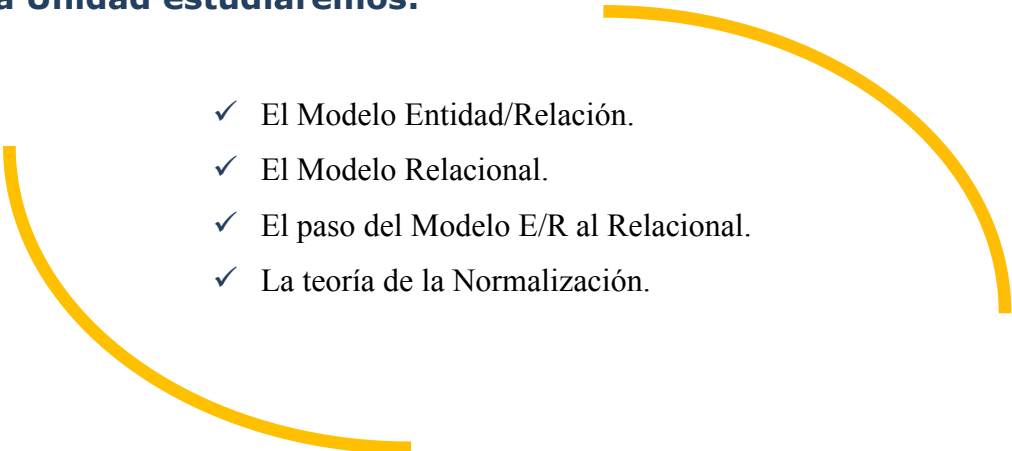


UNIDAD 2: MODELO ENTIDAD/RELACIÓN (E/R) Y MODELO RELACIONAL

**Módulo Profesional:
Bases de Datos**

En esta Unidad estudiaremos:

- 
- ✓ El Modelo Entidad/Relación.
 - ✓ El Modelo Relacional.
 - ✓ El paso del Modelo E/R al Relacional.
 - ✓ La teoría de la Normalización.

CONTENIDO

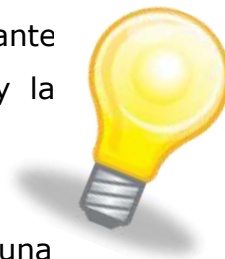
_Toc523074343

Caso introductorio.....	4
1. Modelo Entidad/Relación.....	5
1.1. Elementos.....	6
1.1.1. Entidad.....	6
1.1.2. Atributo.....	7
1.1.3. Clave.....	8
1.1.4. Relación.....	9
1.1.5. Cardinalidad.....	9
1.1.6. Relaciones débiles.....	12
1.1.7. Generalización y Especialización.....	13
1.2. Diagrama.....	15
1.3. Calidad de un esquema. Factores.....	15
2. Modelo Relacional.....	16
2.1. Arquitectura.....	16
2.1.1. Tablas.....	17
2.1.2. Vistas.....	18
2.2. Reglas de Integridad.....	18
2.2.1. Integridad de entidad.....	19
2.2.2. Integridad de clave.....	19
2.2.3. Integridad referencial.....	19
3. Paso del modelo E/R a Modelo Relacional.....	21
3.1. Eliminación de jerarquías de generalización.....	22
3.2. Eliminar los atributos compuestos.....	23
3.3. Eliminar atributos multivalor.....	23

3.4.	Metodología para realizar la conversión.....	24
3.4.1.	Entidades.....	24
3.4.2.	Relaciones Recursivas.....	24
3.4.3.	Relaciones Binarias.....	25
4.	Normalización.....	29
4.1.	Primera Forma Normal (1FN).....	30
4.2.	Segunda Forma Normal (2FN).....	30
4.3.	Tercera Forma Normal (3FN).....	31
5.	Resumen.....	32
ANEXO: Software.....		32
ANEXO: Ejemplos.....		34

Caso introductorio

Al ser parte del equipo de desarrollo de software de una importante consultora internacional, la misión principal es el desarrollo y la puesta en marcha de sistemas. Un ejemplo de ello puede ser aquel que permita gestionar los datos de la intranet de la propia empresa. Para ello, se aprueba realizar el desarrollo de una base de datos MySQL como base del proyecto. Como tal, se deben realizar los bocetos, las estructuras, los esquemas y la definición de datos que contendrá el sistema. En la presente unidad se desarrollarán los conceptos de modelado de datos, el diseño de la base de datos y la teoría de la normalización. Todo ello con el objetivo de obtener una base de datos sólida, estructurada, estandarizada y normalizada, que minimice riesgos y excluya problemas en futuras ampliaciones del software de gestión.



Al finalizar la unidad el alumnado:

- ✓ Conocerá la técnica de modelado de datos a través del Modelo Entidad/Relación.
- ✓ Aprenderá la importancia del desarrollo del Modelo Relacional, necesario para la lógica de predicados y la teoría de conjuntos.
- ✓ Sabrá realizar la transformación entre los diferentes modelos.
- ✓ Será capaz de comprender la teoría de la Normalización o estandarización de la base de datos, aplicando una serie de reglas obtenidas tras la transformación entre los modelos.

1. Modelo Entidad/Relación

El modelo conceptual de datos más ampliamente conocido es el de entidades y relaciones. Permite realizar el diseño conceptual de una base de datos. Como se ha estudiado, este paso, en la fase de diseño, es previo a la elección del modelo lógico de datos y, por supuesto, al sistema gestor de base de datos. El **Modelo Entidad/Relación** es el que incorporan la mayoría de las herramientas de software para el diseño de sistemas.

Es una **representación gráfica y lingüística** de los objetos que forman parte del mundo real. Describe los datos importantes en un entorno determinado y proporciona una visión abstracta de la realidad, sin hacer alusión a formas de almacenamiento, tiempos de ejecución, sistemas operativos, sistemas gestores de bases de datos, etc.

Probablemente, su éxito es fruto del equilibrio conseguido en las cualidades deseables en un modelo de representación de datos:



Ilustración: Cualidades deseables en un modelo de representación de datos

Es expresivo porque ofrece varios mecanismos de abstracción de datos. Es rico en la descripción de conceptos, lo que pone de manifiesto su simplicidad. No obstante, determinados conceptos que complican el modelo, facilitan enormemente el paso al Modelo Relacional, jerárquico o de redes, lo que compensa el esfuerzo por entender y acostumbrarse a utilizar dichos conceptos. El modelo es mínimo, ya que ninguno de los elementos que intervienen puede ser sustituido por la combinación de otros.

Fue definido formalmente por **Peter Chen** a finales de los años setenta. Es fácilmente legible y los diagramas son completos. La legibilidad de un Diagrama Entidad/Relación se ve perjudicada cuando se pretenden incluir todos los elementos del modelo en un único diagrama. Por ello, es conveniente realizar varios esquemas detallados diferentes.

1.1. Elementos

El Modelo Entidad/Relación, tiene sus estructuras propias, que son conocidas como Diagramas Entidad/Relación. De hecho, para describir el esquema conceptual de la base de datos se debe construir su diagrama a través de una serie de elementos. Cada uno de estos elementos tiene asociado un símbolo gráfico específico, que lo representa y lo distingue del resto de elementos. En los siguientes apartados se describen cada uno de estos elementos, sus características y simbología.

