BASES DE DATOS
TEMA 4
DISEÑO DE BASES DE DATOS RELACIONALES

4.3 Normalización

Uno de los objetivos de una estructura de tabla normalizada es minimizar el número de "celdas vacías". Grupos de información son almacenados en distintas tablas que luego pueden ser "juntadas" (relacionadas) basándose en los datos que tengan en común.

Es necesario que al realizar la estructura de una base de datos, esta sea flexible. La **flexibilidad** está en el hecho que se puedan agregar datos al sistema posteriormente sin tener que rescribir lo que ya se tiene. Por lo tanto, no tendremos que modificar la estructura de nuestras tablas actuales, simplemente agregar lo que hace falta.

La **eficiencia** se refiere al hecho de que no se tiene duplicación de datos, y tampoco tienen grandes cantidades de "celdas vacías". El objetivo principal del diseño de bases de datos es generar tablas que modelan los registros en los que se guardara la información. Es importante que esta información se almacene sin redundancia para que se pueda tener una recuperación rápida y eficiente de los datos.

A través de la normalización se trata de evitar defectos que conduzcan a un mal diseño y que lleven a un procesamiento menos eficaz de los datos. Se puede decir que los principales objetivos de la normalización son:

- ✓ Controlar la redundancia de la información.
- ✓ Evitar pérdidas de información.
- ✓ Capacidad para representar toda la información.
- ✓ Mantener la consistencia de los datos.

La **normalización** es una técnica para diseñar la estructura lógica de los datos de un sistema de información en el modelo relacional, desarrollada por E. F. Codd en 1972. Es una estrategia de diseño de abajo a arriba: se parte de los atributos y éstos se van agrupando en relaciones (tablas) según su afinidad. Es como una etapa posterior a la correspondencia entre el esquema conceptual y el esquema lógico, que elimine las dependencias entre atributos no deseadas. Las ventajas de la normalización son las siguientes:

- ✓ Evita anomalías en inserciones, modificaciones y borrados.
- ✓ Mejora la independencia de datos.
- ✓ No establece restricciones artificiales en la estructura de los datos.

Uno de los conceptos fundamentales en la normalización es el de dependencia funcional. Una **dependencia funcional** es una relación entre atributos de una misma relación (tabla). La dependencia funcional es una noción semántica. Si hay o no dependencias funcionales entre atributos lo determinan los modelos mentales del usuario y las reglas de negocio de la organización o empresa para la que se desarrolla el sistema de información. Cada dependencia funcional es una clase especial de regla de integridad y representa una relación de uno a muchos.

En el proceso de normalización se debe ir comprobando que cada relación (tabla) cumpla una serie de reglas que se basan en la clave primaria y las dependencias funcionales. Cada regla que se cumple aumenta el grado de normalización. Si una regla no se cumple, la relación se debe descomponer en varias relaciones que sí la cumplan.

La normalización se lleva a cabo en una serie de pasos. Cada paso corresponde a una forma normal que tiene unas propiedades. Conforme se va avanzando en la normalización, las relaciones tienen un formato más estricto (más fuerte) y, por lo tanto, son menos vulnerables a las anomalías de actualización. El modelo relacional sólo requiere un conjunto

de relaciones en primera forma normal. Las restantes formas normales son opcionales. Sin embargo, para evitar las anomalías de actualización, es recomendable llegar al menos a la tercera forma normal. Uno de los retos en el diseño de la base de datos es el de obtener una estructura estable y lógica tal que:

1. El sistema de base de datos no sufra de anomalías de almacenamiento.
2. El modelo lógico pueda modificarse fácilmente para admitir nuevos requerimientos.

Una DB implantada sobre un modelo bien diseñado tiene mayor esperanza de vida aun en un ambiente dinámico, que una DB con un diseño pobre. En promedio, una DB datos experimenta una reorganización general cada seis años, dependiendo de lo dinámico de los requerimientos de los usuarios. Una DB bien diseñada tendrá un buen desempeño aunque aumente su tamaño, y será lo suficientemente flexible para incorporar nuevos requerimientos o características adicionales.

Existen diversos riesgos en el diseño de las DB relacionales que afecten la funcionalidad de la misma, los riesgos generalmente son la redundancia de información y la inconsistencia de datos. La normalización es el proceso de simplificar la relación entre los campos de un registro. Por medio de está un conjunto de datos en un registro se reemplaza por varios registros que son más simples y predecibles y, por lo tanto, más manejables. La normalización se lleva a cabo por cuatro razones:

1. Estructurar los datos de forma que se puedan representar las relaciones pertinentes entre los datos.
2. Permitir la recuperación sencilla de los datos en respuesta a las solicitudes de consultas y reportes.
3. Simplificar el mantenimiento de los datos actualizándolos, insertándolos y borrándolos.
4. Reducir la necesidad de reestructurar o reorganizar los datos cuando surjan nuevas aplicaciones.

