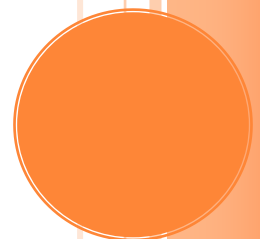


TEMA 5: NORMALIZACIÓN DE LAS RELACIONES

OBJETIVOS:

- Utilizar herramientas gráficas para representar el diseño lógico
- Aplicar las reglas de normalización hasta un nivel adecuado



NORMALIZACIÓN

Introducción:

Habitualmente, el diseño de una base de datos termina en el paso del modelo entidad-relación al modelo relacional. No obstante, siempre que se diseña un sistema, no sólo una base de datos, sino también cualquier tipo de solución informática, se ha de medir la calidad de la misma, y si no cumple determinados criterios de calidad, hay que realizar, de forma iterativa, sucesivos *refinamientos* en el diseño, para alcanzar la calidad deseada. Uno de los parámetros que mide la calidad de una base de datos es la *forma normal* en la que se encuentra su diseño. Esta forma normal puede alcanzarse cumpliendo ciertas restricciones que impone cada forma normal al conjunto de atributos de un diseño. El proceso de obligar a los atributos de un diseño a cumplir ciertas formas normales se llama *normalización*.

Deficiencias como existir datos ambiguos, anormales o disfuncionales, las relaciones sin tener sentido, y algunas otras causas pueden llevarnos a obtener un **Modelo de Datos** desnormalizado, con el cual la aplicación a construir basada en el mismo arrastrará los errores de análisis y diseño; muchas veces, sin otra solución que la de rehacer el Modelo de Datos.

La solución a este tema, es abordada en este capítulo y brinda los conocimientos teóricos más importantes para obtener un Modelo Normalizado y refinado, pero queda en el sentido común del diseñador gran parte de este trabajo.

La experiencia práctica de los diseñadores ha demostrado que cuando las estructuras de datos están normalizadas, el modelo de datos que de ellos deriva es más estable y permite acomodarse mejor a los cambios y ajustes.

Definición:

Normalización es un proceso de refinamiento de la estructura de datos de una Organización / sistema / negocio que consiste en la agrupación de los datos según ciertas reglas o "normas".

¿Por qué normalizamos?

La normalización minimiza la REDUNDANCIA de datos. Los datos redundantes causan problemas de integridad y consistencia de la información.

Normalizar, ayuda a identificar a entidades que en realidad no son más que atributos, identificar relaciones sin sentido, reclasificar a los atributos como obligatorios, no obligatorios, claves primarias, secundarias, etc.

Las formas normales pretenden alcanzar dos objetivos:

1. Almacenar en la base de datos cada hecho sólo una vez, es decir, evitar la redundancia de datos. De esta manera se reduce el espacio de almacenamiento.
2. Que los hechos distintos se almacenen en sitios distintos. Esto evita ciertas anomalías a la hora de operar con los datos.

En la medida que se alcanza una forma normal más avanzada, en mayor medida se cumplen estos objetivos. Hay definidas 6 formas normales, cada una agrupa a las anteriores, de forma que, por ejemplo, la forma normal 3 cumple la forma normal 2 y la forma normal 1.

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

Dependencia funcional: Se dice que un atributo Y depende funcionalmente de otro atributo X o que $X \rightarrow Y$, si cada valor de X tiene asociado en todo momento un único valor de Y. También se dice que X implica Y, y por tanto, que X es el *implicante*.

Por ejemplo:

PRODUCTOS (CódigoProducto, Nombre, Precio, Descripción)

CódigoProducto \rightarrow nombre, puesto que un código de producto sólo puede tener asociado un único nombre, dicho de otro modo, a través del código de producto se localiza un único nombre.

Dependencia funcional completa: Dado una combinación de atributos X (X_1, X_2, \dots) se dice que Y tiene dependencia funcional completa de X, o que $X \twoheadrightarrow Y$, si depende funcionalmente de X, pero no depende de ningún subconjunto del mismo.