1.1.1. Entidad

Una **entidad** es un objeto del mundo real, acerca del cual se puede y se quiere obtener una determinada información. Suelen ser los **sustantivos** empleados al describir las actividades de una empresa o institución (Empresa, Empleado, Proyecto, etc.). Asociado al concepto de **entidad** surge el concepto de ocurrencia de entidad, que es una realización concreta de una entidad. Por ejemplo, si se tiene la entidad **Empleado**, una ocurrencia de la misma serán los datos referentes al empleado Sánchez. La representación gráfica de una entidad es un **rectángulo etiquetado**.

Las reglas que debe cumplir una entidad son:

- Tiene que tener existencia propia.
- Cada ocurrencia de un tipo de entidad debe poder distinguirse de las demás.
- Todas las ocurrencias de un tipo de entidad deben tener los mismos tipos de características (**atributo**).

Se distinguen dos clases de entidad:

- **Fuertes:** Son independientes, no necesitan la existencia de otras entidades. Son la mayoría de ellas, por ejemplo, Empleados.
- **Débiles:** Su existencia está ligada a la existencia de otra entidad. Por ejemplo, HijoEmpleado. Las débiles se representan gráficamente con un rectángulo con doble línea.

1.1.2. *Atributo*

Un **atributo** es una propiedad o característica asociada a una determinada entidad y, por tanto, común a todas las ocurrencias de esa entidad. Por ejemplo, los atributos de la entidad Empleado podrían ser: Código Empleado, DNI, Nombre, Fecha Nacimiento, Dirección, Teléfono, Departamento o Sueldo. El atributo se corresponde con el término **campo** utilizado en el ámbito de los sistemas de gestión de archivos. Un **atributo** se representa gráficamente por medio de una **elipse** y en el interior de la misma se escribe el nombre que lo identifica.

Asociado al concepto de atributo surge el concepto de **dominio**, que es el conjunto de valores permitidos para un atributo. Distintos tipos de dominios son: enteros, cadenas de texto, fecha, etc. Se dice que un **atributo es compuesto** si puede dividirse en atributos independientes. Por ejemplo, el atributo dirección puede dividirse en los atributos: calle, número y código Postal. Un **atributo se dice que es obligatorio** si todas las ocurrencias de atributo son valores no nulos y no vacíos, mientras que es **opcional** si puede estar vacío o contener el valor nulo.

Existe un caso especial llamado **multivalor**, cuando el atributo puede tener un conjunto de valores. Por ejemplo, si se quieren registrar varios Teléfonos para el Empleado. Más adelante se comprobará que esto se puede implementar de varias formas, dependiendo del caso concreto. Por ejemplo, si el número de teléfonos es pequeño y fijo, se pueden incluir varios campos individuales (Teléfono 1, Teléfono 2, etc.), pero si no está determinado, habrá que crear una nueva entidad que contenga los teléfonos asociados a cada empleado.

1.1.3. Clave

De entre todos los atributos de un tipo de entidad, se debe elegir uno o varios que **identifiquen unívocamente** cada una de las ocurrencias de ese tipo de entidad. Este atributo o conjunto de atributos se denomina **identificador principal** y es la **clave primaria de la entidad**. Puede ocurrir que existan otros atributos que verifiquen la condición de ser identificador único. Estos atributos se denominan **atributos identificadores alternativos** y representan las **claves alternativas o candidatas**. Para representar gráficamente los atributos que son **clave primaria**, el nombre que identifica al atributo debe **aparecer subrayado**.

Una clave es **simple** si está formada por un solo atributo, y **compuesta**, si tiene más de un atributo.

Por ejemplo, para la entidad Empleado se tienen dos atributos que lo pueden identificar (Código Empleado y NIF). Se podría elegir el atributo Código Empleado como clave primaria y NIF como clave candidata.

Cuando existen varias claves candidatas, puede resultar conflictivo elegir una de ellas como primaria. La decisión afectará a la rapidez de acceso físico a las ocurrencias de la entidad. Para facilitar esta tarea pueden tenerse en cuenta los siguientes criterios:

- Elegir la clave que sirva para acceder directamente a las ocurrencias en el mayor número de operaciones.
- Es preferible escoger una clave simple antes que una compuesta.

1.1.4. Relación

Una **relación** es una **asociación entre dos o más entidades**. Suelen identificarse **por verbos** que unen las entidades. Por ejemplo, se podría tener la relación Trabaja, que une las entidades Empleado y Empresa. La relación se representa gráficamente por medio de un **rombo**, en cuyo interior se escribe la etiqueta que identifica la relación. Una relación queda caracterizada por:

- **Nombre:** Como todo objeto del Modelo Entidad/Relación, las relaciones deben tener un nombre que las identifique unívocamente.
- **Grado:** Número de tipos de entidad sobre las que se realiza la asociación. La relación Trabaja del ejemplo será binaria, es decir, su grado es dos. Las relaciones de grado 3 se denominan ternarias.
- **Cardinalidad:** Número máximo de ocurrencias de cada tipo de entidad que pueden intervenir en una ocurrencia del tipo de relación.

1.1.5. Cardinalidad

Hace referencia a la cantidad de elementos pertenecientes a una entidad que se pueden relacionar con otra cantidad determinada de elementos de otra entidad:

- **(1:M)** → Un elemento de una entidad se relaciona con varios elementos de la otra entidad.
- **(1:1)** → Un elemento de una entidad se relaciona sólo con otro elemento de la otra entidad.
- **(M:M)** → Son posibles todas las relaciones entre individuos pertenecientes a ambas entidades.

Uno a uno. Se representa **1:1**. Si a cada ocurrencia de A le corresponde como máximo una ocurrencia de B, y viceversa.

Uno a muchos. Se representa **1:M**. Si a cada ocurrencia de A le pueden corresponder varias de B, pero a cada ocurrencia de B sólo le corresponde una de A como máximo. Si la asociación se entendiera de B con A, la relación sería M:1.

Muchos a muchos. Se representa **M:M**. A cada ocurrencia de A le pueden corresponder varias de B, y viceversa.



Ilustración: Relación 1:1



Ilustración: Relación 1:M



Ilustración: Relación M:M

Además, la **cardinalidad** mide la obligatoriedad de correspondencia entre las ocurrencias de dos entidades en una relación. Se dice que una entidad A tiene un tipo de participación **obligatoria** en una relación con otra entidad B, si a cada ocurrencia de A le corresponde al menos una de B. En cambio, A tiene participación **opcional**, si pueden existir ocurrencias de A que no tengan correspondencia en B. Según esto, las relaciones binarias se clasifican en:

- **Obligatoria-Obligatoria.** Todas las ocurrencias de cada entidad tienen correspondencia con como mínimo una ocurrencia de la otra. Por ejemplo, Alumno Cursa Asignatura, todos los alumnos cursan al menos una asignatura, y cada asignatura es cursada por al menos un alumno.