En términos más sencillos la normalización trata de simplificar el diseño de una base de datos, esto a través de la búsqueda de la mejor estructuración que pueda utilizarse con las entidades involucradas en ella.

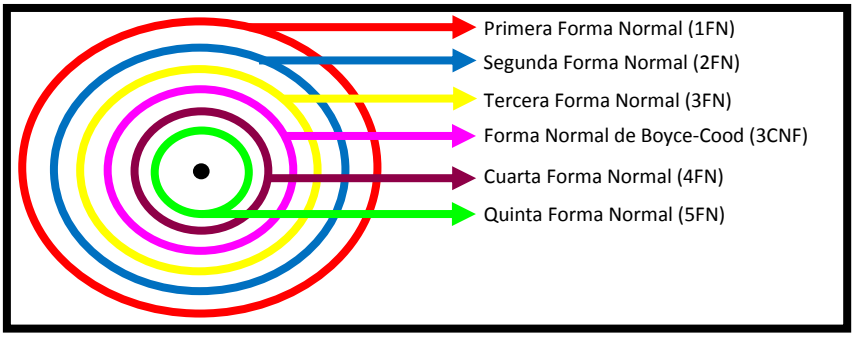
Pasos de la normalización:

1. Descomponer todos los grupos de datos en registros bidimensionales.
2. Eliminar todas las relaciones en la que los datos no dependan completamente de la llave primaria del registro.
3. Eliminar todas las relaciones que contengan dependencias transitivas.

La teoría de normalización tiene como fundamento el concepto de formas normales; se dice que una relación está en una determinada **forma normal** si satisface un conjunto de restricciones.

Formas normales.

Son las técnicas para prevenir las anomalías en las tablas. Dependiendo de su estructura, una tabla puede estar en primera forma normal, segunda forma normal o en cualquier otra. Relación entre las formas normales:



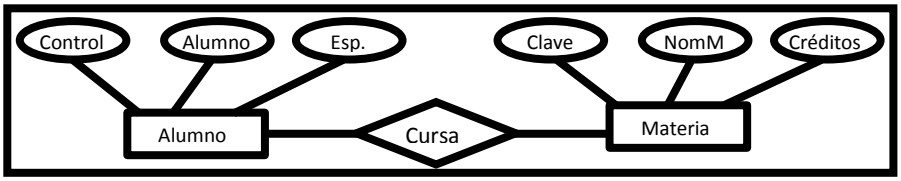
Elaboro: LIA. Blanca García Sánchez

4.3.1 1NF Primera forma normal.

Una relación R se encuentra en 1FN si y solo sí por cada renglón columna contiene valores atómicos. Abreviada como 1FN, se considera que una relación se encuentra en la primera forma normal cuando cumple lo siguiente:

1. 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.
2. Todos los ingresos en cualquier columna (atributo) deben ser del mismo tipo.
3. Cada columna debe tener un nombre único, el orden de las columnas en la tabla no es importante.
4. Dos filas o renglones de una misma tabla no deben ser idénticas, aunque el orden de las filas no es importante.

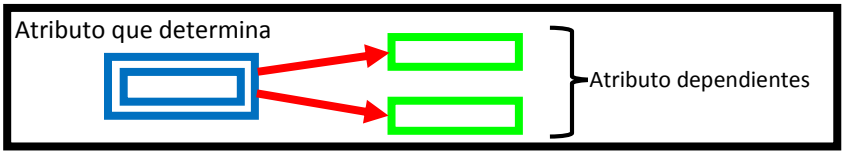
Por lo general la mayoría de las relaciones cumplen con estas características, así que podemos decir que la mayoría de las relaciones se encuentran en la primera forma normal.



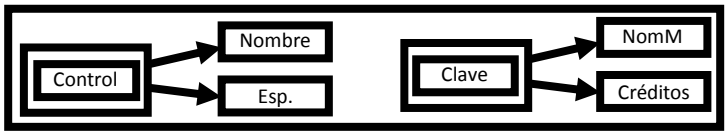
Como esta relación maneja valores atómicos, es decir un solo valor por cada uno de los campos que conforman a los atributos de las entidades, ya se encuentra en primera forma normal, gráficamente así representamos a las relaciones en 1FN.

4.3.2 2NF Segunda forma normal.

Consiste en edificar que atributos dependen de otro(s) atributo(s).



