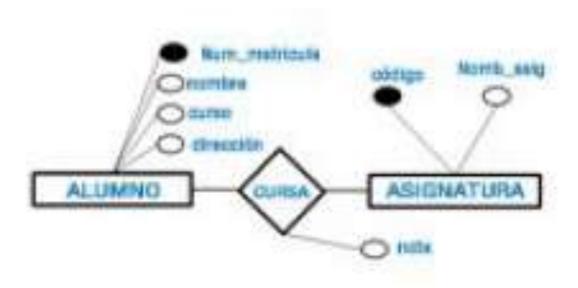


## MODELO ENTIDAD RELACIÓN

#### □Atributos de una relación.

□Una relación puede también tener atributos que la describan. Para ilustrar esta situación, observa el siguiente ejemplo: Consideremos la relación CURSA entre las entidades ALUMNO y ASIGNATURA. Podríamos asociar a la relación CURSA un atributo nota para especificar la nota que ha obtenido un alumno/a en una determinada asignatura.

□Otro ejemplo típico son las relaciones que representan **históricos**. Este tipo de relaciones suele constar de datos como fecha y hora. Cuando se emite una factura a un cliente o se le facilita un duplicado de la misma, es necesario registrar el momento en el que se ha realizado dicha acción. Para ello, habrá que crear un atributo asociado a la relación entre la entidad CLIENTE y FACTURA que se encargue de guardar la fecha de emisión.



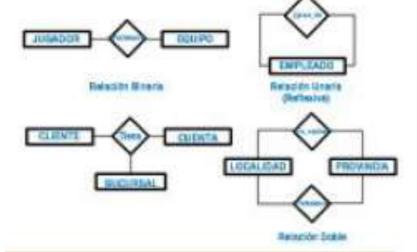


## MODELO ENTIDAD RELACIÓN

- □**Grado de una relación:** número de entidades que participan en una relación. En función del grado se pueden establecer diferentes tipos de relaciones:
- >Relación Unaria o de grado 1: Es aquella relación en la que participa una única entidad. También llamadas reflexivas o recursivas.
- »Relación Binaria o de grado 2: Es aquella relación en la que participan dos entidades. En general, tanto en una primera aproximación, como en los sucesivos refinamientos, el esquema conceptual de la base de datos buscará tener sólo este tipo de relaciones.
- >Relación Ternaria o de grado 3: Es aquella relación en la que participan tres entidades al mismo tiempo.
- >Relación N-aria o de grado n: Es aquella relación que involucra n entidades. Este tipo de relaciones no son usuales y deben ser simplificadas hacia relaciones de menor grado.

»Relación doble: ocurre cuando dos entidades están relacionadas a través de dos relaciones. Este tipo de relaciones son

complejas de manejar.



- MODELO ENTIDAD RELACION

  Cardinalidad de una relación: Es el número máximo de ocurrencias de cada entidad que pueden intervenir en una ocurrencia de relación. La cardinalidad vendrá expresada siempre para relaciones entre dos entidades. Dependiendo del número de ocurrencias de cada una de las entidades pueden existir relaciones uno a uno, uno a muchos, muchos a uno y muchos a muchos.
- >Relaciones uno a uno (1:1). Sean las entidades A y B, una instancia u ocurrencia de la entidad A se relaciona únicamente con otra instancia de la entidad B y viceversa. Por ejemplo, para cada ocurrencia de la entidad ALUMNO sólo habrá una ocurrencia relacionada de la entidad EXPEDIENTE y viceversa. O lo que es lo mismo, un alumno tiene un expediente asociado y un expediente sólo pertenece a un único alumno.
- >Relaciones uno a muchos (1:N). Sean las entidades A y B, una ocurrencia de la entidad A se relaciona con muchas ocurrencias de la entidad B y una ocurrencia de la entidad B sólo estará relacionada con una única ocurrencia de la entidad A. Por ejemplo, para cada ocurrencia de la entidad DOCENTE puede haber varias ocurrencias de la entidad ASIGNATURA y para varias ocurrencias de la entidad ASIGNATURA sólo habrá una ocurrencia relacionada de la entidad DOCENTE (si se establece que una asignatura sólo puede ser impartida por un único docente). O lo que es lo mismo, un docente puede impartir varias asignaturas y una asignatura sólo puede ser impartida por un único docente.
- >Relaciones muchos a uno (N:1). Sean las entidades A y B, una ocurrencia de la entidad A está asociada con una única ocurrencia de la entidad B y un ejemplar de la entidad B está relacionado con muchas ocurrencias de la entidad A. Por ejemplo, Un JUGADOR pertenece a un único EQUIPO y a un EQUIPO pueden pertenecer muchos jugadores. Relaciones muchos a muchos (M:N). Sean las entidades A y B, un ejemplar de la entidad A está relacionado con muchas ocurrencias de la entidad B y viceversa. Por ejemplo, un alumno puede estar matriculado en varias asignaturas y en una asignatura pueden estar matriculados varios alumnos.

## MODELO ENTIDAD RELACIÓN

□Cardinalidad de una relación: La cardinalidad de las relaciones puede representarse de varias maneras en los esquemas del modelo Entidad/Relación. A continuación, te reflejo un resumen de las notaciones clasificadas por autores, más empleadas en la representación de cardinalidad de relaciones:



# UF-2 MODELO LÓGICO Y ÁLGEBRA RELACIONAL MODELO ENTIDAD RELACIÓN

□La cardinalidad de una entidad se representa con el número mínimo y máximo de correspondencias en las que puede tomar parte cada ejemplar de dicha entidad, entre paréntesis. Su representación gráfica será, por tanto, una etiqueta del tipo (0,1), (1,1), (0,N) o (1,N). El significado del primer y segundo elemento del paréntesis corresponde a (cardinalidad mínima, cardinalidad máxima):

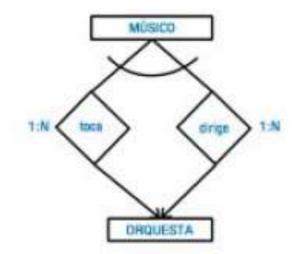
- ▶ Cardinalidad mínima. Indica el número mínimo de asociaciones en las que aparecerá cada ocurrencia de la entidad (el valor que se anota es de cero o uno, aunque tenga una cardinalidad mínima de más de uno, se indica sólo un uno). El valor 0 se pondrá cuando la participación de la entidad sea opcional.
- >Cardinalidad máxima. Indica el número máximo de relaciones en las que puede aparecer cada ocurrencia de la entidad. Puede ser uno, otro valor concreto mayor que uno (tres por ejemplo) o muchos (se representa con n).



## MODELO ENTIDAD RELACIÓN

□Restricciones en las relaciones.

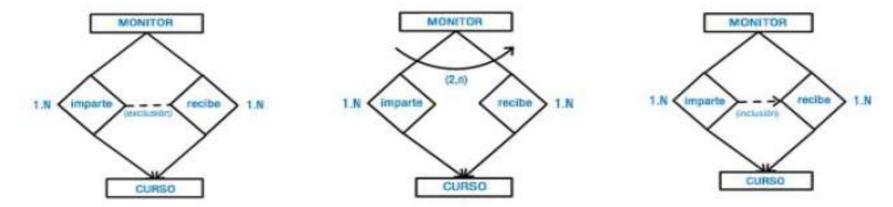
- □La primera extensión que el modelo Entidad/Relación Extendido incluye, se centra en la representación de una serie de restricciones sobre las relaciones y sus ejemplares, vamos a describirlas:
- Restricción de exclusividad: cuando existe una entidad que participa en dos o más relaciones y cada ocurrencia de dicha entidad sólo puede pertenecer a una de las relaciones únicamente, decimos que existe una restricción de exclusividad. Si la ocurrencia de entidad pertenece a una de las relaciones, no podrá formar parte de la otra. O se produce una relación o se produce otra pero nunca ambas a la vez. Por ejemplo, supongamos que un músico puede dirigir una orquesta o tocar en en ella, pero no puede hacer las dos cosas simultáneamente. Existirán por tanto, dos relaciones dirige y toca, entre las entidades MUSICO y ORQUESTA, estableciéndose una relación de exclusividad entre ellas. La representación gráfica en el modelo Entidad/Relación Extendido de una restricción de exclusividad se realiza mediante un arco que engloba a todas aquellas relaciones que son exclusivas.





# UF-2 MODELO LÓGICO Y ÁLGEBRA RELACIONAL MODELO ENTIDAD RELACIÓN

Restricción de inclusividad: Este tipo de restricciones se aplican cuando es necesario modelar situaciones en las que para que dos ocurrencias de entidad se asocien a través de una relación, tengan que haberlo estado antes a través de otra relación. Siguiendo con el ejemplo anterior, supongamos que para que un monitor pueda impartir cursos de cocina sea necesario que reciba previamente dos cursos: nutrición y primeros auxilios. Como puedes ver, es posible que los cursos que el monitor deba recibir no tengan que ser los mismos que luego pueda impartir. Aplicando una restricción de inclusividad entre las relaciones imparte y recibe, estaremos indicando que cualquier ocurrencia de la entidad MONITOR que participa en una de las relaciones (imparte) tiene que participar obligatoriamente en la otra (recibe). Se representará mediante un arco acabado en flecha, que partirá desde la relación que ha de cumplirse primero hacia la otra relación. Se indicará junto al arco la cardinalidad mínima y máxima de dicha restricción de inclusividad. En el ejemplo, (2,n) indica que un monitor ha de recibir 2 cursos antes de poder impartir varios.



