**BASES DE DATOS Normalizadas**

El diseño de una BD relacional se puede realizar aplicando al mundo real, en una primera fase, un modelo como el modelo E/R, a fin de obtener un esquema conceptual; en una segunda fase, se transforma dicho esquema al modelo relacional mediante las correspondientes reglas de transformación como henos visto hasta ahora.

Una vez obtenido el esquema relacional es necesario aplicar un conjunto de reglas, conocidas como Teoría de Normalización, que nos permiten asegurar que un esquema relacional cumple unas ciertas propiedades, de forma que evite:

• La redundancia de los datos: repetición de datos en un sistema.

• Anomalías de actualización: inconsistencias de los datos como resultado de datos redundantes y actualizaciones parciales.

• Anomalías de borrado: pérdidas no intencionadas de datos debido a que se han borrado otros datos.

• Anomalías de inserción: imposibilidad de adicionar datos en la base de datos debido a la ausencia de otros datos.

En la práctica, si la BD se ha diseñado haciendo uso del modelo E/R ya suele cumplir las formas normales.

La normalización es un proceso que consiste en comprobar que las tablas (relaciones en terminología propia del modelo relacional de datos) definidas cumplen unas determinadas condiciones. Mediante la normalización se pueden solucionar diversos errores en el diseño de la base de datos, así como mejorarlo. También se facilita el trabajo posterior del administrador de la base de datos y de los desarrolladores de aplicaciones.

Los pasos fundamentales en el desarrollo de una base de datos serían:

● El Modelo conceptual: Entidad-Relación

● Modelo lógico

○ Modelo Relacional

○ Normalización

○ Optimización

● Modelo físico

**Normalización de modelos relacionales**

El punto de partida del proceso de normalización es un conjunto de tablas con sus atributos, el denominado esquema relacional. Se pretende mejorar dicho esquema de datos. Se dice que una tabla está en una determinada forma normal si satisface un cierto número de restricciones impuestas por la correspondiente regla de normalización.

Si una tabla no satisface una determinada regla de normalización, se procede a descomponerla en otras dos nuevas que sí las satisfagan. Esto usualmente requiere decidir qué atributos de la tabla original van a residir en una u otra de las nuevas tablas. La descomposición tiene que conservar una propiedad fundamental que es la no pérdida de información.

La teoría de la normalización se basa en las Formas Normales.

Formas Normales son una serie de restricciones que el diseño deberá cumplir para satisfacer dichas formas.

**Dependencia funcional**

Una vez diseñado el modelo relacional a partir del modelo ER o ER-E, el siguiente paso es buscar las dependencias funcionales (df) entre atributos para comprobar las FN.

El concepto de dependencia funcional es que se debe conocer un dato concreto para llegar a otro.

Un atributo Y se dice que depende funcionalmente de otro X si, y sólo si, a cada valor de X le corresponde un único valor de Y, expresa de la siguiente forma: X → Y (también se dice que X implica a Y).

Ejemplo: Si A es el DNI, y B el Nombre, está claro que para un número de D.N.I, siempre aparece el mismo nombre de titular.

Cuando se analiza el nivel de redundancia de un esquema de relación es posible considerar diferentes tipos de dependencias funcionales de los datos.

El conocimiento de las reglas de negocio es necesario para identificar las dependencias que se deben reflejar en el esquema; es decir, tener en cuenta las restricciones de datos que se presentan cotidianamente.

**Normalización - Formas normales**

Edgar F. Codd en 1970 definió las tres primeras formas normales (1FN, 2FN, y 3FN).

**Estas formas normales se han resumido deforma que una tabla cumple la 1,2 y 3 forma normal si cumples estas tres condiciones:**

**-Todos los atributos son atómicos.**

**-Dependen de la clave completa.**

**-Dependen de la clave en forma directa (no transitiva)**.

**1ª Forma Normal - 1FN**

Una relación está en 1FN si todos los atributos contienen valores atómicos (tomados de un único dominio y con un único valor). En cada fila de la tabla no va a haber más de un valor por cada columna. Puede contener valor cero o Null.

Ejemplo: Supongamos que tenemos la siguiente tabla con datos de alumnado de un centro de enseñanza secundaria.



Como se puede observar, esta tabla no está en 1FN puesto que el campo Teléfonos contiene varios datos dentro de una misma celda y por tanto no es un campo cuyos valores sean atómicos.

Esto lo solucionamos descomponiendo en dos tablas. Dejamos la tabla principal y creamos otra tabla con el atributo multivaluado y la clave de la primera tabla. En esta nueva tabla la clave primaria será la Pk de la tabla primera junto con el atributo multivaluado.