Una relación R está en 2FN si y solo si está en 1FN y los atributos no primos dependen funcionalmente de la llave primaria. Una relación se encuentra en segunda forma normal, cuando cumple con las reglas de la primera forma normal y todos sus atributos que no son claves (llaves) dependen por completo de la clave. De acuerdo con esta definición, cada tabla que tiene un atributo único como clave, está en segunda forma normal. La segunda forma normal se representa por dependencias funcionales como:

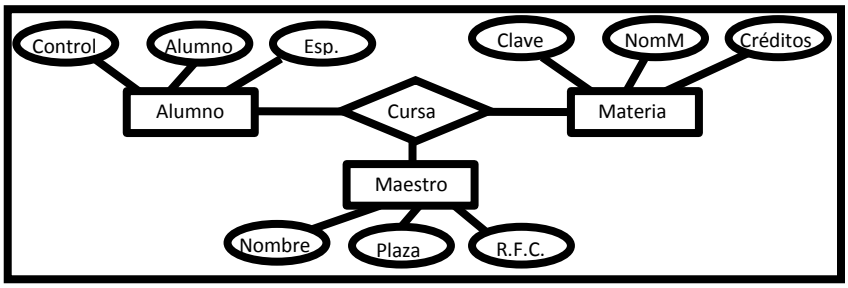


Las llaves primarias están representadas con doble cuadro, las flechas indican que de estos atributos se puede referenciar a los otros atributos que dependen funcionalmente de la llave primaria.

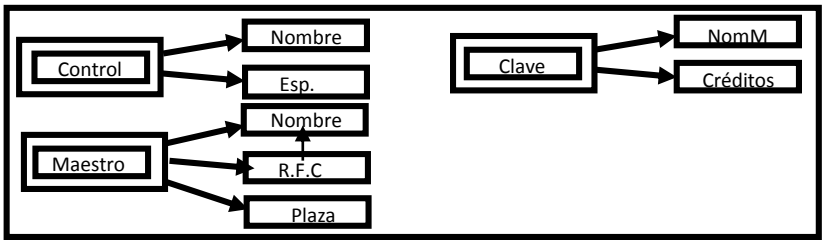
4.3.3 3NF Tercera forma normal.

Para definir formalmente la 3FN se requiere definir dependencia transitiva: En una afinidad (tabla bidimensional) que tiene por lo menos 3 atributos (A,B,C) en donde A determina a B, B determina a C pero no determina a A.

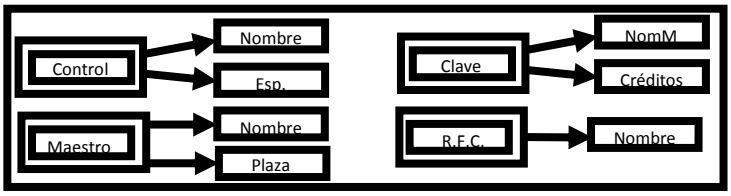
Una relación R está en 3FN si y solo si está en 2FN y todos sus atributos no primos dependen no transitivamente de la llave primaria. Consiste en eliminar la dependencia transitiva que queda en una segunda forma normal, en pocas palabras una relación está en tercera forma normal si está en segunda forma normal y no existen dependencias transitivas entre los atributos, **dependencias transitivas** es cuando existe más de una forma de llegar a referencias a un atributo de una relación.



Se tiene la relación alumno-cursa-materia manejada anteriormente, pero ahora considerando al elemento maestro, gráficamente se puede representar de la siguiente manera:



Se encuentra graficado en segunda forma normal, es decir que todos los atributos llave están indicados en doble cuadro indicando los atributos que dependen de dichas llaves, sin embargo en la llave Maestro tiene como dependientes a 3 atributos en el cual el nombre puede ser referenciado por dos atributos: Maestro y RFC (Existe dependencia transitiva), se puede solucionar esto aplicando la tercera forma normal que consiste en eliminar las dependencias separando los atributos:



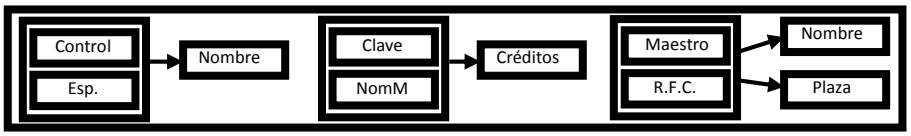
4.3.4 BCNF Forma normal de Boyce-Cood.

Una relación R está en FNBC si y solo si cada determinante es una llave candidato. Determinado que una DB es una colección de archivos interrelacionados creados con un DBMS. El contenido de una DB esta almacenada de tal manera que los datos estén disponibles para los usuarios, una finalidad de la DB es eliminar la redundancia o al menos minimizarla.

La DB es solo un “almacén” de datos, lo que ha hecho indispensable el desarrollo de sistemas que los administren y procesen, siendo estos los DBMS.