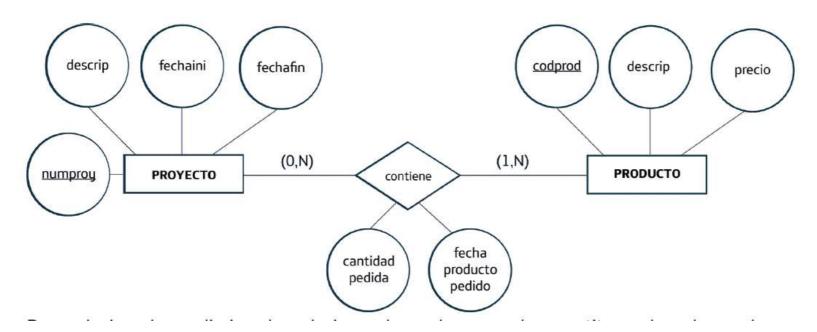


#### Transformar entidades y/o interrelaciones

Cada entidad, normalmente, se transforma en una estructura de datos, denominada **relación o tabla.** A veces cada entidad corresponde a una tabla, pero vamos a ver que no siempre esto es real.

En este paso, se eliminan o crean de cada esquema conceptual las estructuras de datos que los sistemas relacionales no modelan directamente:

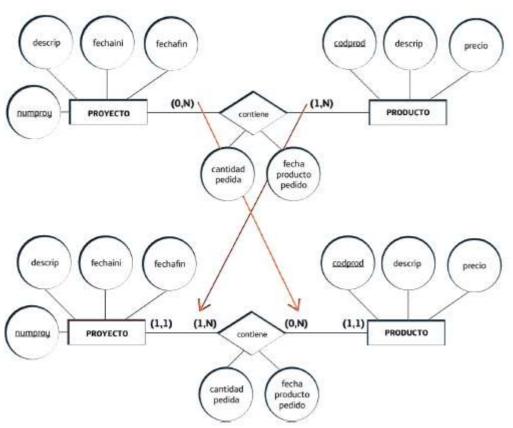
- »Revisar las relaciones uno a uno: porque es posible que se hayan identificado dos entidades que representen el mismo objeto (sinónimos) y si así fuera, ambas entidades deben integrarse en una sola.
- >Transformar las relaciones muchos a muchos: En el modelo relacional, las interrelaciones muchos a muchos no existen y además son imposibles de implementar.





#### Transformar entidades y/o interrelaciones

□Para solucionarlo, se eliminan las relaciones de muchos a muchos, sustituyendo cada una de ellas por una nueva entidad intermedia y dos relaciones de uno a muchos de esta nueva entidad con las entidades originales. La nueva entidad será débil, ya que sus ocurrencias dependen de la existencia de ocurrencias en las entidades originales.



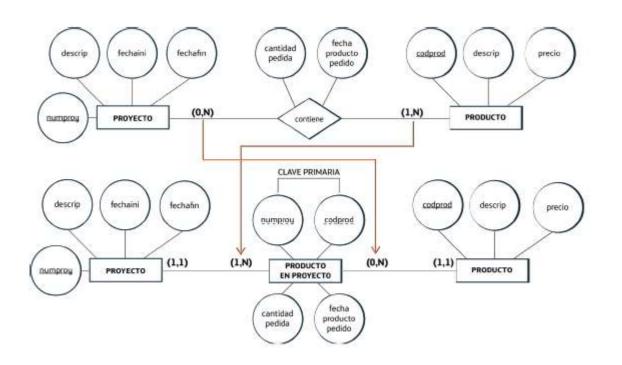
Fíjate en la figura,

□En los extremos de las tablas base (PROYECTO Y PRODUCTO) la cardinalidad siempre es 1:1.

□A ambos lados de la entidad renacida (PRODUCTO EN PROYECTO), la cardinalidad se intercambia (ver figura).

□Después de la transformación, no se puede perder las relaciones semánticas, ni se puede perder las reglas del negocio, y por supuesto los atributos que figuraban en la relación ahora estarán en la tabla renacida:

- >En la tabla Proyecto, están todos los proyectos realizados.
- >En la tabla Producto están todos los productos que tengo en la compañía.
- >En la **tabla renacida (Producto en Proyecto)**, están todos los productos que intervienen en cada proyecto.



□En los extremos de las tablas base (PROYECTO Y PRODUCTO) la cardinalidad siempre es 1:1.

□A ambos lados de la entidad renacida (PRODUCTO EN PROYECTO), la cardinalidad se intercambia (ver figura). □Después de la transformación, no se puede perder las relaciones semánticas, ni se puede perder las reglas del negocio, y por supuesto los atributos que figuraban en la relación ahora estarán en la tabla renacida:

- >En la tabla Proyecto, están todos los proyectos realizados.
- >En la tabla Producto están todos los productos que tengo en la compañía.
- >En la **tabla renacida (Producto en Proyecto)**, están todos los productos que intervienen en cada proyecto.



- **Expandir las claves:** Una vez que hemos transformado el modelo de entidades e interrelaciones, en un modelo lógico de tablas, con eliminación y creación de otras tablas con sus relaciones, debemos de expandir las claves primarias, hacia las tablas con las que tengo relación.
- Cuando una clave primaria migra a otra tabla, a este atributo migrado se le denomina clave ajena, y está tendrá los mismos valores que la clave primaria de la tabla con la que está relacionada. La expansión se lleva a cabo de una forma metodológica, y depende de las interrelaciones.

#### □Interrelaciones 1:1

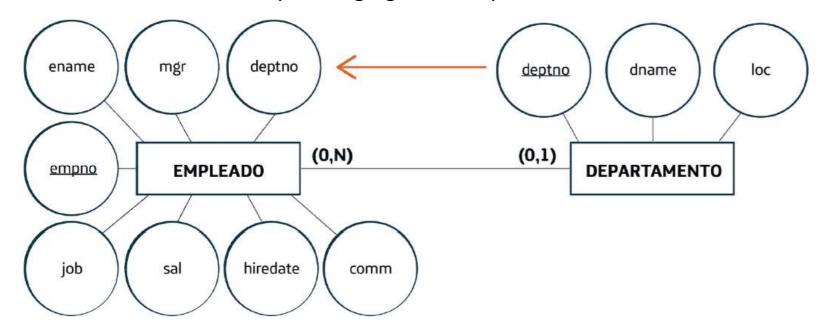
- >Si son del mismo tipo, base o renacidas, se asigna la clave de la más importante para el sistema a la menos importante.
- Si una es base y la otra renacida se asigna la clave de la base a la estructura renacida.

#### □Interrelaciones 1:N

»Siempre se agrega la clave primaria de la tabla del lado 1 a la tabla del lado N.



□Interrelaciones 1:N=> Siempre se agrega la clave primaria de la tabla del lado 1 a la tabla del lado N.



□Cuando un empleado tiene asignado departamento, este tiene que ser obligatoriamente cualquier código de departamento (deptno) de los dados de alta en la tabla de departamentos:

- La columna deptno es clave primaria, única en la tabla DEPARTAMENTO
- >La columna deptno es clave ajena en la tabla de EMPLEADOS.



□Interrelaciones 1:N=> Siempre se agrega la clave primaria de la tabla del lado 1 a la tabla del lado N.

Paso de relaciones 1 a 1 del Esquema E/R al Esquema Relacional



Se ha de tener en cuenta las cadinalidades de las entidades que intervienen. Existen dos soluciones:

Si las entidades poseen cardinalidades (0,1), la relación se convertirá en una tabla.

Si una de las entidades posee cardinalidad (0,1) y la otra (1,1), se propagará la clave de la entidad con cardinalidad (1,1) a la tabla resultante de la entidad de cardinalidad (0,1).

Si ambas entidades tienen cadinalidad (1,1) se puede propagar la clave de cualquiera de ellas a la tabla de la otra entidad.