Ilustración: Relación Obligatoria-Obligatoria

- **Obligatoria-Opcional.** Cada ocurrencia de la primera entidad tiene asociada al menos una ocurrencia de la segunda. Sin embargo, puede haber ocurrencias de la segunda entidad que no tengan asociadas ninguna en la primera entidad. Por ejemplo, en la relación Cliente Ocupa Habitación, suponiendo que sólo interesan los clientes alojados en el hotel, todos los clientes ocupan una habitación, pero puede haber habitaciones vacías y, por tanto, no ocupadas por ningún cliente.



Ilustración: Relación Obligatoria-Opcional

Si la relación hubiera sido enunciada a la inversa, Habitación es Ocupada por Cliente, la relación sería opcional-obligatoria.

- **Opcional-Opcional.** Alguna ocurrencia de ambas entidades puede no tener correspondencia con ninguna ocurrencia de la otra entidad. Por ejemplo, en la relación Lector Saca Libro, un lector socio de una biblioteca puede no haber sacado nunca un libro prestado y también es posible que ciertos libros nunca hayan sido prestados a lectores.



Ilustración: Relación Opcional-Opcional

1.1.6. Relaciones débiles

Las relaciones débiles se caracterizan por tener que tomar parcial o totalmente su clave principal a partir de la o las claves principales de otras entidades. Por tanto, no tiene sentido hablar de una relación débil si no es por el hecho de que determinada entidad (en el gráfico la E2) requiere del atributo presente en otra entidad (la E1) para formar su propia clave. La entidad E2 es por tanto entidad débil, así como su relación R también es débil.



Ilustración: Relación débil

Los atributos de la entidad E2 que forman clave se denominan **discriminante** de la entidad, ya que permiten discriminar diferentes elementos de la entidad por cada uno de los elementos de la entidad E1. Se debe recalcar que sólo existen relaciones binarias débiles uno a muchos y uno a uno, ya que la necesidad de tomar la clave de otra entidad impide el cruce múltiple entre elementos.



Ilustración: Relación débil

En este caso, la entidad E2 tomará la clave de E1 igualmente para formar su clave, pero dicha entidad no podrá tener atributos clave propios de la entidad, a diferencia del caso anterior, ya que de hacerlo ello implicaría la posibilidad de existir varios elementos por cada uno de los elementos existentes en E1. Claramente, en este caso la entidad E2 no tiene discriminante.

Suponga, por ejemplo, el caso de una entidad E1 (Grupos) y una entidad E2 (Discos). Es evidente que los discos pertenecen a grupos musicales. Sin embargo,

en cada entidad se guardarán características determinadas de cada caso. Así de los grupos se puede guardar su nombre, nacionalidad, año de fundación del grupo y miembros del grupo, mientras que de los discos se puede guardar el título del álbum, el año de publicación y el número de temas que componen el disco.

Una de las características que ayudan bastante a identificar a las entidades débiles es el caso de que en la mayoría de las ocasiones ayuda el poner como sufijo de la entidad débil a la entidad fuerte. Así en el ejemplo anterior, se puede pensar que los discos, son Discos (la entidad débil) de Grupos (la entidad fuerte de la que se toma clave). Por tanto, la clave de la entidad Grupos es el nombre del grupo (no pudiendo existir dos grupos con el mismo nombre) y la clave de Discos será el título del grupo, junto con el nombre del grupo. De esta forma sí se pueden tener, en la entidad Discos, dos discos con el mismo título, siempre y cuando, eso sí, pertenezcan a Grupos diferentes. En este ejemplo también se puede ver que el título del álbum discrimina (o diferencia) los discos de cada grupo, pudiendo existir discos del mismo título, siempre que pertenezcan a grupos diferentes.



Ilustración: Ejemplo de relación débil

1.1.7. Generalización y Especialización

La **generalización** es el resultado de la unión de dos o más conjuntos de entidades (de bajo nivel) para producir un conjunto de entidades de más alto nivel. La generalización se usa para hacer resaltar los parecidos entre tipos de entidades de nivel más bajo y ocultar sus diferencias. La generalización consiste en identificar todos aquellos atributos iguales de un conjunto de entidades para formar una/s entidad/es global/es con dichos atributos semejantes. Dicha/s entidad/es global/es quedaran a un nivel más alto al de las entidades origen.

Por ejemplo, se supone que se tienen las entidades CtaAhorro y CtaCheques, y que ambas tienen los atributos semejantes de NumCta y Saldo, aunque además de estos dos atributos, la CtaAhorro tiene el atributo TasaInteres y CtaCheques el atributo SaldoDeudor. De estos atributos se pueden unir (generalizar) NumCta y Saldo que son iguales en ambas entidades.

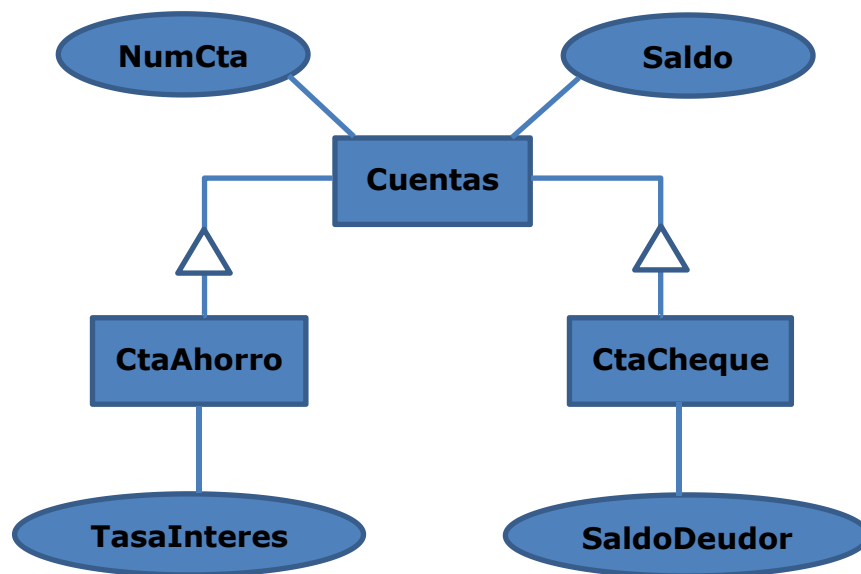


Ilustración: Ejemplo de generalización

Se puede leer esta gráfica como: La entidad CtaAhorro hereda de la entidad Cuentas los atributos numCta y saldo, además del atributo de TasaInteres, de forma semejante CtaCheque posee los atributos de numCta, saldo y saldoDeudor. Como se puede observar, la Generalización trata de eliminar la redundancia (repetición) de atributos, al englobar los atributos semejantes. La/s entidad/es de bajo nivel cuentan (heredan) todos los atributos correspondientes.

La **especialización** coincide con el concepto contrario a la generalización. Es el resultado de tomar un subconjunto de entidades de alto nivel para formar un conjunto de entidades de más bajo nivel. Por tanto, es lo mismo decir que, en el ejemplo anterior, las Cuentas son una generalización de las CtaAhorro y las CtaCheque, que decir que dichas entidades son una especialización de las Cuentas.