El propósito general de los DBMS es el de manejar de manera clara, sencilla y ordenada, los datos de una DB que posteriormente se convertirán en información relevante, para un buen manejo de los datos.

Denominada por sus siglas en inglés como BCNF; Una tabla se considera en esta forma si y sólo sí cada determinante o atributo es una llave candidato. Continuando con el ejemplo anterior, considerando que en la entidad alumno sus atributos control y nombre pueden hacer referencia al atributo esp., entonces dichos atributos pueden ser llaves candidato. Gráficamente así se puede representar la forma normal de Boyce Codd:



A diferencia de la tercera forma normal, se agrupan todas las llaves candidato para formar una global (representadas en el recuadro) las cuales hacen referencia a los atributos que no son llaves candidato.

4.4 Criterios para normalización

Codd se percató de que existían bases de datos en el mercado las cuales decían ser relacionales, pero lo único que hacían era guardar la información en las tablas, sin estar estas tablas literalmente normalizadas; entonces éste publicó 12 reglas que un verdadero sistema relacional debería tener, en la práctica algunas de ellas son difíciles de realizar. Un sistema podrá considerarse "más relacional" cuanto más siga estas reglas.

Regla No. 1 - La Regla de la información

Toda la información en un RDBMS está explícitamente representada de una sola manera por valores en una tabla. Cualquier cosa que no exista en una tabla no existe del todo. Toda la información, incluyendo nombres de tablas, nombres de vistas, nombres de columnas, y los datos de las columnas deben estar almacenados en tablas dentro de las DB. Las tablas que contienen tal información constituyen el Diccionario de Datos. Esto significa que todo tiene que estar almacenado en las tablas.

Regla No. 2 - La regla del acceso garantizado

Cada ítem de datos debe ser lógicamente accesible al ejecutar una búsqueda que combine el nombre de la tabla, su clave primaria, y el nombre de la columna. Esto significa que dado un nombre de tabla, dado el valor de la clave primaria, y dado el nombre de la columna requerida, deberá encontrarse uno y solamente un valor. Por esta razón la definición de claves primarias para todas las tablas es prácticamente obligatoria.

Regla No. 3 - Tratamiento sistemático de los valores nulos

La información inaplicable o faltante puede ser representada a través de valores nulos.

Regla No. 4 - La regla de la descripción de la base de datos

La descripción de la base de datos es almacenada de la misma manera que los datos ordinarios, esto es, en tablas y columnas, y debe ser accesible a los usuarios autorizados.

La información de tablas, vistas, permisos de acceso de usuarios autorizados, etc., debe ser almacenada exactamente de la misma manera.

Regla No. 5 - La regla del sub-lenguaje Integral

Debe haber al menos un lenguaje que sea integral para soportar la definición de datos, manipulación de datos, definición de vistas, restricciones de integridad, y control de autorizaciones y transacciones.

Regla No. 6 - La regla de la actualización de vistas

Todas las vistas que son teóricamente actualizables, deben ser actualizables por el sistema mismo. La mayoría de las RDBMS permiten actualizar vistas simples, pero deshabilitan los intentos de actualizar vistas complejas.

Regla No. 7 - La regla de insertar y actualizar

La capacidad de manejar una base de datos con operandos simples aplica no sólo para la recuperación o consulta de datos, sino también para la inserción, actualización y borrado de datos.

Regla No. 8 - La regla de independencia física

El acceso de usuarios a la base de datos a través de terminales o programas de aplicación, debe permanecer consistente lógicamente cuando quiera que haya cambios en los datos almacenados, o sean cambiados los métodos de acceso a los datos.

Regla No. 9 - La regla de independencia lógica

Los programas de aplicación y las actividades de acceso por terminal deben permanecer lógicamente inalteradas cuando quiera que se hagan cambios (según los permisos asignados) en las tablas de la base de datos.

Regla No. 10 - La regla de la independencia de la integridad

Todas las restricciones de integridad deben ser definibles en los datos, y almacenables en el catálogo, no en el programa de aplicación. Las reglas de integridad son:

1. Ningún componente de una clave primaria puede tener valores en blanco o nulos.
2. Para cada valor de clave foránea deberá existir un valor de clave primaria concordante.
3.

Regla No. 11 - La regla de la distribución

El sistema debe poseer un lenguaje de datos que pueda soportar que la base de datos esté distribuida físicamente en distintos lugares sin que esto afecte o altere a los programas de aplicación.

Regla No. 12 - Regla de la no-subversión

Si el sistema tiene lenguajes de bajo nivel, estos lenguajes de ninguna manera pueden ser usados para violar la integridad de las reglas y restricciones expresadas en un lenguaje de alto nivel.