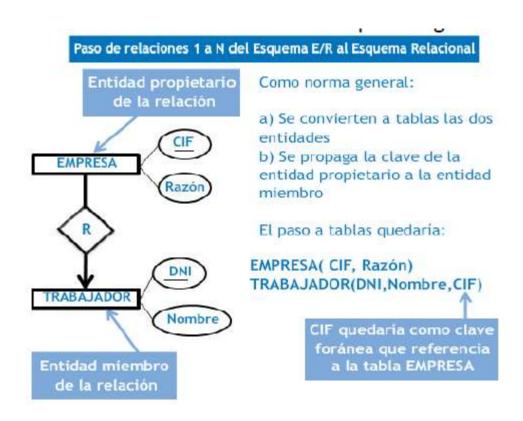
Nuestro ejemplo quedaría:

PUESTO( Cód\_Puesto, Ubicación, DNI)
TRABAJADOR(DNI, Nombre)

DNI quedaría como clave foránea que referencia a la tabla TRABAJADOR



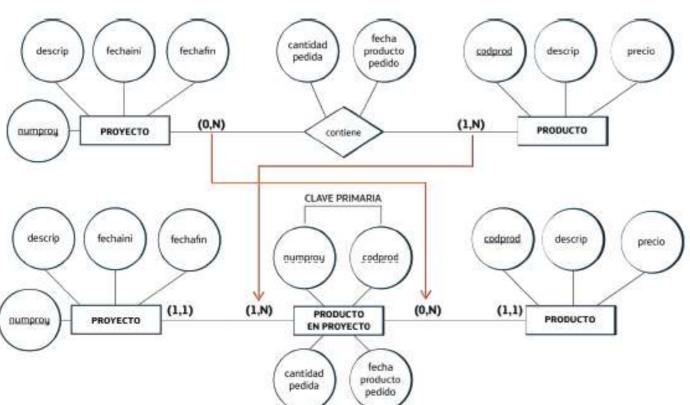
□Interrelaciones 1:N=> Siempre se agrega la clave primaria de la tabla del lado 1 a la tabla del lado N.





#### **Interrelaciones N:N**

□Si hemos realizado la transformación de relaciones N:N a dos relaciones binarias 1:N con la aparición de una renacida. No estaríamos en esta situación.



□Se migra la clave primaria (numproyecto) de la entidad PROYECTO, como clave ajena en la entidad renacida.

aSe migra la clave primaria (codproducto) de la entidad PRODUCTO, como clave ajena en la entidad renacida.

□Estas dos claves ajenas, forman la clave primaria, con las siguientes consideraciones:

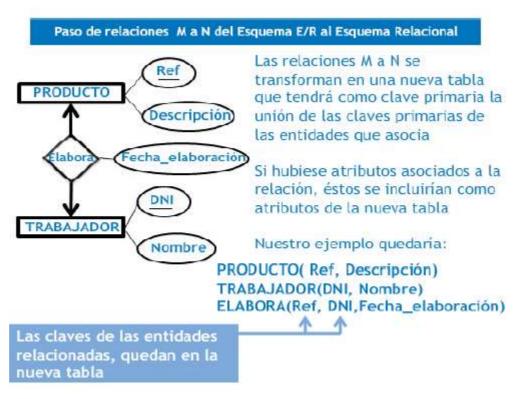
□Si la suma de las dos claves ajenas da valores únicos, éstas dos forman la clave principal.

□Si no dan valores únicos: Añadir una columna a la clave que de valor único.

□Crear una columna denominada "artefacto" (inventada) que proporcione unicidad. Por ejemplo una columna autonumérica, que queda reflejado en el diccionario de datos. Este tipo de columnas se actualizan automáticamente con valores consecutivos cada vez que insertamos un registro en la tabla.



□Las **relaciones N:M Muchos a Muchos** se transforman en una tabla que tendrá como clave primaria las claves primarias de las entidades que asocia:

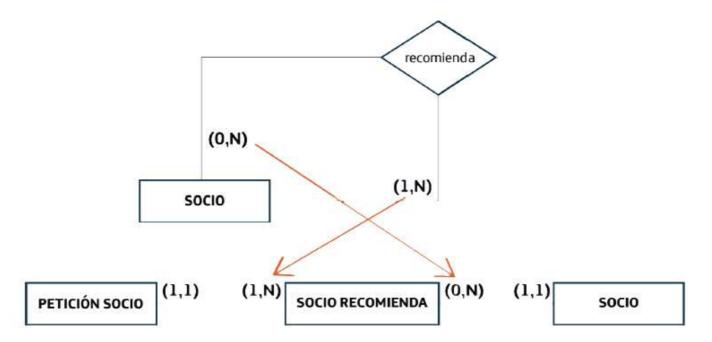




#### Eliminar las relaciones recursivas

□En las relaciones recursivas de muchos a muchos, hay que sustituir cada una de ellas por una nueva entidad (débil) y dos relaciones binarias de esta nueva entidad con la entidad original. La cardinalidad de estas relaciones dependerá de su significado.

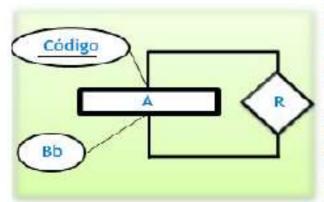
□En la figura se representa el caso en que para entrar en una entidad te tiene que recomendar uno o más socios. En el modelo conceptual tenemos una entidad con una interrelación recursiva. Al pasarlo a lógico tenemos dos tablas y una renacida.





#### Eliminar las relaciones recursivas

#### Paso de relaciones Reflexivas del Esquema E/R al Esquema Relacional



Para este tipo de relaciones hemos de tener muy en cuenta la cardinalidad. Podremos tener los siguientes casos:

a) Si la relación es de 1 a 1, la clave de la entidad se repetirá, aunque debe ser con identificadores diferentes. Un identificador actuará como clave primaria y el otro, como clave foránea de ella misma.

La clave primaria Código pasa a tener dos identificadores diferentes. Uno de ellos actuará como clave primaria y el otro como foránea Para nuestro ejemplo, el paso a tablas quedaría:

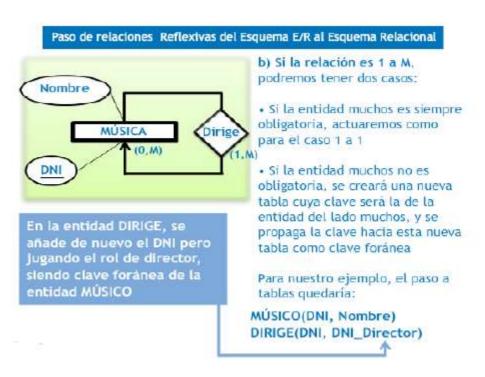
A(Código1, Código2, Bb)





#### Eliminar las relaciones recursivas

□Las relaciones reflexivas o cíclicas podrán generar una o varias tablas en función de la cardinalidad de la relación la cardinalidad de la relación si es 1:M, N:M:



#### Paso de relaciones Reflexivas del Esquema E/R al Esquema Relacional c) Si la relación es M a N, se trataría como una relación binaria Descripcion entre dos entidades. Se crearía (1.N una nueva tabla que tendrá dos veces la clave primaria de la PIEZA ompone entidad. Si hubiese atributos (1,M) N,M) asociados a la relación, se Cod asociarían a la nueva tabla. La clave de esta tabla será la combinación de ambas claves primarias. En la nueva tabla PIEZA COMPUESTA aparecerá Para nuestro ejemplo, el paso a dos veces la clave primaria de tablas quedaria: PIEZA, pero jugando dos roles. Por un lado el de clave de la PIEZA(Cod, Descripción) pieza en cuestión y, por otro, el PIEZA\_COMPUESTA(Cod,Cod\_com de la/s pieza/s que la componen ponente)



#### Eliminar las relaciones entre tres o más entidades

□Sustituir cada una de ellas por una nueva tabla (débil) intermedia que se relaciona con cada una de las tablas originales. La cardinalidad de estas nuevas relaciones binarias dependerá de su significado.

#### Eliminar las relaciones redundantes

□Una relación es redundante cuando se puede obtener la misma información que ella aporta mediante otras relaciones. El hecho de que haya dos caminos diferentes entre dos entidades no implica que uno de los caminos corresponda a una relación redundante, eso dependerá del significado de cada relación.

#### Definición de atributos

- □Para cada atributo de las entidades del modelo conceptual, que forman parte ahora de las tablas, se termina de definir:
- >Asignar a cada atributo un nombre: corto, fácil de recordar, significativo y único.
- >Especificar su tipo de dato y su dominio: si es semántico se indicará además su validación.
- »Si es un código: indicar su tipo y validación.
- >El atributo puede ser simple o compuesto y puede ser simple o multievaluado.
- »El tipo del campo es el tipo de campo de atributo al que se puede asociar el dominio.
- □El tipo de campo se puede establecer en cualquiera de los siguientes valores:
- >Corto: enteros cortos
- >Largo: enteros largos
- >Flotante: números de coma flotante de precisión simple
- »Doble: números de coma flotante de precisión doble
- Texto (solo los dominios codificados): caracteres alfanuméricos
- >Fecha: datos de fecha y hora

### **Expandir las claves**

□Una vez que hemos transformado el modelo de entidades e interrelaciones, en un modelo lógico de tablas, con eliminación y creación de otras tablas con sus relaciones, debemos de **expandir las claves primarias**, hacia las tablas con las que tengo relación.

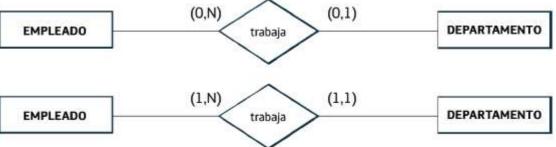
□Recordar que el modelo relacional, precisamente se caracteriza porque la relación entre dos tablas es a través de un campo (el mismo) en común.