Por ejemplo:

COMPRAS (CódigoProducto, CódigoProveedor, Cantidad, FechaCompra)

CódigoProducto, CódigoProveedor \twoheadrightarrow FechaCompra, puesto que la FechaCompra es única para la combinación de CódigoProducto y CódigoProveedor (se puede hacer un pedido al día de cada producto a cada proveedor), y sin embargo, se pueden hacer varios pedidos del mismo producto a diferentes proveedores, es decir, CódigoProducto $\not\rightarrow$ Fecha.

Dependencia funcional transitiva: Dada la tabla T, con atributos (X, Y, Z), donde $X \rightarrow Y$, $Y \rightarrow Z$ e $Y \not\rightarrow X$, se dice que X depende transitivamente de Z que, $X \twoheadrightarrow Z$.

Ejemplo 1:

PRODUCTOS (CódigoProducto, Nombre, Fabricante, País)

CódigoProducto \rightarrow Fabricante

Fabricante \rightarrow País

CódigoProducto \rightarrow País, es decir, CódigoProducto depende transitivamente de País.

Formas Normales:

La Normalización se discute habitualmente en términos de formas: la primera, la segunda y la tercera forma normal, son las más comunes, donde la tercera representa el estado más normalizado. Algunas aplicaciones complejas pueden llegar hasta la cuarta forma, la quinta forma, o la forma normal "Boyce-Codd". Existen también otras formas, pero son objetivo de este curso únicamente las primeras tres formas normales nombradas.

Primera Forma Normal:

"Una entidad no puede contener grupos repetitivos"

Se deberán identificar los grupos repetitivos y crear una nueva estructura para cada grupo repetitivo. Esto se hace poniendo los datos en tablas separadas de manera que los datos de cada tabla son de un tipo similar y dando a cada tabla una clave principal (PK).

En esta forma normal se prohíbe que en una tabla haya atributos que puedan tomar más un valor. Esta forma normal es inherente al modelo relacional, puesto que las tablas gestionadas por un SGBD relacional, están construidas de esta forma

Segunda Forma Normal:

"Siempre una columna NO UID debe ser dependiente únicamente de todas las partes del UID (PK)".

Un diseño se encuentra en FN2 si está en FN1 y además, cada atributo que no forma parte de la clave tiene dependencia completa de la clave principal.

Ejemplo:

COMPRAS (CódigoProducto, CódigoProveedor, NombreProducto, Cantidad, FechaCompra).

CódigoProducto → NombreProducto, por tanto, al no ser dependencia funcional completa, no está en FN2.

La segunda forma, debe analizarse si la clave principal es compuesta por dos o más atributos (PK Composite o compuesta).

La solución a esta forma es identificar los grupos con claves concatenadas. Si cualquiera de los atributos no depende de la totalidad de la clave concatenada crear una o más estructuras nuevas para estos atributos.

La clave para estas nuevas estructuras se extrae de la clave concatenada del grupo original.

En esta forma se pretende eliminar dependencias parciales.

Tercera Forma Normal:

"Una columna NO UID debe ser solamente dependiente del UID".

Un diseño se encuentra en FN3 si está en FN2 y además, no hay ningún atributo no clave que depende de forma transitiva de la clave.

Ejemplo:

PRODUCTOS (CódigoProducto, Nombre, Fabricante, País).

CódigoProducto → Fabricante

Fabricante → País

CódigoProducto --//→ País

País depende transitivamente de CódigoProducto, por tanto, no está en tercera forma normal.

Se deberán examinar todas las estructuras y si cualquier atributo depende de otro que no es clave, crear estructuras separadas para estos atributos.

La clave de esta nueva estructura es el atributo del cual depende. Aquí se pretende eliminar las dependencias transitivas.

FNBC: Esta forma normal, llamada Forma Normal de Boyce-Codd, exige que el modelo esté en FN3, y que además, todo implicante de la tabla, sea una clave candidata.

Ejemplo:

NOTAS (DNIAlumno, DNIProfesor, NombreProfesor, Nota).

