UNIVERSIDAD DE SONORA
Departamento de Matemáticas

TEORIA DE DISEÑO PARA BASES DE DATOS RELACIONALES Y UN EJEMPLO PARA UNA BASE DE DATOS DEL CENTRO DE INVESTIGACION EN ALIMENTACION Y DESARROLLO, A. C.

T E S I S

Que para obtener el Título de

LICENCIADO EN MATEMATICAS

Presenta

Sergio Salazar Aganza

CENTRO DE INVESTIGACION EN ALIMENTACION Y DESARROLLO, A. C.

Hermosillo, Sonora Diciembre de 1984

Universidad de Sonora

Repositorio Institucional UNISON

"El saber de mis hijos hará mi grandeza"

Excepto si se señala otra cosa, la licencia del ítem se describe como openAccess

A todos los que sientan la curiosidad de abrir este trabajo.

AGRADECIMIENTOS:

Agradezco la comprensión y estímulo del personal del CIAD (Centro de Investigación en Alimentación y Desarrollo, A.C.), en general, y en particular al personal de la Unidad de Estadística y Cómputo, especialmente al jefe de esta unidad el Sr. Elías Robles Sotelo.

Agradezco también al CIAD como Institución sin cuyo apoyo económico y técnico este trabajo no hubiera sido posible, y en particular a su Director el Dr. Carlos Enrique Peña Limón.

Finalmente, agradezco a mi asesor de tesis y amigo Fernando Avila M. por su confianza, apoyo y valiosos consejos.

INDICE

0 PRESENTACION.

Antes de entrar propiamente en materia es indispensable dar un breve bosquejo de los objetivos del trabajo, de la metodología seguida, de la estructura general del escrito y de su relación con las matemáticas.

Originalmente este trabajo formó parte de un proyecto del Centro de Investigación en Alimentación y Desarrollo, A.C. (en adelante lo abreviaremos CIAD), actualmente suspendido, el cual consistía en el desarrollo de un Banco de Datos (también conocido como Base de Datos y que en adelante abreviaremos BD) que contuviera información Socioeconómica de los municipios del Estado de Sonora. Los objetivos generales de dicho proyecto eran:

1o. Centralizar la información periódica disponible sobre los municipios del Estado, en los relativo a datos Socioeconómicos.

2o. Permitir hacer cualquier tipo de cruces entre los datos.

3o. Facilitar la consulta y análisis de los datos a personas no conocedoras sobre computadoras.

Como podemos fácilmente concluir de los anteriores objetivos, el BD-CIAD, se desenvolvería en un entorno de consulta exclusivamente, es decir sólo el personal encargado podría agregar nuevos datos, pues los usuarios finales serían consultantes exclusivamente.

Todo lo anterior se debía lograr utilizando una microcomputadora IBM PC, un disco duro con capacidad de almacenar 20 millones de caracteres (20 Mb) y un paquete comercial de programas manejadores de bases de datos (DBMS), llamado dBASE II, que en ese entonces era el mejor y más popular en el mercado para microcomputadoras, además de ser según sus vendedores un sistema relacional (ver Modelos de Datos en el siguiente capítulo).

Inicialmente el objetivo particular de quién esto escribe, consistía en encontrar una forma de organizar los distintos datos, a fin de lograr los objetivos 2o. y 3o. del BD-CIAD, a dicha organización la denominamos Estructura Logica de la BD, y seleccionamos el enfoque relacional para ella por ser el más nuevo, y según se decía en la literatura especializada era el más simple y poseía una gran capacidad para establecer cruces no previstos entre los datos.

A fin de diseñar dicha esctructura lógica, se decidió, investigar al respecto en las obras marcadas [1], [2] y [3] en la Bibliografía del presente escrito.

Pronto se hizo evidente que los BD a los que se referían tales obras, correspondían a sistemas implementados en computadoras grandes, no obstante se continuó su estudio pues se buscaba una posible adaptación de ellas a los BD para micros; en estas circunstancias cabe aclarar que los sistemas de BD que se manejan en los dos primeros capítulos corresponden casi exclusivamente a BD implementados en grandes computadoras, que se encuentran en un entorno no sólo de consulta sino también de inserción de datos por parte de los usuarios finales, no programadores.

Mucho más tarde se encontró que lo que se conoce en términos de la teorla existente como Esquema Conceptual Ideal, cumple con el segundo objetivo del BD-CIAD, y que si bien el tercer objetivo era propiciado por la adopción del enfoque relacional para dicho esquema, esto finalmente era más bien cuestión del DBMS a usar.

Este Esquema Conceptual Ideal, sólo podla obtenerse siguiendo la teorla para BD relacionales, a través de lo que se denomina Proceso de Normalización.

El interés matemático en dicho Proceso de Normalización, consiste en que se basa en el desarrollo axiomático de ciertas caracterlsticas de dependencia entre los valores posibles de los datos y este desarrollo representa una aplicación no trivial del concepto de relación, demostrandose además la consistencia y la completez de dichos axiomas. Este contenido matemático se encuentra expuesto en el capltulo 3 del presente trabajo.

El capltulo cuarto presenta un ejemplo de la aplicación del proceso de normalización, para el BD-CIAD.

Finalmente el capltulo quinto contiene algunas conclusiones y experiencias obtenidas durante la elaboración del presente escrito, para los interesados en profundizar en el tema.

1. INTRODUCCION.

1.1 Conceptos Fundamentales sobre el Proceso de Datos por Computadora.

Las definiciones siguientes no pretenden ser rigurosas o precisas sino solo dar una idea intuitiva de lo que se trata.

SISTEMA OPERATIVO:

El sistema operativo de una computadora es el conjunto de programas escritos en lenguaje de mâquina que permiten que êsta empiece a recibir instrucciones y haga un uso eficiente de los recursos computacionales disponibles; constituye, por decirlo de un modo poêtico, "la conciencia de la computadora".

DISPOSITIVOS DE ALMACENAMIENTO MASIVO:

Se trata de las cintas perforadas, tarjetas perforadas, cintas, tambores y discos magnêticos, mâs generalmente, cualquier medio que permita conservar datos y programas en forma mâs o menos permanente y que, desde luego, la computadora pueda leer râpidamente.

LENGUAJE DE PROGRAMACION:

Podemos ver un lenguaje de programaciôn como un conjunto de comandos que pueden usarse conjuntamente para "programar" una computadora, distintos del lenguaje de mâquina.

ARCHIVO FISICO:

Un archivo fîsico o simplemente archivo, es un conjunto de datos contenidos en un dispositivo de almacenamiento masivo y que son definidos al sistèma operativo mediante algûn lenguaje de programaciôn como pertenecientes a un mismo conjunto, para lo cuâl se le asigna un nombre, dicho nombre debe reunir ciertas caracterîsticas impuestas por el propio sistema operativo y desde luego ser distinto de todos los demâs nombres de archivos.

REGISTRO:

Es un conjunto de valores relacionados entre sî, que constituyen una unidad para la computadora. Es la unidad de entrada/salida de datos de un archivo fîsico.

ESTRUCTURAS DE ALMACENAMIENTO FISICAS :

Es la descripciôn de la forma en que los registros de un archivo fîsico son almacenados fîsicamente, dicha definiciôn podemos dividirla en:

Mêtodo de Acceso o Bûsqueda.
Formato del Archivo Fîsico.
LLaves o Claves del Archivo (si las hay).

Secuenciación de los registros (si la hay).
Técnica(s) de Compresión [si la(s) hay].

La parte más importante de una estructura de almacenamiento física de datos es sin duda el Método de Acceso.

METODOS DE ACCESO O BUSQUEDA :

Es la forma en que se localiza un determinado registro dentro de un archivo físico. Existe una gran variedad de métodos de acceso como por ejemplo:

Secuencial Físico.
Secuencial Indicado.
Aleatorio Indicado.
Inversión de Archivo.
Directo.
Hash., etc.,...

FORMATO DE ARCHIVO O DE REGISTRO:

Para grabar o leer datos en un archivo es necesario especificar al sistema operativo además de un nombre con el que identificamos el archivo que se desea usar, cuál será la forma general que tienen los datos o registros en dicho archivo. Esta descripción se hace en general definiendo una variable para cada elemento del tipo de registro que se desea, a estas variables se les conoce como "campos", esta descripción se conoce como "formato de archivo" o "formato de registro".

ARCHIVO LOGICO:

Son archivos que no se encuentran físicamente almacenados, sino que cada uno de sus registros (llamados lógicos), se construye mediante programación a partir de los archivos físicos. En el contexto de las BD pueden construirse inclusive a partir de otros archivos lógicos. Los archivos lógicos pueden tener registros de varios formatos.

LLAVES O CLAVES DE ARCHIVO:

Se le llama "llave" o "clave" de un archivo al conjunto de campos que identifican unívocamente los registros de dicho archivo. Todo archivo tiene al menos una llave, trivialmente todos los campos de un registro lo identifican unívocamente.

SOFTWARE:

Conjunto de programas escritos para trabajar en forma interrelacionada, generalmente sin intervención consciente del usuario.

1.2 Conceptos Fundamentales en la Teoría de las Bases de Datos.

ENTIDAD:

Entidad es todo aquello que posee atributos que permiten identificarlo unívocamente de entre todos los de su misma especie. Obviamente esta definición tan general nos indica que corresponde a los diseñadores de cada BD particular decidir cuáles partes del mundo real tienen el papel de "sujetos activos" generadores de la información que desean guardar

ASOCIACIONES ENTRE ENTIDADES:

Una vez elegidas las entidades o más bien los tipos de entidades que nos interesan y cuáles serán los atributos que de ellas manejaremos, es inevitable que quede información importante que no corresponda propiamente a una sola entidad, sino que surja de la interacción de dos o más . A esta clase de interacciones significativas es a lo que llamamos una asociación entre entidades.

BASE DE DATOS (BD):

Es todo conjunto de datos contenidos en un grupo de archivos físicos o lógicos (pero no de ambos a la vez), manipulados mediante rutinas más o menos comunes.

SISTEMA DE BASE DE DATOS:

Un sistema de base de datos, llamado con frecuencia simplemente "Base de Datos", puede concebirse como un depósito central de toda la información necesaria para las distintas aplicaciones de un grupo heterogéneo de usuarios y en el que dicha información es manejada a través de una ó más computadoras.

Los principales componentes de un sistema de BD son: Datos, Equipo, Programas y Usuarios. En la Fig. 1, se muestra la estructura de un sistema de BD típico.

LENGUAJE DE DATOS:

Es el lenguaje que un sistema de base de datos pone a disposición de uno o más tipos de usuarios según el sistema particular, y se puede dividir en dos partes, la parte destinada a formular preguntas que se conoce como "Lenguaje de Interrogación" o "Lenguaje de Manipulación de Datos", DML por sus siglas en inglés, y el "Lenguaje de Definición de Datos" o DDL, que permite crear los respectivos archivos lógicos o internos. Según el nivel de abstracción en que dicho lenguaje esté actuando, hablamos de un SDML, lenguaje de manipulación de datos para subesquemas, SDDL lenguaje de definición de datos para subesquemas, DML interno o conceptual, DDL interno o conceptual, aunque convendremos que DDL y DML se refieren al nivel conceptual a menos que especifiquemos otra cosa.

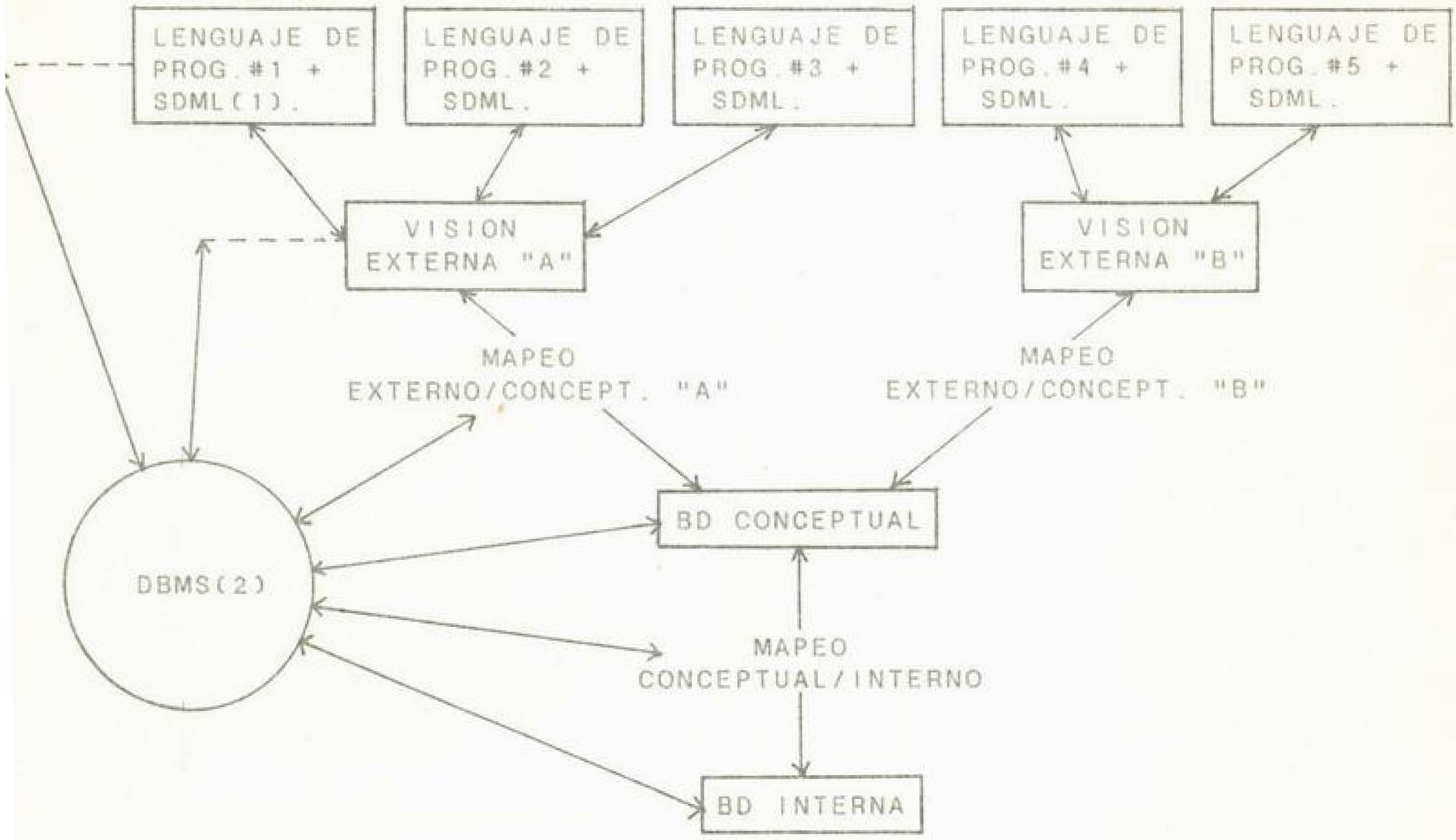

FIG.1: Arquitectura de un Sistema de Base de Datos propuesta por la ANSI/SPARC (American National Standars Institute/Systems Planning And Requirements Committee), que es la dependencia gubernamental encargada de definir los estandares para uso de la industria en los E.E.U.U.

Aunque en la prâctica no existe ningûn sistema comercial que se apegue estrictamente a esta estructura, al menos a la fecha.

(1) SDML: Lenguaje Manejador de Datos en Subesquemas (ver en hoja 5 "Lenguajes de Datos").
(2) DBMS: Sistema Administrador de Bases de Datos (ver en hoja 7 "DBMS").

DBMS:

Es la abreviación por Database Management System, es decir el sistema administrador de Bases de Datos. Se trata del conjunto de programas que se encargan entre otras cosas de:

* Manejar una ó más visiones lógicas de la totalidad ó parte de la información contenida a un muy alto nivel de abstracción y de acuerdo a un modelo de datos específico.

Para ello debe proporcionar al menos un lenguaje de datos que permita definir los tipos de "archivos lógicos" que constituyen un subesquema, los que constituyen la BD conceptual y en cierta medida, que depende del DBMS usado, la estructura de almacenamiento para los "archivos lógicos conceptuales".

* Proporcionar rutinas para el establecimiento de controles de acceso a partes de la información contenida, esto es, con el fin de restringir el acceso a cierta información.

* Permitir el establecimiento de rutinas que comprueben automáticamente, al momento de cargar datos, que se cumplan determinadas condiciones en los valores de sus atributos.

* Dar facilidades para la obtención de copias de la totalidad de la BD, con el fin de obtener respaldos periódicos de la misma para evitar su pérdida total en caso de alguna falla o error.

NIVELES DE ABSTRACCION DE UNA BASE DE DATOS:

Para que un DBMS presente a cada grupo de usuarios una adecuada visión de la BD de acuerdo a sus necesidades particulares, y que dichas visiones no se vean afectadas por modificaciones a la implementación interna de la BD, es necesario que maneje la información contenida en el sistema a tres distintos niveles de abstracción.

El nivel más alto es el que corresponde a la forma en que uno o más grupos de usuarios consideran que es la BD, a cada una de estas visiones se le llama "visión externa"; al plano o estructura de dichas visiones se le llama "subesquema", estos están definidos usando el SDDL (Subschema Data Definition Language) y las visiones manejan sus datos con el SDML (Subschema Data Manipulation Language).

Por su parte al nivel más bajo, más cercano a la forma en que físicamente están almacenados los datos, se le conoce como "Base de Datos Interna" y a su modelo o plano se le llama "Esquema Interno", la descripción del esquema interno se hace en su mayor parte automaticamente por las rutinas del DDL durante la definición del esquema conceptual. A un nivel intermedio se encuentra la "Base de Datos Conceptual", que es la representación de la totalidad de la información contenida en el Banco de Datos, sin consideraciones de la forma en que estos datos están almacenados físicamente. Para la definición del esquema conceptual se usa el DDL y para manejar los datos o registros conceptuales se emplea el DML.

A la descripción de la forma en que cada nivel es representado en el nivel inmediato inferior se le conoce como "mapeo". Es importante aclarar que de todas estas bases de datos la única que existe todo el tiempo es la BD física que es la implementación de la BD interna hecha por el sistema operativo y de cuya existencia el DBMS no esta consciente, al menos en la mayoría de los DBMS.

MODELO DE DATOS:

Es la forma o estructura en que "parecen" estar almacenados los datos para un usuario ya sea del nivel conceptual ó del nivel de subesquemas, los primeros son los "programadores de aplicaciones" y los segundos los "usuarios externos", existe un tercer tipo de usuario que se conoce como "administrador de la Base de Datos" y que utiliza la BD física por medio del DDL.

Los modelos también llamados "enfoques" o "aproximaciones" mejor conocidos son:

* Modelo Relacional
* Modelo Jerárquico
* Modelo de Redes

En realidad podemos encontrar DBMS que manejan una combinación de ellos e inclusive algunos que utilizan una estructura más propiamente del nivel de implantación física como es la "Inversión de Archivos" que utiliza el DBMS "ADABAS" (Adaptable Data Base System).

La principal diferencia entre estos enfoques o modelos es la forma en que se representan las asociaciones entre entidades.

MODELO RELACIONAL:

Este enfoque representa a cada entidad y/o asociación entre entidades de la misma forma: mediante una "tabla" es decir una representación tabular de los registros en el archivo en los que cada columna representa un atributo del tipo de entidad del que se conserva información en el archivo, es decir, cada registro es un mapeo de los nombres de los atributos a los valores posibles de sus atributos, dicha tabla recibe el nombre de "relación" ya que originalmente era estrictamente una relación matemática. Posteriormente estudiaremos con más detalle este modelo.

MODELO JERARQUICO:

El modelo Jerárquico de una BD se presenta ante los ojos del usuario como uno o más árboles jerárquicos con un solo nodo en el nivel superior (o nivel 1), llamado nodo raíz; los nodos representan tipos de entidades, esto es, cada nodo está formado por un conjunto de atributos pertenecientes al tipo de entidad que representan.

Como sabemos de las estructuras de árbol, cada nodo que no esté en el nivel 1 está conectado mediante un arco a un y sólo un nodo en el nivel superior y se dice entonces (con respecto a dos nodos unidos por un arco) que están "asociados"; al nodo en el nivel superior se le conoce como "nodo padre" y al del nivel inferior "nodo hijo"; la asociación entre entidades establecida por un arco en el modelo jerárquico es del tipo "una-a-muchas" del nodo padre al nodo hijo, esto es, para cada ocurrencia del nodo padre existirá un número arbitrario de ocurrencias asociadas del nodo hijo y a su vez toda ocurrencia de un nodo hijo tendrá asociada una y sólo una ocurrencia del nodo padre, es por esto que no se nombran las asociaciones en el modelo jerárquico y que las asociaciones del tipo "muchas-a-muchas" no puedan implementarse en un sólo árbol , esto es porque además de lo anterior no se permite volver a poner un nodo padre en un nivel inferior de la rama que le corresponde.