□Cuando una clave primaria migra a otra tabla, a este atributo migrado se le denomina clave ajena, y está tendrá los mismos valores que la clave primaria de la tabla con la que está relacionada.

□La expansión se lleva a cabo de una forma metodológica, y depende de las interrelaciones.



□Tratamiento de opcionalidad



□En la imagen vemos las dos reglas de negocio de departamento frente a empleado. Ambas son de tipo 1:N, por lo tanto en ambas migraríamos el código de departamento (clave primaria) de la tabla de departamentos, a la tabla de empleados como clave ajena.

□Pero vamos a analizar la relación de la derecha en ambos casos:

- >(1,1): Un empleado pertenece a un departamento y solo a uno, es decir el valor de la clave ajena, cuando vaya a dar de alta el empleado es obligatorio, y además tiene que ser uno de los valores existentes en la tabla de departamentos.
- >(0,1): Un empleado puede que no pertenezca a ningún departamento, y como mucho pertenece a uno. Es decir, su valor es opcional. Pero esto plantea un problema: Imagina que tenemos los departamentos 10,20,30, como códigos de departamento. Cualquier empleado que pertenezca a un departamento, tendría que tener uno de estos valores. Pero que ocurre si el empleado no tiene departamento asociado. En este caso si no le diera alguno de esos tres valores nos daría un error.
- En el modelo relacional, la opcionalidad en la cardinalidad mínima de una relación, se solventó asignando a la columna que es clave ajena la característica o propiedad NULL a la definición del atributo, de esta manera el modelo no comprueba el valor de la clave ajena respecto de la primaria de la tabla con la que está relacionada porque el valor sería irrelevante.



- dadmite nulos la clave ajena?: Cada clave ajena expresa una relación. Si la participación de la entidad hijo en la relación es total, entonces la clave ajena no admite nulos; si es parcial, la clave ajena debe aceptar nulos.
- □Borrar una ocurrencia de la entidad padre que tiene algún hijo o lo que es lo mismo, ¿qué hacer cuando se quiere borrar una fila que está siendo referenciada por otra fila a través de una clave ajena? Hay varias respuestas posibles:
- >Restringir: no se pueden borrar filas que están siendo referenciadas por otras filas.
- >Propagar: se borra la fila deseada y se propaga el borrado a todas las filas que le hacen
- >referencia.
- >Anular: se borra la fila deseada y todas las referencias que tenía se ponen, automáticamente, a nulo (esta respuesta sólo es válida si la clave ajena acepta nulos).
- »Valor por defecto: se borra la fila deseada y todas las referencias toman, automáticamente, el valor por defecto (esta respuesta sólo es válida si se ha especificado un valor por defecto para la clave ajena).
- »No comprobar: se borra la fila deseada y no se hace nada para garantizar que se sigue cumpliendo la integridad referencial (Este concepto se vE en el siguiente apartado).



- □En las reglas de borrado y modificación para una clave ajena se tienen las siguientes opciones, cuando se crea la tabla:
- >RESTRICTED: modo por defecto de todos los gestores relacionales. No se permite borrar un registro maestro (padre) si tiene detalle (hijo).
- >NULLIFY: Si se borra un registro maestro y contiene detalles el valor de la clave ajena se pone a NULL.
- >CASCADE: Si se borra un registro maestro y contiene detalles se borran en cascada ambos.
- □Modificar la clave primaria: ¿Qué hacer cuando se quiere modificar la clave primaria de una fila que está siendo referenciada por otra fila a través de una clave ajena?=> Las respuestas posibles son las mismas que en el caso anterior. Cuando se escoge propagar, se actualiza la clave primaria en la fila deseada y se propaga el cambio a los valores de clave ajena que le hacían referencia.



- □**Reglas de negocio:** Cualquier operación que se realice sobre los datos debe cumplir las restricciones que impone el funcionamiento de la empresa:
- >Todas las restricciones de integridad establecidas en este paso se deben reflejar en el diccionario de datos para que puedan ser tenidas en cuenta durante la fase del diseño físico.
- »En realidad cuando se usa SQL para definir las columnas la opción por defecto es que todas las columnas admiten NULOS, por ello esta característica no se pone. Si queremos que una columna admita valores en todo momento, es decir, sea obligatorio darle valor cuando se crea la fila, entonces hay que especificar la característica NOT NULL.
- »El identificador de una entidad, no puede ser nulo, por lo tanto, las claves primarias de las relaciones (tablas) no admiten nulos.



- □**Repasar las relaciones de integridad:** En el modelo hay que repasar las siguientes cuestiones, para asegurar la integridad.
- >Integridad de entidades: Toda tabla debe tener una columna, o conjunto de columnas, denominada clave primaria, que identifiquen unívocamente a cada fila de la tabla.
- >Integridad referencial: Una clave ajena es aquella que tiene el mismo valor que la clave primaria de la tabla con la que está relacionada.
- √Si la participación de la entidad hijo en la relación es **total**, entonces la clave ajena no admite nulos; **si es parcial**, la clave ajena debe aceptar nulos.
- Restricciones de borrado de la clave ajena, cuando se borra la clave primaria de referencia.
- >Integridad del dominio de la información
- ✓ Datos requeridos. Algunos atributos deben contener valores en todo momento, es decir, no admiten nulos.
- Restricciones de dominios. Todos los atributos tienen un dominio asociado, que es el conjunto de los valores que cada atributo puede tomar.
- √Reglas de negocio. Cualquier operación que se realice sobre los datos debe cumplir las restricciones que impone el funcionamiento de la empresa.



#### Normalización

- □**Objetivos de la normalización** En esta fase de diseño, una vez agrupados los atributos en torno a las entidades y formadas las estructuras, se procederá a la normalización de las mismas. Es un proceso fundamentalmente de comprobación.
- □El objeto de la normalización es:
- >Eliminar la información redundante, reduce la probabilidad de errores en la introducción de datos y aumenta la efectividad de las correcciones.
- >Reducir el tamaño de las bases de datos
- >Simplificar las consultas. Reducir la complejidad y simplificar las estructuras todo lo que sea posible.
- »Aunque las formas normales son abundantes y su estudio, a veces es complicado, vamos a ver las tres primeras formas normales, de una forma lo más clara posible, sin recurrir a fórmulas matemáticas que existen en otras documentaciones.



- □Normalización: Proceso que consiste en imponer a las tablas del modelo Relacional una serie de restricciones a través de un conjunto de transformaciones consecutivas. Este proceso garantizará que las tablas contienen los atributos necesarios y suficientes para describir la realidad de la entidad que representan, permitiendo separar aquellos atributos que por su contenido podrían generar la creación de otra tabla.
- □Formas normales(FN): conjunto de restricciones sobre tablas que evitan problemas de redundancia y anomalías de modificación, inserción y borrado de datos
- □A principios de la década de los setenta, concretamente en 1972, Codd establece una técnica para llevar a cabo el diseño de la estructura lógica de los datos representados a través del modelo relacional, a la que denominó **normalización**. Pero esta técnica no ha de utilizarse para el diseño de la base de datos, sino como un proceso de refinamiento que debe aplicarse después de lo que conocemos como "paso a tablas", o lo que formalmente se denomina traducción del esquema conceptual al esquema lógico. Este proceso de refinamiento conseguirá los siguientes objetivos:
- »Suprimir dependencias erróneas entre atributos.
- >Optimizar los procesos de inserción, modificación y borrado en la base de datos.

El proceso de normalización se basa en el análisis de las dependencias entre atributos. Para ello tendrá en cuenta los conceptos de: dependencia funcional, dependencia funcional completa y dependencia transitiva.



### Normalización de los modelos relacionales

- □Una **dependencia funcional** de la forma X->Y, que se lee "X determina Y", quiere decir que el valor de X determina unívocamente el valor de Y: un profesor sólo puede dar clase de una asignatura profesor -> asignatura.
- □Clasificación de dependencias funcionales:
- >Una dependencia funcional X -> Y es trivial si Y es un subconjunto de X.

**Ejemplo1**COD\_LIBRO -> COD\_LIBRO

Ejemplo2
ARTICULO , REVISTA -> REVISTA

Ejemplo3 ARTICULO -> REVISTA ⊀

>Una dependencia funcional X -> Y es elemental si Y es un atributo único y la dependencia funcional no es trivial:

```
Ejemplo1
PROFESOR -> AREA ✓

Ejemplo2
PROFESOR -> AREA , DEPARTAMENTO ×
```



▶Una **dependencia funcional X -> Y** es **transitiva** cuando un atributo tiene dependencia de otro y éste a su vez de un tercero. Si se tiene los elementos A,B,C y si A es funcionalmente dependiente de B, y B es funcionalmente dependiente de C, entonces A es transitivamente dependiente de C:

#### Ejemplo1

PROFESOR -> AREA
AREA -> DEPARTAMENTO

Entonces se obtiene la dependencia funcional transitiva:

PROFESOR -> DEPARTAMENTO ✓

>Una dependencia funcional X -> Y es completa cuando un atributo tiene una dependencia funcional de un conjunto de atributos, pero ninguno de ellos en particular :

$$R(T,L) = \{T = (A,B,C,D) , L = \{A -> B,C,D\}\} \checkmark$$

$$R(T,L) = \{T = (A,B,C,D) , L = \{(A -> B,C,D) , (A -> B\}\} \checkmark$$



□La normalización es un proceso que se realiza en varias etapas secuenciales. Cada etapa está asociada a una forma normal, que establece unos requisitos a cumplir por la tabla sobre la que se aplica. Existen varias formas normales: Primera, Segunda, Tercera, FNBC, Cuarta, Quinta.