DNIProfesor → NombreProfesor

NombreProfesor → DNIProfesor

DNIProfesor, DNIAlumno → Nota

En este caso, la tabla está en 3FN porque no hay dependencias funcionales transitivas, y sin embargo no está en FNBC porque NombreProfesor y DNIProfesor son implicantes, y no son claves candidatas. Para obtener la tabla en FNBC habría que quitar de la tabla los atributos DNIProfesor y Nombreprofesor.

Otras formas normales: Existen más formas normales (FN4, F 5, FNDK, FN6) cuyo alcance excede el de este tema y cuya aplicación en el mundo real es únicamente teórica). Las

formas normales 4 y 5, se ocupan de las dependencias entre atributos multivaluados, la Forma Normal Dominio Clave (FNDK) trata las restricciones y los dominios de los atributos, y finalmente la FN6 trata ciertas consideraciones con las bases de datos temporales.

Sintetizando: para que se cumplan las tres formas normales anteriores:

"Todo atributo NO UID debe depender de la UID, de toda la UID, y de nada más que la UID".

Una vez normalizado el Modelo de Datos; será sencillo ver como el usuario podrá desplazarse de una tabla a la otra obteniendo mediante las relaciones adecuadas la información precisa que requería.

Tenga en cuenta que normalizar no es sinónimo de confiabilidad, ya que garantiza integridad y elimina redundancia pero no asegura siempre una buena performance de las aplicaciones.

Ejemplo de Normalización:

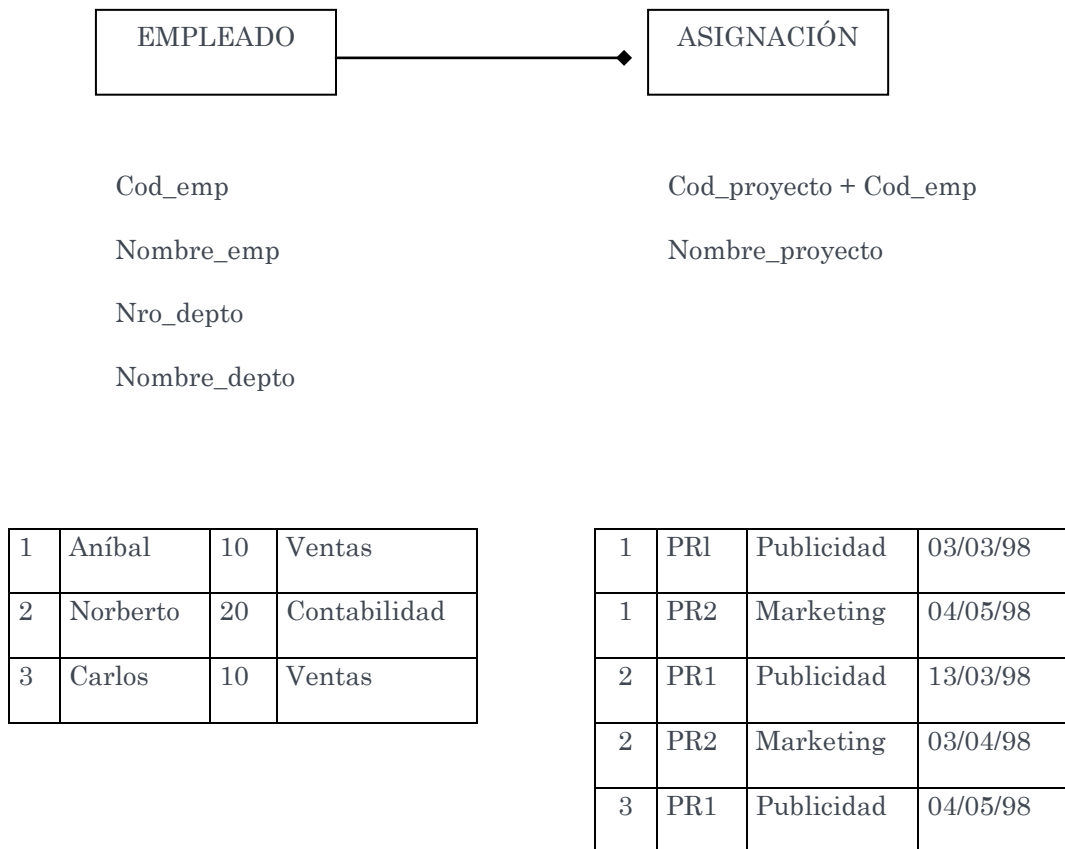
Dada la siguiente entidad; verifique que la misma este en 3ra. Forma normal.