A cada ocurrencia de un nodo se le considera iniciadora de un "registro lógico", el cual consiste de todas las ocurrencias asociadas a ella en los niveles inferiores. Al registro lógico determinado por una ocurrencia del nodo raíz se le conoce como "registro lógico de la base de datos", a cada uno de los cuáles le podemos asignar un árbol jerárquico, es por ello que una ocurrencia o instancia de una BD jerárquica se presenta al usuario como un "bosque", esto es, un conjunto de árboles, cada uno de ellos asociado a un registro lógico de la base de datos.

MODELO DE REDES:

Este modelo organiza los datos en forma de lo que conocemos como gráfica dirigida. El modelo o esquema de una BD de Redes usa dos tipos de archivos lógicos: los nodos que representan tipos de entidades (llamamos "registros" a las entidades en este modelo) y los arcos (llamados "ligaduras) que representan asociaciones una-a-muchas entre los tipos de entidades que unen.

En este modelo se asigna un nombre a los arcos, en vista de que un nodo es libre de tener arbitrario número de asociaciones con los demás.

1.3 Bosquejo de la Evolución del Proceso de Datos.

A mediados de la década de los 50 aparecen las primeras computadoras comerciales, mismas que sólo manejaban archivos con registros físicamente secuenciados (esto es, contiguos dentro del dispositivo de almacenamiento), en ese entonces cada "usuario" debía producir sus propios programas por lo que éstos eran encaminados a usos muy específicos como "Nóminas", "Inventarios", "Facturación", etc. Luego, a principios de los 60 se evidencian las ventajas de la integración de las aplicaciones; esto condujo a mediados de dicha década al concepto de "Sistema Integrado para Manejo de Información" (IMIS por sus siglas en Iglés), cuyo enfoque era la integración de las aplicaciones de un "usuario", pero todavía cada una de ellas necesitaba sus propios archivos, el resultado era un complejo y frágil sistema que involucraba muchos archivos, reordenaciones y comparaciones, y que se detenía si cualquier archivo o programa fallaba.

Se pensó que la solución a los problemas del IMIS estaban en la integración de los datos mas bien que en la de las aplicaciones y aparecieron sistemas que manejaban estructuras más adecuadas para el encadenamiento de datos.

El término Base de Datos apareció a fines de los 60, sin embargo, los sistemas comerciales para manejo de información carecían de coordinación y flexibilidad.

A principios de los 70 el enfoque de las bases de datos fué ampliamente reconocido como el deseable para el procesamiento de datos. Empezaron a ofrecerse sistemas más avanzados y el término "Sistema Administrador de Base de Datos" (DBMS por sus siglas en inglés) empezó a usarse.

Por primera vez el enfoque de las bases de datos fué una práctica alternativa para una mayoría de usuarios.

1.3.1 Ventajas de las Bases de Datos.

Uno de los objetivos fundamentales de un sistema de base de datos, es el lograr la "Integración de los Datos", esto es, un control centralizado de los datos, lo cual proporciona las siguientes ventajas:

* Reducción de redundancia en el almacenamiento físico de los datos.
* Como consecuencia del punto anterior, existe un menor riesgo de inconsistencia en los datos almacenados.
* Se pueden aplicar criterios de estandarización, de acuerdo a normas de la compañía, de la industria, nacionales o internacionales, con el fin de lograr un eficiente intercambio de información entre sistemas.
* Es más fácil establecer controles para la integridad de los datos. Por Ejem. comprobar que ningún empleado tenga más de 2.50 m. de estatura, o que las horas trabajadas en una semana no sean más de, digamos 55, etc.

Otro de los objetivos fundamentales en el desarrollo de lo que hoy son las bases de datos es el de obtener la llamada "Independencia de Datos". Existen dos niveles de independencia de datos: La independencia física y la independencia lógica.

La independencia física consiste en que las modificaciones al modelo físico, ya sea cambios en la disposición física de los archivos o en el método de acceso, no afecten al modelo conceptual ni a las visiones externas, aunque desde luego modifiquen su rapidez de operación.

La indepencia lógica consiste en que los cambios al modelo conceptual no afectan las visiones externas, es decir, no es necesario modificar las aplicaciones existentes.

1.4 Etapas del Proceso de Diseño de una BD.

La figura 2 muestra en forma de diagrama de flujo los pasos o etapas generales del proceso de diseño de una BD.

A continuación describiremos brevemente en qué consiste el diseñar un modelo conceptual ideal.

Este es un proceso para el cual la teoría de diseño para BD relacionales resulta especialmente útil, ya que es la única que cuenta con una técnica que prodríamos calificar como "rigurosa" y se ocupa de problemas en la representación de datos en una estructura tabular llamada relación, misma que resulta adecuada para su "traducción" a una representación jerárquica o de redes, por lo que muy bien puede ser el primer paso para ver cual modelo de datos resulta más adecuado de acuerdo a los datos que pretendemos manejar.

El modelo conceptual ideal se distingue del modelo conceptual en que no toma en cuenta las limitaciones impuestas a la información "almacenable" por un DBMS particular, sino que se define en términos de un modelo de datos general teórico, mientras que el modelo conceptual debe definirse en términos del modelo de datos que maneje un DBMS mediante su DDL.

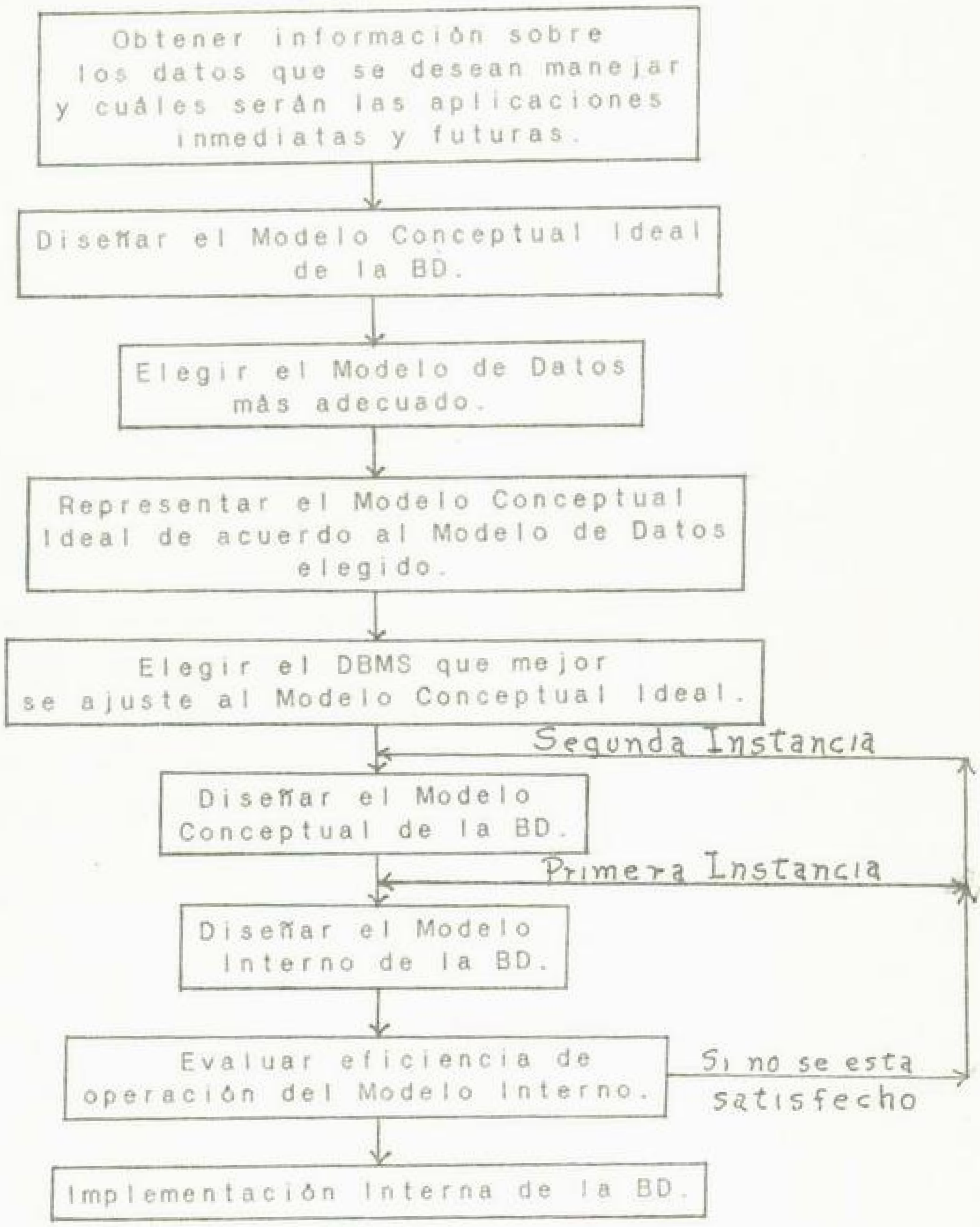

Fig. 2 Etapas en el Diseño de una Base de Datos.

2. EL MODELO DE DATOS RELACIONAL.

Ya hemos dicho que un "modelo de datos" es la forma en la que los datos "parecen" estar almacenados ante los ojos de algún usuario, y dicha "estructura de almacenamiento" no presenta detalles de su disposición física, ni de la manera como la computadora tiene acceso a ellos.

Por su parte el modelo de datos Relacional presenta los datos en forma de "tablas" o "tabulares" que se acostumbra llamar "relaciones" por su semejanza con la noción matemática de relación.

2.1 Conceptos Básicos.

RELACION: Es una colección de mapeos que asocian a una serie de variables llamadas "atributos" uno o más valores o elementos de sus respectivos dominios (los cuáles no son necesariamente distintos para atributos distintos).

Equivalen en el nivel conceptual a los archivos internos del nivel interno. Se representan en forma tabular con los mapeos como "renglones" y los atributos como "columnas" (ver en Fig. 2 un ejemplo de una relación llamada "DISTRIBUCION DE GRUPOS" con 8 atributos y solo 2 mapeos.

DISTRIBUCION DE GRUPOS.

M	G	T	P	H	A	E	S
100-01	01	78-101	JUAN PEREZ	14-15	K-101	78532-1	R
						78296-1	I
						78438-1	R
100-01	02	81-141	HECTOR LOPEZ	7-8	K-101	77083-2	R
						76942-2	R

CLAVES DE ATRIBUTOS:

M : Clave de la Materia
G : No. de Grupo
T : Clave del Maestro
P : Nombre del Maestro
H : Horario
A : Aula
E : Clave del Estudiante
S : Estatus del Estudiante

Fig. 3 Ejemplo de una relación.

RELACION NORMALIZADA: Es aquélla que satisface las siguientes condiciones:

1.- Todo mapeo asocia solo conjuntos elementales a cada atributo.

2.- Todo Dominio es "simple", es decir sus elementos son "atómicos", o sea que no pueden descomponerse en entidades más elementales dentro de una Base de Datos determinada.

En la Fig. 3 presentamos la relación normalizada "DISTRIBUCION DE GRUPOS", semánticamente equivalente a la relación de la Fig. 2.

DISTRIBUCION DE GRUPOS

M	G	T	P	H	A	E	S
100-01	01	78-101	JUAN PEREZ	14-15	K-101	78532-1	R
100-01	01	78-101	JUAN PEREZ	14-15	K-101	78296-1	I
100-01	01	78-101	JUAN PEREZ	14-15	K-101	78438-1	R
100-01	02	81-141	HECTOR LOPEZ	7-8	K-101	77083-2	R
100-01	02	81-141	HECTOR LOPEZ	7-8	K-101	76942-2	R

Fig. 4 Una relación normalizada.

Las ventajas de la normalización son:

1.- La simplicidad de las relaciones normalizadas permite su estudio en forma más sencilla.

2.- Facilita la manipulacidón de las relaciones mediante operadores más simples y generales.

3.- Simplifica su implantación física en una computadora.

En una Base de Datos relacional sólo se trabaja con relaciones normalizadas, en adelante cuando digamos "relación" se entenderá que es normalizada.

ESQUEMA RELACIONAL: Es un conjunto de atributos, junto con una serie de restricciones que deben satisfacer las relaciones para ser consideradas instancias válidas de dicho esquema, dentro de alguna interpretación determinada.

Un esquema relacional incluye el nombre de la relación y de sus atributos, así como una descripción de sus respectivos dominios y un conjunto de resticciones sobre la validez de sus mapeos dado que coinciden o no en determinados atributos a estas restricciones se les conoce como dependencias.

Los denotaremos de varias formas de acuerdo a las necesidades prácticas de cada caso, las ilustraremos con el esquema relacional correspondiente al ejemplo de la Fig. 2 :

1.- DISTRIBUCION DE GRUPOS*(Cve. Mat., No. Gpo., Cve. Maestro, Nmbe. Maestro, Hora, Aula, Cve. Estudiante, Estatus).
2.- DISTRIBUCION DE GRUPOS*
3.- MGTPHAES

ENEADA: Es la forma en que llamaremos à los mapeos de una relación.

2.2 Ventajas del Modelo de Datos Relacional.

* Es el más sencillo de entender para el usuario.

* Hace que el DDL y el DML sean más simples y poderosos.

* Como no existe dependencia posicional entre relaciones, las preguntas no se ven complicadas por la "colocación" de los datos.

* Es el modelo más abstracto ya que no impone métodos de acceso de acuerdo a la posición de los nodos, como en los modelos jerárquicos y de redes.

* Es el único que posee una teoría de diseño rigurosa.
* Está basada en un concepto más sencillo.
* Maneja una sola "estructura de almacenamiento lógica".
* Sus lenguajes de manipulación de datos son más interesantes para las matemáticas.

2.3 Representación de Entidades y Asociaciones Mediante Relaciones.

Relaciones que representan entidades:

Ya hemos sugerido la forma en que se hace esta representación, es simplemente asignando al esquema de la relación los atributos que nos interesen del tipo de entidad en cuestión.

Relaciones que representan asociaciones:

El esquema de una relación que representa una asociación entre los tipos de entidades E1,E2,...,En, consiste de una clave para cada tipo de entidad (la clave de una entidad es un conjunto de atributos que la determinan unívocamente de todas las de su mismo tipo) más los atributos propios de la asociación de que se trata. Así una eneada "t" en este tipo de relación representa una lista de entidades que están en la asociación de que se trata.

2.4 Lenguajes Relacionales para Bases de Datos.

La parte más interesante de cualquier lenguaje para Bases de Datos es el "Lenguaje de Interrogación" que consiste en el conjunto de comandos que permiten formular preguntas a la BD, dicho lenguaje de interrogación forma parte del DML el cuál a su vez es parte del "Lenguaje de la Base de Datos".

Actualmente existen dos enfoques básicos en los lenguajes relacionales de interrogación y son:

a) Algebra Relacional: Este enfoque nos produce los llamados Lenguajes Algebraicos, donde las preguntas se expresan aplicando operadores especializados a las relaciones, entre los que se encuentran los correspondiente a la teoría de conjuntos.

b) Cálculo Relacional: Consiste en formular las preguntas usando una notación parecida al cálculo de predicados de la lógica. En atención al elemento de las relaciones que se tome como "fundamental" o "mínimo", podemos hablar de dos clases de cálculos relacionales:

i) Cálculo Relacional de Eneadas.

ii) Cálculo Relacional de Dominios.

No veremos estos últimos lenguajes puesto que su estudio es de interés especialmente para los diseñadores de DBMS y no tanto para los diseñadores de su aplicación a una BD particular, pues estos últimos se limitan a elegir el DBMS comercial que más se adapte a sus necesidades.

2.4.1 Algebra Relacional.

A continuación hacemos una breve descripción de los principales operadores del álgebra relacional, debido al uso que haremos de ellos en el presente trabajo. Los operandos de dichos operadores son las relaciones y/o sus esquemas relacionales.

Los operadores básicos del álgebra relacional son:

1.- Unión
2.- Diferencia
3.- Producto Cartesiano
4.- Proyección
5.- Selección

Además de estos se acostumbra manejar los siguientes como útiles taquigrafías:

6.- Intersección
7.- Juntura Natural

Para los operadores booleanos tradicionales, es necesario que los esquemas relacionales de los operandos sean iguales.

1.- Unión:

RUS=RS La relación cuyas eneadas están en el conjunto de {t/ t es una eneada en la relación R ó en la relación S}.

2.- Diferencia:

R -S= La relación con eneadas en {t/ t ∈ R y t ∉ S}.

3.- Producto Cartesiano:

R x S= Una relación tal que posee una eneada para cada concatenación t + u con t ∈ R y u ∈ S.

4.- Proyección:

Este operador produce un subconjunto "vertical" de la relación dada, esto es, dicho subconjunto se obtiene seleccionando los atributos deseados en un orden especificado. La relación proyectada se denota con el nombre de la relación original seguida entre paréntesis rectangulares de los atributos sobre los cuáles se efectuó la proyección. Por ejemplo:

ALUMNOS

ALMN.#	NOMBRE	CARR.	CRDTS.	FECHA INGR.
1325	FCO.MARTINEZ R.	LM	190	20/02/81
5775	ALMA FIGUEROA S.	LF	105	01/09/81
8003	HECTOR PARRA V.	IQ	0	20/02/84
8569	HILDA LOPEZ M.	IC	0	20/02/84

ALUMNOS[ALMN.#,NOMBRE]

ALMN.#	NOMBRE
1325	FCO.MARTINEZ R.
5775	ALMA FIGUEROA S.
8003	HECTOR PARRA V.
8569	HILDA LOPEZ M.

5.- Selección:

Este operador obtiene un subconjunto "horizontal" de la relación en que opera, esto es, un conjunto de eneadas que satisfacen una condición especificada. La notación que usaremos por comodidad será: S[p(A1,A2,...,An)](R), donde R es una relación y p(A1,...,An) una proposición sobre los atributos A1,...,An.

Por ejemplo :

S[CRDTS.= 0](ALUMNOS)

ALMN.#	NOMBRE	CARR.	CRDTS.	FECHA INGR.
8003	HECTOR PARRA V.	IQ	0	20/02/84
8569	HILDA LOPEZ M.	IC	0	20/02/84

6.- Intersección

R S= La relación con eneadas {t/ t ∈ R y t ∈ S}.

7.- Juntura Natural Generalizada:

Cabe aclarar que existen muchos tipos de operadores juntura. La juntura natural generalizada de las relaciones R1,R2,...,Rr Ri∩Rj=/∅, y es denotada ¦x¦Ri es una relación cuyo conjunto de atributos es la unión de los conjuntos de atributos de las Ri para toda i y cuyas eneadas "t" son tales que t[Ri*] es una eneada en Ri para toda i. Por ejemplo:

R:

ALUMN.#	NOMBRE	FECHA INGR.
1325	FCO.MARTINEZ R.	20/02/81
5776	ALMA FIGUEROA S.	01/09/81
8003	HECTOR PARRA V.	20/02/84
8569	HILDA LOPEZ M.	20/02/84

S:

ALUMN.#	CARR.	CRDTS.
1325	LM	190
5776	LF	105
8003	IQ	0
8569	IC	0

Podemos ver que R¦x¦S= ALUMNOS. A los operadores que actúan sobre las relaciones para producir otras se les llama operadores de conjuntos o de alto nivel y a los lenguajes que los poseen se les conoce como lenguajes de "alto nivel".

3. TEORIA DE OPTIMIZACION DE ESQUEMAS RELACIONALES.

Esta teorla se ocupa del problema de encontrar dado un conjunto de esquemas relacionales otro conjunto de esquemas (o sea el Esquema Conceptual Ideal de una BD-relacional) que tengan las mejores propiedades en un medio de inserción y borrado de información, además de eliminar redundancias y propiciar una completa flexibilidad en la recuperación de la información.

Por otro lado, esta teorla no se ocupa de:

1.- Como accesar la información contenida.

2.- Como obtener el conjunto inicial de esquemas relacionales.

3.- Como almacenar flsicamente los esquemas resultantes.

4.- Métodos de evaluación de la eficiencia esperada de un diseño conceptual ideal determinado, dadas las caracterlsticas de operación del futuro BD.

5.- Como implementar los esquemas resultantes y las visiones externas que estos soporten.

