NORMAS DE DISEÑO DE BASE DE DATOS

Julio 2014

ESTRUCTURA DEL DOCUMENTO.

1 CONSIDERACIONES GENERALES. 3

2 MODELO CONCEPTUAL DE DATOS (MCD) 3

2.1 PROPIEDADES 4
2.2 ENTIDADES 4
2.3 CLAVE DE UNA ENTIDAD 5
2.4 RELACIONES 5
2.5 IDENTIFICADOR DE UNA RELACIÓN 6
2.6 CARDINALIDADES 7
2.7 INTEGRIDAD REFERENCIAL 7

3 NORMALIZACIÓN DEL MODELO CONCEPTUAL 7

4 TRANSFORMACIÓN DEL MCD EN UN MODELO LÓGICO DE DATOS (MLD) 8

4.1 DEFINICIÓN DE UN MLD 8
4.2 TRASPASO DE ENTIDADES A TABLAS 8
4.3 TRASPASO DE RELACIONES M:N A TABLAS 8
4.4 TRASPASO DE RELACIONES 1:N A TABLAS 8

5 ELEMENTOS DE DATOS DB2 9

5.1 DEFINICIÓN DE ELEMENTOS 9
5.2 DATOS ESPECIALES (DATE, TIME, VARCHAR....) 9

1 CONSIDERACIONES GENERALES.

Disponibilidad del Entorno - 24 horas, 365 días

El concepto de un entorno disponible constantemente implica el desarrollo de software que potencie al máximo la compatibilidad entre procesos on-line y minimice las necesidades para 'ventanas Batch' o de 'Down-time' para las reorganizaciones u otras utilidades.

Filosofía de diseño de las bases de datos

Toda aplicación nueva debería tener en consideración los requerimientos de Multi-entidad, Multi-divisa y Multi-lenguaje. Estos obviamente influyen en el diseño de las bases de datos.

Filosofía de mantenimiento de las bases de datos

Cuando se genere una tabla DB2 o cualquier fichero estable debería diseñarse un procedimiento (batch) o una transacción (on-line) mediante la cual se pueda actualizar el fichero.

Tratamiento de datos históricos

Toda aplicación susceptible de tener históricos, debería, en tiempo de diseño, contemplar los procedimientos de carga de estos así como los procedimientos de recuperación de la información en históricos.

Errores controlados por programa

El tratamiento de errores controlados por programa debería contemplar que, en todas las ocasiones que fuera posible, el programa debe continuar, tras grabar un log con la información necesaria para solucionar el problema.

2 MODELO CONCEPTUAL DE DATOS (MCD)

El objetivo perseguido al elaborar un Modelo Conceptual de Datos (MCD) es obtener un conjunto de datos interdependientes:

Sin redundancia

Sin dependencia entre datos y aplicación

Común a todos los usuarios.

Es evidente que para diseñar un conjunto de datos que responda a estos objetivos es necesario prescindir en un primer paso de los tratamientos particulares de los usuarios, e intentar organizar los datos únicamente en función de su "naturaleza intrínseca".

La metodología que se resume a continuación se basa en los trabajos de E.F. CODD y sus definiciones de las distintas Formas Normales.

El Modelo Conceptual de Datos es la representación gráfica y descriptiva de una parte de la Base de Datos de la empresa u organización. Su función es proporcionar una visión global y esquemática de las Entidades,

Relaciones y Propiedades que forman parte del subconjunto de los datos de estas empresas visto desde un proyecto o aplicación.

El MCD lo integran los siguientes documentos:-

El modelo gráfico, que representa los vínculos entre las entidades a través de sus relaciones, con una indicación de su cardinalidad y de la naturaleza de la dependencia entre las entidades.

Las descripciones de las Entidades que componen la BD. Debe existir un documento por cada entidad especificando la Clave y los Atributos de la entidad, junto con una descripción de los atributos y unas estimaciones de los volúmenes de datos previstos para la entidad.