1.2. Diagrama

Una vez analizado en detalle cada uno de los elementos que forman el modelo, se observan algunos ejemplos de cómo se representarían en un diagrama.

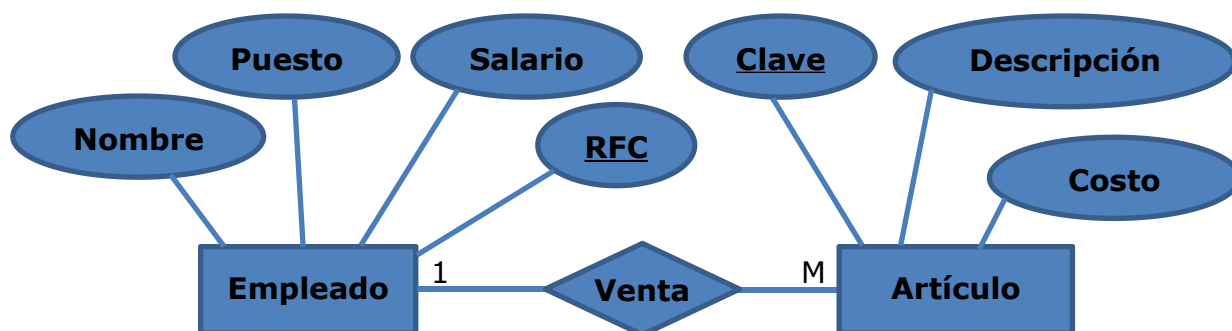


Ilustración: Ejemplo de generalización

La entidad Empleado tiene como clave primaria el atributo RFC (se muestra subrayado); para la entidad Artículo, su clave es el atributo Clave. Se observa que este diagrama representa una relación binaria (de grado 2), ya que relaciona dos entidades. La **cardinalidad** de este ejemplo es de uno a muchos (1:M), ya que un empleado puede vender varios artículos, pero un artículo solo puede ser vendido por un empleado. Si los artículos pudiesen ser vendidos por varios empleados, esta relación sería del tipo muchos a muchos (M:M).

1.3. Calidad de un esquema. Factores

A la hora de crear un esquema conceptual utilizando el Modelo Entidad/Relación, se debe intentar que el diagrama resultante cumpla los siguientes factores:



Ilustración: Factores de un esquema Entidad/Relación

- **Completo:** En el modelo se reflejan todas las características de la realidad que pretende representar.
- **Correcto:** Utiliza correctamente todos los elementos necesarios del Modelo Entidad/Relación.
- **No Redundante:** No existen propiedades repetidas en el esquema.
- **Legible:** Que una persona que no ha participado en la elaboración del esquema sea capaz de comprenderlo.
- **Extensible:** Se adapta a futuros cambios que no afecten totalmente a las estructuras del diagrama.
- **Normalizado:** El esquema está en la forma normal de Boyce-Codd, esta propiedad puede posponerse hasta que el diagrama se transforme al modelo lógico de datos elegido (relacional, jerárquico o de redes).

2. Modelo Relacional

En el Modelo Relacional los datos se estructuran lógicamente en forma de relaciones (tablas), siendo el objetivo fundamental del modelo mantener la independencia de la estructura lógica respecto de la estructura física.

2.1. Arquitectura

Para una arquitectura en 3 niveles cada nivel tiene asociado un tipo de esquema:

- **Esquema conceptual:** Se define mediante relaciones que se representan con tablas. Cada entidad y relación se corresponde con una tabla bidimensional. Las columnas son los atributos y las filas las ocurrencias.
- **Esquema externo:** Se describe mediante vistas que se corresponden con los subesquemas de la arquitectura estándar. Una vista es una tabla virtual que se forma a partir de las tablas del esquema conceptual, pero que no tiene correspondencia en el nivel interno.
- **Esquema interno:** Cada tabla del esquema conceptual se almacena en un archivo. Para cada clave candidata se crea un índice para el posterior acceso directo a los datos del archivo.

2.1.1. Tablas

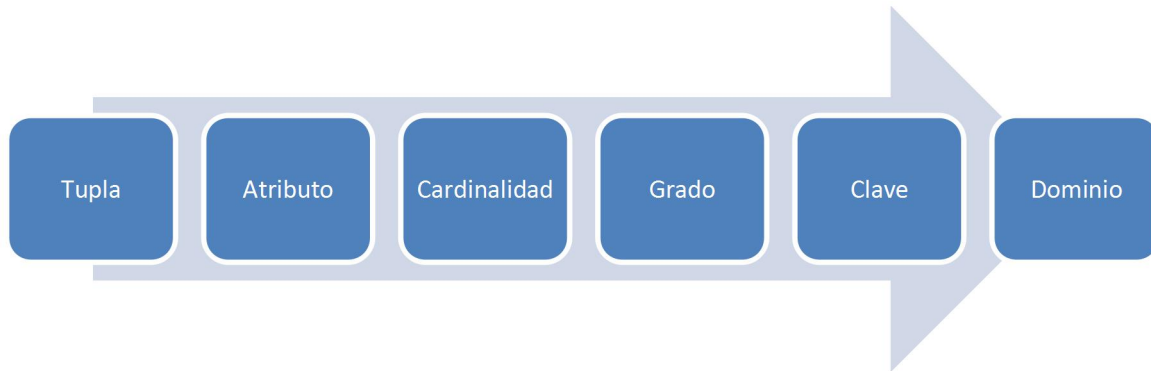


Ilustración: Conceptos que se derivan de las tablas

- **Tupla:** Es una fila de la tabla que se corresponde con cada ocurrencia de la relación.
- **Atributo:** Es cada una de las columnas de la tabla. Es equivalente al concepto de atributo del Modelo Entidad/Relación.
- **Cardinalidad:** Es el número de filas de la tabla.
- **Grado:** Es el número de atributos de la tabla. Todas las tuplas tienen el mismo número de atributos.
- **Clave:** Este concepto está definido de forma similar al expuesto en el Modelo Entidad/Relación. Es un conjunto de atributos que identifican unívocamente cada **tupla** de la relación. Cada tabla debe tener una sola clave primaria y puede tener otras claves alternativas o candidatas. Si un conjunto de atributos es clave en otra tabla distinta, se considera **clave foránea** o externa.
- **Dominio de un atributo:** Es el conjunto de valores que puede tomar dicho atributo. Pueden ser:
 - **Generales.** Los valores están comprendidos entre un mínimo y un máximo. Por ejemplo, la garantía de un vehículo es un número que puede estar comprendido entre 3 y 18 meses.
 - **Restringidos.** Los valores pertenecen a un conjunto discreto. Por ejemplo, sexo puede tomar los valores masculino y femenino.

En general, una tabla debe reunir los siguientes requisitos:

- Tener un número fijo de atributos para todas las tuplas.
- Cada atributo tiene un único dominio.
- Las tuplas no tienen por qué tener una secuencia determinada.
- El orden de los atributos no importa.
- No pueden existir tuplas ni atributos repetidos.
- Cada intersección fila-columna debe contener un valor único perteneciendo al dominio de la columna correspondiente.