3.1 Dependencias Funcionales.

DEFINICION 1 (Ullman [1]): Sea R(A1,...,An) un esquema relacional y sean X y Y subconjuntos de {A1,...An}. Decimos que "X determina funcionalmente a Y" o que "Y es funcionalmente dependiente de X", denotado : "X-->Y", si siempre que la relación R sea una instancia de R*, no es posible que R posea dos eneadas que coincidan en su valor para los atributos de X y que sean distintas en uno o más atributos en el conjunto de atributos Y.

Antes de presentar una definición mucho más concisa necesitamos acordar la siguiente notación: Sea R una relación con conjunto de atributos @ y $X \subset @$, entonces si r es una eneada en R diremos que $r \in R$ y por r[X] entenderemos la X-eneada con los mismos valores que r para los atributos de X.

DEFINICION 2 (Stout & Woodworth [4]): Si R* es un esquema relacional cuyo conjunto. de atributos es @ con X y Y subconjuntos de @. En R*, X-->Y, si $r1,r2 \in R$ y r1[X]=r2[X] implican que r1[Y]= r2[Y], para toda instancia R de R*.

3.1.1 Axiomas de las Dependencias Funcionales.

En atención a las propiedades de las dependencias funcionales podemos abstraer los siguientes axiomas conocidos como Axiomas de Amstrong.

Sea U un conjunto de atributos y X,Y,Z y W subconjuntos de U. Entonces:

FD1.(Proyectividad): Si $Y \subset X$ entonces X-->Y.

FD2.(Transitividad): Si X-->Y y Y-->Z, entonces X-->Z.

FD3.(Aumento): Si X-->Y, entonces XZ-->YZ.

En adelante usaremos estos axiomas como ùnicas reglas de inferencia para dependencias funcionales.

OBSERVACIONES: 1.- Las dependencias funcionales se basan en la semàntica de los atributos y no en la de los esquemas relacionales a los que pertenezcan.

2.- Posteriormente cuando definamos dependencia multivaluada entre atributos veremos que son un modelo que satisface FD1 y FD3, pero no FD2, demostrando asì la independecia de este ùltimo.

AMBITO DE APLICACION DE LOS AXIOMAS: Los axiomas se usarân solo para deducir si un conjunto dado de dependencias funcionales F implican lògicamente otras dependencias G.

Los problemas relacionados con la descomposiciòn sin pèrdida son de otra naturaleza, aunque como veremos despuès se halla estrechamente relacionado con las dependencias funcionales.

Por lo tanto, separaremos los resultados implicados por los axiomas de los que no lo son llevando numeraciòn natural para los primeros y numeraciòn Romana para los demâs resultados.

3.1.2 LLaves o Claves.

Ya hemos definido "llave" en relaciòn con archivos, ahora definiremos un concepto anâlogo para esquemas relacionales. Sea R* un esquema relacional con atributos @={A1,...,An}. Decimos que $X \subset @$ es una llave para R* sì:

1.- X-->Ai, i=1,...,n
2.- No existe $Y \subsetneq X$, tal que Y-->Ai, i=1,...,n

Es decir una llave es "minimal" en cierta forma.

3.1.3 Cerradura de un Conjunto de Dependencias.

DEFINICION: La cerradura de un conjunto de dependencias F denotada F+ es el conjunto de dependencias "lògicamente implicadas por F".

Decimos que "F implica lògicamente X-->Y" si siempre que un esquema relacional satisfaga las dependencias F se tiene que tambièn satisface la dependencia X-->Y.

DEFINICION: La cerradura de un conjunto de atributos X con respecto a un conjunto de dependencias F, contenidos en los atributos U de F, denotada X+, es el conjunto de atributos A tales que X-->A puede deducirse de F y de los Axiomas de Amstrong.

3.1.3.1 Propiedades de las Dependencias Funcionales.

LEMA 1. Las dependencias funcionales satisfacen las siguientes propiedades:

i) (Aditividad): X-->Y, X-->Z ==> X-->YZ.

ii) (Pseudotransitividad): X-->Y, WY-->Z ==> WX-->Z.

iii) (Descomposición): X-->Y, Z⊂Y ==> X-->Z.

DEMOSTRACION:

i) (Aditividad). X-->Y, X-->Z ==> (Por FD3) X-->XY, XY-->Z ==> (Por FD2) X-->Z. L.Q.Q.D.

ii) (Pseudotransitividad). X-->Y, WY-->Z ==> (Por FD3) WX-->WY y con la hipótesis: WY-->Z ==> (Por FD2) WX-->Z. L.Q.Q.D.

iii) (Descomposición). X-->Y, Z Y ==> (Por FD1) Y-->Z y con la hipótesis: X-->Y ==> (Por FD2) X-->Z. L.Q.Q.D.

LEMA 2. (Ullman [1]): Dadas las dependencias F, un conjunto de atributos U y un subconjunto X de U decimos que X-->Y se sigue de F y los Axiomas de Amstrong sí y sólo sí Y⊂X+. Donde X+ es el conjunto de atributos A en U tales que la dependencia X-->A puede deducirse de F y los Axiomas de Amstrong.

DEMOSTRACION (Ullman [1]):

(==>) Supongamos que X-->Y se sigue de F y los Axiomas de Amstrong. Para cada i X-->Ai se cumple por la propiedad de descomposición demostrada en el lema 1, así Y⊂X+. L.Q.Q.D.

(<==) Sea Y = A1...An para los atributos A1,...,An, y supongamos Y⊂X+. Entonces X-->Ai es obtenida de F y los Axiomas de Amstrong para toda i. Por la propiedad de aditividad demostrada en el lema 1 se sigue X-->Y. L.Q.Q.D.

TEOREMA 1 (Ullman [1]): Los Axiomas de Amstrong son consistentes y completos.

DEMOSTRACION (Ullman [1]):

i) Consistencia:

Sea X-->Y una dependencia que se deduce mediante los axiomas de Amstrong de un conjunto de dependencias F con atributos U. LLamemos R* al esquema con dependencias F y atributos U y R una instancia cualquiera de R*.

Si X-->Y se sigue de F por proyectividad.

Entonces si t,s∈ R con t[X]=s[X], se sigue de Y⊂X que t[Y]=s[Y], es decir X-->Y se cumple en R.

Si X-->Y se sigue de F por transitividad.

Esto implica que existen dependencias X-->Z, Z-->Y en F.

Por lo tanto si $t,s \in R$ con t[X]=s[X], tenemos de X-->Z que t[Z]= s[Z] y a su vez de Z-->Y tenemos que t[Y]=s[Y], es decir X-->Y se cumple en R.

Si X-->Y se sigue de F por aumento.

Entonces deben existir: V-->W en F y $Z \subset U$, tales que X=ZV y Y=ZW. Por lo que si $t,s \in R$ con t[X]=s[X], tenemos que t[V]=s[V] y t[Z]=s[Z] puesto que X=ZV. Ahora bien, de V-->W tenemos que t[W]=s[W] y dado que t[Z]=s[Z] tenemos que t[WZ]=s[WZ]=t[Y]=s[Y], es decir X-->Y se cumple en R.

Dado que R es una instancia cualquiera de R*, tenemos que R* satisface X-->Y, luego los axiomas de Amstrong son reglas de inferencia consistentes para las dependencias funcionales.

L.Q.Q.D.

ii) Completez:

Sea F un conjunto de dependencias sobre el conjunto de atributos U y supongamos que X-->Y no puede deducirse de los axiomas. Consideremos la instancia R con dos eneadas:

Atributos de X+							Otros Atributos						
1	1	.	.	.	1	1	1	1	.	.	.	1	1
1	1	.	.	.	1	1	0	0	.	.	.	0	0

Primero mostraremos que todas las dependencias en F son satisfechas por R.

Supongamos que V-->W está en F pero no es satisfecha por R. Entonces $V \subset X+$, o las dos eneadas de R difieren en algún atributo de V lo que implicaría que R satisface V-->W por vacuidad. Además, W no puede ser subconjunto de X+ pues V lo es y entonces trivialmente V-->W pues las eneadas coincidirían en sus atributos de W.

Sea A un atributo de W que no está en X+. Puesto que $V \subset X+$, X-->V se sigue de los axiomas. La dependencia V-->W está en F, así que por transitividad X-->W. Por reflexividad W-->A y por transitividad de nuevo X-->A esto implica que A X+ CONTRADICCION!

Por lo tanto V-->W se cumple en R.

Ahora falta demostrar que X-->Y no se satisface en R. Supongamos que es satisfecha. Como $X \subset X+$, entonces $Y \subset X+$ ó las dos eneadas coinciden en X pero difieren en Y lo cual no puede ser por nuestra suposición. Por lo tanto el lema 1 de consistencia nos dice que en este caso X-->Y puede inferirse de F y los Axiomas de Amstrong CONTRADICCION! Pues ya habíamos supuesto inicialmente lo contrario.

Por lo tanto X-->Y no se cumple en R aún cuando se cumplen todas las otras dependencias de F.

Concluimos que siempre que X-->Y no se siga de F por los Axiomas de Amstrong, F no implica lógicamente X-->Y. Esto es, los axiomas son completos. L.Q.Q.D.

OBSERVACIONES : Nótese que en la demostración anterior se utiliza la noción de "dependencia funcional en una relación" y nosotros no la hemos definido así; sin embargo, de nuestra definición es muy fácil deducir esta "nueva" noción por lo que no nos molestaremos en definirla explícitamente.

ALGORITMO 1 (Ullman [1]): Computación de la Cerradura de un conjunto de Atributos con Respecto a un Conjunto de Dependencias Funcionales.

ENTRADA: Un conjunto finito de atributos U, un conjunto de dependencias funcionales F en U y un subconjunto X de U.

SALIDA: X+ la cerradura de X con respecto a F.

METODO: Obtenemos una sucesión de conjuntos de atributos X(0),X(1),... mediante:

1.- X(0) es X.

2.- X(i+1) es X(i) más el conjunto de atributos sencillos "A" tales que existe alguna dependencia Y-->Z en F, A en Z y Y en X(i). Puesto que $X=X(0)\subset X(1)\subset X(2)\subset \ldots \subset X(i)\subset \ldots \subset U$, y U es finito, debemos eventualmente alcanzar i tal que X(i)=X(i+1).

Entonces se sigue que X(i)=X(i+1)= ... =X+. No hay necesidad de continuar más allá de X(i+1) una vez que descubramos que X(i)=X(i+1).

TEOREMA 2. Ullman [1]: El algoritmo 1 obtiene correctamente X+.

DEMOSTRACION (Ullman [1]):

L1.- Si A está en X(j), entonces A está en X+.

DEMOSTRACION L1 (Por Inducción):

Se cumple para j=0. Si $A\in X(0)=X$ obviamente X-->A.

Hipótesis de Inducción: El conjuto X(j-1) consiste sólo de atributos en X+, para algún j > 0.

Por demostrar X(j) sólo tiene atributos en X+.

Supongamos que $A\in X(j)$ porque $A\in Z$, $Y\text{-->}Z\in F$ y $Y\subset X(j-1)$. Como Y X(j-1) entonces $Y\subset X+$ por nuestra hipótesis de inducción Así X-->Y por el lema 1 y por transitividad X-->Z. Por reflexividad Z-->A y X-->A de nuevo por transitividad. Así $A\in X+$.

L.Q.Q.D.

L2.- Si X1 y X2 son dos conjuntos de atributos y $X1\subset X2$, entonces para toda j, $X1(j)\subset X2(j)$.

DEMOSTRACION L2 (Por Inducción):

$j=0$, $X1(0)=X1\subset X2=X2(0)$.

Hipótesis de Inducción: $X1(i)\subset X2(i)$ Para toda i >= 0, i <= j-1.

Sea $A\in X1(j)$ tal que $A\in Z$, $Y\text{-->}Z\in F$ y $Y\subset X1(j-1)$.

Entonces $Y\subset X2(j-1)$ por nuestra hipótesis.

Por lo tanto $A\in X2(j)$ de acuerdo al algoritmo 1. L.Q.Q.D.

L3.- Si $A\in X+$ entonces $A\in X(j)$ para alguna j >= 0.

DEMOSTRACION L3 (Por Inducción):

Demostraremos que si X-->Y se sigue de F por los axiomas de Amstrong, entonces todo atributo de Y está en algún X(j).

La demostración es por inducción sobre el número de líneas en la prueba de que X-->Y se obtiene de F mediante los axiomas de Amstrong, donde una "línea" es una dependencia que está en F o se sigue por reflexividad, aumento o transitividad, la última línea es X-->Y.

Sea p=1; entonces X-->Y se sigue por reflexividad, en cuyo caso, $Y \subset X$ = X(0) ó X-->Y está en F, y claramente todo atributo de Y está en X(1) por el algoritmo 1.

Hipótesis de Inducción: Si la demostración de X-->Y tiene menos de "p líneas" entonces todo atributo de Y está en algún X(j).

Sea X-->Y con p líneas en su demostración. Si X-->Y está en F o se sigue de X por reflexividad podemos aplicar el caso p=1.

Si X-->Y se sigue por transitividad de las líneas anteriores en la prueba, digamos X-->Z y Z-->Y, entonces ambas dependencias tendrán pruebas con menos de p líneas. Por la hipótesis de inducción existe algún X(j) que incluye todos los atributos de Z, así: $Z \subset X(j)$. Ahora apliquemos el algoritmo 1 con Z en lugar de X. Por la hipótesis de inducción existe k tal que Z(k) contiene todos los atributos de Y pues al menos existe una demostración de Z-->Y con menos de p líneas que sería la obtenida en el desarrollo de la demostración de X-->Y que ya sabemos se obtiene en p líneas.

Ahora por el resultado L2 con X(j) en lugar de X2 y Z como X1 tenemos que $Y \subset (X(j))(k)$, puesto que $Z \subset X(j)$ entonces $Z(k) \subset (X(j))(k)$. Pero (X(j))(k) no es más que X(j+k) por la forma en que el algoritmo 1 obtiene los X(j).

Por último consideremos cuando X-->Y se sigue por aumento de algúna línea previa V-->W con el conjunto de atributos Z.

Entonces VZ=X y WZ=Y. Sabemos que V-->W tiene una prueba de menos de p líneas, así por la hipótesis de inducción si corremos el algoritmo 1 empezando con V, existe algún j tal que $W \subset V(j)$. Por L2, si corremos el algoritmo 1 para X=VZ entonces $V(j) \subset X(j)$.

Por lo tanto $W \subset X(j)$, como $Z \subset X$, seguramente $Z \subset X(j)$, así WZ = Y es un subconjunto de X(j). Esto completa la inducción.

Por el lema 1 si $A \in X+$ entonces existe una prueba de X-->A a partir de F mediante los axiomas de Amstrong.

Luego por la inducción anterior A está en algún X(j), y así, A está en el conjunto obtenido mediante el algoritmo 1 como X+.

Por lo tanto el algoritmo 1 obtiene ni más ni menos que X+.

L.Q.Q.D.

3.2 Descomposición de Esquemas Relacionales

DEFINICION (Ullman [1]): El esquema relacional R(A1,A2,...,Ak) se dice que se "descompone" en los esquemas R1*,R2*,...,Rk*, de subconjuntos de R*, sí R*=R1*R2*...Rk*. No es necesario que los Ri* sean ajenos. No toda descomposición de un esquema relacional garantiza que cualquier instancia de él sea unívocamente reconstruida a partir de las proyecciones en sus "esquemas componentes".

3.2.1 Descomposiciones Sin Pérdida.

DEFINICION (Ullman [1]): La descomposición del esquema relacional R* con dependencias F, en R1*,R2*,...,Rk* se dice "Sin Pérdida con respecto a F" si para toda instancia R se tiene:

R=R[R1*]¦x¦R[R2*]¦x¦ . . . ¦x¦R[Rk*]

Es decir R es la juntura natural de sus proyecciones sobre los Ri*.

DEFINICION (Ullman [1]): Si @={R1*,R2*,...,Rk*} es una descomposición de R* entonces:

$$M@(R)=\underset{i=1}{\overset{k}{¦x¦}}R[Ri*]$$

Para toda instancia R de R*.

LEMA I (Ullman [1]): Sea R* un esquema relacional @={R1*,...,Rk*} una descomposición de R*, R una instancia del mismo y Ri=R[Ri*]. Entonces se cumplen:

i) R⊂M@(R)
ii) Sí S=M@(R), entonces S[Ri*]=Ri
iii) M@(M@(R))=M@(R)

DEMOSTRACION en Ullman [1] p.181.

ALGORITMO I (Ullman [1]): Comprobación de Descomposiciones Sin Pérdida:

ENTRADAS: Un esquema relacional R*=A1A2...An, un conjunto de dependencias funcionales F y una descomposición @=(R1*, R2*,...,Rk*) de R*.

SALIDA: Si @ es o no una descomposición sin pérdida.

METODO: Se construye una tabla con n columnas y k renglones; la columna j corresponde al atributo Aj y el renglón i corresponde al esquema relacional Ri*.

En el renglón i y la columna j ponemos el símbolo Aj sí el atributo Aj está en Ri*, si no, ponemos el símbolo b(i,j).

"Considerar" una y otra vez cada una de las dependencias X-->Y en F, hasta que no se puedan hacer más cambios. Cada vez que "consideremos" X-->Y buscamos renglones que coincidan en todas las columnas para los atributos de X. Si encontramos dos de tales renglones, igualamos los símbolos en los atributos de Y de la siguiente manera: Sí uno de ellos es Aj hacemos el otro Aj; si no, igualamos arbitrariamente.

Si después de hacer todas las modificaciones posibles a la tabla, encontramos que algún renglón tiene solamente Aes como elementos, entonces, la descomposición es sin pérdida, de otra forma es con pérdida.

TEOREMA I (Ullman [1]): "El algoritmo 2 determina correctamente si una descomposición es sin pérdida."

DEMOSTRACION (Ullman [1]):

Supongamos que la tabla final del algoritmo 2 no tiene un renglón de Aes. Podemos considerar dicha tabla como una relación (instancia) R de R*=A1...An, donde cada renglón es una eneada y los Ai y b(i,j) son los valores para los atributos Ai. La relación R satisface las dependencias F, puesto que el algoritmo 2 modifica la tabla siempre que encuentra una violación a F. Afirmamos que R=/M@(R). Claramente R no contiene la eneada A1A2...An. Pero para cada Ri, existe la eneada t(i) en R, que es el i-ésimo renglón correspondiente a Ri, entonces t(i)[Ri*] consiste sólo de Aes. Así la juntura de los Ri tiene una eneada que contiene sólo Aes, ya que cada Ri tiene la eneada t(i)[Ri*] y la juntura de todas estas eneadas nos produce un renglón con sólo Aes. Concluimos que M@(R)=/R, es decir, la descomposición en los Ri no es sin pérdida.

Desgraciadamente la demostración recíproca que da Ullman [1] utiliza ampliamente el cálculo relacional de dominios que es uno de los lenguajes más importantes para BD relacionales, sin embargo su estudio sería una desviación demasiado grande en comparación con el presente trabajo como para justificar incluirlo como herramienta en la demostración de un teorema por importante que este pudiera ser.

3.2.2 Descomposiciones que Preservan Dependencias.

Antes de definir lo que llamaremos una "descomposición que preserva dependencias" debemos definir "proyección de dependencias".

DEFINICION Proyección de Dependencias: Sea F un conjunto de dependencias sobre el conjunto de atributos U, decimos que el conjunto de dependencias X-->Y en F+ tales que XY⊆Z⊆U constituyen la "proyección de F sobre Z" y lo escribimos: F[Z].

DEFINICION Descomposición que Preserva Dependencias: Decimos que la descomposición @ de un esquema R*, con @={R1*,...,Rk*} "preserva" un conjunto de dependencias F si la unión de todas las F[Ri*] implica lógicamente todas las dependencias de F.

OBSERVACIONES: La importancia de que una descomposición preserve dependencias es que podemos tener una descomposición sin pérdida de una relación R con respecto a un conjunto de dependencias F y sin embargo encontrar en un momento dado que la juntura natural de las proyecciones de R contiene eneadas que violan algunas dependencias en F.

DEFINICION Cubierta de un Conjunto de Dependencias: Decimos que un conjunto de dependencias F "cubre" a un conjunto de dependencias G si G+⊆F+. Si además F+⊆G+, decimos que ambos conjuntos son "equivalentes".

DEFINICION Cubierta Minimal:(Ullman [1]) " Decimos que un conjunto de dependencias F es una "cubierta minimal" si:

1.- Todo lado derecho de una dependencia en F es un atributo sencillo.

2.- Para ninguna X-->A en F el conjunto {F-{X-->A}} es equivalente a F.

3.- Para ninguna X-->A en F y un subconjunto propio Z de X, {F-{X-->A}}{Z-->A} es equivalente a F.