Las descripciones de las Relaciones entre Entidades donde se puede determinar la existencia de Atributos que dependen de más de una entidad, (es decir, que la relación es de tipo N:M). La Relación entre Cuentas y Clientes es un ejemplo de una Relación N:M.

2.1 PROPIEDADES

Una Propiedad es el elemento lógico de información mas pequeño que manipula una empresa, derivado de sus reglas de gestión, teniendo un sentido en sí mismo. Las propiedades se utilizan para describir Entidades y Relaciones. Ejemplos de Propiedades son "Nº Empleado", "Sueldo", "Apellido", "Fecha de Nacimiento". En tablas DB2, las propiedades son representadas por Columnas.

Las Propiedades deben ser siempre datos "conceptuales" y elementales, esto es, prescindir de datos calculados que se deriven de propiedades ya definidas, no tomando en consideración datos a nivel físico existentes en los procesos mecanizados que se vayan a concebir de nuevo (indicadores, contadores, etc.) ni tampoco considerar datos derivados de la existencia de una organización determinada, que seguramente será cambiada por otra.

De entre las propiedades que constituyen una Entidad, al menos una de ellas debe permitir caracterizar cada una de sus ocurrencias de manera única. Esta o éstas propiedades se denominan Identificador o Clave de la Entidad. La Clave siempre ocupa el primer lugar en la lista de las propiedades de la Entidad. P.e. "Nº Empleado"; "Nº Cuenta" Las ocurrencias de Nº Cuenta son las ocurrencias de la Entidad CUENTA

Es muy importante que el MCD sea RELACIONAL y no jerárquico. Esto debe reflejarse tanto en el modelo gráfico como en la definición de las claves de las entidades. P.e. las entidades "Cliente", "Cuenta", "Movimiento" son correctas. En cambio las entidades "Persona-Cuenta", "Persona-Cuenta-Fecha de Movimiento_Nº de Movimiento" son incorrectas.

2.2 ENTIDADES

Podemos definir una Base de Datos como un conjunto de Entidades de Datos relacionadas íntimamente entre sí. Cada una de estas Entidades tendrá, por lo general, una serie de Propiedades inherentes a la misma que tomarán un conjunto de valores (finito o infinito). Es decir, una Entidad será un conjunto de n-tuplas, cada una de las cuales será un conjunto de n valores, uno para cada una de las propiedades que tenga esa Entidad de Datos.

Una vez que ha sido normalizada, los atributos de una Entidad dependen totalmente de un único Identificador o Clave. Una Entidad debe tener un nombre y descripción concreto acordes con la descripción de su Clave. Ejemplos de Entidades incluyen "Clientes", "Empleados", "Cuentas" y "Movimientos" pero no "Saldos" (¿de que tipo?) ni "Fechas".

Observar que si la entidad se llama P.e. "Persona", la Clave de la entidad debería ser "Nº Persona" y no una combinación de atributos que en su conjunto garanticen la unicidad como P.e. "Nombre" + "Apellido" + "Color de ojos" + "Talla" + "Fecha 1ª Comunión".....

2.3 CLAVE DE UNA ENTIDAD

La Clave de una Entidad es aquella Propiedad o conjunto de propiedades que permiten identificar de manera única cada una de sus ocurrencias. Esta o éstas propiedades forman la Clave de tal Entidad.

El hecho de que una Entidad tenga una existencia propia, permite asegurar que siempre será posible hallar una Clave. Si ésta no se hallara entre la lista de propiedades, será necesario crearla o renunciar a la concepción de tal Entidad.

Se trata de un medio cómodo de asegurarse que las Entidades concebidas son verdaderas entidades. Por otra parte, el diseñador deberá siempre hacerse la pregunta: ¿cual es el auténtico identificador de esta Entidad? Si existe más de un candidato, se elegirá aquel que, en el dominio estudiado, sea el que semánticamente mejor se adapte y, evidentemente, el que sea estable en su valor.