EMPLEADO	Cod_emp
	Nombre_emp
	Nro_depto
	Nombre_depto
	Cod_proyecto
	Nombre_proyecto
	Fecha_Asignación

1	Aníbal	10	Ventas	PR1	Publicidad	03/03/98
1	Aníbal	10	Ventas	PR2	Marketing	04/05/98
2	Norberto	20	Contabilidad	PR1	Publicidad	13/03/98
2	Norberto	20	Contabilidad	PR2	Marketing	03/04/98
3	Carlos	10	Ventas	PR1	Publicidad	04/05/98

Se visualiza que el grupo de datos dado viola la primera forma normal; ya que existen grupos repetitivos. Puede afirmarse que está en **forma normal cero**.

Llevando el esquema anterior a **primera forma normal**, resulta:



Como se observa, el atributo Nombre_proyecto está relacionado con una parte del UID (Cod_proyecto) y no con Cod_emp (que también pertenece al UID Compuesto), **con lo cual la entidad que llamamos asignación está violando la segunda forma normal.**

Si existe PR1 → existe Publicidad (DEPENDENCIA FUNCIONAL)

Luego, el siguiente esquema está en **segunda forma normal**:

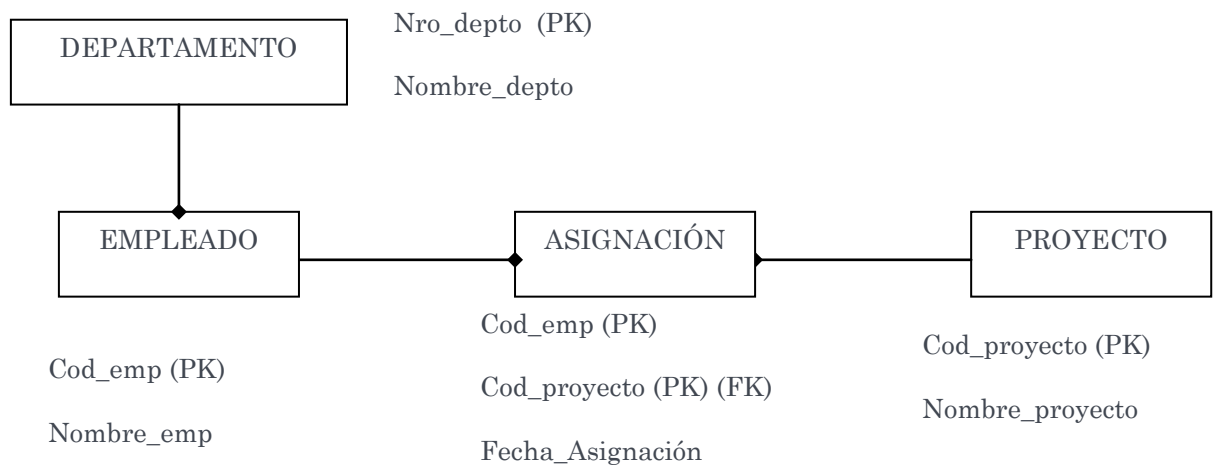


Pero, si analizamos la entidad EMPLEADO:

1	Aníbal	10	Ventas
2	Norberto	20	Contabilidad
3	Carlos	10	Ventas

Nos damos cuenta de que Nombre_depto que es un atributo NO UID depende de Nro_depto pero no de Cod_emp (que es el UID de la entidad) **violando de esta forma la tercera forma normal**.

Resolviendo :



Quedando de esta forma el MR NORMALIZADO en **tercera forma normal**.