**Alumnos** (**DNI**, Nombre, Curso, Fecha-Matricula, Tutor, LocalidadAlumno, ProvinciaAlumno)

**Telefonos\_Alumnos** (**DNI, Telefono)**



**2ª Forma Normal - 2FN**

Esta forma normal sólo debe ser considerada para aquellas tablas en las que alguna de sus claves candidatas sean compuestas. Una tabla está en segunda forma normal si se cumple:

● Está en 1FN

● Todo atributo secundario (que no pertenezca a la clave principal) tiene una dependencia funcional total de la clave principal, y no de una parte de ella.

Así se pretende garantizar una correcta elección de claves y eliminar redundancias. Si las claves están formadas por un único atributo entonces ese esquema estará seguro en segunda forma normal.

Ejemplo: Seguimos con el ejemplo anterior. Trabajaremos con la tabla obtenida una vez eliminado el atributo multivaluado:



Dependencias funcionales:

Los atributos Nombre y LocalidadAlumno no dependen funcionalmente de Curso, y el atributo Tutor no depende funcionalmente de DNI.

Siempre que aparece un Curso aparecerá el Tutor correspondiente.

Por tanto Curso → Tutor. El único atributo que sí depende de forma completa de la clave compuesta DNI y Curso es FechaMatrícula:

(DNI,Curso) → FechaMatrícula.

A la hora de establecer la Clave Primaria de una tabla debemos escoger un atributo o conjunto de ellos de los que dependan funcionalmente el resto de atributos. Además debe ser una dependencia funcional completa.

Así, aquí la clave sería (Dni, Curso) que nos permite saber todos los atributos.



La posible descomposición sería:

**Alumnos** (**DNI**, Nombre, LocalidadAlumno, ProvinciaAlumno)

**Matriculas** (**DNI, Curso**, Fecha-Matricula)

**Cursos** (**Curso**, Tutor)



**3ª Forma Normal - 3FN**

Una tabla está en tercera forma normal si se cumplen

● Está en 2FN

● No existen atributos no clave que sean transitivamente dependientes de cada posible clave de la tabla. Es decir, un atributo secundario sólo puede ser conocido a través de la clave principal y no por medio de un atributo no primario

Una relación está en 3FN si y sólo si está en 2FN y no existen dependencias transitivas. Todas las dependencias funcionales deben ser respecto a la clave principal.

3ª Forma Normal - 3FN

Ejemplo: Seguimos con el ejemplo anterior.

Para que la tabla esté en 3FN no pueden existir dependencias funcionales transitivas.

Para solucionar el problema se debe crear una nueva tabla.



La tabla

**Alumnos** (**DNI**, Nombre, LocalidadAlumno, ProvinciaAlumno)

No está en tercera forma normal, ya que Provincia no depende del DNI (la clave), si no que depende de localidad.



Por eso pasamos a su descomposición como

Alumnos (**DNI**, Nombre, Localidad)

Provincia (**Localidad**, Provincia)

|  |  |
| --- | --- |
| **Localidad** | **Provincia** |
| Ècija | Sevilla |
| El Villar | Córdoba |

**Resultado final:**

La tabla Alumnos la descomponemos en:

|  |  |  |
| --- | --- | --- |
| **Dni** | **Nombre** | **Localidad** |
| 11111111A | Eva | Écija |
| 22222222B | Ana | Écija |
| 3333333C | Susana | El Villar |
| 44444444D | Juan | El Villar |
| 55555555E | José | Ècija |

|  |  |  |
| --- | --- | --- |
| **Dni** | **Curso** | **FechaMatricula** |
| 11111111A | 1ESO-A | 01-Julio-2016 |
| 22222222B | 1ESO-A | 09-Julio-2016 |
| 33333333C | 1ESO-B | 11-Julio-2016 |
| 44444444D | 2ESO-A | 05-Julio-2016 |
| 55555555E | 2ESO-A | 02-Julio-2016 |

|  |  |
| --- | --- |
| **Dni** | **Telefono** |
| 11111111A | 660111222 |
| 22222222B | 660222333 |
| 22222222B | 660333444 |
| 22222222B | 660444555 |
| 55555555E | 660000111 |
| 55555555E | 660000222 |

|  |  |
| --- | --- |
| **Localidad** | **Provincia** |
| Ècija | Sevilla |
| El Villar | Córdoba |

Todas estas tablas están en 1,2 y 3ª FN.