□El paso de una forma normal a otra es consecutivo, si no se satisface una determinada forma normal no puede pasarse al análisis de la siguiente. Según vamos avanzando en la normalización, los requisitos a cumplir serán cada vez más restrictivos, lo que hará que nuestro esquema relacional sea cada vez más robusto.

□Como norma general, para garantizar que no existan problemas en la actualización de datos, es recomendable aplicar el proceso de normalización hasta Tercera Forma Normal o incluso hasta Forma Normal de Boyce-Codd.





#### □Una relación R se encuentra **en 1FN**:

- »Si y solo sí por cada renglón columna contiene valores atómicos.
- Las celdas de las tablas poseen valores simples y no se permiten grupos ni arreglos repetidos como valores, es decir, contienen un solo valor por cada celda.
- >Todos los ingresos en cualquier columna (atributo) deben ser del mismo tipo.
- >Cada columna debe tener un nombre único, el orden de las columnas en la tabla no es importante.
- >Dos filas o renglones de una misma tabla no deben ser idénticas, aunque el orden de las filas no es importante.

Ne Gr	focts	nom recepcionsta	cod: art	cart art	nom an
0000010	01/06/2010	Marcos-Goncales	AB410	12	pientalde
0.0000011	02/00/2010	Nee Noon	ASAII	36	polos
61/000013	01/04/2010	Samuel IIIndnes	A8412	24	Medias
KEDODOVA	04/06/2010	Affairual Pairez	A8413	36	politic

Ejemplo 1:

En esta Guía de Pedido, la PK es el Nro\_Gl (número de guía) quién determina a los demás atributos de

la tabla.



#### □Una relación R se encuentra **en 1FN**:

## Ejemplo 2:

En este caso de la biblioteca, la PK es el CodLibro, quién determina a los demás atributos de la tabla.

Contident	- France	1000	Editorial	Femme	Material	Steadown	Fechalier
ini.	Variable averable	Name has all	SteGew intt	Progr	Consus	tell (	Total State
1014	Viceal Bear 5	E. Provinciano	Anny	Pare .	less.	Auto	Shipper
juit.	Tombien	Mintery Spread	Midles Mil	Property.	Probas	Name .	TANADAY
Little	George Cornelly	Page Hollies	tivali Orp.	Circle	Ninger .	CHE	Dent Des
mir.	Clarer	Knowles	Stations (16)	Pensi.	Giner,	No.	(Allaigne)

NOMORE	(Clave) (DESTUDIANTE	PROMERO	(Clave) (D-CLASE	CALFICACION
Hearten, J.	81254	5.4	F15-1A	
Formers, A.	20046	5.1	F15-5A	44
Serioto, P.	11349	48	GUIM 201	A
Muerton, J.	01234	5.4	GRHW 362	A
Clements, C.	68349	5.9	MU5-5	10
Pirez, R.	40472	5.1	ARTE 3A	100
Ferroro, A.	22546	3.1	GUIM 1A	C
Wester, J.	81254	3.4	6605.5	in in
Yaupes, S.	33465	4.9	ARTE:3A	
Printe, 8	83472	5.1	MUST	

## Ejemplo 3:

En esta Informe de Notas, la PK esta conformada por el ID-Estudiante y el ID-Clave, quienes determinan a los demás atributos de la tabla.



□Una relación R que no se encuentra **en 1FN**:

usuarios				
nombre	empresa	direccion_empresa	url1	url2
Joe	ABC	1 Work Lane	abc.com	xyz.com
Jill	XYZ	1 Job Street	abc.com	xyz.com

### Observa los campos url1 y url2 :

- >¿Qué haremos cuando en nuestra aplicación necesitemos una tercera url?
- >¿ Quieres tener que añadir otro campo/columna a tu tabla y tener que reprogramar toda la entrada de datos de tu código PHP
- ? Obviamente no, tu quieres crear un sistema funcional que pueda crecer y adaptarse fácilmente a los nuevos requisitos.



□Una relación R que no se encuentra **en 1FN**:

usuarios				
nombre	empresa	direccion_empresa	url1	url2
Joe	ABC	1 Work Lane	abc.com	xyz.com
Jill	XYZ	1 Job Street	abc.com	xyz.com

□Volvemos a repasar las reglas del Primer Nivel de Formalización-Normalización, y las aplicaremos a nuestra tabla

- >Eliminar los grupos repetitivos de la tablas individuales.
- >Crear una tabla separada por cada grupo de datos relacionados.
- >Identificar cada grupo de datos relacionados con una clave primaria. ¿ Ves que estamos rompiendo la primera regla cuando repetimos los campos url1 y url2 ? ¿ Y que pasa con la tercera regla, la clave primaria ?
- La regla tres básicamente significa que tenemos que poner una campo tipo contador autoincrementable para cada registro.



□De otra forma, ¿ Qué pasaría si tuviéramos dos usuarios llamados Joe y queremos diferenciarlos. Una vez que aplicáramos el primer nivel de F/N nos encontraríamos con la siguiente tabla:

usuarios					
userId	nombre	empresa	direccion_empresa	url	
1	Joe	ABC	1 Work Lane	abc.com	
1	Joe	ABC	1 Work Lane	xyz.com	
2	Jill	XYZ	1 Job Street	abc.com	
2	Jill	XYZ	1 Job Street	xyz.com	

□Ahora diremos que nuestra tabla está en el primer nivel de F/N. Hemos solucionado el problema de la limitación del campo url. Pero sin embargo vemos otros problemas....Cada vez que introducimos un nuevo registro en la tabla usuarios, tenemos que duplicar el nombre de la empresa y del usuario. No sólo nuestra BD crecerá muchísimo, sino que será muy fácil que la BD se corrompa si escribimos mal alguno de los datos redundantes.



### □Segundo nivel de F/N

- »Crear tablas separadas para aquellos grupos de datos que se aplican a varios registros.
- »Relacionar estas tablas mediante una clave externa. Hemos separado el campo url en otra tabla, de forma que podemos añadir más en el futuro si tener que duplicar los demás datos. También vamos a usar nuestra clave primaria para relacionar estos campos:

usuarios					
userId	non	nbre	empresa	dir	eccion_empresa
1	Joe		ABC	1 V	Vork Lane
2	Jill		XYZ	1 J	ob Street
urls					
urlId		relUs	serId		url
1		1	1		abc.com
2	1			xyz.com	
3	2			abc.com	
4		2			xyz.com

□Vale, hemos creado tablas separadas y la clave primaria en la tabla usuarios, userId, esta relacionada ahora con la clave externa en la tabla urls, relUserId. Esto esta mejor. ¿ Pero que ocurre cuando queremos añadir otro empleado a la empresa ABC ? ¿ o 200 empleados ? Ahora tenemos el nombre de la empresa y su dirección duplicándose, otra situación que puede inducirnos a introducir errores en nuestros datos. Así que tendremos que aplicar el tercer nivel de F/N:



□Segundo nivel de F/N= 1 FN + NO EXISTEN DEPENDENCIAS FUNCIONALES PARCIALES

**Dependencia Funcional.** Una relación está en 2FN si está en 1FN y si los atributos que no forman parte de ninguna clave dependen de forma completa de la clave principal. Es decir que no existen dependencias parciales. (Todos los atributos que no son clave principal deben depender únicamente de la clave principal); en otras palabras podríamos decir que la segunda forma normal está basada en el concepto de dependencia completamente funcional. Una dependencia funcional es completamente funcional si al el $x \to y$ r los atributos A de X significa que la dependencia no es mantenida, esto es que  $A \in X, X - \{A\} \nrightarrow Y$ 

Considere una tabla describiendo las habilidades de los empleados:

#### Habilidades de los empleados

Empleado	<u>Habilidad</u>	Lugar actual de trabajo
Jones	Mecanografía	114 Main Street
Jones	Taquigrafía	114 Main Street
Jones	Tallado	114 Main Street
Bravo	Limpieza ligera	73 Industrial Way
Ellis	Alquimia	73 Industrial Way
Ellis	Malabarismo	73 Industrial Way
Harrison	Limpieza ligera	73 Industrial Way



La única clave candidata de la tabla es {Empleado, Habilidad}.