2.1.2. Vistas

Los conceptos que se han expuesto hasta el momento hacían referencia al esquema conceptual del Modelo Relacional. Para describir el esquema externo se utilizan las **vistas**. Una vista es una tabla virtual que el usuario puede crear y manejar, y que no tiene que corresponderse con ningún archivo del nivel interno. Las tuplas que pertenecen a una vista se obtienen como resultado de consultas a las tablas del nivel conceptual. Las vistas pueden formarse:

- Eliminando atributos de una tabla.
- Uniendo tablas por atributos comunes.
- De ambas formas.
- También pueden definirse a partir de otras vistas.

Al crearlas se deben de tener en cuenta las restricciones impuestas por el administrador de la base de datos para cada usuario.

2.2. Reglas de Integridad

Al enumerar los objetivos que debe cumplir un sistema gestor de bases de datos se citaba la integridad. Parte de este objetivo se cumple si se consigue mantener la coherencia y la veracidad de la información en la base de datos. Las operaciones que pueden afectar a la **integridad** son la inserción, la modificación y el borrado de registros en la base de datos.

Existen algunas restricciones que deben cumplirse para mantener la base de datos íntegra, aunque no todos los sistemas gestores de bases de datos relacionales facilitan mecanismos de especificación de tales restricciones. Las tres restricciones principales que forman parte del Modelo Relacional de datos son las siguientes:

- Integridad de entidad.
- Integridad de clave.
- Integridad referencial.

2.2.1. Integridad de entidad

Ningún valor de la clave primaria de una relación puede ser nulo o desconocido. Tampoco se podrá desconocer parte de dicha clave primaria si está estuviera formada por varios atributos. Es razonable que si, por definición, la clave primaria permite distinguir las tuplas de una relación, las tuplas con valor nulo en la clave primaria no estarán identificadas. Esta restricción se consigue comprobando en cada inserción y modificación que ningún valor de los introducidos para la clave primaria es nulo o no forma parte de su dominio.

2.2.2. Integridad de clave

Los valores de claves candidatas en una relación deben ser únicos para cada tupla. El razonamiento es análogo al de la restricción anterior. Para evitar violar esa regla, se comprobará en todas las inserciones y modificaciones que el valor de todas las claves candidatas no existe en alguna tupla de esa relación.

2.2.3. Integridad referencial

Una tupla de una relación R_1 que haga referencia a otra relación R_2 debe referirse a una tupla existente en R_2 . Formalmente, cada valor de un atributo A que forma parte de una clave foránea en una relación R_2 , R_3 , etc., en la que A forma parte de la clave primaria. Esta regla puede verse afectada en la inserción, modificación y borrado de tuplas.

Para las dos primeras operaciones, el **SGBD** relacional debe asegurarse de que, al introducir un valor de una clave foránea, ese valor sea nulo o exista en todas las relaciones en las que la clave foránea sea clave primaria. En cuanto a la eliminación, no se permitirá borrar una tupla de una relación cuyo valor de clave primaria exista como valor de la clave foránea de otra relación.

PROFESORES

NOMBRE	EDAD	TELÉFONO
JUAN	27	911111111
JOSE	31	912222222
ROSA	31	913333333
JAVIER	33	914444444
LUIS	21	915555555

ASIGNATURAS

NOMBRE	HORAS	TIPO
MATEMÁTICAS	5	OB
INGLÉS	3	OP
FÍSICA Y QUÍMICA	5	OB
LENGUA Y LITERATURA	4	OP
TECNOLOGÍA	5	OP

IMPARTE

PROFESOR	ASIGNATURA
JUAN	MATEMÁTICAS
JOSE	TECNOLOGÍA
ROSA	INGLÉS
JAVIER	FÍSICA Y QUÍMICA
LUIS	LENGUA Y LITERATURA

Ilustración: Ejemplo de integridad referencial

Además de estas restricciones formales, es conveniente que los sistemas gestores de bases de datos relacionales, tengan en cuenta lo siguiente:

- No se permitirá introducir valores de atributos que no pertenezcan a sus dominios o sean de distintos tipos.
- Se podrán definir atributos obligatorios y opcionales. Los obligatorios, sean claves o no, deberán contener siempre valores válidos.
- Se podrán formular reglas de gestión de la base de datos (reglas semánticas), tales como: la fecha de devolución será mayor que la de préstamo, el mínimo de unidades suministradas por un proveedor será tres, todos los valores numéricos deberán ser positivos, un árbitro no podrá participar en dos partidos consecutivos de un mismo equipo, etc.

3. Paso del modelo E/R a Modelo Relacional

En el proceso de diseño de una base de datos, el primer paso consiste en realizar el diseño conceptual para, posteriormente, abordar el diseño lógico que genera el esquema de la base de datos en el modelo lógico elegido (relacional, jerárquico, de redes u orientado a objetos). Se presenta un procedimiento para transformar el esquema conceptual resultante de aplicar el Modelo Entidad/Relación al problema de diseño de una base de datos en un esquema relacional.



Ilustración: Paso del modelo Entidad/Relación al Modelo Relacional

Para poder llevar a cabo esta transformación, es necesario realizar previamente determinadas conversiones que eliminen elementos del Modelo Entidad/Relación no representable en el Modelo Relacional. Antes de pasar al Modelo Relacional, es conveniente analizar el diagrama Entidad/Relación y comprobar que no existen elementos que no puedan ser representados directamente en el Modelo Relacional. Para ello pueden seguirse los siguientes pasos.

3.1. Eliminación de jerarquías de generalización

Al hacerlo hay que tener en cuenta que no debe existir pérdida de información.

- **Integrar todas las entidades en una única:** Esta nueva entidad contendrá todos los atributos de la entidad genérica, los de las subentidades y un atributo discriminativo para distinguir a qué subentidad pertenecen las tuplas. Los atributos de las subentidades son tratados como opcionales. Todas las relaciones que tuvieran subentidades se mantienen ligadas a la nueva entidad recién creada, reconsiderando la cardinalidad de cada relación. Esta alternativa presenta el inconveniente de generar demasiados valores nulos en los atributos opcionales. También ralentiza el proceso de búsqueda al tener en cuenta todas las tuplas en vez de las que pertenecen a la subentidad deseada.
- **Considerar cada subentidad como entidad:** Para ello, se añaden los atributos de la entidad genérica a la subentidad y la clave primaria de la genérica pasa a serlo de las nuevas entidades creadas. Inconvenientes:
 - Se pierde el concepto de la entidad genérica.
 - Los accesos a la entidad genérica deben convertirse en accesos a todas las subentidades.
 - Los atributos de la entidad genérica son repetidos en cada subentidad.
 - Sólo es válida para jerarquías totales y exclusivas.
 - Si la entidad genérica tiene alguna relación, ésta debe propagarse a cada subentidad.

En consecuencia, esta alternativa es válida cuando la jerarquía es total o exclusiva, no importa el concepto de la entidad genérica en las operaciones y no hay relación entre la entidad genérica y otras entidades.

