Una vez obtenido el esquema relacional resultante del esquema entidad/relación que representa la base de datos, normalmente tendremos una buena base de datos.
Entonces, el proceso suele terminar. Pero siempre que se realiza cualquier tipo de solución informática hace falta medir la calidad de la misma. La mayor parte de problemas se agravan si no se sigue un modelo conceptual y se decide crear directamente el esquema relacional. En ese caso, el diseño tiene una garantía casi asegurada de funcionar mal.
Si no cumple unos determinados parámetros realizar de forma reiterada sucesivos refinamientos en el diseño. Uno de los parámetros que mide esta calidad es la **forma normal**.
Son una serie de reglas agrupadas en niveles consecutivos de tal forma que un nivel agrupa al precedente
Estas reglas suelen forzar la división de una tabla en dos o más tablas para arreglar ese problema.

4.1. Objetivos.

a) Almacenar en la base de datos cada hecho solamente una vez, es decir evitar la redundancia de datos. De esta forma se reduce el espacio de almacenamiento.

b) Que los hechos distintos se almacenen en distintos sitios. Esto evita ciertas anomalías a la hora de operar con los datos.

Cuanto más avancemos en el cumplimiento de las formas normales mayor nivel de cumplimiento alcanzaremos.

Antes de abordar las distintas formas normales, es necesario definir los siguientes conceptos

4.2. Conceptos previos

4.2.1 Dependencia funcional

Se dice que un conjunto de atributos (**Y**) depende funcionalmente de otro conjunto de atributos (**X**) si para cada valor de **X** hay un único valor posible para **Y**. También se dice que **X** implica **Y**.

Simbólicamente se denota por **X** → **Y**.

Ejemplo:
Tenemos la tabla: DATOSPERSONALES(***DNI , nombre, dirección***)
El *nombre* de una persona depende funcionalmente del *DNI.*

DNI → *nombre*

La dirección no tiene dependencia funcional, ya que para un mismo DNI puede haber más de una dirección posible.

Al conjunto **X** del que depende funcionalmente el conjunto **Y** se le llama **implicante**. Al conjunto **Y** se le llama **implicado.**

4.2.2 Dependencia funcional completa

Un conjunto de atributos (**Y**) tiene una dependencia funcional completa sobre otro conjunto de atributos (**X**) si **Y** tiene dependencia funcional de **X** y además no se puede obtener de **X** un conjunto de atributos más pequeño que consiga una dependencia funcional de **Y** es decir, no depende funcionalmente de ningún subconjunto.
Una dependencia funcional completa se denota como **X** → **Y**

Ejemplo:

CLIENTES(Nombre, DNI, Apellidos)

El conjunto nombre y el dni producen una dependencia funcional sobre el atributo apellidos.
Pero no es completa ya que el dni individualmente, también produce una dependencia funcional sobre apellidos.

COMPRAS(CódigoProducto, CódigoProveedor, Cantidad, FechadeCompra)
CódigoProducto, CódigoProveedor → FechadeCompra

La FechadeCompra necesita de los dos códigos para tener dependencia funcional de ellos. Sin embargo ningún subconjunto, es decir en este caso ni uno ni otro código son capaces de determinar de forma única la fecha.

4.2.3 Dependencia funcional transitiva

Es más compleja de explicar, pero tiene también utilidad. Se produce cuando tenemos tres conjuntos de atributos **X, Y** y **Z. Y** depende funcionalmente de **X** (**X→Y**), **Z** depende funcionalmente de **Y** (**Y→ Z**). Además **X** no depende funcionalmente de **Y** (**Y!** → **X**). Entonces ocurre que **X** produce una dependencia funcional transitiva sobre **Z**.
Esto se denota como: (**X-** → **Z**)

Ejemplo:
PRODUCTOS(CódigoProducto, Nombre, Fabricante, País)
CódigoProducto →Fabricante
Fabricante → País
CódigoProducto **-** → País

PRODUCTOS(CódigoProducto, Nombre, Fabricante, País)
CódigoProducto → Nombre
Nombre→ CódigoProducto
CódigoProducto **-!**→País

4.3. FORMAS NORMALES

4.3.1 FN1

Se da cuando todos los valores para cada atributo de la relación son atómicos, es decir, cada atributo toma un valor único dentro del dominio del atributo.
Esta forma normal es inherente al modelo relacional, las tablas en un SGBD relacional están constituidas de esta forma. No se admiten los atributos multivaluados.