### □Segundo nivel de F/N= 1 FN + NO EXISTEN DEPENDENCIAS FUNCIONALES PARCIALES

#### Considere una tabla describiendo las habilidades de los empleados:

#### Habilidades de los empleados

<b>Empleado</b>	<u>Habilidad</u>	Lugar actual de trabajo
Jones	Mecanografía	114 Main Street
Jones	Taquigrafía	114 Main Street
Jones	Tallado	114 Main Street
Bravo	Limpieza ligera	73 Industrial Way
Ellis	Alquimia	73 Industrial Way
Ellis	Malabarismo	73 Industrial Way
Harrison	Limpieza ligera	73 Industrial Way

La única clave candidata de la tabla es {Empleado, Habilidad}.

- El atributo restante, Lugar actual de trabajo, es dependiente solo en parte de la clave candidata, llamada Empleado. Por lo tanto la tabla no está en 2NF.
- Observe la redundancia de la manera en que son representadas los Lugares actuales de trabajo: nos dicen tres veces que Jones trabaja en la 114 Main Street, y dos veces que Ellis trabaja en 73 Industrial Way.
- Esta redundancia hace a la tabla vulnerable a anomalías de actualización: por ejemplo, es posible actualizar el lugar del trabajo de Jones en sus registros "Mecanografía" y "Taquigrafía" y no actualizar su registro "Tallado". Los datos resultantes implicarían respuestas contradictorias a la pregunta "¿Cuál es el lugar actual de trabajo de Jones?".



□Un alternativa 2NF a este diseño representaría la misma información en dos tablas:

Habilidades de los empleados				
Habilidad	Lugar actual de trabajo			
Mecanografía	114 Main Street			
Taquigrafia	114 Main Street			
Tallado	114 Main Street			
Limpieza ligera	73 Industrial Way			
Alquimia	73 Industrial Way			
Malabarismo	73 Industrial Way			
Limpieza ligera	73 Industrial Way			
	Habilidad Mecanografia Taquigrafia Tallado Limpieza ligera Alquimia Malabarismo			

Las anomalías de actualización no pueden ocurrir en estas tablas, las cuales están en 2NF.

### **Empleados**

<b>Empleado</b>	Lugar actual de trabajo
Jones	114 Main Street
Bravo	73 Industrial Way
Ellis	73 Industrial Way
Harrison	73 Industrial Way

#### Habilidades de los empleados

Empleado	<u>Habilidad</u>
Jones	Mecanografía
Jones	Taquigrafía
Jones	Tallado
Bravo	Limpieza ligera
Ellis	Alquimia
Ellis	Malabarismo
Harrison	Limpieza ligera



Sin embargo, no todas las tablas 2NF están libres de anomalías de actualización. Un ejemplo de una tabla 2NF que sufre de anomalías de actualización; Aunque el Ganador y la Fecha de nacimiento del ganador están determinadas por una clave completa {Torneo, Año} y no son partes de ella, particularmente las combinaciones Ganador/ Fecha de nacimiento del ganador son mostradas redundantemente en múltiples registros. Este problema es tratado por la tercera forma normal (3NF):

#### Ganadores del torneo

<u>Torneo</u>	<u>Año</u>	Ganador	Fecha de nacimiento del ganador
Des Moines Masters	1998	Chip Masterson	14 de marzo de 1977
Indiana Invitational	1998	Al Fredrickson	21 de julio de 1975
Cleveland Open	1999	Bob Albertson	28 de septiembre de 1968
Des Moines Masters	1999	Al Fredrickson	21 de julio de 1975
Indiana Invitational	1999	Chip Masterson	14 de marzo de 1977



□3NF=2 FN + La tabla se encuentra en 3FN si es 2FN y si no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

□Un ejemplo de este concepto sería que, una dependencia funcional X->Y en un esquema de relación R es una dependencia transitiva si hay un conjunto de atributos Z que no es un subconjunto de alguna clave de R, donde se mantiene X->Z y Z->Y.

□Un ejemplo de una tabla 2NF, donde la única clave candidata es {Torneo, Año} y donde se falla en satisfacer los requerimientos de la 3NF :

Ganadores	del	torneo
-----------	-----	--------

<u>Torneo</u>	<u>Año</u>	Ganador	Fecha de nacimiento del ganador
Indiana Invitational	1998	Al Fredrickson	21 de julio de 1975
Cleveland Open 1999		Bob Albertson	28 de septiembre de 1968
Des Moines Masters	1999	Al Fredrickson	21 de julio de 1975
Indiana Invitational	1999	Chip Masterson	14 de marzo de 1977

□La violación de la 3NF ocurre porque el atributo no primario Fecha de nacimiento del ganador es dependiente transitivamente de {Torneo, Año} vía el atributo no primario Ganador.

□El hecho de que la Fecha de nacimiento del ganador es funcionalmente dependiente en el Ganador hace la tabla vulnerable a inconsistencias lógicas, pues no hay nada que impida a la misma persona ser mostrada con diferentes fechas de nacimiento en diversos registros.

□Para expresar los mismos hechos sin violar la 3NF, es necesario dividir la tabla en dos; las anomalías de actualización no pueden ocurrir en estas tablas, las cuales están en 3NF.:

#### Ganadores del torneo

<u>Torneo</u>	<u>Año</u>	Ganador	Fecha de nacimiento del ganador
Des Moines Masters	1998	Chip Masterson	14 de marzo de 1977
Indiana Invitational	1998	Al Fredrickson	21 de julio de 1975
Cleveland Open	1999	Bob Albertson	28 de septiembre de 1968
Des Moines Masters	1999	Al Fredrickson	21 de julio de 1975
Indiana Invitational	1999	Chip Masterson	14 de marzo de 1977

#### Fecha de nacimiento del jugador

<u>Jugador</u>	Fecha de nacimiento
Chip Masterson	14 de marzo de 1977
Al Fredrickson	21 de julio de 1975
Bob Albertson	28 de septiembre de 1968

#### Ganadores del torneo

<u>Torneo</u>	<u>Año</u>	Ganador
Indiana Invitational	1998	Al Fredrickson
Cleveland Open	1999	Bob Albertson
Des Moines Masters	1999	Al Fredrickson
Indiana Invitational	1999	Chip Masterson



#### **RESUMEN Normalización**

- □Primera Forma Normal (1FN): Se dice que una entidad está en 1º FN cuando;
- >Tiene un identificador único principal
- »No existen grupos repetitivos de datos. Por ejemplo un cliente con dos o más direcciones, un empleado con dos o más directores...
- >Los grupos repetitivos de datos no los contempla el modelo relacional, ni la mayoría de los gestores de bases de datos relacionales actuales=> No existen atributos multievaluados.
- Las soluciones a este caso son:
- ▶Crear columnas: Saber cuál es el número máximo de datos que se repiten (hasta 4 teléfonos, hasta 5 directores, hasta 20 direcciones...), y crear tantas columnas del mismo dato como valor máximo tenga el dato. El problema es que se crean muchos huecos en blanco, y la fila al ser más grande, ocupa más espacio en los almacenamientos.
- >Crear tabla: Crear una tabla distinta, que tendrá como clave ajena, la clave primaria de la tabla de referencia, y ésta a su vez formará parte de la clave principal, más un artefacto, es decir una columna que marque el orden de la repetición de las columnas con distintos valores=> MEJOR OPCIÓN QUE LA ANTERIOR.



#### **RESUMEN Normalización**

- □**Segunda Forma Normal (2FN)=>**Una tabla está en segunda forma normal cuando:
- »Está en 1º FN, es decir no tiene grupo de datos repetitivos y
- >Todos los atributos que no forman parte del identificador único principal, dependen de todo el identificador único principal.
- >Esta forma normal se aplica, generalmente, a tablas renacidas. Recordamos que el campo clave de estas tablas está formado por las claves ajenas de las tablas con las que está relacionada.
- >Una vez creada la clave principal, se ve el resto de columnas y se mirará que depende de toda la clave principal.



- □Imagínate, que por error hemos puesto en esta tabla renacida la columna "Descrip producto".
- □Esta columna depende de codProducto (clave ajena), pero no depende de codProyecto (la otra clave ajena, que conforman ambas la clave principal). Por tanto pasaríamos esta columna a su tabla correspondiente.



#### **RESUMEN Normalización**

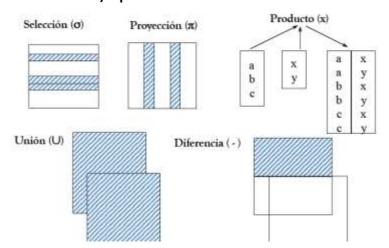
- □Forma Normal (3FN)=> Una tabla está en tercera forma normal cuando:
- ⊳Está en 2ª FN
- >Y todos los campos que no son claves, son independientes entre sí

### Ejemplo

- □Supongamos que tenemos la siguiente tabla que contiene información acerca de suministradores:
- »Código de Suministrador + Nombre + Código de Ciudad + Ciudad
- □Donde:
- »Nombre, depende de la clave (Código de Suministrador)
- >Código de Ciudad, depende de la clave.
- >Ciudad depende de Código ciudad y no de la clave.
- □Para solucionar este caso procedemos dividiendo la tabla en dos tablas:
- >Tabla Suministradores: Código de Suministrador + Nombre + Código de Ciudad
- >Tabla Ciudades: Código de Ciudad + Ciudad.