Siempre que sea posible, la Clave de una Entidad debería estar formada por una sola propiedad, incluso cuando requiere la creación de una propiedad "artificial". P.e. definir la propiedad "Nº Movimiento" para identificar de forma única las distintas ocurrencias de la Entidad MOVIMIENTOS EN CUENTA.

No obstante, existen algunas Claves que se traducen en más de un elemento de datos en la Clave Primaria de la tabla correspondiente. El ejemplo clásico es la propiedad conceptual "Nº Cuenta" que se convierte en los datos lógicos de Banco, Oficina y Cuenta en las tablas DB2.

La Clave de una Entidad se convierte en la Clave Primaria de una tabla DB2.

Nota importante

Se desprende de lo anterior que hay limitaciones en los tipos de datos que pueden formar parte de la Clave de una Entidad. Por lo tanto se deberían evitar aquellas Claves que contengan campos de los siguientes tipos:

Fecha (incluyendo campos como año o mes)

Timestamp

Switch / Estado / Indicadores

Tipo (excepto en tablas de descripción de los distintos tipos de una aplicación. P.e. TIPO DE MOVIMIENTO, descripción)

Porcentaje

Nombre (P.e. Apellido, Municipio, Empresa, Agencia o Producto)

Texto descriptivo (P.e. Dirección, Mensaje, Motivo)

Importe o Saldo

2.4 RELACIONES

Una Relación es la representación de asociaciones entre Entidades, desprovista de existencia propia y derivada de las reglas de gestión.

Una vez establecidas las Entidades de una Base de Datos se pueden analizar las relaciones o vínculos que existen de forma natural entre ellas. Estas relaciones pueden o no tener atributos propios que dependen de la combinación de las Claves de más de una Entidad y deben tener una descripción sencilla.

Entidad	**Relación**	**Entidad**
Cliente	es titular de	Cuentas
Cuenta	genera	Movimientos
Pólizas	aseguran	Clientes
Clientes	son tomadores de	Pólizas
Clientes	son beneficiarios de	Pólizas
Cliente	esta relacionado con	Cliente

Hay varios tipos de relaciones posible.

Uno a muchos (1:N).
Muchos a muchos (N:M).
Uno a uno (1:1).

Las relaciones casi siempre unen dos entidades. En algunas ocasiones excepcionales puede darse el caso que una relación una tres entidades. Normalmente una relación que une tres o más entidades puede resolverse en dos o más relaciones distintas.

Una entidad puede presentar una relación consigo misma. P.e. Cliente con Cliente.

2.5 IDENTIFICADOR DE UNA RELACIÓN

El identificador de una relación es el resultado de combinar los identificadores de las Entidades que asocia.

Hay que subrayar que esta noción de identificador de una relación contiene ambigüedades importantes. Se puede, en efecto, concebir dos relaciones distintas entre las mismas Entidades. P.e., entre las entidades CLIENTE y CUENTA pueden existir las relaciones siguientes:

Cliente es el primer titular de unas cuentas.
Clientes avalan cuentas

En la práctica, esta situación puede dar lugar a la creación de dos tablas distintas, una para cada relación y ambas con el mismo identificador. Por esta razón, no se debería utilizar el identificador de una relación y se manipulará la relación por su nombre.

En una tabla DB2 que se deriva de una relación, el identificador puede o no pasar a ser la Clave Primaria de la tabla, según las reglas de gestión referentes a la posibilidad de actualización de los valores de los campos que forman el identificador. Por ejemplo, la relación Cliente-Cuenta tiene como Identificador de la relación los atributos Nº Cliente y Nº Cuenta. Las reglas dicen que no se debe poder cambiar la titularidad de una cuenta. En este caso se definiría el Identificador como Clave Primaria de ésta tabla para así impedir operaciones de UPDATE contra estos campos (incluso desde SPUFI). En otros casos las reglas de gestión suponen una relación más flexible. Un ejemplo sería la asignación de un Empleado a unos Clientes.