- **Insertar una relación entre la entidad genérica y cada subentidad:** Los atributos se mantienen y a las subentidades se las puede identificar con la clave foránea de la entidad genérica. Su inconveniente principal es que el esquema resultante es bastante complejo e, incluso puede resultar redundante. En cambio, permite representar todos los tipos de jerarquías.

3.2. Eliminar los atributos compuestos

El Modelo Relacional sólo permite representar atributos simples, por lo que deben convertirse los atributos compuestos. Existen dos alternativas:

- Considerar como atributo simple cada componente del compuesto; por ejemplo, el atributo FechaNacimiento compuesto de Día, Mes y Año convertirlo en tres atributos DíaNacimiento, MesNacimiento y AñoNacimiento.
- Considerar el atributo compuesto como simple. En este caso es responsabilidad de la aplicación identificar cada parte del atributo. Para el atributo FechaNacimiento, pasaría a ser simple y, en vez de tener tres partes de dos dígitos, se representaría con seis dígitos manteniendo el significado de cada dígito antes de la conversión.

3.3. Eliminar atributos multivalor

Pueden darse dos casos:

- El atributo multivalor pertenece a una entidad. En este caso, se crea una nueva entidad que contiene el atributo multivalor como monovalor y la clave primaria de la entidad original. La clave primaria de la nueva entidad es la suma de sus atributos. Por ejemplo, el atributo Méritos en una entidad Candidato con atributos DNI..., Méritos se dividirá en dos entidades: Candidato con todos sus atributos, excepto Méritos, y la entidad Currículum con los atributos DNI y Méritos.
- El atributo multivalor pertenece a una relación R entre dos entidades E1 y E2. También se crea una nueva entidad que contiene el atributo multivalor como simple y:
 - La clave primaria de una de las entidades de la relación es 1:1.
 - La clave primaria de la entidad del lado muchos si la relación es 1:M.
 - Las claves primarias de ambas relaciones si la relación es M:M.

3.4. Metodología para realizar la conversión

Se detallan las entidades, las relaciones recursivas y las relaciones binarias.

3.4.1. Entidades

Este elemento es equivalente en los 2 modelos. Por lo tanto, basta con cambiar el nombre de la entidad por el de la relación, que será representada por una tabla. La clave primaria de la entidad también lo es de la relación.



Ilustración: Paso del Modelo E/R al Modelo Relacional con Entidades

3.4.2. Relaciones Recursivas

Una relación R de una entidad E consigo misma, se convierte en dos relaciones del Modelo Relacional, una tabla para la entidad E y otra para R. La nueva tabla incluirá dos veces la clave primaria de E (una vez por cada papel de la relación) y los atributos de R. La clave primaria será una de ellas si la relación es 1:M o la combinación de las dos si es M:N.

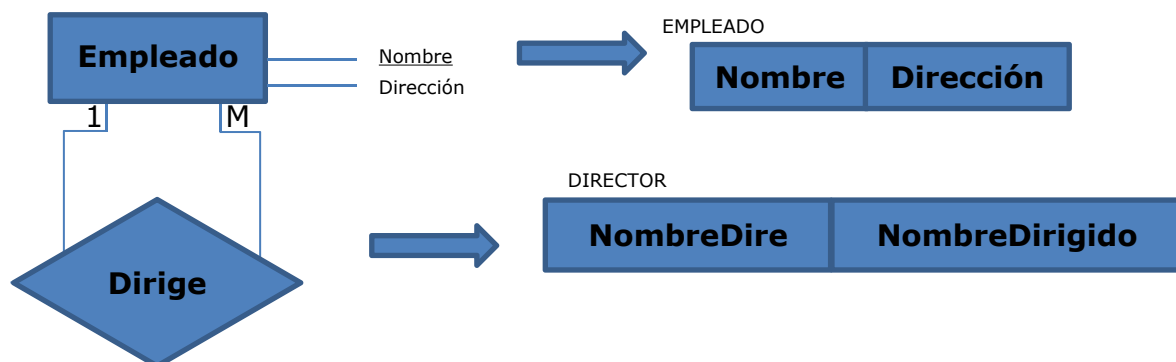


Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Recursiva

Si la relación fuera 1:1 se repetiría la clave primaria con el segundo papel y no necesitaría la segunda tabla.

Debe aclararse que siendo la relación 1:M, es posible utilizar una única relación con la clave primaria duplicada en los dos papeles. En este caso, el ejemplo se transformaría en la siguiente tabla:

EMPLEADO

NombreEmpleado	Dirección	NombreDirector
----------------	-----------	----------------

Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Recursiva

Esta alternativa no se aconseja en relaciones no obligatorias, ya que podrían generarse valores nulos (los directores no tienen por qué estar dirigidos).

3.4.3. Relaciones Binarias

Cuando se implemente el modelo en un SGBD, una vez diseñado gráficamente, las entidades se convertirán en tablas, pero ¿qué ocurrirá con las relaciones? El Modelo Relacional establece un sistema en el que las relaciones, según su cardinalidad, se convierten en unos u otros elementos del modelo.

RELACIÓN 1:M: Un atributo 'extra' añadido, a la tabla que tiene la cardinalidad 'M', al cual, y por convenio de algunos SGBD, se le podrá el sufijo '_fk' de Foreign Key (Clave Foránea). En general, se modelan añadiendo la clave primaria de la entidad del lado uno a la del lado muchos. Sin embargo, al aparecer alguna entidad con participación opcional es conveniente elegir otra solución.

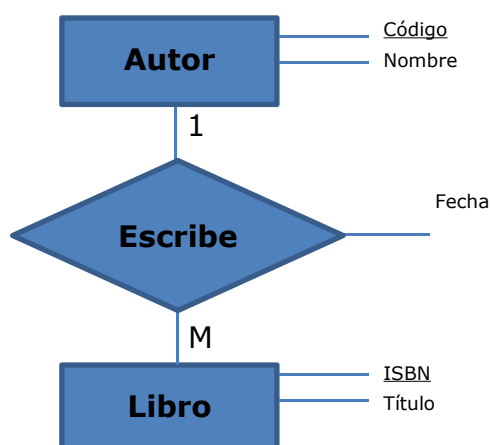


Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Binaria 1:M

Obligatoria-obligatoria. Cada entidad se convierte en una relación con su clave primaria, y la clave primaria de la entidad del "lado uno" se añade a la del lado muchos. Puede darse el caso que se quisiera añadir atributos a la relación; se incluyen en la entidad del lado muchos. El ejemplo quedaría de esta forma:

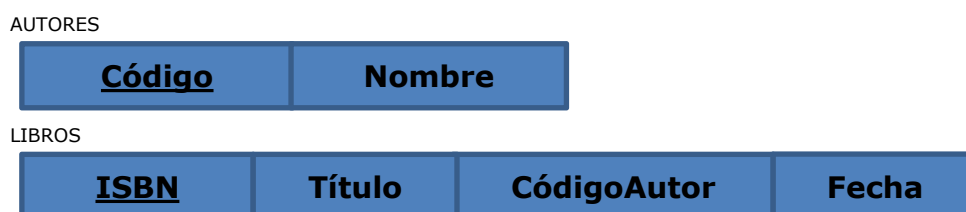


Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Binaria 1:M

Obligatoria-opcional. Se necesitan tres relaciones, una para cada entidad y otra para la correspondencia. Las entidades se transforman convirtiéndolas en tablas, y la nueva relación debe contener las claves primarias de las dos entidades y los posibles atributos de la correspondencia R. La clave primaria de la relación creada será la combinación de las claves primarias de E1 y E2:

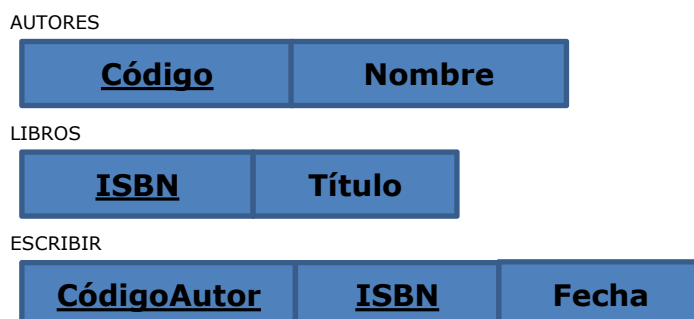


Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Binaria 1:M

Opcional-opcional: Este caso se reduce al de relación obligatoria-opcional.

RELACIÓN 1:1: Un atributo 'extra' añadido, a cualquiera de las tablas (preferiblemente, a la que supuestamente vaya a tener menos registros), al cual, y por convenio de algunos SGBD, se le podrá el sufijo '_fk' de Foreign Key (Clave Foránea) y al que se le fijará la condición de Unicidad o sin Duplicados (esta condición nada tiene que ver con la de requerido, la cual obliga a rellenar el campo, o evitar valores nulos).

En principio puede modelarse con una única relación que incluya los atributos de E1, E2 y R. Sin embargo, pueden generarse demasiados valores nulos cuando alguna de las participaciones no sea obligatoria. Por ello, se deben tratar por separado.

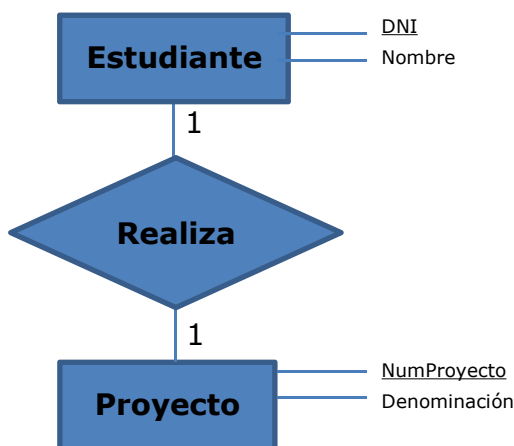


Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Binaria 1:1

Obligatoria-obligatoria. Se integran las dos entidades en una tabla que contiene los atributos, y la clave primaria es cualquiera de las de E1 y E2. Si coincidiera, sólo se incluiría una vez en la nueva relación:



Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Binaria 1:1

Obligatoria-opcional. Cada entidad se convierte en una relación representada por una tabla, y a la que tiene participación obligatoria se añadiría la clave primaria de la opcional. Las claves primarias de ambas relaciones se mantienen. Suponiendo ESTUDIANTE opcional, el ejemplo resultaría:

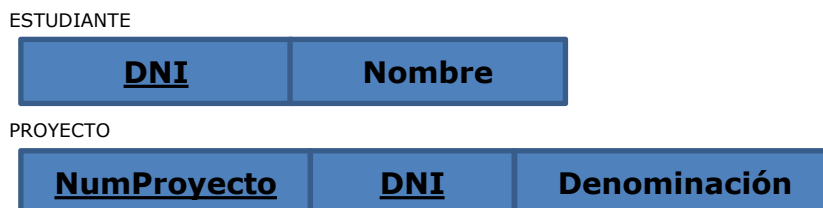


Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Binaria 1:1

Opcional-opcional. En este caso, se generarán tres relaciones: una para cada entidad y otra para la correspondencia entre ambas. Las entidades no sufrirían cambios y la nueva relación incluirá las claves primarias de E1 y E2, así como los atributos de R, si los hubiera. La clave primaria de esta relación recién creada será cualquiera de las de E1 o E2. El ejemplo, sería:



Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Binaria 1:M

RELACIÓN M:N: Se crea una nueva tabla en el modelo, cuyas claves principales serán tantos atributos extra clave foránea como sean necesarios hacia las tablas implicadas en la relación. Para estas relaciones no importa la cardinalidad. Siempre se transforman en tres relaciones, una para entidad y otra para la correspondencia. Las entidades se convierten en tablas y la relación de la correspondencia incluye las claves primarias de las entidades y los atributos de la asociación, siendo su clave primaria la combinación de las claves primarias de las entidades. Por ejemplo, observar la siguiente relación M:N que muestra la figura:

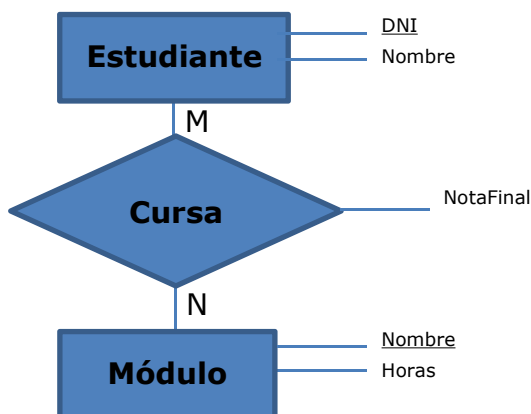


Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Binaria M:N

ESTUDIANTE

<u>DNI</u>	Nombre
-------------------	---------------

MÓDULO

<u>Nombre</u>	Horas
----------------------	--------------

CURSAR

<u>DNI</u>	<u>Nombre</u>	NotaFinal
-------------------	----------------------	------------------

Ilustración: Paso del Modelo E/R al Modelo Relacional con Relación Binaria M:N

4. Normalización

El proceso de normalización de bases de datos consiste en aplicar una serie de reglas a las relaciones obtenidas tras el paso del Modelo Entidad/Relación al Modelo Relacional. Las bases de datos relacionales se normalizan para:

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

En el Modelo Relacional es frecuente llamar **tabla a una relación**, aunque para ello tiene que cumplir con algunas restricciones:



Ilustración: Restricciones del Modelo Relacional

Las formas normales son aplicadas a tablas de una base de datos. Decir que una base de datos está en la **forma normal N** es decir que todas sus tablas están en la forma normal N. En general, las primeras tres formas normales son suficientes para cubrir las necesidades de la mayoría de las bases de datos. El creador de estas 3 primeras formas normales (o reglas) fue Edgar F. Codd.