- □En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:
- >UNION: Consiste en obtener una tabla formada por el total de filas de otras "n" tablas.
- >INTERSECCIÓN: Nos permite obtener una tabla con las filas coincidentes de otras "n" tablas.
- »DIFERENCIA: De una tabla y otra es una tabla formada por todas las filas de la primera que no son coincidentes con alguna fila de la segunda.
- >SELECCIÓN: Nos permite obtener una tabla formada por una parte de las filas de una tabla. Para obtener unas filas y no otras es precisa la utilización de un criterio de selección que decida que filas son válidas y cuáles no.
- »PROYECCIÓN: Forma una nueva tabla extrayendo un subconjunto de columnas de una tabla. Esta operación también necesita algún criterio de selección capaz de distinguir entre columnas válidas y no válidas.
- »JOIN: Es una operación que nos permite combinar filas de una tabla con filas de otras tablas basándonos en los valores de una determinada columna. Se obtienen por tanto filas con columnas de una tabla y columnas de las otras.
- □En el modelo relacional **la relación** que existe entre dos tablas, se hace a través de un campo en común, es decir una columna (la misma, **deptno en DEPT y deptno en EMP**) que se encuentra en dos tablas distintas.





- □Dos relaciones son compatibles en su esquema si:
- »Si tienen el mismo grado.
- >Si se puede hacer una correspondencia de cada uno de los atributos de las dos relaciones y si estos están definidos sobre el mismo dominio.

#### Autor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan

#### **Editor**

Nombre	Nacionalidad	Institución
Chen	Norteamericana	ER Institute
Yao	Norteamericana	U.N.Y
Bertino	Italiana	U. Milan



En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

**UNION** (**U**): La unión de dos relaciones R1 y R2, compatibles en su esquema, es otra relación definida sobre el mismo esquema de relación, cuya extensión estará constituida por el conjunto de tuplas que pertenezcan a R1, a R2 o a ambas (sin duplicar).

»Se escribe con la notación Rx=R1 U R2:

#### Autor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan

#### Editor

Nombre	Nacionalidad	Institución
Chen	Norteamericana	ER Institute
Yao	Norteamericana	U.N.Y
Bertino	Italiana	U. Milan

#### **Autor** ∪ **Editor**

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan
Chen	Norteamericana	ER Institute
Yao	Norteamericana	U.N.Y



En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

□**INTERSECCIÓN** ( $\cap$ ): La intersección entre dos relaciones con la misma estructura (grado y dominio) proporciona una tabla con la misma estructura y cuyos elementos son el conjunto de los elementos comunes a ambas relaciones iniciales >Se escribe con la notación  $Rx=R1 \cap R2$ 

$$R1 \cap R2 = R1 - (R1 - R2)$$
  
 $R1 \cap R2 = R2 - (R2 - R1)$ 

■ Intersección (∩ ): La sintaxis de la intersección es:

 La intersección de dos relaciones con los mismos atributos proporcionará como resultado una relación que contendrá las tuplas presentes en ambas relaciones.

Ejemplo:	<b>R</b> : <u>A</u>	В	C	<b>S</b> : <u>A</u>	В	C	(R ∩ S): A	1	В	C
O	1	2	3	2	3	4	1	1	2	3
- 11	4	5	6	4	5	5				
	7	8	9	7	7	7				

#### **AUTOR**

NOMBRE	NACIONALIDAD	INSTITUCION
Date, C.J.	Norteamericana	Relational Inst.
Saltor, F.	Española	U.P.C.
Ceri, S.	Italiana	Politéc. Milán

#### **EDITOR**

NOMBRE	NACIONALIDAD	INSTITUCION
Chen, P.	Norteamericana	ER Institute
Yao, L.	Norteamericana	U.N.Y.
Ceri, S.	Italiana	Politéc. Milán



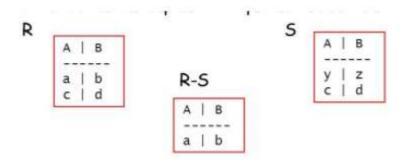
AUTOR 🤦 EDITOR

NOMBRE	NACIONALIDAD	INSTITUCION	
Ceri, S.	Italiana	Politéc. Milán	

En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

DIFERENCIA (-): La diferencia entre dos relaciones con la misma estructura ( grado y dominio) proporciona una tabla con la misma estructura y cuyos elementos son el conjunto de los elementos que están en la primera relación y no están en la segunda.

»Se escribe con la notación Rx=R1 - R2





En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

**PRODUCTO CARTESIANO (X)**: El producto cartesiano de dos relaciones R1 y R2 de cardinalidades m1 y m2 respectivamente, es una relación definida sobre la unión de los atributos de ambas relaciones y cuya extensión estará constituida por las m1 x m2 tuplas formadas concatenando cada tupla de la primera relación con cada una de las tuplas de la segunda relación:

»Se escribe con la notación Rx=R1 X R2

#### LIBRO

Código	Título	Idioma	Nombre_e
	Bases de Datos	Español	Ra-ma
003	Diseño de BD	Español	Ra-ma

#### **EDITORIAL**

ENombre	Dirección	Ciudad	País
Ra-ma	Pez, 20	Madrid	España
	24 Lennon	London	UK

#### LIBRO x EDITORIAL

Código	Título	Idioma	Nombre_e	ENombre	Dirección	Ciudad	País
001	BD	Español	Ra-ma	Ra-ma	Pez, 20	Madrid	España
001	BD	Españo1	Ra-ma	Addison- Wesley	24 Lennon	London	UK
003	Diseño de BD	Españo1	Ra-ma	Ra-ma	Pez, 20	Madrid	España
003	Diseño de BD	Españo1	Ra-ma	Addison- Wesley	24 Lennon	London	UK



En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

**División (:)**: la división de una relación R1(dividendo) por otra relación R2 (divisor) es una relación R (cociente) tal que, al realizarse su combinación con el divisor, todas las tuplas resultantes se encuentran en el dividendo. Se escribe con la notación Rx=R1: R2

R1 : R2 = 
$$\pi_{C}(R1)$$
 -  $\pi_{C}(R2 \times \pi_{C}(R1)-R1)$ 

## AUTOR\_EDITORIAL

NOMBRE	NACIONALIDAD	EDITORIAL
Date, C.J.	Norteamericana	Addison
Cervera, J.	Española	Rama
Saltor, F.	Española	Paraninfo
Ceri, S.	Italiana	Clup
Costilla, C.	Española	Diaz de Santos
Codd, E.	Norteamericana	Prentice Hall
Cervera, J.	Española	Addison

### **EDITORIAL**

EDITORIAL Addison Rama

## AUTOR\_EDITORIAL: EDITORIAL

NOMBRE	NACIONALIDAD
Cervera, J.	Española



En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

SELECCIÓN-RESTRICCIÓN (σ): Nos permite obtener una tabla formada por una parte de las filas de una tabla. Para obtener unas filas y no otras es precisa la utilización de un criterio de selección que decida que filas son válidas y cuáles no: >Se escribe con la notación Rx= σ(c)

### Selección (σ):

La selección de una relación mediante una expresión lógica (predicado de selección) da como resultado una relación formada por el conjunto de tuplas que satisfacen dicha expresión.

#### Autor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan



Nombre	Nacionalidad	
Saltor	Española	U.P.C.



SELECT \*
FROM autor
WHERE nacionalidad="Española";

En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

**PROYECCIÓN**: forma una nueva tabla extrayendo un **subconjunto de columnas** de una tabla, por lo que vamos a extraer un grupo de atributos, donde tendremos un número de n-tuplas compuesta por los valores asociados a dichos atributos. Esta operación también necesita algún criterio de selección capaz de distinguir entre columnas válidas y no válidas.

Se escribe con la notación  $Rx = \sigma(c)$ 

Proyección (π):

La proyección de una relación sobre un conjunto de sus atributos es otra relación definida sobre ellos, eliminando las tuplas duplicadas que hubieran podido resultar.