Donde una relación se transforma o bien en una tabla DB2 o bien en una columna adicional de la tabla "hija" de una relación, debería estar definida como Clave Ajena (Foreign Key) de la(s) tabla(s) padre(s) con las correspondientes reglas de Integridad Referencial.

2.6 CARDINALIDADES

La cardinalidad de una Relación indica el número de Ocurrencias que pueden existir de una Entidad con referencia a otra. P.e. una persona puede (o no) tener una (o más) cuentas. En cambio una cuenta puede tener (mínimo uno) varios titulares.

Conviene examinar cuidadosamente las relaciones entre entidades para establecer correctamente las cardinalidades, siendo este punto una importante causa de errores en el diseño de una BD.

2.7 INTEGRIDAD REFERENCIAL

Es muy importante que el diseño de la BD contemple desde el principio las dependencias entre las distintas tablas para asegurar un diseño que permita que el sistema de gestión de la BD cuide de la coherencia de los datos de la misma.

La Integridad Referencial de DB2 debe asegurar dos cosas

> Cuando se inserta o actualiza una fila en una tabla, comprueba que los valores entrados en campos designados como claves foráneas de otras tablas existan en aquellas tablas referenciadas. De no ser así, se rechazará la operación.
>
> Cuando se elimina una fila de una tabla referenciada desde otras tablas, comprueba la existencia o no de ocurrencias de este valor de Clave en las tablas relacionadas, y en el caso de encontrar este valor podrá tomar una de estas tres acciones; - impedir la eliminación de la fila (RESTRICT); - borrar todas las filas relacionadas que tengan este valor (CASCADE); - Sustituir cada ocurrencia de este valor en las tablas relacionadas por el valor Nulo (SET NULL).

3 NORMALIZACIÓN DEL MODELO CONCEPTUAL

Las entidades y relaciones de un MCD deberían cumplir como mínimo la tercera forma normal. Un modelo puede incumplir la cuarta forma normal solo si hay suficientes razones como para justificarlo.

Está en **Primera Forma Normal** (1ª FN) si para una ocurrencia de la Clave no hay múltiples ocurrencias de alguno de sus atributos.

> P.e. Nº CUENTA, Saldo Enero, Importe Enero, Saldo Febrero, Importe Febrero, Saldo ... no cumple con la 1ª FN porque para una ocurrencia de la clave hay múltiples ocurrencias de los datos "Saldo" e "Importe".

Está en **Segunda Forma Normal** (2ª FN) si está en 1ª FN y además cada Atributo depende completamente de la Clave.

> P.e. Nº CLIENTE, Nº CUENTA, Plan de la cuenta no cumple con la 2ª FN porque el Plan depende de la cuenta pero no del cliente.

Está en **Tercera Forma Normal** (3ª FN) si lo está en 2ª FN y cada Atributo no es transitivamente dependiente de la Clave. Dicho de otra manera, no cumple con la 3ª FN si un Atributo es dependiente de otro Atributo.

P.e. Nº CUENTA, Oficina gestora, Código Postal Oficina no cumple con la 3ª FN porque el atributo Código Postal es dependiente del atributo Oficina gestora y solo indirectamente de la clave Nº Cuenta.

Está en la **Cuarta Forma Normal** si lo está en 3ª FN y, si existe una dependencia multievaluada de B respecto a A (un valor determina un número finito de valores de B), entonces el resto de atributos depende funcional y completamente de A*B. Si incumple la 4ª FN hace falta separar la relación en dos o mas relaciones distintas.

P.e. Nº EMPLEADO, Nº TELEFONO, IDIOMA, donde los tres campos forman el Identificador de la Relación, incumple la cuarta forma normal dado que un Empleado puede tener varios telefonos y hablar varios idiomas pero no necesariamente el mismo número de cada. Además, queda obvio que no hay una relación directa entre los teléfonos que tiene un empleado y los idiomas que sabe. En este caso conviene claramente dividir esta relación en dos relaciones distintas: Nº EMPLEADO, Nº TELEFONO, y Nº EMPLEADO, IDIOMA.