Ejemplo:

No es FN1.

Alumno	Asignatura
Pepe	Matemáticas
	Física
Juan	Historia
	Química

Sí es FN1.

Alumno	Asignatura
Pepe	Matemáticas
Pepe	Física
Juan	Historia
Juan	Química

4.3.2 FN2:

Se da cuando:

1. Es FN1.

2. Cada atributo no principal, que no forma parte de la clave, tiene dependencia funcional completa de la clave principal.

Ejemplo:

PEDIDOS (CódigoProducto, CódigoEncargo, NombreProducto, Cantidad, FechaCompra)

Solamente nos hace falta tener informado CódigoProducto para obtener el NombreProducto.

CódigoProducto→ NombreProducto

No tenemos dependencia funcional completa, por tanto tampoco FN2.

Para arreglarlo tendríamos que dividir nuestra tabla en:

Dividimos en dos tablas, en cada una ponemos los atributos que sí tienen una dependencia funcional completa.

PRODUCTOS (CódigoProducto, NombreProducto)

ENCARGOS (CódigoProducto, CódigoEncargo, Cantidad, FechaCompra)

Ejemplo:

El DNI y el código de curso forman una clave principal, sólo la nota tiene dependencia funcional completa. El nombre y los apellidos dependen de forma completa del DNI. La tabla no es 2FN.

DNI	Cod Curso	Nombre	Apellido1	Nota
12121219A	34	Pedro	Valiente	9
12121219A	25	Pedro	Valiente	8
3457775G	34	Ana	Fernández	6
5674378J	25	Sara	Crespo	7
5674378J	34	Sara	Crespo	6

Para arreglarlo:

DNI	Nombre	Apellido1
12121219A	Pedro	Valiente
12121219A	Pedro	Valiente
3457775G	Ana	Fernández
5674378J	Sara	Crespo
5674378J	Sara	Crespo

DNI	Cod Curso	Nota
12121219A	**34**	**9**
12121219A	**25**	**8**
3457775G	**34**	**6**
5674378J	**25**	**7**
5674378J	**34**	**6**

4.3.3 FN3:

Se da cuando:

1. Es FN2.

2. Ningún atributo que no sea clave depende transitivamente de las claves de la tabla.

PRODUCTOS(CódigoProducto, Nombre, Fabricante, País)

CódigoProducto→ Fabricante

Fabricante→ País

CódigoProducto - →País

La Provincia depende funcionalmente del código de provincia, lo que hace que no esté en 3FN.

DNI	Nombre	Apellido1	Nota	Cod Provincia	Provincia
12121219A	Pedro	Valiente	9	39	Cantabria
12121219A	Pedro	Valiente	8	39	Cantabria
3457775G	Ana	Fernández	6	05	Ávila
5674378J	Sara	Crespo	7	05	Ávila
5674378J	Sara	Crespo	6	08	Barcelona

Para arreglarlo:

DNI	Nombre	Apellido1	Cod Provincia
12121219A	Pedro	Valiente	39
12121219A	Pedro	Valiente	39
3457775G	Ana	Fernández	05
5674378J	Sara	Crespo	05
5674378J	Sara	Crespo	08

Cod Provincia	Provincia
39	Cantabria
05	Ávila
08	Barcelona

4.3.4 FNBC (Boyce-Codd):

Caso especial de FN3:

Ocurre si la tabla está en FN3 y además, todo determinante es clave candidata.

Determinante: Atributo del que depende totalmente algún otro atributo.

NOTAS (DniAlumno, DniProfesor, NombreProfesor, Nota)

DniProfesor→ NombreProfesor

NombreProfesor→ DniProfesor

DniProfesor, DniAlumno→Nota

En este caso la tabla está en FN3 porque no hay dependencias transitivas, pero no está en FNBC porque el atributo NombreProfesor y DniProfesor son determinantes y no son claves candidatas de la tabla.

Para obtener la tabla en FNBC habría que quitar de la tabla los atributos DniProfesor y NombreProfesor.