Intuitivamente: (2) garantiza que ninguna dependencia en F es redundante. (3) garantiza que ningún atributo en el lado izquierdo es redundante y (1) que ningún atributo del lado derecho es redundante."

TEOREMA 3 (Ullman [1]): "Todo conjunto de dependencias F es equivalente a un conjunto F' que es minimal."

DEMOSTRACION (Ver Ullman [1]): Sólo daremos una idea intuitiva de la demostración: Para que se satisfaga la propiedad (1) de las cubiertas minimales es fácil ver que se puede aplicar la propiedad de descomposición para las DF y obtener un conjunto equivalente de dependencias con un sólo atributo en el lado derecho. Para satisfacer la condición (2) sólo nos queda ir eliminando las dependencias una a una y verificar si el conjunto restante es equivalente a F esto al menos en teoría siempre se puede hacer y lo mismo podemos hacer con los atributos del lado izquierdo para satisfacer la condición (3).

OBSERVACIONES : 1.- La cubierta minimal de un conjunto de dependencias F no es única.

2.- Esta construcción altamente teórica nos puede dar una idea de lo incipiente de la teoría sobre BD relacionales, pues no tenemos un método para obtener una cubierta minimal.

3.3 Dependencias Multivaluadas.

DEFINICION 1 (Ullman [1]): "X->->Y se cumple en R* si siempre que R es una instancia de R* y s,t∈R, con t[X]=s[X], entonces existen eneadas u y v en R tales que:

1.- u[X]=v[X]=t[X]=s[X]

2.- u[Y]=t[Y] y u[R-X-Y]=s[R-X-Y]

3.- v[Y]=s[Y] y v[R-X-Y]=t[R-X-Y]"

DEFINICION 2 (Stout & Woodworthy [4]): "Si el esquema relacional R* tiene conjunto de atributos U y X,Y⊂U, entonces X->->Y si siempre que t,s∈R (instancia de R*) y t[X]=s[X], entonces existe u∈R tal que:

u[XY]=t[XY] y u[U-Y]=s[U-Y]"

DEFINICION 3 (Fagin [5]): Si X y Y son subconjuntos de atributos de un esquema relacional R*, y si W es es el conjunto de atributos de R^* que no están en X ni en Y, entonces la dependencia multivaluada X->->Y se cumple en R^*, si toda instancia R es la juntura de sus proyecciones R[XY] y R[XW]."

OBSERVACIONES : 1.- En la DEFINICION 1 la condición (3) se deriva de (1) y (2) simplemente intercambiando t y s, además no es necesario que X y Y sean ajenos ni que t=/s ó u=/v.

2.- Las Dependencias Multivaluadas son simétricas con respecto al lado izquierdo pues X->->Y sí y sólo sí X->->R^*-Y

3.- Véase que las DM son propiedades de los esquemas relaciones como un todo, a diferencia de las DF que radican en la semántica de los conjunto de atributos específicos.

4.- Es fácil demostrar que las definiciones (1) y (2) son equivalentes por lo que sólo demostraremos que (3) es equivalente a (1):

1) ==> 3):

Sea t $\in$ R, entonces t[XY]$\in$R[XY] y t[R-Y]$\in$R[R-Y] y de la definición de juntura tenemos que t$\in$R[XY]|x|R[R-Y].

Por lo tanto hemos mostrado que R$\subset$R[XY]|x|R[R-Y].

Ahora sea t R[XY]|x|R[R-Y] entonces t[XY] $\in$ R[XY] y t[R-Y]$\in$ R[R-Y] esto implica que existen eneadas u,v $\in$ R tales que u[XY]= t[XY] y v[R-Y]=t[R-Y] entonces dado que X->->Y se satisface en R y que u[X]=v[X] tenemos que existe w $\in$ R tal que w[Y]=u[Y] y w[R-X-Y]=v[R-X-Y] pero u[Y]=t[Y] por lo tanto w=t $\in$ R. L.Q.Q.D.

3) ==> 1):

Sí R[XY]|x|R[R-Y]=R sean s,t $\in$ R tales que s[X]=t[X] pero s[Y]=/t[Y] esto implica que existen u1,v1 $\in$ R[XY] tales que u1= s[XY] y v1=t[XY] además existen u2,v2 $\in$ R[R-Y] tales que u2[X]= v2[X] y u2=s[R-Y], v2=t[R-Y] entonces por la definición de |x| tenemos que deben existir eneadas u,v R tales que u[XY]=v1, u[R-Y]=u2 y v[XY]=u1,v[R-Y]=v2 es decir:

1.-u[X]=v[X]=t[X]=s[X]

2.-u[Y]=v1[Y]=t[Y], u[R-X-Y]=u2[R-X-Y]=s[R-X-Y]

3.-v[Y]=u1[Y]=s[Y], v[R-X-Y]=v2[R-X-Y]=t[R-X-Y]. L.Q.Q.D.

3.3.1 Axiomas para Dependencias Multivaluadas.

En abstracción a las propiedades de las dependencias multivaluadas podemos obtener los siguientes axiomas:

Sea U un conjunto de atributos, en el que se satisfacen las hipótesis de los siguientes axiomas, con X,Y,V y W$\subset$U.

MD1.- (Complementación): X->->Y ==> X->->U-X-Y.

MD2.- (Aumento): X->->Y y V$\subset$W ==> WX->->YV.

MD3.- (Transitividad): X->->Y y Y->->Z ==> X->->Z-Y.

MD4.- X-->Y ==> X->->Y.

MD5.- X->->Y, Z$\subset$Y, W$\cap$Y=$\emptyset$, W-->Z ==> X-->Z.

En adelante usaremos estos axiomas como únicas reglas de inferencia para las dependencias multivaluadas.

OBSERVACIONES: 1.- El axioma MD4 nos dice que las dependencias funcionales son estrictamente un caso especial de las DM.

2.- Las dependencias multivaluadas son un modelo que cumple solamente con dos axiomas de Amstrong.

Ya que la proyectividad de las dependencias multivaluadas se sigue del axioma MD4, mientras que el aumento es prácticamente el mismo para ambos tipos de dependencias, sin embargo, la transitividad para dependencias funcionales no es satisfecho por las dependencias multivaluadas.

AMBITO DE APLICACION DE LOS AXIOMAS: Estos axiomas al igual que los de Amstrong, solo los aplicaremos para deducir nuevas dependencias a partir de un conjunto dado de dependencias G en un conjunto de atributos U, los problemas que surjan para encontrar las dependencias válidas en un subconjunto Z de U (es decir de una descomposición del esquema relacional definido por U y las dependencias G), son de otra naturaleza por lo que continuaremos la numeración llevada hasta el momento, la justificación para no usar otro tipo de numeración será dada pronto por el teorema de consistencia y completez para un conjunto de dependencias funcionales y multivaluadas D, que reune los axiomas de Amstrong con los de dependencias multivaluadas.

LEMA 3 (Ullman [1]): Las dependencias multivaludas tienen las siguientes propiedades:

i) (Unión) X->->Y y X->->Z ==> X->->YZ.

ii) (Pseudotransitividad) X->->Y y WY->->Z ==> WX->->Z-WY.

iii) (Pseudotransitividad Mixta) X->->Y y XY-->Z ==> X-->Z-Y

iv) (Descomposición) X->->Y y X->->Z ==> X->->Y $\cap$ Z, X->->Y-Z y X->->Z-Y.

DEMOSTRACION (Beeri,Fagin y Howard [4]):

i) (Unión) X->->Y y X->->Z ==> X->->YZ.

X->->Y ==>(MD2) X->->XY, X->->Z ==>(MD2) XY->->YZ ==>(MD1) XY->->U-X-YZ.

Por lo tanto X->->XY y XY->->U-X-YZ==>(MD3) X->->(U-X-YZ)-XY =U-X-YZ ==>(MD2) X->->U-YZ ==> X->->YZ por MD1 y MD2. L.Q.Q.D.

ii) (Pseudotransitividad) X->->Y y WY->->Z ==> WX->->Z-WY.

X->->Y ==>(MD2) WX->->WY, entonces X->->Y y WY->->Z ==> WX->->WY y WY->->Z ==>(MD3) WX->->Z-WY. L.Q.Q.D.

iii) (Pseudotransitividad Mixta) X->->Y y XY-->Z ==> X-->Z-Y.

X->->Y ==> (MD1): X->->U-X-Y, Z-X-Y $\subset$ U-X-Y, XY $\cap$ U-X-Y= $\emptyset$, XY-->Z-X-Y (de XY-->Z-Y y la propiedad de descomposición para DF) ==> (MD5): X-->Z-X-Y ==> (FD3): X-->Z-Y. L.Q.Q.D.

iv) (Descomposición) X->->Y y X->->Z ==> a) X->->Z-Y b) X->->Y-Z y c) X->->Z $\cap$ Y.

a) X->->Z-Y.

X->->Y y X->->Z ==>(por i)) X->->ZY ==> (MD1) X->->(U-X-YZ)

==> (MD2) X->->(U-YZ). Por lo tanto X->->(U-YZ) y X->->Y ==> (por i)) X->->(U-YZ)Y=U-(Z-Y) ==> (MD1) X->->U-X-(U-(Z-Y)) ==> (MD2) X->->U-(U-(Z-Y))=Z-Y. Por lo tanto X->->Z-Y. L.Q.Q.D.

b) X->->Y-Z

Lo mismo que (a). L.Q.Q.D.

c) X->->Y$\cap$Z.

X->->Y y X->->Z ==> X->->Y-Z y X->->Z-Y ==> (por i)) X->->(Y-Z)(Z-Y), además X->->Y y X->->Z ==> (por i)) X->->ZY. Por lo tanto X->->(Y-Z)(Z-Y) y X->->ZY ==> X->->ZY-(Y-Z)(Z-Y)=Z$\cap$Y.

L.Q.Q.D.

TEOREMA 4 (Ullman [1]): "Sean D un conjunto de DF y DM, válidas en un conjunto finito de atributos U y X$\subset$U. Entonces podemos partir U-X en conjuntos Y1,Y2,...,Yk, tales que si Z$\subset$U-X entonces X->->Z sí y sólo si Z es la unión de algunos de los Yi."

DEMOSTRACION (Ullman [1]):

Construiremos la partición deseada, empezando con el propio U-X como primera aproximación. Supongamos que en algún punto tenemos la partición W1,W2,...,Wn y X->->Wi para toda i $\in$ [1,..., n]. Sí X->->Z, y Z no es la unión de algunos Wi. reemplacemos cada Wi tal que Wi$\cap$Z y Wi-Z=/$\emptyset$ por Wi$\cap$Z y Wi-Z. Por la propiedad de la descomposición X->->Wi$\cap$Z y X->->Wi-Z. Como no podemos partir un conjunto finito de atributos indefinidamente, este proceso debe terminar, de tal modo que para todo conjunto Z tal que X->->Z existen elementos de la partición tales que su unión es igual a Z. Por la regla de la unión, X multidetermina la unión de cualquier conjunto de elementos de esta partición. Por último, por la regla de descomposición tenemos que Yi$\cap$Yj= $\emptyset$ si i=/j ya que si Yi$\cap$Yj=/$\emptyset$ entonces existe Z= Yi$\cap$Yj tal que X->->Z pero Z=/ U Yi, i $\in$ I para toda I elemento del conjunto potencia de las Yi. CONTRADICCION! L.Q.Q.D.

DEPENDENCIAS BASICAS: Al conjunto de Yi a que se refiere el teorema 5 se le conoce como "las dependencias básicas para X con respecto a D".

TEOREMA 5 (Ullman [1]): "Los Axiomas MD1-MD5 y de Amstrong forman un conjunto consistente y completo de axiomas para dependencias funcionales y multivaluadas. Es decir, si D es un conjunto de dependencias funcionales y multivaluadas sobre un conjunto de atributos U y D+ el conjunto de DF y DM que se siguen lógicamente de D (en el sentido de que toda relación que satisface D satisface también D+), entonces D+ es exactamente el conjunto de DF y DM obtenidas mediante los axiomas MD1-MD5 y de Amstrong."

DEMOSTRACION Consistencia (Ullman [1]):

Sea D un conjunto de DF y DM de un esquema relacional R* con atributos U. Sea R una instancia cualquiera de R* y sea f una dependencia multivaluada o funcional, obtenida de D mediante los axiomas de MD1-MD5.

Si se usó proyectividad (FD1), entonces f es de la forma $X \longrightarrow Y$ con $Y \subset X$, $X \subset U$, si $t,s \in R$ con $t[X]=s[X]$, entonces y puesto que $Y \subset X$: $t[Y]=s[Y]$, es decir $X \longrightarrow Y$ se cumple en R.

Si se usó transitividad (FD2), entonces f es de la forma $X \longrightarrow Y$ y existen dependencias $X \longrightarrow Z$, $Z \longrightarrow Y$ en D, luego si $t,s \in R$ con $t[X]=s[X]$, tenemos que $t[Z]=s[Z]$, y $t[Y]=s[Y]$, es decir $X \longrightarrow Y$ se cumple en R.

Si se usa aumento (FD3), entonces f es de la forma $X \longrightarrow Y$ y existe $V \longrightarrow W$ en D y $Z \subset U$, tales que $VZ=X$ y $WZ=Y$. Entonces si $t,s \in R$ y $t[X]=s[X]$ tenemos que $t[V]=s[V]$ y $t[Z]=s[Z]$ pues $X=VZ$, además $t[W]=s[W]$ pues $V \longrightarrow W$, entonces tenemos que $t[VZ]=t[X]=s[VZ]=s[X]$ y $t[ZW]=t[Y]=s[ZW]=s[Y]$, es decir $X \longrightarrow Y$ se cumple en R.

Si usamos MD1, f es de la forma $X \to\to W$ y existe $X \to\to Y$ en D con $W=U-X-Y$.

Si $t,s \in R$ y $t[X]=s[X]$ entonces existe $u \in R$ tal que $u[X]=s[X]=t[X]$, $u[Y]=t[Y]$ y $u[U-X-Y]=s[U-X-Y]$, claramente la definición de DM es simétrica con respecto a X de donde $X \to\to W$ es válida en R.

Si usamos MD2, f es de la forma $X \to\to W$ y existen $V \to\to Y$ en D y $Z \subset T \subset U$, con $W=YZ$.

Si $t,s \in R$ con $t[TV]=s[TV]$, como $V \to\to Y$ entonces existe u1 tal que $u1[V]=t[V]=s[V]$, $u1[Y]=t[Y]$ y $u1[U-V-Y]=s[U-V-Y]$, por lo tanto $u1[TV]=t[TV]=s[TV]$ ya que:

a) Si $T \subset U-V-Y$ y $u1[U-V-Y]=s[U-V-Y]$ y tenemos que $t[TV]=s[TV]$, de donde $u1[TV]=t[TV]=s[TV]$.

b) Si $T \not\subset U-V-Y$ entonces de $T \cap (U-V-Y)$ se encarga el inciso (a) y sólo nos ocuparemos de $T \cap VY$, pero $t[TV]=s[TV]$ y $t[VY]=u1[VY]$, Por lo tanto $u1[T \cap VY]=t[T \cap VY]=s[T \cap VY]$, luego $u1[TV]=t[TV]=s[TV]$. Por lo tanto $u1[YZ]=t[YZ]$ y $u1[U-TV-YZ]=s[U-TV-YZ]$.

Esto es $TV=X \to\to YZ=W$ es válida en R.

Si se sigue de MD3 (Ullman [1]), f es de la forma $X \to\to W$ y entonces existen $X \to\to Y$ y $Y \to\to Z$ en D con $Z-Y=W$. Supongamos que $X \to\to Z-Y$ no se cumple ==> Existe $t,s \in R$ tales que $t[X]=s[X]$ pero la eneada "u" tal que $u[X]=t[X]=s[X]$, $u[Z-Y]=t[Z-Y]$ y $u[U-X-(Z-Y)]=s[U-X-(Z-Y)]$ no está en R.

Como $X \to\to Y$ ==> Existe $v \in R$ tal que $v[X]=t[X]=s[X]$, $v[Y]=s[Y]$ y $v[U-X-Y]=t[U-X-Y]$ además como $Y \to\to Z$ ==> Existe $w \in R$ tal que $w[Y]=s[Y]=v[Y]$, $w[Z]=v[Z]$ y $w[U-Y-Z]=s[U-Y-Z]$.

$w[X]=t[X]$, ya que en los atributos de Z X, w coincide con v la cuál coincide con t en X, y en los atributos de X-Z, w coincide con t puesto que:

$X-Z \subset U-Y-Z$ ó $X-Z \subset Y$ ya que si $X-Z \not\subset (U-Y-Z)Y$ ==> $X-Z \not\subset U-Z$ ==> $X \not\subset U$ CONTRADICCION!

a) $X-Z \subset U-Y-Z$, como $w[U-Y-Z]=s[U-Y-Z]$ ==> $w[X-Z]=s[X-Z]$ pero $s[X]=t[X]$ por lo tanto $w[X-Z]=t[X-Z]$.

b) $X-Z \subset Y$, como $w[Y]=v[Y]$ y $v[X]=t[X]$ ==> $w[X-Z]=t[X-Z]$.
Por lo tanto $w[X]=t[X]$.

w[Z-Y]=t[Z-Y], puesto que w coincide con v en Z-Y pues w[Z]=v[Z] y v coincide con t en Z-Y ya que:

a) v[U-X-Y]=t[U-X-Y] y Z-Y U-X-Y

ó b) $Z-Y \subset X$ y ya vimos como w[X]=t[X].

Sea V=U-X-(Z-Y).

w[V-Z]=s[V-Z], puesto que V-Z=U-X-Z y w[U-Z]=s[U-Z] pues w[Y]= s[Y]. Además $V \cap Z=(Y\ \ Z)-X$, pero w coincide con v en Z y v[Y]=s[Y] ==> $w[Y \cap Z]=v[Y \cap Z]=s[Y \cap Z]$. Así $w[V \cap Z]=s[V \cap Z]$.

Por lo tanto w[V]=s[V]. Si vemos la definición de u nos damos cuenta que u=w CONTRADICCION! pues $w \in R$.

De donde X->->Z-Y=W es válida en R.

Si se sigue de MD4, f es de la forma X->->W y X-->W está en D.

Si X-->W se cumple entonces no existen t,s U tales que t[X]=s[X] y t[W]=/s[W]. Supongamos que X->->W no se cumple, es decir, existen $t,s \in U$ tales que t[X]=s[X] pero no existe $u \in U$ tal que u[X]=t[X]=s[X] con u[W]=t[W] y u[U-X-W]=s[U-X-W]; como X-->W ==> haciendo u=s tenemos s[X]=t[X]=s[X], s[W]=t[W] y s[U-X-W]= s[U-X-W]. CONTRADICCION!. Por lo tanto X->->W se cumple en R.

Si se usa MD5, f es de la forma X-->W y se cumplen las siguientes condiciones: X->->Y en D, $Y \cap V=\emptyset$, $W \subset Y$, V-->W.

Como X->->Y se cumple en R, si $t,s \in R$ con t[X]=s[X], tenemos que existen $u,v \in R$ tales que:

1.- u[X]-t[X]=v[X]=s[X].

2.- t[Y]=u[Y], s[U-X-Y]=u[U-X-Y].

3.- s[Y]=v[Y], t[U-X-Y]=v[U-X-Y].

Como V-->W y $V \subset U-Y$, $W \subset Y$ ==> t[W]=u[W], s[V]=u[V], s[W]= v[W], t[V]=v[V].

Por demostrar: s[W]=v[W]=t[W]=u[W].

Podemos suponer sin pérdida de generalidad de que $X \cap W=\emptyset$, pues s[X]=v[X]=t[X]=u[X].

Si t[W]=/s[W] ==> t[V]=/s[V], pero u[V]=s[V] y por V-->W ==> u[W]=s[W] pero u[W]=t[W] CONTRADICCION!.

Por lo tanto t[W]=s[W]=u[W]=v[W]. Es decir, X-->W es válida en R.

Como esto se cumple para cualquier instancia R de R* tenemos que los axiomas son consistentes.

L.Q.Q.D.

Completez (Beeri,Fagin y Howard [6]):

El teorema de completez "débil" fué establecido por Beeri, Fagin y Howard como sigue:

"Sean F y G conjuntos de DF y DM respectivamente (en un conjunto U). Para cada DF o DM que no pertenece a (F,G)+, existe una relación R con atributos U tal que todas las dependencias en (F,G)+ son válidas, pero la dependencia dada no lo es."