4 TRANSFORMACIÓN DEL MCD EN UN MODELO LÓGICO DE DATOS (MLD)

4.1 DEFINICIÓN DE UN MLD

Si al nivel conceptual se determina **qué** hay que controlar en la base de datos de la aplicación, al nivel lógico se establece el **cómo**. El objetivo del traspaso del MCD a un MLD es convertir las Entidades y Relaciones en tablas lógicas con las definiciones de las columnas de cada tabla y relacionadas entre si por sus vínculos (Claves Foráneas) y sus Reglas de Integridad.

4.2 TRASPASO DE ENTIDADES A TABLAS

Una Entidad correctamente identificada de transforma en una sola tabla.

El Identificador de la Entidad se convierte en la Clave Primaria de la tabla.

Las Propiedades de la Entidad se convierten en columnas de la tabla. Una Propiedad puede dar lugar a mas de una columna (p.e. Nº Cuenta).

Una Propiedad opcional o que puede no conocerse al momento de dar de alta la fila debe convertirse en una columna Nula.

4.3 TRASPASO DE RELACIONES M:N A TABLAS

Una Relación correctamente normalizada y de tipo m:n se convierte en una sola tabla.

La combinación de los Identificadores de las Entidades vinculadas por esta Relación determina las columnas de la Clave Primaria de la tabla. Debe definirse un índice único para el conjunto de estas columnas.

Para cada una de las Entidades relacionadas se establece cual debe ser su regla de Integridad Referencial (RESTRICT, SET NULL o CASCADE) y se definen las Claves Foraneas correspondientes.

4.4 TRASPASO DE RELACIONES 1:N A TABLAS

Una relación de tipo 1:n entre dos Entidades se convierte en una Clave Foranea en la tabla 'hija' compuesta de la(s) misma(s) columna(s) que forma(n) la Clave Primaria de la tabla 'padre'.

De este modo, una tabla se compone de las columnas que forman su Clave Primaria más las columnas que representan sus propiedades mas las columnas correspondientes a todas las Claves Primarias de aquellas tablas de las cuales es 'hija'.

Para cada una de las Entidades relacionadas se establece cual debe ser su regla de Integridad Referencial (RESTRICT, SET NULL o CASCADE) y se definen las Claves Foraneas correspondientes.

5 ELEMENTOS DE DATOS DB2

5.1 DEFINICIÓN DE ELEMENTOS

Como indica su nombre, un elemento representa un dato cuya información no es divisible en otros datos, sino que representa un único concepto elemental.

Esto implica que no se deberían definir elementos que incorporen otros datos, ni la utilización de técnicas de SUBSTR para tratar filas sobre la base de una parte del valor de un campo. P.e. no se permite la definición de un elemento que, en realidad, sea la fusión de dos datos distintos que, de forma individual, serían dos elementos por si mismos. .

5.2 DATOS ESPECIALES (DATE, TIME, VARCHAR....)

Se recomienda que los elementos de tipo fecha se encuentren definidos como DATE en DB2 y como Pic X(10) en las copies.

Los elementos que llevan la fecha y hora deberían estar definidos como TIMESTAMP en DB2 y como Pic X(26) en las copies.

Los elementos de tipo hora deberían estar definidos como TIME en DB2 y como Pic X(8) en las copies.

Cuando la longitud de la información de una columna de tipo CHAR es altamente variable o donde en la mayoría de las ocurrencias de una determinada columna de tipo CHAR la información es nula o de escasa longitud, conviene definir la columna como VARCHAR.

La definición de una columna como VARCHAR o LONG VARCHAR es obligatoria cuando la longitud máxima de la información es superior a 254 caracteres.

La definición de una columna como LONG VARCHAR es en función del tamaño de la página del espacio de tablas y la longitud de toda la fila.