4.1. Primera Forma Normal (1FN)

Esta forma normal elimina los **valores repetidos** dentro de una Base de Datos. Una tabla está en Primera Forma Normal si (ver ejemplo en Anexo):

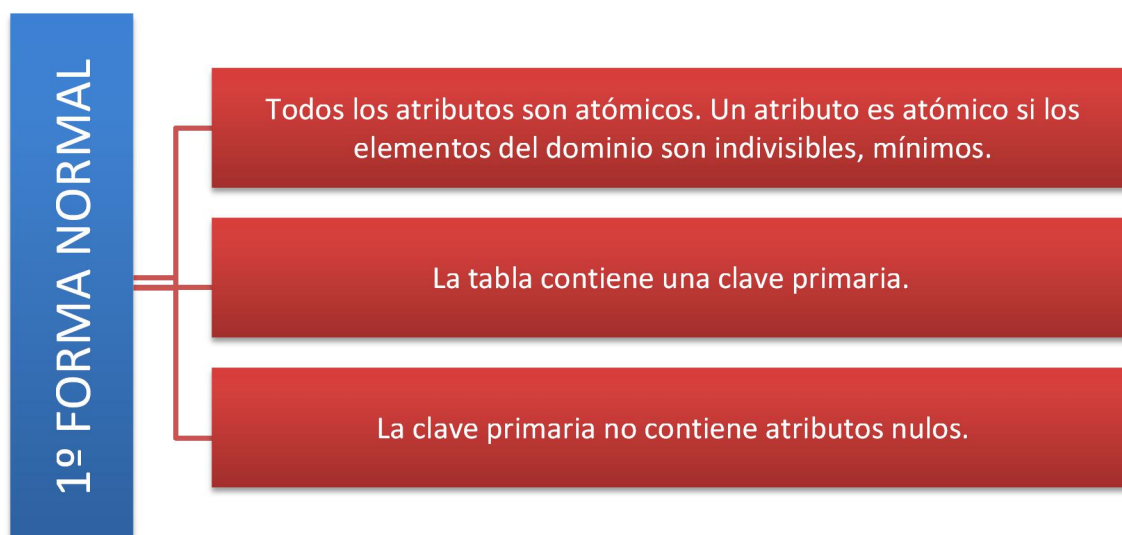


Ilustración: Primera Forma Normal (1FN)

4.2. Segunda Forma Normal (2FN)

Una relación está en 2FN si (ver ejemplo en Anexo):

- Está en 1FN.
- Los atributos que no forman parte de la clave dependen de forma completa de la clave principal, y no solo de una parte de ella. En otras palabras, los atributos no clave tienen una dependencia funcional completa con respecto a la clave, y no existen dependencias parciales.

Por tanto, se observa que cuando una tabla que está 1FN y su clave tiene un solo atributo, la tabla está automáticamente en 2FN. Por ejemplo, si se tiene una tabla cuya clave es IdProyecto y DNI, y tiene los atributos HorasTrabajo y NombreEmpleado:

- El atributo HorasTrabajo tendría dependencia funcional completa respecto de la clave, ya que si se quiere saber cuántas horas trabaja un empleado en un proyecto no basta solo con el DNI o el IdProyecto, sino que es necesario conocer ambos.
- Por el contrario, el atributo NombreEmpleado tiene dependencia parcial ya que solo depende del campo clave DNI, y no del IdProyecto.

Para organizar la información de manera que cumpla la 2FN, NombreEmpleado debería pertenecer a la tabla de empleados, y no a la de proyectos.

4.3. Tercera Forma Normal (3FN)

La tabla se encuentra en 3FN si (ver ejemplo en Anexo):

- Lo está 2FN.
- Se cumple que no existe ninguna dependencia funcional transitiva entre los atributos que no son clave.

Una **dependencia funcional transitiva** se produce cuando un atributo A depende de otro atributo B, y éste atributo B depende del atributo clave C. Por tanto, A también depende de C. Por ejemplo, se tiene una tabla con los campos Etapa, Ganador y NacimientoGanador (fecha de nacimiento del ganador de la etapa), donde la clave es el campo Etapa. Esta tabla tendría dependencia funcional transitiva entre los atributos que no son clave, ya que el campo NacimientoGanador se podría obtener a partir del campo Ganador, y no sería necesario almacenarlo tantas veces como etapas haya ganado un mismo corredor. Para expresar los mismos hechos sin violar la 3FN, es necesario dividir la tabla en dos, una tabla que contenga los ganadores de las etapas, y otra que contenga la fecha de nacimiento de cada corredor.

5. Resumen

En la presente unidad se han expuesto una serie de conceptos muy importantes en el ámbito de las bases de datos. En primer lugar, el modelo Entidad/Relación, como modelo de referencia para el diseño conceptual de las bases de datos, y tratando de analizar sus elementos (entidades, relaciones, atributos, etc.). En segundo lugar, el modelo Relacional, como modelo mucho más cercano a la futura implementación de la base de datos en el sistema. Y, por último, el paso entre ambos, analizando las diferentes posibilidades que pudiesen aparecer en situaciones reales. Para finalizar, se ha estudiado la teoría de la normalización, que ayuda al diseñador a comprender los diferentes modelos y a adecuar cada situación a su forma normal correspondiente.

ANEXO: Software

Existe en el mercado diverso software tanto comercial como libre para desarrollar los diagramas entidad relación y los modelos relacionales.

La principal herramienta a utilizar a lo largo del presente módulo será MySQL Workbench, una potente herramienta de gestión de bases de datos.

Web

Todo lo necesario sobre MySQL Workbench se encuentra en la siguiente web:

<https://www.mysql.com/products/workbench/>

Se dispone de herramientas de modelado de bases de datos, por lo que será más sencilla la realización de los diversos modelos aprendidos en la presente unidad.

Enlaces de

Para conocer cómo utilizar la herramienta MySQL Workbench como editor de diagramas, se puede obtener más información en el siguiente enlace:

<https://www.pedroventura.com/gestion-de-proyecto/crear-un-esquema-eer-desde-el-gestor-de-base-de-datos-mysql-workbench/>

Videos de

En los siguientes videos tutoriales se puede conocer cómo realizar los esquemas entidad relación gracias a la herramienta MySQL Workbench:

https://www.youtube.com/watch?v=hfE0_Mme32k

<https://www.youtube.com/watch?v=zAbFFEFeoBs>

ANEXO: Ejemplos

Video de

Seis ejercicios iniciales explicados paso a paso:

<https://www.youtube.com/watch?v=mSjG6XXRwZI>

Video de

Modelo conceptual de Bases de Datos. Modelo Entidad/Relación:

<https://www.youtube.com/watch?v=OfO5pPSP0a8>

Video de

Modelo conceptual de Bases de Datos. Modelo Entidad/Relación:

<https://youtu.be/N9-Z9ui7seE>

Videos de

Ejemplo Diseño de Bases de datos. Caso: Línea Aérea:

<https://www.youtube.com/watch?v=maSg-Ju0q7M>

Ejemplo de creación de diagramas y Base de Datos. Caso: Tienda Musical:

<https://www.youtube.com/watch?v=i6v4gFBeZC4>

Enlaces de

Primera, segunda y tercera forma normal. Teoría de la Normalización:

https://es.wikipedia.org/wiki/Primera_forma_normal

https://es.wikipedia.org/wiki/Segunda_forma_normal

https://es.wikipedia.org/wiki/Tercera_forma_normal