Notación: Por Z(R,x) entenderemos $\{z \in Z : t \in R,\ t[Z]=z \text{ y } t[X]=x\}$

Sea X el lado izquierdo de la dependencia que no estå en (F,G)+. El conjunto X+ es un subconjunto propio de U, ya que, de otra forma, toda dependencia (F ó M) con el lado izquierdo X, pertenecerían a (F,G)+.

Sean W1,...,Wm (m >=1) el conjunto de las dependencias båsicas de X que cubren U-X+ (esto siempre se puede hacer puesto que las dependencias båsicas cubren U, son ajenas e incluyen todas las DF en que X aparece al lado izquierdo esto es si A ∈ X+ ==> X-->A ==> X->->A. Por lo tanto "A" es una dependencia båsica de X). Así X+, W1,...,Wm forman una partición de U.

Las DM X->->X+,W1,W2,...,Wm estån en (F,G)+ y además, si una DM en (F,G)+ tiene a X como su lado izquierdo entonces su lado derecho es la unión de un subconjunto de X+ y algunos de los conjuntos W1,...,Wm (Por la definición de dependencias båsicas y X+).

La relación R se construye como sigue: Escogemos al conjunto {0,1} como el dominio de cada uno de los atributos en U. La relación R tiene dos a la m renglónes, un renglón para cada secuencia de ceros y unos de longitud m. En el renglón correspondiente a la secuencia <a1,...,an> (con a(i) {0,1}), cada uno de los atributos en Wi tiene asignado el valor a(i), con i∈ {1,...,m}.

Cada atributo en X+ tiene asignado el valor 1 en todos los renglónes. Esto es, por ejemplo si m=3 a la secuencia <0,0,1> le corresponde el renglón con ceros en los atributos de W1 y W2 y unos en los de W3.

Ahora mostraremos que R satisface las condiciones del teorema de completez. En lo que sigue usaremos los axiomas y demás reglas de inferencia derivadas de ellos para dos propósitos:

1.- Mostrar que si algunas dependencias están en (F,G)+, entonces dicho conjunto también contiene alguna otra dependencia.

El que podamos usar los axiomas y reglas de inferencia en esta forma se sigue directamente de la definición de (F,G)+.

2.- Mostrar que si algunas dependencias están en R entonces existe otra dependencia que es válida en R. Podemos hacer esto puesto que hemos probado que son reglas de inferencia válidas para la "familia de dependencias", es decir, su aplicación a dependencias que son válidas en esa instancia produce dependencias válidas en esa instancia. Indicaremos nuestra intención cada vez que usemos estos axiomas y reglas.

Ahora probaremos 3 afirmaciones sobre la relación R que hemos construido:

(1) Si el lado derecho de una DF es un subconjunto no vacío de Wi, entonces la DF es válida en R sí y sólo sí su lado izquierdo intersecta Wi para i {1,...,m}.

(2) Toda DM que tenga a Wi como lado derecho es válida en R i {1,...m}.

(3) Si el lado derecho de una DM es un subconjunto propio no vacío de Wi entonces la DM es válida en R sí y sólo sí su lado izquierdo intersecta Wi para i {1,...,m}.

Primero probaremos en una dirección (1). Para cada renglón dado, todos los atributos en Wi tienen el mismo valor. Se sigue que todo atributo en Wi es funcionalmente dependiente en R de cualquier otro atributo en Wi (y por aumento de cualquier otro conjunto que contenga dicho atributo). Así, hemos probado que si el lado izquierdo de la DF intersecta a Wi entonces la DF es válida en R. De aquí también se sigue la verdad en la misma dirección de la afirmación (3), ya que toda DF es también DM.

Ahora probaremos (2) y las otras direcciones de las afirmaciones (1) y (3). La relación R es el producto cartesiano de sus proyecciones R[Wi] y R[U-Wi]. Se sigue inmediatamente de la definición de DM que $\emptyset$->->Wi es válida en R, y por aumento Y->->Wi es válida en R para todo conjunto Y. Esto prueba la afirmación (2). Ahora bien, sean Y y Z conjuntos tales que Y es ajeno a Wi y Z subconjunto no vacío de Wi. Se sigue de la anterior construcción de R que para cada valor de Y digamos "y", el conjunto Z(R,y) contiene dos valores de Z (una eneada de ceros y otra de unos). En particular la DF: Y-->Z no es válida en R; esto concluye la prueba de (1), pues esta vez hemos probado que no existe DF si el lado izquierdo no intersecta a Wi. Por otra parte, si Z es un subconjunto de Wi, sea "A" un atributo de Wi-Z. Entonces si "a" es un valor de A, Z(R,ya) contiene únicamente un valor de Z puesto que los atributos de Z deben tener asignado el mismo valor que el atributo A. Por lo tanto la DM: Y->->Z no es válida en R, ya que si lo fuera aplicando MD5 (pues $A \cap Z = \emptyset$, $Z \subset Z$, A-->Z, Y->-> Z) ==> Y-->Z CONTRADICCION! a (1) de este teorema. Esto concluye la prueba de (3).

Ahora mostraremos que R satisface las condiciones del teorema. Sea "f" una DF en (F,G)+. Por FD6 podemos suponer que "f" es de la forma Y-->B donde B es un solo atributo. Ahora, si B está en X+ entonces la DF es claramente válida en R, ya que en R todo atributo de X+ toma el valor 1, y es por lo tanto funcionalmente depen. de cualquier otro conjunto. Si B no está en X+ entonces pertenece a algún Wi. Si Y es ajeno a dicho Wi entonces de la DM X->->Wi y de la DF Y-->B, las cuales están en (F,G)+ se seguiría por MD5 que X-->B está en (F,G)+ CONTRADICCION! ya que $B \notin X+$. Luego, $Y \cap Wi = \emptyset$ y Y-->B es válida en R por (1).

Sea ahora g:Y->->Z una DM en (F,G)+ notemos que $Y \text{-->} Z \cap X+$ es válida en R pues los atributos de X+ sólo toman un valor en R.

Demostraremos que para cada i, $Y \text{->->} Z \cap Wi$ es válida en R.

Supongamos que, para alguna i, el conjunto Z Wi es vacío ó todo Wi. Por (2) Y->->Wi es válida en R (ya que Y->-> $\emptyset$ siempre es válida para toda Y). Enseguida, supongamos que, para alguna i, $Z \cap Wi$ es un subconjunto propio no vacío de Wi. Si Y no intersecta a Wi podemos usar aumento sobre Y->->Z para obtener que U-Wi->->Z está en (F,G)+ esto contradice el hecho de que Wi sea una dependencia básica de X. Así Y debe intersectar a Wi y por (3) $Y \text{->->} Z \cap Wi$ es válida en R. Por lo tanto hemos demostrado que para cada i, $Y \text{->->} Z \cap Wi$ es válida en R y que también lo es $Y \text{->->} Z \cap X+$. Tomando la unión para DM (propiedad (i)), se sigue que Y->->Z es válida en R.

Finalmente, consideremos la depen. (con lado izquierdo X) que se sabe no está en (F,G)+. Si es una DF: X-->Y entonces Y no es un subconjunto de X+, así Y intersecta Wi para alguna i. Por FD6 si X-->Y es válida en R también lo es $X \text{-->} Y \cap Wi$ y esto contradice (1) pues $X \cap Wi = \emptyset$. Por lo tanto X-->Y no puede ser válida en R.

Si la dependencia es X->->Y, entonces para alguna i, $Y \cap Wi$ debe ser un subconjunto propio no vacío de Wi (de otra forma puesto que $X \text{-->} Y \cap X+$ y $X \text{->->} Y \cap Wi = Wi$ o X->-> $\emptyset$ están en (F,G)+

para toda i=1,...,m, la DM X->->Y estaría en (F,G)+). Para esta i, X->->Y∩Wi no es válida en R por (3). Puesto que X->->Wi es válida en R si X->->Y fuera también válida en R podríamos aplicar la propiedad de intersección para DM y obtener X->->Y∩Wi contradiciendo al hecho de que Wi está en las dependencias básicas de X. Así X->->Y no es válida en R. L.Q.Q.D.

OBSERVACIONES : Dado un conjunto de DF y DM siempre podremos encontrar su "cerradura" es decir, el conjunto de DF y DM lógicamente implicadas por él usando los axiomas para DF y DM.

ALGORITMO II (Ullman [1]): Comprobación de Descomposiciones Sin Pérdida para Esquemas con Dependencias Multivaluadas.

ENTRADA: Un esquema relacional R* con dependencias funcionales y multivaluadas D y una descomposición @ de R*.

SALIDA: Si @ es o no una descomposición sin pérdida.

METODO: 1.- Construir la tabla T de aes y bes del algoritmo para descomposiciones sin pérdida de esquemas con DF.

2.- Si se halla una dependencia funcional X-->Y, seguir el algoritmo mencionado antes. Si se encuentra una DM: X->->Y tómense dos renglones t1 y t2 tales que t1[X]=t2[X] y agréguese el renglón u, donde u[X]=t1[X]=t2[X], u[Y]=t1[Y] y u[U-X-Y]= t2[U-X-Y], si u no está en T.

3.- Como no se crean aes o bes este proceso debe terminar. Si existe un renglón formado con aes, entonces la descomposición es sin pérdida, de otra forma no.

TEOREMA II : El algoritmo anterior determina si una descomposición es sin pérdida con respecto a un esquema R* con atributos U y dependencias funcionales y multivaluadas D.

DEMOSTRACION: La demostración en su parte correspondiente a la consistencia es similar a la del teorema 3 y por supuesto presenta las mismas dificultades en la segunda parte.

TEOREMA III (Ullman [1]): "Sea R* un esquema relacional y @=(R1*,R2*) una descomposición en dos partes de R*. Sea D un conjunto de DF y DM sobre los atributos de R*. Entonces @ es una descomposición sin pérdida si y sólo si R1*∩R2*->->R1*-R2* o R1*∩R2*->->R2*-R1*."

DEMOSTRACION (Ullman [1]): @ es una descomposición sin pérdida si y sólo si para cualquier relación R que satisfaga R*, y cualesquiera dos eneadas t y s en R, la eneada u tal que u[R1*]=t[R1*] y u[R2*]=s[R2*] está en R. Ya que de otro modo u ∈ M@(R) pero u ∉ R y la descomposición no sería sin pérdida. Pero "u" existe si y sólo si t[R1*] ∈ R[R1*], s[R2*] ∈ R[R2*] y (por la definición de juntura natural) t[R1*∩R2*]=s[R1*∩R2*]. Así, la condición de que "u" esté siempre en R es exactamente la

condición de que R1*∩R2*->-> R1*-R2* (ó equivalentemente R1*∩R2*->->R2*-R1*), ya que si R1*∩R2*->->R1*-R2* y t[R1*∩R2*]=s[R1*∩R2*] con t,s ∈ R ==> Existe u∈R tal que:
u[R1*∩R2*]=t[R1*∩R2*]=s[R1*∩R2*], u[R1*-R2*]=t[R1*-R2*] y u[R*-(R1*∩R2*)-(R1*-R2*)]=s[R*-(R1*∩R2*)-(R1*-R2*)] y como: R*-(R1*∩R2*)-(R1*-R2*)=R*-R1* ==> u[R*-R1*]=s[R*-R1*].
Además u[R1*∩R2*]=t[R1*∩R2*]=s[R1*∩R2*] tenemos que:

a) u[R1*]=t[R1*] pues
u[(R1*∩R2*)(R1*-R2*)]=t[(R1*∩R2*)(R1*-R2*)].

b) u[R2*]=s[R2*] pues
u[(R1*∩R2*)(R*-R1*)]=s[(R1*∩R2*)(R*-R1*)]. L.Q.Q.D.

3.4 FORMAS NORMALES DE ESQUEMAS RELACIONALES

Las llamadas "formas normales de esquemas relacionales" son definiciones de esquemas relacionales basadas en el hecho de que satisfacen determinadas condiciones con respecto a algún tipo de dependencia entre sus atributos, en esta sección estudiaremos únicamente las formas normales basadas en las dependencias funcionales y multivaluadas.

PRIMERA FORMA NORMAL (1FN): Un esquema relacional R* está en 1FN <==> Para cualquier instancia R, todas las eneadas tienen componentes con un sólo valor (i.e. un conjunto no puede ser componente de una eneada).

SEGUNDA FORMA NORMAL (2FN): Un esquema relacional R* está en 2FN <==> Está en 1FN y no existe un atributo no primo "A" y una llave X con un subconjunto propio Y que satisfaga Y-->A. Se dice que la 2FN elimina las "dependencias parciales".

TERCERA FORMA NORMAL (3FN): Un esquema relacional R* está en 3FN <==> Está en 2FN y no existe un atributo no primo "A" y un conjunto Y de atributos de R* tales que A∉Y y Y-->A sin que Y contenga una llave para R*. Se dice que la 3FN elimina las "dependencias transitivas funcionales".

FORMA NORMAL BOYCE & CODD (BCFN): Un esquema relacional R* está en BCFN si para todo conjunto de atributos X de R* tales que X--->A, con A en R* y A X ==> X contiene al menos una llave de R*.

CUARTA FORMA NORMAL (4FN): Un esquema relacional R* está en 4FN <==> Para todos los conjuntos propios X,Y de R* con Y⊄X, Y=/∅, XY=/R* y X->->Y ==> X contiene al menos una llave de R*.

OBSERVACIONES: 1.- De acuerdo a nuestra definición de relación no puede haber un esquema relacional que no esté al menos en 1FN.

2.- Nótese la "independencia" de la definición de BCFN con respecto a las anteriores formas normales, la justificación para colocarla en ese orden se da en el lema 5 siguiente.

3.- Si XY=R* tendremos que X->->Y para cualquier "partición" de R* en dos subesquemas. Puesto que: ∅ ⊂ X ==> (FD1): X-->∅ ==> (MD4): X->->∅ ==> (MD1): X->->U-X-∅ = U-X = Y-X, pero Y∩

$X \subset X$ ==> (MD2): X->->Y. L.Q.Q.D.

4.- Si un esquema R* sólo tiene DF y si está en BCFN entonces también está en 4FN (Ullman [1]), ya que si X-->Y se cumple en R* entonces también se cumple X->->Y y por estar en BCFN, X incluye una llave.

LEMA 4 (Ullman [1]): "Si un esquema relacional R* con dependencias funcionales F está en BCFN, entonces está en 3FN."

DEMOSTRACION (Ullman [1]):(Por Contradicción).

Supongamos que R* está en BCFN pero no en 3FN entonces existe una dependencia parcial o transitiva X-->Y-->A, donde X es una llave para R*, "A" no está en X ni en Y y Y-->X no está en F+. Luego Y no incluye una llave para R*, ya que si la incluyera, Y-->X estaría en F+, pues una llave determina cualquier conjunto de atributos de su esquema relacional. Pero "A" no está en Y, por lo que Y-->A viola la condición de BCFN. CONTRADICCION! pues ya habíamos supuesto que R estaba en BCFN.

L.Q.Q.D.

LEMA 5 (Ullman [1]): "Si un esquema relacional R* está en 4FN, entonces está en BCFN."

DEMOSTRACION:(Por contradicción):

Supongamos que el teorema es falso, es decir, que existe un esquema R* que está en 4FN pero no en BCFN.

SI R* no está en BCFN entonces existe en R* una dependencia funcional no trivial X-->Y, y un atributo "A" tal que X-->A no se cumple en R* CONTRADICCION! ya que por el axioma MD4 : X-->Y ==> X->->Y y por estar R* en 4FN tenemos que X debe contener una llave.L.Q.Q.D.

Es decir 4FN es estrictamente más fuerte que BCFN.

3.4.1 Ventajas de las Primeras Formas Normales:

A continuación presentaremos las mejoras sucesivas de las formas normales.

1FN: Permite el empleo de los operadores relacionales y es la base para el proceso de optimización.

2FN: Al eliminar las dependencias parciales ahorra espacio, ya que si tenemos la dependencia parcial Y-->A, donde X es una llave y $Y \subset X$, entonces todas las eneadas con el mismo valor para los atributos Y repiten la misma asociación Y-->A.

Por ejemplo: La siguiente relación posee la dependencia parcial: CVE. ALUMNO --> NOMBRE ALUMNO.

MATERIAS POR ALUMNO

SEMESTRE	CVE. ALUMNO	NOMBRE ALUMNO	CVE. MATERIA
84-84	847345-233	HECTOR PARRA	101000-001
84-84	847345-233	HECTOR PARRA	203040-002
84-85	845788-311	LIDIA LOPEZ	621077-001
84-85	845788-311	LIDIA LOPEZ	601101-002

3FN: Ataca cierto tipo de "restricciones" impuestas a la información que desea guardarse, por esquemas relacionales que permiten la existencia de las llamadas dependencias transitivas. Se trata de dependencias funcionales de atributos no primos A con respecto a conjuntos que no contienen una llave Y ($Y \not\subset X$) y esto obliga a conocer la asociación Y-->A cada vez que se pretenda guardar información de la dependencia X-->Y, para alguna llave X. Por ejemplo: La siguiente relación posee la dependencia transitiva: CVE. MAESTRO --> NOMBRE MAESTRO.

CLASES IMPARTIDAS EN EL SEMESTRE

CVE. MATERIA	CVE. GRUPO	CVE. MAESTRO	NOMBRE MAESTRO
44017221-001	01	8754316-9	DANIEL ESTRADA C.
44017221-001	02	8132991-9	LETICIA VERDUGO M.
10010012-001	01	7815132-10	HERNAN QUIJADA S.
10010012-001	02	8412513-10	IRMA SANDOVAL T.

BCFN: Va más allá de la 3FN pues elimina las dependencias transitivas incluyendo la de los atributos primos, con lo cuál elimina aún más las falsas restricciones inducidas por los esquemas relacionales mal definidos a la información "almacenable".

Por ejemplo: La siguiente relación en 3FN no está en BCFN por la dependencia entre atributos primos: MATERIA --> DEPTO.

MATERIAS POR DEPTO. Y CARRERA

DEPTO.	CARRERA	MATERIA
MATEM.	LM	CALCULO I
MATEM.	LF	CALCULO I
FISICA	LM	MECANICA
FISICA	LF	MECANICA

4FN: Su ventaja es que ahorra espacio al considerar un tipo más general de dependencia entre atributos como base para la descomposición de los esquemas relacionales.

Por ejemplo: La siguiente relación está en BCFN pero no en 4FN, pues se da la dependencia: (CVE.MAT.,CVE.MAESTRO) ->-> (DIA-HR.,SALON).

ALUMNOS POR GRUPO POR MAESTRO

CVE. MAT.	CVE. MAESTRO	DIA-HR.	SALON	CVE. ALUMNO
101000-001	7512310-109	LUN-17	E-201	8404121-9
101000-001	7512310-109	VIE-16	E-201	8404121-9
101000-001	7512310-109	LUN-17	E-201	8404733-11
101000-001	7512310-109	VIE-16	E-201	8404733-11

3.4.2 Proceso de Normalización:

LLamamos "proceso de normalización" al proceso de descomponer un esquema relacional no normalizado o normalizado en una forma normal menos "eficiente", en un conjunto de esquemas relacionales que satisfacen una forma normal superior. Como vemos sería más propio hablar de "Procesos de Normalización" puesto que se trata de varios algoritmos de descomposición, al menos uno para cada forma normal. A continuación presentamos dichos algoritmos y algunos resultados con respecto a las formas normales que aparecen en Ullman [1].

LEMA II (Ullman [1]): "No todo esquema relacional puede descomponerse en esquemas BCFN y que preserve dependencias."

DEMOSTRACION (Por Contradicción):

Supongamos que el lema es falso. Consideremos el esquema ABC con dependencias funcionales F={AB-->C,C-->A}. Claramente no está en BCFN puesto que C-->A pero C no es una llave de ABC ya que C-/->B. Consideremos todas las posibles descomposiciones de ABC (que no incluyan al propio ABC y que estén en BCFN):

{A,B,C}=@1, {AB,AC}=@2, {AB,BC}=@3, {A,BC}=@4, {B,AC}=@5, {C,AB}=@6, {AC,BC}=@7

Probaremos que ninguna de las descomposiciones satisface ser preservadora de dependencias.

Afirmamos que F+={AB-->C, C-->A, CB-->AB, AB-->CB, AB-->CA, CB-->A, + dependencias triviales}

Enseguida analizaremos cada una de las particiones, obteniendo las dependencias proyectadas para luego ver si su unión determina lógicamente F.

@1: F[A]={A-->A},F[B]={B-->B},F[C]={C-->C}

Claramente la unión de estas dependencias no puede implicar lógicamente a F.

@2: F[AB]={AB-->AB,AB-->A,AB-->B,B-->B,A-->A}.

F[AC]={C-->A,AC-->A,AC-->C,AC-->AC,A-->A,C-->C}

Vemos que la unión de estas dependencias no pueden implicar AB-->C que está en F

@3: F[AB] Como en @2.