### Autor

Nombre	Nacionalidad	Institución
Date	Norteamericana	Relat. Institute
Saltor	Española	U.P.C.
Bertino	Italiana	U. Milan



Nacionalidad
Norteamericana
Española
Italiana



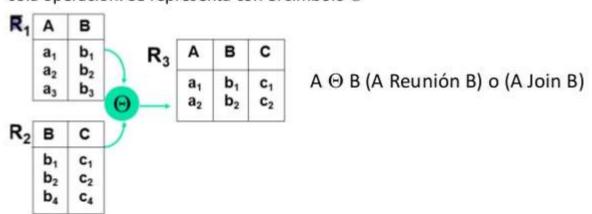
SELECT distinct (nacionalidad) FROM autor;

En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

- **JOIN O COMINACIONES:** es una operación que nos permite **combinar filas** de una tabla con filas de otras tablas basándonos en los valores de una determinada columna. Se obtienen por tanto filas con columnas de una tabla y columnas de las otras.
- »Se escribe con la notación Rx=S1 JOIN(CONDICION) S2
- >Hay tres tipos de combinaciones:
- »Combinación-theta: es una comparación de dos atributos.
- »Equi-combinación: la condición es una igualdad entre dos atributos.
- >Combinación natural: es una combinación de igualdad entre atributos del mismo nombre

### La operación reunión natural (⊕ Join)

 Dadas dos relaciones específicas A y B, construye una tercera relación C que combina ciertas selecciones, proyección y un producto cartesiano en una sola operación. Se representa con el símbolo Θ





En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

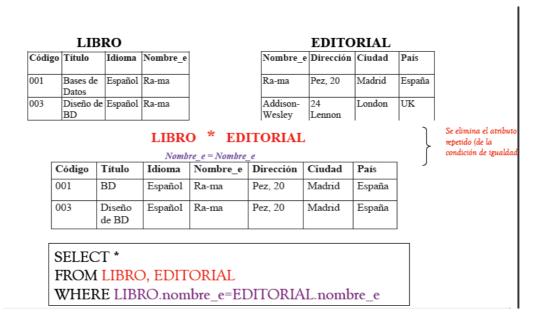
- □Combinación Natural o Equi-JOIN (\*): Si se trata de una condición de combinación simple por igualdad se denomina Combinación Natural (denotada con \*).
- »En esta combinación por igualdad se elimina uno de los dos atributos cuyos valores son idénticos. Es el caso más utilizado de combinación para relaciones que tienen un atributo común.
- La combinación natural puede hacerse entre relaciones que tengan más de un atributo común. En este caso, la combinación natural se realizarán sobre el conjunto de atributos comunes:

>R1 \* R2

>A1i = A2j

□Cuando el atributo común tiene el mismo nombre en ambas relaciones, se suele omitir la condición de combinación:

>R1 \* R2





En el modelo relacional las operaciones que podremos realizar y que están basadas en la teoría de conjuntos son:

## **□Combinación Natural o Equi-JOIN (\*):**

#### LIBRO \* EDITORIAL

Nombre\_e = Nombre\_e

	LIBRO	)	
Código	Título	Idioma	Nombre e
			_
001	Bases de	Español	Ra-ma
	Datos	_	
003	Diseño de	Español	Ra-ma
	BD	_	

#### **EDITORIAL**

Nombre_e	Dirección	Ciudad	País
Ra-ma	Pez, 20	Madrid	España
	24 Lennon	London	UK

π Libro.Código, Libro.Título, Libro.Nombre\_e, Editorial.Dirección, Editorial.Ciudad, Editorial.País(

our Libro.Nombre\_e=Editorial.Nombre\_e(LIBRO x EDITORIAL))

Código	Título	Idioma	Nombre_e	Nombre_e	Dirección	Ciudad	País
001	BD	Español	Ra-ma	Ra-ma	Pez, 20	Madrid	España
 001 -	BD	Español-	Pa-ma -	Addison-	24	London -	UK _
				Wesley	Lennon		
003	Diseño de BD	Español	Ra-ma	Ra-ma	Pez, 20	Madrid	España
 -0 <del>03</del>	Diseño de BD	Español	Ra-ma	-Addison-	24— — - Lennon	Lenden –	UK
	uc DD			*****	Lemon		



□Combinación Externa o OUTER JOIN: es un operador especial para el tratamiento de los valores nulos. Impide que desaparezcan tuplas por no tener correspondencia con ninguna de la otra relación (cuando se aplica la combinación *interna*). Por lo tanto, evita que las tuplas de una relación que no *casan* con ninguna tupla de la otra desaparezcan en el resultado. La combinación externa entre dos relaciones R1 y R2 consiste en variantes de combinación que conservan en el resultado, todas las tuplas de R1 (izquierda), todas las tuplas de R2 (derecha) o de ambas relaciones.

>Combinación Externa Izquierda o Left Outer JOIN (/\* o  $\theta$ i): La combinación externa izquierda entre dos relaciones R1 y R2 conserva en el resultado todas las tuplas de R1 (de la relación de la izquierda)=> R1 /\* R2 o R1  $\theta$ i R2.

Combinación Externa Izquierda o Left Outer JOIN (/\* o  $\theta$ i):

#### **AUTOR**

Cod_Autor	Nombre	Año_Nac	Libro
A1	Date	1939	L1
A2	Piattini	1965	
A3	De Miguel	1940	

#### LIBRO

Cod_Libro	Titulo	Año_P	ISBD
L1	BD	1980	1-10-80
L2	Ing.Sw.	1965	2-20-80
L3	EI ME/R	1940	3-30-80

Autor /\* Libro
Libro = Cod\_Libro

Cod_Autor	Nombre	Año_Nac	Libro	Título	Año_P	ISBN
A1	Date	1939	L1	BD	1980	1-10-80
A2	Piattini	1965	NULL	NULL	NULL	NULL
A3	De Miguel	1940	NULL	NULL	NULL	NULL



>Combinación Externa Completa/Plena o Full Outer JOIN (/\*/ o  $\theta$ p): la combinación externa plena entre dos relaciones R1 y R2 conserva en el resultado todas las tuplas de R1 y todas las tuplas de R2=> R1 /\*/ R2 o R1  $\theta$ p R2.

# Combinación Externa Plena o Full Outer JOIN (/\*/ o $\theta_p$ ):

### **AUTOR**

Cod_Autor	Nombre	Año_Nac	Libro
A1	Date	1939	L1
A2	Piattini	1965	
A3	De Miguel	1940	

#### LIBRO

Cod_Libro	Titulo	Año_P	ISBD
L1	BD	1980	1-10-80
L2	Ing.Sw.	1965	2-20-80
L3	EI ME/R	1940	3-30-80

Autor /\*/ Libro

Cod_Autor	Nombre	Año_Nac	Libro	Cod_Libro	Título	Año_P	ISBN
A1	Date	1939	L1	L1	BD	1980	1-10-80
A2	Piattini	1965	NULL	NULL	NULL	NULL	NULL
A3	De Miguel	1940	NULL	NULL	NULL	NULL	NULL
A1— — —	—⊕ate— —	<b>-1</b> 939	_L1— —	<b>#</b>	<del></del>	<del>49</del> 80— —	<del></del>
NULL	NULL	NULL	NULL	L2	Ing.Sw.	1965	2-20-80
NULL	NULL	NULL	NULL	L3	EI ME/R	1940	3-30-80



# **EJEMPLOS:**

CLIOESTE				
NUMEROC LI	NOMBRECLI	DIRECCIO N		
15	LMV	VIGO		
35	JMP	ORENSE		
152	JORGE	CORUÑA		

□CLIENTES=CLIOESTE U CLICENTRO

CLIENTES				
NUMEROC LI	NOMBRECLI	DIRECCIO N		
20	RSV	TOLEDO		
138	JMP	MADRID		
152	JORGE	CÁCERES		
36	DEL PINO	TOLEDO		
15	LMV	VIGO		
35	JMP	ORENSE		
152	JORGE	CORUÑA		

	CLICENTRO		
	NUMERO CLI	NOMBRE CLI	DIRECCIO N
	20	RSV	TOLEDO
	138	JMP	MADRID
	152	JORGE	CÁCERES
CLIENTES=CLIOES	TE O ČL	CENTRO	TOLEDO

CLIENTES

NUMEROC NOMBRECLI DIRECCIO
LI N

## **EJEMPLOS:**

CLIOESTE			
NUMEROC LI	NOMBRECLI	DIRECCIO N	
15	LMV	VIGO	
35	JMP	ORENSE	
152	JORGE	CORUÑA	

□CLIENTES=CLIOESTE - CLICENTRO

CLIENTES				
NUMEROC LI	NOMBRECLI	DIRECCIO N		
15	LMV	VIGO		
35	JMP	ORENSE		
152	JORGE	CORUÑA		

CLICENTRO					
NUMERO CLI	NOMBRE CLI	DIRECCIO N			
20	RSV	TOLEDO			
138	JMP	MADRID			
152	JORGE	CÁCERES			

CLIEORENSE=O(DIRECCIÓN=DE ORENSTENSIONELIDESTE

CLIOESTE			
NUMEROC LI	NOMBRECLI	DIRECCIO N	
35	JMP	ORENSE	



# **EJEMPLOS:**

CLIOESTE				
NUMEROC LI	NOMBRECLI	DIRECCIO N		
15	LMV	VIGO		
35	JMP	ORENSE		
152	JORGE	CORUÑA		

NCLIENTE=σ(NUMEROCLI>15 AND NUMEROCLI<152) CLIOESTE

NCLIENTE				
NUMEROC LI	NOMBRECLI	DIRECCIO N		
35	JMP	ORENSE		

CLICENTRO				
NUMERO CLI	NOMBRE CLI	DIRECCIO N		
20	RSV	TOLEDO		
138	JMP	MADRID		
152	JORGE	CÁCERES		
36	DEL PINO	TOLEDO		