F[BC]={BC-->BC,BC-->B,BC-->C,B-->B,C-->C}

Trata sólo de dependencias triviales por lo que no pueden generar a F.

@4: Ya tenemos F[A],F[BC] y obviamente estas no implican F.

@5: F[B]F[AC] =/=> F

@6: F[C]F[AB] =/=> F

@7: F[AC]F[BC]=/=> F. L.Q.Q.D.

LEMA III (Ullman [11]):

"a) Sea R^* un esquema relacional con dependencias funcionales F. Sea @= {R1*,...,Rk*} una descomposición sin pèrdida de R* con respecto a F. Para cada i sea Fi=F[Ri*] y sea &={S1*,...,Sm*} la descomposición sin pèrdida de Ri con respecto a Fi. Entonces la descomposición de R* en {R1*,...,R(i-1)*, S1*,...,Sm*,R(i+1)*,...,Rk*} es sin pèrdida con respecto a F.

b) Sean R*,F y @ como en a), y sea &={R1*,...,Rk*, R(k+1)*,...,Rn*} una descomposición de R* en un conjunto de esquemas relacionales que incluyen los de @. Entonces & es sin pèrdida con respecto a F."

DEMOSTRACION:

a) Escribamos la definición de Juntura Natural Generalizada en forma conjuntista:

$$\underset{i=1}{\overset{r}{:x:}} Ri \;=\; \{t/t[Ri^*] \;\; Ri \text{ para toda } i \text{ de } 1 \text{ a } r\}$$

Trivialmente vemos que es una operación conmutativa, puesto que el orden de los Ri es indiferente.

Demostraremos ahora que también es asociativa.

Sea k<r entonces:

$$(\underset{i=1}{\overset{k}{:x:}} Ri) :x: (\underset{n=k+1}{\overset{r}{:x:}} Rn) = R1 :x: R2$$

$$\begin{aligned}
R1 &= \{u/u[Ri^*] \in Ri \text{ para toda } i \text{ de } 1 \text{ a } k\} \\
R2 &= \{s/s[Rn^*] \in Rn \text{ para toda } n \text{ de } (k+1) \text{ a } r\} \\
\text{Pero } R1:x:R2 &= \{u/u[R1^*] \in R1 \text{ y } u[R2^*] \in R2\} \\
&= \{u/(u[R1^*])[Ri^*] \in Ri \text{ para toda } i \text{ de } 1 \text{ a } k \text{ y} \\
&\qquad (u[R2^*])[Rn^*] \in Rn \text{ para toda } n \text{ de } (k+1) \text{ a } r\} \\
&= \{u/u[Ri^*] \in Ri \text{ para toda } i \text{ de } 1 \text{ a } k \text{ y } u[Rn^*] \\
&\qquad Rn \text{ para toda } n \text{ de } (k+1) \text{ a } r\} \\
&= \{u/u[Ri^*] \in Ri \text{ para toda } i \text{ de } 1 \text{ a } r\} = R.
\end{aligned}$$

Por demostrar: (u[R1*])[Ri*] = u[Ri*].

Esta igualdad se sigue inmediatamente del hecho de que los atributos en u[R1*] son subconjunto de los atributos en u.

De la asociatividad de la juntura natural se sigue directamente el inciso (a) del lema. L.Q.Q.D.

b) Sea &1={R1*,...,Rk*} y &2={R(k+1)*,...,Rn*}. Entonces:

$$M\&(R) = M\&1(R) :x: M\&2(R) = R :x: R'$$

Por lo tanto:

$$\begin{aligned}
R:x:R' &= \{t/t[R^*] \in R,\ t[R'^*] \in R'\} = \{t/t \in R,\ t[R'^*] \in R'\} \\
&= \{t/t \in R\} = R.
\end{aligned}$$

Ya que: R' = {u/u[Rj*] ∈ Rj=R[Rj*], j ∈ (k+1)-n}.

Por lo tanto si u R entonces u[R'*] ∈ R'.

Luego {t/t R, t[R'*] ∈ R'} = {t/t ∈ R}= R = M&(R). L.Q.Q.D.

ALGORITMO III (Ullman [1]): Descomposición sin pérdida en BCFN.

ENTRADA: Esquema relacional R* con dependencias funcionales F.

SALIDA: Una descomposición @ de R* sin pérdida, tal que todo esquema Ri* @ está en BCFN.

METODO: 1.- @(0)={R*}

2.- Si S* @(i) tal que S* no está en BCFN, entonces debe existir una dependencia X-->A, donde A ∉ X y X no contiene una llave de S*, hacemos:

@(i+1)=(@i{S1*,S2*})-S* con S1*=XA, S2*=S*-A.

3.- Continuar hasta que no exista S* ∈ @(k) que tenga una dependencia transitiva. En ese caso @(k) será una descomposición sin pérdida en BCFN.

TEOREMA IV: El algoritmo 3 obtiene correctamente una descomposición sin pérdida de un esquema R* con dependencias F en esquemas en BCFN.

DEMOSTRACION: Como la descomposición @(0) consiste de R* mismo, trivialmente @(0) es sin pérdida con respecto a F. Además, si @(i) es una descomposición sin pérdida de R* con respecto a F y existe S* @(i) tal que S* no está en BCFN, entonces existe X-->A en S* y la descomposición de S* en S1*=XA, S2*=S*-A es sin pérdida, de acuerdo al algoritmo para descomposición sin pérdida en dos esquemas relacionales, y que, como ya vimos, nos dice que la descomposición de R* en R1* y R2* es sin pérdida sí y sólo sí R1* ∩ R2*-->R1*-R2* ó R1* ∩ R2*-->R2*-R1* y en este caso S1* ∩ S2*=X-->S1*-S2*=A.

Como R* tiene un número finito de atributos, este proceso terminará y no quedará ningún determinante no llave en los esquemas de alguna descomposición @(k) que será sin pérdida. L.Q.Q.D.

ALGORITMO IV (Ullman [1]): Descomposición Preservadora de Dependencias en 3FN

ENTRADA: Un esquema relacional R* con un conjunto de dependencias F, las cuáles sin pérdida de generalidad suponemos constituyen una cubierta minimal para las dependencias de R*.

SALIDA: Una desc. de R* que preserva dependencias y tal que cada esquema relacional está en 3FN con respecto a la proyección de F sobre esos esquemas.

METODO: Si hay un atributo que no se encuentra en ninguna de las dependencias F entonces forma un esquema relacional por sí mismo por lo que podemos "eliminarlo" de R*. Pero si por alguna razón es conveniente presentarlo junto con otro(s) atrbuto(s) X, entonces podemos "inventar" un atributo Y tal que AX-->Y, para forzar formalmente esta asociación muchos a muchos. Si una de las dependencias en F involucra todos los atributos de R* la salida será R*; de otra forma la descomposición @ de salida constará de los esquemas XA para cada dependencia X-->A en F.

OBSERVACIONES : En general si tenemos las dependencias X-->A1,X-->A2,..., X-->An, usualmente es preferible substituirlos por el esquema único XA1A2...An, siempre que esto pueda hacerse sin contravenir la definición de 3FN.

TEOREMA V (Ullman [1]): "El algoritmo 3 conduce a una descomposición preservadora de dependencias en 3FN."

DEMOSTRACION (Ullman [1]):

Puesto que las dependencias obtenidas incluyen una cubierta para F, la descomposición claramente preserva dependencias. Supongamos que hay una dependencia X-->A en F que incluye todos los atributos, después de eliminar aquéllos que no están involucrados en ninguna dependencia. En este caso, afirmamos que R* está en 3FN. Primero observemos que puesto que F es una cubierta minimal, X no puede tener atributos innecesarios y como XA=R* entonces X-->R*.

Si algún subconjunto propio Y de X es tal que Y-->A contradiría la definición de cubierta minimal ya que X-->A está en F. Luego X es una llave para R*. Tenemos los casos siguientes:

CASO 1.- Supongamos que existe una dependencia transitiva V-->W-->B en R ==> $W \not\subset V$, $B \notin V$ y $B \notin W$ y de nuevo pueden ocurrir:

a) $A \notin V$ ==>$V \subset X$. Sea Z=V(X-W-B). ==> B ó algún atributo de W-V no es A por la definición de dependencia transitiva y $A \notin VX=X$ ==> $Z \subset X$.

Pero Z=V(X-W-B) y V-->W-->B, por lo tanto Z-->X CONTRADICCION! al hecho de que X es una llave.

b) $A \in V$. Sea Z como antes, de nuevo Z-->X aunque $Z \not\subset X$ ==> Z-->R, entonces existe $Z' \subset Z$ tal que es una llave de R*. SI A Z'==> R* sólo tiene atributos primos ya que también "A" forma parte de una llave. Luego R* está en 3FN por "vacuidad", esto es no existe un atributo no primo "A" tal que pueda presentarse una dependencia transitiva.

SI $A \notin Z'$ ==> $Z' \subset X$ y Z'-->X CONTRADICCION! al hecho de que X sea una llave para R*.

CASO 2.- Existe una dependencia parcial en R*, es decir un conjunto $Y \subset X$ y $B \in R*$ tales que Y-->B, Y-/->X y $B \notin Y$. entonces:

a) A=/B ==> $X-B \subset X$ y X-B-->X pués $Y \subset X-B$ y Y-->B CONTRADICCION! de nuevo al hecho de que X es una llave.

b) A=B ==> Y-->A CONTRADICCION! otra vez a que X es llave y también a que F es cubierta minimal.

Por lo tanto no existe dependencia parcial ó transitiva entre los atributos no primos de R*, de donde R* está en 3FN.

SI X-->A está en F y XA=/R entonces por el algoritmo 4 formamos el esquema XA, afirmamos que dicho esquema está en 3FN con respecto a las dependencias proyectadas de F sobre XA ya que volvemos al caso en que R=XA demostrado antes. L.Q.Q.D.

TEOREMA VI (Ullman [11]): Descomposición en 3FN Sin Pérdida y que Preserva Dependencias : "Sea @ la descomposición en 3FN de R* preservadora de dependencias obtenida con el algoritmo 4, y sea X una llave para R*. Entonces &=@U{X} es una descomposición sin pérdida de R* y que preserva dependencias."

DEMOSTRACION (Ullman [11]):

"Supongamos que existe una dependencia parcial en X, es decir, existen Y,{B}$\subset$X tales que Y-->B, y Y$\cap${B}=ϕ ==> X-B-->X pues Y$\subset$X-B y Y-->B contradiciendo el que X sea una llave de R*.

Ahora supongamos que existe una dependencia transitiva en X, es decir, existen Y, W, {B} en X tales que YUW$\subset$X-B. Por lo que X-B-->B y una vez más se contradice el que X es una llave.

Así X debe estar en 3FN como lo están todos los esquemas de @. Claramente &=@{X} preserva dependencias pues @ lo hace.

Para demostrar que & es sin pérdida usaremos la prueba tabular del algoritmo 2. Podemos demostrar que el renglón para X se vuelve todo aes como sigue:

Consideremos el orden A1,A2,...,Ak en el cual los atributos de R*-X son agregados al renglón correspondiente a X durante el algoritmo 1. Seguramente todos los atributos serán agregados, puesto que X es una llave. Mostraremos por inducción sobre "i" que la columna correspondiente a Ai en el renglón para X es convertida en a(i) durante la aplicación del algoritmo 2.

Para i=0 tenemos aes sólo en los atributos de X.

Hipótesis de Inducción: Supongamos que a(i-1) es colocada en el renglón de X en el lugar correspondiente al atributo A(i-1). Tenemos que Ai es agregado a X+ a causa de alguna dependencia funcional Y-->Ai, donde Y$\subset$X{A1,...,A(i-1)}.

Entonces YAi$\in$@ y los renglones para YAi y X coinciden en Y después de que las columnas A1,...,A(i-1) son hechas aes para el renglón X. Así estos renglones se hacen coincidir en Ai durante la ejecución del algoritmo 2. Como el renglón YAi tiene a(i) en el lugar correspondiente al atributo A(i), también lo debe tener el renglón para X. L.Q.Q.D.

TEOREMA VII (Ullman [11]): Proyección de Dependencias Funcionales y Multivaluadas.: "Sean un esquema relacional R* con dependencias funcionales y multivaluadas D y un subesquema S* de R*. El conjunto de dependencias que se cumplen en S*, es decir, la proyección de D sobre S*, puede obtenerse como sigue:

1.- Compute D+

2.- Para cada X-->Y en D+, si X$\subset$S*, entonces X-->Y$\cap$S* se cumple en S*.

3.- Para cada X->->Y en D+, si X$\subset$S*, entonces X->->Y$\cap$S* se cumple en S*.

4.- Ninguna otra dependencia para S* puede deducirse del hecho de que D se cumpla para R*.

DEMOSTRACION:

2.- Para toda X-->Y$\in$D+ y X$\subset$S* ==> X-->Y$\cap$S* está en el esquema S*. Este inciso es trivial a partir del axioma FD6 de descomposición para DF.

3.- Para toda X->->Y ∈ D+ y X⊂S* ==> X->->Y∩S* en el subesquema S*.

Supongamos que el punto 3 es falso. Entonces existe una instancia R de R* y un subesquema S* en el cuál X->->Y∩S* no se cumple ==> Existen t,s∈R[S*]=S tales que t[X]=s[X] pero no existe u ∈ S tal que u[X]=t[X]=s[X], u[Y∩S*]=t[Y∩S*] y u[S*-X-(Y∩S*)]= s[S*-X-(Y∩S*)].

Pero dado que X->->Y se cumple en R tenemos que existe u1∈R tal que u1[X]=t1[X]=s1[X], u1[Y]=t1[Y], u1[R*-X-Y]=s1[R*-X-Y], donde t1[S*]=t, s1[S*]=s, claramente estas eneadas siempre existen pues existen t y s en S y S=R[S*]. Por lo tanto si u1[S*]=u2 tenemos que u2[Y∩S*]=t[Y∩S*] y puesto que S*⊂R* ==> S*-X-(Y∩S*) ⊂ R*-X-Y ==>u2[S*-X-(Y∩S*)]=s[S*-X-(Y∩S*)] pero u2=u1[S*] ∈ S CONTRADICCION! pues habíamos supuesto que no existía tal u. Por lo tanto X->->Y∩S* se satisface en S.

4.- Ninguna otra dependencia en S* puede deducirse del hecho de que D se cumple en R*.

Sea D[S*] el conjunto de dependencias en S* obtenidas mediante los pasos 1,2 y 3 del teorema 12.

Demostraremos que D[S*]=D[S*]+.

La demostración es por inducción sobre el número de líneas en la prueba de que X->->Y a partir de dependencias en D[S*]. Claramente esta dependencia no es funcional ya que trivialmente por el paso 2 del teorema y la propiedad de descomposición para dependencias funcionales: X-->Y ∈ D+ <==> X-->Y∈ D[S*].

Cada "línea" es una dependencia que está en D[S*] o se sigue de los axiomas MD1-MD3 (descartamos los axiomas FD1-FD3 y MD4, MD5 pues estos producen dependencias funcionales y ya vimos que siempre estarán en D[S*]), la última línea será X->->Y.

Sea i=1; trivialmente X->->Y ∈ D[S*].

Hipótesis de Inducción: X->->Y∈ D[S*] siempre que X->->Y tenga una prueba de menos de k líneas.

Si la demostración de X->->Y tiene k líneas, entonces se obtuvo mediante alguno de los axiomas MD1-MD3.

Si se usó MD1 (complemento), entonces tenemos X->->W en la línea k-1 y X->->S*-W-X = Y, como la demostración de X->->W tiene k-1 líneas entonces X->->W ∈D[S*] ==> Existe X-->W'∈ D+ con W'∩S*=W ==> X->->R*-W'-X ∈ D+ ==> X->->(R*-X-W')∩S* ∈ D[S*].

Luego X->->S*-X-W' = S*-X-W = Y ∈ D[S*] pues W'∩S* = W.

Si usamos MD2. Entonces V->->W se demuestra en k-1 pasos, luego por nuestra hipótesis V->->W∈ D[S*] y VZ=X, WT=Y con T⊂Z.

V->->W ∈ D[S*] ==> V->->W' ∈ D+ con W'∩S* = W, como T, Z⊂R* ==> VZ=X->->W'T=Y' ∈ D+.

Luego X->->Y'∩S* = WT = Y∈ D[S*].

Si usamos MD3, entonces X->->Z, Z->->V tienen pruebas con menos de k pasos y por nuestra hipótesis ambas están en D[S*].

X->->Z, Z->->V ∈ D[S*] ==> X->->Z', Z'->->V' ∈ D+ con Z'∩S*=

Z, V'∩S*=V ==> X->->V'-Z' ∈ D+ ==> X->->(V'-Z')∩S* ∈ D[S*], pero (V'-Z')∩S*= V'∩S*-Z' = V-Z' = V-Z pues Z'∩S*=Z.

Por lo tanto X->->V-Z = Y ∈ D[S*], luego D[S*]=D[S*]+.

L.Q.Q.D.

Ahora demostraremos el inciso 4.

Si X->->Y ∉ D[S*] con XY=/S*, se cumple en S* por el hecho de que D se cumple en R*.

Es claro que X-->Y ∉ D+ pues trivialmente X->->Y ∈ D[S*] por el paso 3.

Usando el mismo procedimiento que el de la prueba de completez para los axiomas MD1-MD5 y de Amstrong (Teorema 6), construyamos una instancia R en la que especificamente no se cumpla X->->Y y sea S=R[S*]. Podemos suponer sin pérdida de generalidad que X+∩Y= ∅ ya que X-->Y∩X+ y trivialmente X->->Y∩X+ ∈ D[S*] por el paso 3 del teorema.

Sean Z(1),...,Z(k)_ las dependencias básicas de X que cubren R*-X+. Dado que X->->Y no se cumple en D+ ==> Existe una subcolección @ = [Z(i) / Z(i)∩Y=/∅, i ∈ 1-k] tal que la unión de los elementos de @ llamemosle Z contiene propiamente a Y.

Ahora bien, dado que S*∩Z =/ Y o de lo contrario X->->Y ∈ D[S*], contradiciendo nuestra suposición, tenemos :

S*∩Z = Y'==> Y⊂Y', X->->Y' ∈ D[S*] por el paso 3.

Sea W' = S*-X-Y' ==> X->->W' ∈ D[S*].

Como X->->Y se cumple en S* entonces X->->Y'-Y debe valerse en S* por la propiedad de descomposición, pero X->->Y'-Y ∉ D[S*] pues la misma propiedad de descomposición implicaría que X->->Y∈ D[S*] pues D[S*]=D[S*]+.

Sea @'= [Z(i) ∈ @/ Z(i)∩(Y'-Y) =/∅] claramente ningún Z(i)⊂ (Y'-Y) por la forma en que elegimos @.

Pero Z(i)∩S* = Z(i)∩(Y'-Y) por la construcción de Y'.

Luego aplicando la propiedad aditiva para los Z(i) ∈ @' tenemos que:

X->->Y'-Y ∈ D[S*] luego por descomposición X->->Y'-(Y'-Y)=Y∈ D[S*]. CONTRADICCION! a nuestra suposición inicial. L.Q.Q.D.

OBSERVACIONES : A este tipo de dependencias multivaluadas que no se cumplen en R* pero sí en un subesquema S* se les llama dependencias "inmersas" o "embebidas" en R*.

ALGORITMO V (Ullman [1]):Descomposición Sin Pérdida en 4FN.

ENTRADA: Un esquema R* y un conjunto de dependencias D.

SALIDA: @ descomposición sin pérdida de R* con esquemas en 4FN.

METODO: 1.- La primer descomposición @(0) será R*.

2.- Si existe un esquema S* ∈ @ que no esté en 4FN con respecto a D[S*], entonces debe existir en S* una dependencia X->->Y, donde X no incluye una llave para S*, Y=/∅, Y⊄X y XY=/S*. Podemos suponer que X∩Y= ∅ ya que si X->->Y entonces X->->Y-X se sigue de la regla de descomposición para DM. Después reemplácese S* por S1*=XY y S2*=S*-Y, los cuáles son esquemas relacionales con menos atributos que S* y por el teorema 8 sobre descomposición de esquemas relacional en dos esquemas tenemos que S1* ∩ S2*->->S1*-S2*. Por lo tanto la descomposición de S* en S1* y S2* es sin pérdida con respecto a DIS*1, continuando este proceso hasta que todos los esquemas estén en 4FN obtendremos la descomposición buscada.

3.4.2.1 Otras Formas Normales.

La intensa actividad de investigación sobre Bases de Datos y especialmente del modelo relacional ha producido algunas nuevas formas normales la mayoría de las cuales están en la etapa de "proposición" y aún no cuentan con la aceptación general.
A continuación definiremos brevemente las más importantes y enunciaremos algunas de sus mayores ventajas.

CUARTA FORMA NORMAL DEBIL (W4FN): Un esquema R* con atributos U está en W4FN si está en 3FN y siempre que X->->Y se tiene que U=XY.

OBSERVACIONES : Es fácil demostrar que 4FN ==> W4FN ==> 3FN.
Ya que W4FN ==> 3FN trivialmente de la definición de W4FN y 4FN ==> W4FN pues si un esquema relacional está en 4FN, entonces las únicas dependencias multivaluadas propiamente dichas serán aquéllas que permite la W4FN.

VENTAJAS DE LA W4FN : La mayor ventaja es que permite obtener descomposiciones "sin pérdida" que preservan dependencias y que sus esquemas estarán en 4FN, casi siempre [4].

Hasta aquí llegan las formas normales que usan dependencias multivaluadas, para continuar necesitamos definir un nuevo tipo más general de dependencia que llamaremos "Dependencia en Juntura" (JD) [7].

DEPENDENCIA EN JUNTURA (JD): "Sean X1,...,Xr subconjuntos (no necesariamente ajenos) de los atributos en el esquema R*, donde cada atributo de R* está contenido en al menos un Xi, i en {1,...,r}. Decimos que R* obedece la "Dependencia en Juntura": *[X1,...,Xr], si para cualquier instancia R de R* la juntura de sus proyecciones R[X1],..., R[Xr] nos da nuevamente R."

OBSERVACIONES: El hecho de que DM ==> JD, se sigue de la definición de DM dada por Fagin, pues podemos decir que la DM X->->Y se cumple en R* si y sólo si la JD *[XY, R*-Y] se cumple en R.

QUINTA FORMA NORMAL (5FN) : Un esquema relacional R* está en 5FN si toda JD : *[X1,...,Xr] de R* es "implicada lógicamente" por el conjunto de llaves de R*.

OBSERVACIONES : 1.- Fagin en [8] da un algoritmo mediante el cuál es posible decidir si una JD dada es implicada por un conjunto de llaves.

2.- Fagin en [8] demuestra que toda DM implicada lógicamente por una llave de una relación R debe ser también una DF con lado izquierdo igual a dicha llave. Por lo tanto se sigue que 5FN ==> 4FN.

3.- A la 5FN también se le llama "Forma Normal Proyección-Juntura" (PJ/FN).

VENTAJAS DE LA 5FN : Elimina las dependencias más generales posibles dentro de los límites de las operaciones de proyección para descomposición y juntura para reconstitución de relaciones (de ahí el nombre de PJ/FN).

En [7] se presenta un ejemplo (un poco rebuscado) de un esquema en 4FN pero no en 5FN, que presenta una "anomalía de inserción" con respecto a una restricción de JD.

ANOMALIAS DE INSERCION Y DE BORRADO : Se presentan cuando la inserción ó borrado de una eneada cuyos valores de los distintos atributos que la componen han sido obtenidos de los correspondientes dominios, provoca la violación de una restricción de el esquema relacional al que pertenecen.

La posibilidad de descomponer una relación separándola "horizontalmente" (en contraste a la descomposición obtenida mediante proyección que es "vertical"), de tal forma que la relación original pueda ser recuperada vía la operación de unión, condujo a Smith a lo que el llama (3,3)FN.

(3,3)FN: Un esquema R* en 1FN está en (3,3)FN si las DF embebidas en cualquier subrelación S obtenida de una instancia también arbitraria R de R* mediante la selección de todas aquellas eneadas t que satisfagan alguna condición p(t) son consecuencia lógica de los dominios y llaves válidas para R* (ver [7]).

OBSERVACIONES: (3,3)FN implica BCFN, pero no tiene relación alguna con la 4FN ni la 5FN por lo que podemos verla como un desarrollo "ortogonal" a dichas formas normales (Date [2]). Es fácil de la definición dada arriba "intuir" la verdad de estas afirmaciones, ya que si toda DF es consecuencia lógica de las restricciones de dominio y de llaves válidas en el esquema de que se trate (definición de (3,3)FN), entonces en toda dependencia X-->A válida en ese esquema, X debe contener una llave y esta es en escencia la definición de BCFN. Por parte de la falta de relación con las Cuarta y Quintas formas normales, podemos atribuirla a que en la definición para (3,3)FN, no se hace referencia alguna a las DM (a menos que también sean DF).

VENTAJAS DE LA (3,3)FN: En [7] se proporciona un ejemplo en el cual un esquema en 5FN pero no en (3,3)FN presenta una anomalía de inserción.

Otra importante forma normal es la llamada Forma Normal/Dominio-LLave que abreviamos DK/FN siguiendo la literatura al respecto, definida por Fagin en [7] como sigue:

FORMA NORMAL/DOMINIO-LLAVE (DK/FN): Sea R* un esquema relacional en 1FN, y sea "T" el conjunto de "dependencias de dominio" y de "dependencias de llave" del esquema R*. R* está en Forma Normal/Dominio-LLave (DK/FN) si T implica lógicamente "@" para cualquier restricción "@" de R*.

DEPENDENCIAS DE DOMINIO (DD) : Es la declaración de que un atributo "A" en un esquema R*, toma valores en un conjunto "S" y se denota "IN(A,S)".

DEPENDENCIAS DE LLAVE (KD): Es la declaración de que un conjunto "X" de atributos en un esquema R*, es una llave para dicho esquema y se denota "KEY(X)"

RESTRICCION: Es cualquier condición que todas las instancias de un esquema relacional deben satisfacer.
Son ejemplos de restricciones las dependencias de dominio, las dependencias de llave, las DF, las DM, las JD, y cualquier otra que pueda establecerse sobre las instancias de un esquema relacional.

Intuitivamente un esquema relacional está en DK/FN si toda restricción puede ser conocida por el simple hecho de saber los nombres de los atributos en dicho esquema con sus correspondientes dominios y cuáles son sus llaves.

VENTAJAS DE LA DK/FN : Fagin [7] demuestra que existen anomalías de inserción y de borrado en esquemas relacionales aún en 5FN. Otra ventaja es que se basa en los conceptos mas "naturales" de dominios, llaves y restricciones, a diferencia de las formas normales 4ta. y 5ta. que se basan en conceptos no muy obvios.

4 ESQUEMA CONCEPTUAL PROPUESTO PARA EL BANCO DE DATOS CIAD.

4.1 Consideraciones

Es necesario aclarar que el BD-CIAD se pretende sirva como fuente de información sobre aspectos económicos y sociales de las distintas regiones del Estado, ya sea para decidir la justificación de un proyecto de investigación, ó bien como información para definir el marco general en que se desarrollará un proyecto.

En vista de lo anterior debemos hacer algunas consideraciones sobre la estructura del BD-CIAD y sobre la información disponible.

Sobre la información disponible podemos decir:

a) En su mayor parte es de origen censal. El resto se obtiene de las dependencias de gobierno encargadas de Pesca, Agricultura, Ganaderla y Minerla.

b) La información agrlcola viene por distritos de riego y es prácticamente imposible reordenarla por municipios.

c) La información sobre Industria, Comercio y Servicios es virtualmente inexistente al nivel de municipios.

d) Los datos no se manejan en forma consistente a través de todos los censos.

Por Ejemplo:

El censo del 40 maneja Población por Edad y Sexo, para cada edad entre 0 y 85 ó más años , pero el del 50 sólo maneja rangos de edad.

También es común encontrar diferencias en los criterios de PEA (Población Económicamente Activa) y PEA ocupada, entre otros.

e) La información censal se presenta en forma de tablas las cuáles son bastante independientes entre sl, esto es, casi no presentan información que permita relacionar de alguna manera una tabla con otra.

Sobre la estructura del BD-CIAD, podemos decir:

a) A fin de presentar a cada usuario en forma sencilla la totalidad de la información disponible, no es necesaria la existencia de subesquemas parciales, sino que sólo se necesita un subesquema que abarque toda la información disponible.

b) Con el objeto de que el Subesquema presente a los usuarios no programadores la misma visión de los datos censales que el más detallado censo disponible (el del 70), es necesario que dicho subesquema contenga relaciones que violan la 3FN, por ejemplo las columnas para totales. Sin embargo esto no resulta importante, ya que en el caso del BD-CIAD, los usuarios finales no agregarán ni borrarán información, con lo cuál se eliminan las anomallas de inserción y borrado, al menos a nivel de subesquemas, y como las relaciones en los subesquemas no están flsicamente guardadas, en realidad no existe desperdicio de espacio.

4.2 Subesquema del Banco de Datos CIAD.

El subesquema del BD-CIAD estará constituido por los siguientes esquemas relacionales correspondientes casi directamente a las tablas del censo de 1970.

1.- Población Total por sexo y Densidad de Población.
ATRIBUTOS: .Año, Municipio, Pob. Tot. Hombres y Mujeres, Tot. Hombres, Tot. Mujeres, Sup. Km2, Hab./Km2, % de la Pob.Tot. del Estado, % de la Sup. Total.

2.- Número de Localidades por Grupo de Tamaño.
ATRIBUTOS: Año, Municipio, Gpo. de Tamaño, Núm. de Loc., Pob. Total, Tot.Hombres, Tot.Mujeres.

3.- Familias según Número de Miembros.
ATRIBUTOS: Año, Municipio, Pob. Fam. Total, Total Fams., Fams. de 2,...,Fams. de 9 ó + Miembros.

4.- Estado Civil Población Mayor de 12 Años.
ATRIBUTOS: Año, Municipio, Sexo, Pob. >12 Años, Solteros, Casados, Sólo Civil, Sólo Religioso, Civil y Relig., U. Libre, Viudos, Divorciados, Separados.

5.- Población por Lugar de Nacimiento y Sexo.
ATRIBUTOS: Año, Municipio, Lugar de Nac., Hombres, Mujeres, Total Hombres y Mujeres.

6.- Población de 10 ó más Años por Alfabetismo y Sexo.
ATRIBUTOS: Año, Municipio, Grupo Edad A, Pob. Total, Hombres, Mujeres, Homb. Alfab., Muj. Alfab., Homb. Analf., Muj. Analf.

NOTA: Usaremos una letra mayúscula de la A a la E, para distinguir los distintos agrupamientos de edades.

7.- Grados de Instrucción Aprobados por Edades.
ATRIBUTOS: Año, Municipio, Gpo. Edad B, Pob. >6 Años, Sin Instr., Algún Grado Aprovado, 1er Año,..., 6to Año, Instr. Postprimaria, Instr. Insuficientemente Especificada.

8.- Población de 6 a 15 Años que Asiste a Escuelas Primarias.
ATRIBUTOS: Año, Municipio, Grado, Total Asistentes, de 6 Años,...,de 15 Años.

9.-Población Mayor ó igual a 11 Años que Asiste a Escuelas Postprimarias por Grupo de Edad C
ATRIBUTOS: Año, Municipio, Gpo. Edad C, Capacitación, Secundaria, Prep. ó Voc., Prof. Medio con Sec.,Prof. Medio con Prepa. ó Voc., Prof. Sup., Post Grado.

10.-Mujeres Mayor ó igual a 12 Años por Número de Hijos por Grupo de Edades D.
ATRIBUTOS: Año, Municipio, Núm. Hijos Nac. Vivos, Tot. Mujeres >=12 Años, 12-14 Años,..., 50 ó más Años.

11.-Población Mayor ó igual a 12 Años que Trabajó la Semana Anterior al Censo (PEA1).

ATRIBUTOS: Año, Municipio, Sexo, Gpo. Edad E, Pob. >=12 Años, PEA 1 Total, PEA1 Ocupada, PEA1 Desocup., PEI Total, PEI Hogar, PEI Est., PEI Otros.

12.-Población Mayor ó igual a 12 Años que Trabajó el Año Anterior al Censo (PEA2).

ATRIBUTOS: Año, Municipio, Gpo. Edad D, Sexo, PEA2 Total, PEA2 Ocupada, % de la PEA2 Total.

13.-PEA2 por Ramas de Actividad y Grupo de Edad D.

ATRIBUTOS: Año, Municipio, Sexo, Gpo. Edad D, PEA2 Total, Agricultura Ganaderia Silvicultura Caza y Pesca, Ind. del Petróleo, Ind. Extractiva, Ind. de Transf., Construcción, E. Eléctrica, Comercio, Transporte, Servicios, Gobierno, Insuf. Especificada.

14.-Población Total por Edad y Sexo.

ATRIBUTOS: Año, Municipio, Edad, Total H. y M., Hombres, Mujeres.

15.-PEA2 por Ocupación Principal y Rama de Actividad.

ATRIBUTOS: Año, Municipio, Ocup. Ppal., Total por Ocup. Ppal., Agricultura Ganadería Silvicultura Caza y Pesca, Ind. del Petr., Ind. Extract., Ind. de Transf., Construcción, E. Eléctrica, Comercio, Transporte, Servicios, Gobierno, Insuf. Especifidada.

16.-PEA2 por Posición en el Trabajo por Rama de Actividad y Sexo.

ATRIBUTOS: Año, Municipio, Puesto, Total por Puesto, Agricultura Ganadería Silvicultura Caza y Pesca, Ind. del Petr.,Ind. Extract., Ind. de Transf., Construcción, E. Eléctrica, Comercio, Transporte, Servicios, Gobierno, Insuf. Especificada.

17.-PEA2 por Rama de Actividad por Grupo de Meses Trabajados durante el año anterior al Censo.

ATRIBUTOS: Año, Municipio, Gpo. de Meses, Tot. por Gpo. de Meses, Agricultura Ganadería Silvicultura Caza y Pesca, Ind. del Petr., Ind. Extract., Ind. de Transf., Construcción, E. Eléctrica, Comercio, Transporte, Servicios, Gobierno, Insuf. Especificada.

18.-PEA2 por Posición en el Trabajo por Ocupación Principal por Sexo.

ATRIBUTOS: Año, Municipio, Puesto, Sexo, Tot. por Puesto, Profesionistas ó Técnicos, Funcs. Sup. y Personal Directivo Privado, Personal Administrativo, Comerciantes Vendedores y Similares, Conductores de Vehículos, Trab. en Labores Agropec., Obreros No Agropec., Insuf. Especificada.

19.-PEA2 por Grupo de Ingresos y por Ocupación Principal
ATRIBUTOS: Año, Municipio, Ocupación, Tot. por Ocup., Tot. que Declaró Ingresos, <= $199, $200-499, $500-999, $1000-1499, $1500-2499, $2500-4999, $5000-9999, $10000 ó más.

20.-PEA2 por Grupo de Ingresos y por Rama de Actividad.
ATRIBUTOS: Año, Municipio, Ocupación, Tot. por Ocup., Agricultura Ganadería Silvicultura Caza y Pesca, Ind. del Petr., Ind. Extractiva, Ind. de Transf., Construcción, Transportes, Servicios, Gobierno, Insuf. Especificada.

21.-Población Mayor ó igual a 12 Años que busca Trabajo por Sexo y Grupo Edad E.
ATRIBUTOS: Año, Municipio, Gpo. Edad E.,Sexo, Total Gpo. Edad, Tot. Ocupados, Tot. que han Trab. Antes, Tot. que no han Trab. Antes.

22.-Número de Viviendas y Ocupantes por Tipo por Tenencia.
ATRIBUTOS: Año, Municipio, Tipo Viv.,Tipo Tenencia, Tot. Viviendas, Tot. Ocupantes.

23.-Viviendas por Número de Cuartos por Número de Ocupantes.
ATRIBUTOS: Año, Municipio, Clase Viv. por Núm. de Cuartos, Tot. Viviendas, Tot. Ocupantes.

24.-Viviendas y Ocupantes por Drenaje y Agua Entubada.
ATRIBUTOS: Año, Municipio, Disp. de Agua, Tot. Viv., Tot. Ocup., Tot. Viv. con Drenaje, Tot. Ocup. Viv. con Drenaje, Tot. Viv. sin Drenaje, Tot. Ocup. Viv. sin Drenaje.

25.-Número de Viviendas y Ocupantes por Días que Consumieron diversos Alimentos la semana anterior al Censo.
ATRIBUTOS: Año, Municipio, Alimento, Tot. Viv., Tot. Ocup., Viv. 1 Día, Ocup. Viv. 1 Día, ... , Viv. 7 Días, Ocup. Viv. 7 Días.

Hasta aquí las relaciones correspondientes a datos censales. Continúan las relaciones de producción.

26.-Producción No Agrícola por Municipio.
ATRIBUTOS: Año, Municipio, Producto, Valor Producción, Cantidad Producida, Unidades.

27.-Inventario de Aves por Municipio por Tipo de Ave.
ATRIBUTOS: Año, Municipio, Tipo de Ave, Núm. de Animales.

28.-Producción de Carne de Ave por Municipio por Tipo de Ave.
ATRIBUTOS: Año, Municipio, Tipo de Ave, Producción, Unidades, Valor de la Producción.

29.-Inventario de Bobinos por Municipio por Raza.
ATRIBUTOS: Año, Municipio, Raza Bovina, Núm. de Cabezas.

30.-Producción de Carne por Tipo por Municipio.
ATRIBUTOS: Año, Municipio, Tipo de Carne, Cant. Producida, Unidades, Valor Producción.

31.-Producción de Mariscos por Especie por Municipio.
ATRIBUTOS: Año, Municipio, Especie, Tons. Producidas, Valor de la Producción.

32.-Distritos de Riego por Municipio.
ATRIBUTOS: Año, Distrito de Riego, Municipio, Superficie.

33.-Producción Agrícola por Distrito de Riego.
ATRIBUTOS: Año, Distrito de Riego, Cultivo, Sup. Sembrada, Tons. Producidas, Valor Producción.

4.3 Obtención del Esquema Conceptual Ideal para el Banco de Datos CIAD.

Antes de dar el Esquema Conceptual Ideal presentamos algunas definiciones y reglas de integridad para Esquemas Conceptuales del modelo relacional.

LLAVE PRIMARIA: La LLave Primaria de un esquema relacional es aquella que tiene el menor número de atributos, si existen dos ó más llaves con el mismo número de atributos entonces se escoge alguna como llave primaria.

LLAVE ALTERNATIVA: Es cualquier llave distinta de la primaria.

DOMINIO PRIMARIO: [D] "Un dominio dado puede opcionalmente ser designado como "primario" si y sólo si existe algún atributo definido en él y que por si solo sea llave primaria de alguna relación"

A continuación damos las reglas de integridad para Esquemas Conceptuales en bases de datos relacionales.

REGLA DE INTEGRIDAD 1 (INTEGRIDAD DE ENTIDAD):
Ningún componente de una llave primaria puede ser nulo.

OBSERVACIONES : Es fácil comprender la finalidad de la regla anterior y es la de asegurar la unicidad en la identificación de las eneadas en toda relación.

REGLA DE INTEGRIDAD 2 (INTEGRIDAD REFERENCIAL):
Sea D un dominio primario, y sea R1 una relación con un atributo A que está definido en D. Entonces, en cualquier tiempo dado, cada valor de A en R1 debe ser o nulo o igual a V, donde V es el valor de la llave primaria en alguna relación R2 (R2 no necesariamente distinta de R1), con dominio primario definido en D.

OBSERVACIONES : El objeto de esta regla es impedir que pueda hacerse referencia a una entidad en una relación, sin que dicha entidad esté en la BD.

4.3.1 Esquema Conceptual Ideal para el BD-CIAD.

Las entidades "censales" o que generan la información censal serán:

1.-Censos.
ATRIBUTOS: Año

2.-Municipios.
ATRIBUTOS: Municipio

Las entidades de producción ó que generan la información de producción serán:

3.-Años con Datos de Producción.
ATRIBUTOS: Año.

4.-Productos No Agrícolas.
ATRIBUTOS: Producto

5.-Clasificación de Aves.
ATRIBUTOS: Tipo de Ave

6.-Razas Bovinas Consideradas.
ATRIBUTOS: Raza Bovina

7.-Clasificación de Carne.
ATRIBUTOS: Clase de Carne

8.-Especies Marinas Consideradas.
ATRIBUTOS: Especie Marina

9.-Productos Agrícolas.
ATRIBUTOS: Cultivo

10.-Distrito de Riego.
ATRIBUTOS: Núm. del Distr.; Lugar; Superficie

A continuación presentamos las asociaciones entre dos ó más tipos de entidades.

Asociaciones entre entidades Censales:

11.- Población Total por Sexo y por Densidad de Población
ATRIBUTOS: Año; Municipio; Tot. Hombres; Tot. Mujeres; Sup.

12.- Número de Localidades por Grupo de Tamaño por Municipio
ATRIBUTOS: Año; Municipio; Gpo. de Tamaño; Núm. de Locs.; Tot. Hombres; Tot. Mujeres.

13.- Familias según Número de Miembros.
ATRIBUTOS: Año; Municipio; Núm. de Miembros; Tot. Personas

14.- Estado Civil Población Mayor de 12 Años.
ATRIBUTOS: Año; Municipio; Sexo; Tot. Solteros; Tot. sólo Civil; Tot. sólo Relig.; Tot. Civil y Relig.; Tot. Unión Libre; Tot. Viudos; Tot. Divorciados; Tot. Separados

15.- Población por Lugar de Nacimiento y Sexo.
ATRIBUTOS: Año; Municipio; Lugar de Nac.; Hombres; Mujeres

16.- Población de 10 o más Años por Alfabetismo y Sexo.
ATRIBUTOS: Año; Municipio; Gpo. Edad A; Alfab. Homb.; Alfab. Muj.; Analf. Homb.; Analf. Muj.;

17.- Grados de Instrucción Aprobados por Edades.
ATRIBUTOS: Año; Municipio; Gpo. Edad B; Sin Instr.; 1er. Año; 2o. Año; ... ; 6to. Año; Instr. Postprim.; Instr. Insuf. Especif.

18.- Población de 6 a 15 Años que Asiste a Escuelas Primarias.
ATRIBUTOS: Año; Municipio; Grado; Tot. de 6 Años; ... ; Tot. de 15 Años

19.- Población Mayor ó igual a 11 Años que Asiste a Escuelas Postprimarias por Grupo Edad C.
ATRIBUTOS: Año; Municipio; Gpo. Edad C; Capacitación; Secundaria; Prepa. ó Voc.; Prof. Medio con Sec.; Prof. Medio con Prepa. ó Voc.; Prof. Sup.; Postgrado.

20.- Mujeres de 12 Años o más por de Hijos y Grupo Edad D.
ATRIBUTOS: Año; Municipio; Núm. de Hijos; Muj. >=12 Años; 12-14 Años; ... ; 50 ó más Años.

21.- Población Mayor ó igual a 12 Años que Trabajó la Semana Anterior al Censo (PEA1).
ATRIBUTOS: Año; Municipio; Sexo; Gpo. Edad E; PEA1 Ocup.; PEA1 Desocup.; PEI Hogar; PEI Est.; PEI Otros.

22.- Población Mayor ó igual a 12 Años que Trabajó el Año Anterior al Censo (PEA2).
ATRIBUTOS: Año; Municipio; Gpo. Edad E; PEA2 Homb.; PEA2 Muj.; PEA2 Homb. Ocup.; PEA2 Muj. Ocup.

23.- PEA2 por Ramas de Actividad y Grupo de Edad E.
ATRIBUTOS: Año; Municipio; Sexo; Gpo. Edad E; Agri., Gan., Silvic., Caza y Pesca; Ind. del Petróleo; Ind. Extractiva; Ind. de Transf.; Construcción; E. Eléctrica; Comercio; Transporte; Servicios; Gobierno; Insuf. Especif.

24.- Población Total por Edad y Sexo
ATRIBUTOS: Año; Municipio; Edad; Hombres; Mujeres.

25.- PEA2 por Ocupación Principal y Rama de Actividad.
ATRIBUTOS: Año; Municipio; Ocup. Ppal.; Agri., Gan., Silvic., Caza y Pesca; Ind. del Petróleo; Ind. Extractiva; Ind. de Transf.; Construcción; E. Eléctrica; Comercio; Transporte; Servicios; Gobierno; Insuf. Especif.

26.- PEA2 por Posición en el Trabajo por Rama de Actividad y Sexo.
ATRIBUTOS: Año; Municipio; Puesto; Agri., Gan., Silvic., Caza y Pesca; Ind. del Petróleo; Ind. Extractiva; Ind. de Transf.; Construcción; E. Eléctrica; Comercio; Transporte; Servicios; Gobierno; Insuf. Especif.

27.- PEA2 por Rama de Actividad por Grupo de Meses Trabajados en el año anterior al Censo.
ATRIBUTOS: Año; Municipio; Gpo. de Meses; Agri., Gan., Silvic., Caza y Pesca; Ind. del Petróleo; Ind. Extractiva; Ind. de Transf.; Construcción; E. Eléctrica; Comercio; Transporte; Servicios; Gobierno; Insuf. Especif.

28.- PEA2 por Posición en el Trabajo por Ocupación y Sexo.
ATRIBUTOS: Año; Municipio; Puesto; Sexo; Prof. ó Técnicos; Funcionarios Sups. y Pers. Direct. Privado; Pers. Admvo.; Comerciantes Vendedores y Simlres.; Cond. de Vehículos; Trab. en Labores Agropec.; Obreros No Agropec.; Insuf. Especif.

29.- PEA2 por Grupo de Ingresos y por Ocupación Principal.
ATRIBUTOS: Año; Municipio; Ocupación; Tot. por Ocup.; Ing.<=$199; Ing. $200-499; ... ; Ing. $10000 ó Más.

30.- PEA2 por Grupo de Ingresos y por Rama de Actividad.
ATRIBUTOS: Año; Municipio; Gpo. Ingresos; Agri., Gan., Silvic., Caza y Pesca; Ind. del Petróleo; Ind. Extractiva; Ind. de Transf.; Construcción; E. Eléctrica; Comercio; Transporte; Servicios; Gobierno; Insuf. Especif.

31.- Población Mayor ó igual a 12 Años que Busca Trabajo por Sexo y Grupo Edad E.
ATRIBUTOS: Año; Municipio; Gpo. Edad E; Sexo; Tot. Ocup.; Tot. que Trab. Ant.; Tot. que no han Trab. Ant.

32.- Número de Viviendas y Ocupantes por Tipo. por Tenencia.
ATRIBUTOS: Año; Municipio; Tipo Viv.; Tipo Tenencia; Tot. Viviendas; Tot. Ocupantes.

33.- Viviendas por Número de Cuartos por Número de Ocupantes.
ATRIBUTOS: Año; Municipio; Núm. de Ctos. Viv.; Tot. Viv.; Tot. Ocupantes.

34.- Viviendas y Ocupantes por Drenaje y Agua Entubada.
ATRIBUTOS: Año; Municipio; Tot. Viv. con Agua; Tot. Ocup.

Viv. con Agua; Tot. Viv. sin Agua; Tot. Ocup. Viv. sin Agua; Tot. Viv. con Drenaje; Tot. Ocup. Viv. con Drenaje; Tot. Viv. sin Drenaje; Tot. Ocup. Viv. sin Drenaje

35.- Número de Viviendas y Ocupantes por Días de Consumo diversos Alimentos la semana anterior al censo.
ATRIBUTOS: Año; Municipio; Alimento; Tot. Viv. 1 Día; Ocup. Viv. 1 Día; ... ; Tot. Viv. 7 Días; Ocup. Viv. 7 Días.

Hasta aquí las relaciones correspondietes a datos censales, continuan las de datos de producción:

36.- Producción No Agrícola por Municipio
ATRIBUTOS: Año; Municipio; Producto; Valor Producción; Cantidad Producida.

37.- Inventario de Aves por Municipio por Tipo de Ave.
ATRIBUTOS: Año; Municipio; Tipo de Ave; Núm. de Animales

38.- Producción de Carne de Ave por Municipio por Tipo de Ave.
ATRIBUTOS: Año; Municipio; Tipo de Ave; Producción; Valor Producción.

39.- Inventario de Bobinos por Municipio por Raza.
ATRIBUTOS: Año; Municipio; Raza ; Núm. de Cabezas.

40.- Producción de Carne por Tipo por Municipio.
ATRIBUTOS: Año; Municipio; Tipo de Carne; Cant. Producida; Valor Producción.

41.- Producción de Mariscos por Especie por Municipio.
ATRIBUTOS: Año; Municipio; Especie; Tons. Producidas; Valor Producción.

42.- Distritos de Riego por Municipio.
ATRIBUTOS: Año; Distrito de Riego; Municipio; Sup. por Mpio.

43.- Producción Agrícola por Distrito de Riego.
ATRIBUTOS: Año; Distrito de Riego; Cultivo; Sup. Sembrada; Tons. Producidas; Valor Producción.

OBSERVACIONES: En realidad este esquema conceptual ideal, no es más que un conjunto de relaciones en 5FN, que contienen una relación para cada relación en el subesquema más el conjunto de relaciones que representan a las entidades que se juzgaron necesarias.

De lo anterior podemos concluir que el proceso de normalización presentado alcanza su máxima utilidad en el diseño conceptual de bases de datos que permitan modificación e inserción de información a los usuarios externos, lo cuál no es el caso del BD-CIAD, pues como ya hemos dicho es exclusivamente de consulta, al menos para los usuarios externos.

5. OBSERVACIONES Y CONCLUSIONES.

A continuación presentamos un resumen de las principales características del trabajo anteriormente expuesto.

Ambito de Aplicación:

Si bien los dos primeros capítulos presentan conceptos más bien aplicables a las grandes computadoras, al menos hasta ahora, esto no quiere decir que la teoría expuesta en el capítulo 3 no sea útil para cualquier tamaño de Banco de Datos, solamente debemos tener en cuenta que no existe un verdadero Esquema Conceptual en los DBMS para micros, lo cuál nos obliga a establecer un compromiso entre el esquema conceptual ideal y los subesquemas deseados, al momento de diseñar los subesquemas o visiones de usuario respectivos.

Un hecho que se debe tomar en cuenta, sobre todo al momento de decidir cuáles serán los subesquemas o visiones que presente el Banco de Datos, es el ámbito que va a tener dicho Banco que puede ser:

1o. Consulta y Actualización (Inserción, Modificación y Borrado).

2o. Consulta Exclusivamente.

Como hemos visto en el capítulo 3 sección 3.4.1, las principales ventajas de las sucesivas formas normales y por consiguiente del proceso de normalización se manifiestan en un Banco de Datos que funcione en un ámbito de consulta y actualización. En el caso de que se trate de un ambiente de consulta exclusivamente, es probable que lo mejor sea llevar el proceso de normalización sólo hasta donde represente un ahorro de espacio y en el caso particular de las micros, dicho ahorro debe justificar el esfuerzo adicional de programación que supone el manejar más de un archivo físico, al momento de responder determinadas preguntas.

Posibles Ramificaciones y Profundizaciones del Trabajo.

Como fácilmente se puede apreciar por la bibliografía, este es un campo de aplicación e investigación muy reciente, posee por tanto un enorme campo totalmente nuevo para futuras investigaciones tanto en alguna de sus ramificaciones como en profundizaciones sobre la teoría existente.

Entre las más importantes ramificaciones podemos citar:

1o. Lenguajes de Definición de Datos.

2o. Lenguajes de Manipulación de Datos.

3o. Algoritmos de Optimización en el manejo de las preguntas.

4o. Búsqueda de nuevos Modelos de Datos.

5o. Diseño y Evaluación de Esquemas Conceptuales Reales, conocidos el Esquema Conceptual Ideal, el ámbito y el DBMS a usar.

Por lo que se refiere a las profundizaciones podemos citar entre las ya existentes las siguientes:

1o. Estudio de la Forma Normal Dominio-LLave (DK/FN), pues generaliza y engloba todas las anteriores.

2o. Lo relativo a tiempos esperados de ejecución de los diversos Algoritmos.
3o. Sobre las dependencias Básicas y sus propiedades.

Y por lo que respecta a profundizaciones aún de investigación podemos mencionar:
1o. Estudio de nuevas y más útiles dependencias.
2o. Técnicas para una más uniforme selección de las entidades en un Esquema Conceptual Ideal, a partir de los datos que se desean guardar y su ámbito.
3o. Desarrollo de algortimos más eficientes para la obtención de las descomposiciones en las diversas formas normales, así como para la rápida comprobación de la existencia de dependencias, etc.
4o. Desarrollar una forma eficiente de representar jerarquías en el esquema conceptual usando relaciones.

APENDICE.

APLICACION DE LA TEORIA EXPUESTA A UNA RELACION PARTICULAR.

Consideremos la siguente relación de una Universidad:

DISTRIBUCION DE GRUPOS.

M	G	T	P	H	A	E	S
100-01	01	78-101	JUAN PEREZ	14-15	K-101	78532-1	R
						78296-1	I
						78438-1	R
100-01	01	81-141	HECTOR LOPEZ	7-8	K-101	77083-2	R
						76942-2	R

CLAVES DE LOS ATRIBUTOS:

M : Clave de la Materia
G : No. de Grupo
T : Clave del Maestro
P : Nombre del Maestro
H : Horario
A : Aula
E : Clave del Estudiante
S : Situación del Estudiante.

Fig. A.1

Hipótesis Simplificadoras:

1.- Las Clases son Diarias, siempre a la misma hora y en la misma aula.
2.- La situación del estudiante solo puede ser Regular (R) o Irregular (I).
3.- El No. de Grupo (G), no es en realidad un identificador del grupo, sino un número progresivo del 01 en adelante que va creciendo según el número de grupos que llevan la misma materia.

Procedamos ahora a normalizar la relación, esto es llevarla a la primera forma normal (1FN), la cuál se muestra a continuación:

DISTRIBUCION DE GRUPOS.

M	G	T	P	H	A	E	S
100-01	01	78-101	JUAN PEREZ	14-15	K-101	78532-1	R
100-01	01	78-101	JUAN PEREZ	14-15	K-101	78296-1	I
100-01	01	78-101	JUAN PEREZ	14-15	K-101	78438-1	R
100-01	02	81-141	HECTOR LOPEZ	7-8	K-101	77083-2	R
100-01	02	81-141	HECTOR LOPEZ	7-8	K-101	76942-2	R

Fig. A.2

Describamos el esquema relacional de esta relación:

Esquema Relacional : DISTRIBUCION DE GRUPOS*

ATRIBUTOS:	DOMINIOS:
Clave de la Materia (M)	Cve. de la Materia (Alfanum. 6b)
No. de Grupo (G)	No. de Grupo (Num. 2b)
Clave del Maestro (T)	Cve. del Maestro (Alfanum. 6b)
Nombre del Maestro (P)	Nombre del Maestro (Alfanum. 30b)

ATRIBUTOS:	DOMINIOS:	
Horario (H)	Hora	(Alfanum. 5b)
Aula (A)	Aula	(Alfanum. 5b)
Clave del Estudiante (E)	Cve. del Estudiante	(Alfanum. 7b)
Situación (S)	Situación	(Alfanum. 1b)

Los Dominios los hemos especificado solo en su naturaleza más general diciendo si son numéricos (Num.) o alfanuméricos (Alfanum.) y su longitud en bytes (b).

Las Dependencias que posee este esquema son aparte de las triviales las siguientes:

LLaves: ME-->GTPHAS; HE-->GTPHAS.

Dependencias Parciales: E-->S.

Dependencias Transitivas: T-->P
HA-->MGTP
TH-->MGPA
MG-->TPHA
MT-->GPHA.

Como vemos la existencia de una dependencia parcial en este esquema nos dice que este no esta en 2FN; sin embargo como el algoritmo IV para descomposición en 3FN de esquemas relacionales que preservan dependencias y el Teorema V para descomposición sin pérdida en 3FN que preserve dependencias, no especifican que para aplicarse el esquema deba estar en una forma normal superior a la primera, no eliminaremos esta dependencia parcial sino hasta que la aplicación de estos resultados así lo obliguen.

Para aplicar el algoritmo IV necesitamos una cubierta minimal para las dependencias en el esquema. Como una cubierta minimal no es única y dada la importancia que tiene esta para los esquemas resultantes en la descomposición obtenida mediante el algoritmo IV, debemos "sembrar" ciertas dependencias especialmente útiles según la aplicación que se vaya a dar a la información contenida, por ellos hemos "sembrado" la dependencia ME-->G, en el proceso que sigue.

OBTENCION DE LA CUBIERTA MINIMAL.

Como no existe un algoritmo para la obtención de una cubierta minimal, veremos que nuestra cubierta satisfaga las condiciones de la definición de cubierta minimal.

1.- Primero observemos las dependencias dadas en la definición del esquema relacional, veremos que no existen atributos redundantes en el lado derecho de las dependencias.

2.- ME-->G ==> ME-->MG-->TPHA, E-->S ==>ME es una llave.
MG-->T , T-->P ==> MG-->P
MG-->H , MG-->T ==> MG-->TH, TH-->MGPA ==> MG-->TPHA.
HA-->M , HA-->G ==> HA-->MG, MG-->TPHA ==> HA-->MGTP.
TH-->M , TH-->G ==> TH-->MG, MG-->TPHA ==> TH-->MGPA.

MT-->G ==> MT-->MG, MG-->TPHA ==> MT-->GPHA.

HE-->T ==> HE-->TH, TH-->MG, MG-->TPHA, E-->S ==>HE es una llave.

Como vemos las dependencias: ME--G, E-->S, MG-->T, T-->P, MG-->H, MG--T, HA-->M, HA-->G, TH-->M, TH-->G, MT-->G, HE-->T.

Constituyen una cubierta minimal de las dependencias en el esquema ya que implican lògicamente cualquier otra dependencia vålida en el.

DESCOMPOSICION SIN PERDIDA EN 3FN QUE PRESERVA DEPENDENCIAS.

Como podemos aplicar el algoritmo IV y el teorema V al mismo tiempo tenemos como resultado los siguientes esquemas relacionales:
MEG, ES,(MGT, MGH, MGA) ==> MGTHA, TP,(HAM, HAG) ==> HAMG, (THM, THG) ==>THMG, MTG, HET, ME.

Eliminando los esquemas redundantes, que siempre serân aquêllos cuyos atributos estên contenidos en otro mayor, ya que no podemos eliminar el esquema mayor puesto que obviamente perderlamos dependencias, nos quedan como esquemas resultantes: MEG, ES, TP, MGTHA, HET.

OBSERVACION: Felizmente los esquemas encontrados podemos asegurar que se encuentran al menos en 4FN.

Proyecciòn de DISTRIBUCION DE GRUPOS en los esquemas encontrados:

MESTROS Y ESTUDIANTES POR GRUPO

M	G	E	[illegible]
100-01	01	78532-1	[illegible]
100-01	01	78296-1	[illegible]
100-01	01	78438-1	[illegible]
100-01	02	77083-2	[illegible]
100-01	02	76942-2	[illegible]

SITUACION DE ESTUDIANTES

E	S
78532-1	R
78296-1	I
78438-1	R
77083-2	R
76942-2	R

MAESTROS

T	P
78-101	JUAN PEREZ
81-141	HECTOR LOPEZ

GRUPOS

M	G	T	H	A
100-01	01	78-101	14-15	K-101
100-01	02	81-141	7-8	K-101

MAESTROS Y ESTUDIANTES POR HORARIO

M	E	H
78-101	78532-1	14-15
78-101	78296-1	14-15
78-101	78438-1	14-15
81-141	77083-2	7-8
81-141	76942-2	7-8

OBSERVACIONES: Como vemos a fuerza de conservar dependencias hemos obtenido un esquema poco significativo: MEH (al menos desde nuestro particular punto de vista, no podemos asegurar que no existe una circunstancia particular que haga tal esquema muy importante para una aplicación determinada).

BIBLIOGRAFIA

[1] Ullman, J.D. [1980]. "Principles of Database Systems", Computer Science Press Inc., Rockville, Maryland.

[2] Date, C.J. [1981]. "An Introduction to Database Systems" (Tercera Edición), Addison-Wesley, Reading, Mass.

[3] Atre, S. [1980]. "Database: Structured Techiques for Design, Performance and Management", John Wiley & Sons, Inc., New York, N.Y.

[4] Stout, Q.F., Woodworth, A.P. [1983]. "Relational Databases" American Mathematical Monthly, Feb. 83, pp. 101-117.

[5] Fagin, Ronald [1977]. "Multivalued Dependencies and a New Normal Form for Relational Databases", ACM Transations on Database Systems, Vol. 2, No. 3, Sept. 1977, pp. 262-278.

[6] Beeri, C., Fagin, R. Howard, H.J. [1977]. "A Complete Axiomatization for Functional and Multivalued Dependencies" ACM/SIGMOD International Symposium on Management of Data, pp. 47-61.

[7] Fagin, R. [1981]. "A Normal Form for Relational Databases That is Based on Domains and Keys", ACM Transactions on Database Systems, Vol. 6, No. 3, Sept. 1981, pp. 387-415.

[8] Fagin, R. [1979]. "Normal Forms and Relational Database Operators", ACM/SIGMOD International Symposium on Management of Data, pp. 153-160.