

57
2ej



UNIVERSIDAD NACIONAL AUTÓNOMA DE MÉXICO

FACULTAD DE CIENCIAS

MANIPULACION DE DATOS EN EL MODELO
RELACIONAL CON UNA ORIENTACION AL
CALCULO RELACIONAL

T E S I S

QUE PARA OBTENER EL TITULO DE:

A C T U A R I O

P R E S E N T A :

MARIA VERONICA GARCIA MORALES



DIRECTOR DE TESIS,

M. en C. Javier García García

México, D. F.

TESIS CON
FALLA DE ORIGEN

1991



Universidad Nacional
Autónoma de México



UNAM – Dirección General de Bibliotecas Tesis Digitales Restricciones de uso

DERECHOS RESERVADOS © PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

Todo el material contenido en esta tesis está protegido por la Ley Federal del Derecho de Autor (LFDA) de los Estados Unidos Mexicanos (México).

El uso de imágenes, fragmentos de videos, y demás material que sea objeto de protección de los derechos de autor, será exclusivamente para fines educativos e informativos y deberá citar la fuente donde la obtuvo mencionando el autor o autores. Cualquier uso distinto como el lucro, reproducción, edición o modificación, será perseguido y sancionado por el respectivo titular de los Derechos de Autor.

CONTENIDO

OBJETIVO	5
INTRODUCCION	6
1 BASES DE DATOS RELACIONALES	
1.1 Relaciones.	11
1.2 Normalización.	24
2 MANIPULACION DE DATOS EN EL MODELO RELACIONAL	
2.1 Lenguajes de manipulación.	51
2.1.1 Algebra relacional.	54
2.1.2 Cálculo relacional.	61
2.2 Comparación del álgebra relacional y el cálculo relacional.	70
3 EJEMPLO APLICATIVO	
3.1 El lenguaje manipulador de datos NATURAL.	79
3.2 Caso de estudio: Sistema de Contabilidad General.	86
3.3 Operaciones de consulta	89
3.4 Ejemplos de aplicaciones construidos con NATURAL.	
CONCLUSIONES	123
APENDICE	126
BIBLIOGRAFIA	134

INTRODUCCION

Los sistemas manejadores de bases de datos son sistemas computarizados para el registro y mantenimiento de información y su importancia consiste en que proporcionan un control centralizado de los datos.

Los usuarios de los sistemas manejadores de bases de datos disponen de lenguajes de programación, consulta o de propósitos especiales para la actualización de la información.

La parte de estos lenguajes que concierne a los objetos y operaciones de la base de datos es conocida como sublenguaje de datos y es la combinación de un lenguaje de definición de datos y un lenguaje de manipulación de datos.

El lenguaje de definición de datos describe las entidades de la base de datos tal como la percibe el usuario y el lenguaje de manipulación de datos se encarga del procesamiento de estas entidades. Las entidades son tanto objetos como asociaciones entre los objetos representados en la base de datos.

Los datos son presentados al usuario en una forma o vista en la que se omiten los detalles sobre la manera en la que están representados en el almacenamiento.

Las formas o estructuras de datos soportadas al nivel del usuario son un factor que afecta a los componentes del sistema, porque impone el diseño de los lenguajes de manipulación, ya que cada operación de estos lenguajes se define en términos de su efecto sobre las estructuras de datos.

Los sistemas manejadores de bases de datos se clasifican de acuerdo a la estructura de datos y operadores asociados que soportan. Así, se puede hablar de tres enfoques principales: el enfoque relacional, el enfoque jerárquico y el enfoque de red.

En otras palabras, los distintos enfoques difieren en la manera en que permiten al usuario ver las entidades, así como manipular éstas y las asociaciones, ya que en el enfoque relacional las asociaciones se representan de la misma manera que las entidades y en los enfoques jerárquico y de red se representan por medio de ligas.

En el enfoque de red las ligas se pueden combinar para modelar asociaciones complejas de muchos a muchos y en el enfoque jerarquico esto no es posible.

En la actualidad la tecnología avoca sus esfuerzos al desarrollo basado en el enfoque relacional por las ventajas que a continuación se enumeran:

- * Proporciona un alto grado de independencia de los datos.
- * Proporciona una vista sencilla y comunitaria de la base de datos, de modo que una amplia diversidad de usuarios en una empresa pueda interactuar con una vista común.
- * Simplifica el trabajo del administrador de bases de datos.
- * Introduce un fundamento teórico en la administración de bases de datos.
- * Las relaciones son tratadas como operandos en

lugar de procesarse elemento por elemento.

Todo lo anterior se fundamenta en el hecho de que los archivos que obedecen ciertas restricciones se pueden considerar relaciones matemáticas y por tanto, la teoría elemental de las relaciones se puede aplicar a varios problemas prácticos.

Este trabajo se centra en la manipulación del enfoque relacional, por lo que en el capítulo 1 se presenta la teoría que respalda a dicho enfoque.

Teniendo como base el capítulo 1, en el capítulo 2 se profundiza en la parte manipulativa del modelo, para presentar finalmente en el capítulo 3 un ejemplo de aplicación de un sistema manejador de bases de datos relacionales y de su lenguaje manipulativo.

1. BASES DE DATOS RELACIONALES

1.1 RELACIONES

Un sistema de bases de datos es un sistema computarizado para almacenar, dar mantenimiento y tener acceso a datos. Los componentes principales de un sistema de bases de datos son: los datos, el hardware, el software y los usuarios.

Se entiende por datos los valores registrados en el sistema. El hardware está formado por los componentes físicos de la computadora, así como el software es el conjunto de programas que hacen funcionar al computador. Los usuarios son aquellas personas que utilizan la computadora, ya sea para escribir programas de aplicación que utilicen bases de datos, para consultar bases de datos desde una terminal o como administrador de bases de datos (persona encargada del control general del sistema de base de datos).

Existe una estructura muy general para analizar cualquier sistema de bases de datos. Esta estructura se divide en tres niveles: el nivel interno, que es el más cercano al almacenamiento físico de los datos; el nivel externo, que se refiere a la manera en que cada usuario ve

los datos; y el nivel conceptual, que es un nivel de mediación entre el nivel interno y externo.

El nivel conceptual consta de una descripción abstracta de los diversos tipos de entidades que se necesitan procesar de alguna manera. Si el nivel conceptual no está bien definido, entonces es probable que las aplicaciones y los esquemas externos tampoco estén bien definidos, lo que confundiría al usuario, aumentaría la necesidad de reprogramar y entrañaría un riesgo mayor de error.

Por todo lo anterior, el nivel conceptual no debe cambiar a menos que algún ajuste en el mundo real demande ajustar también alguna definición para que se siga reflejando la realidad. Este nivel no debe de estar influido por consideraciones relativas a la manera como se van a almacenar y tener acceso físicamente los datos, ni por la manera como van a usarse en aplicaciones específicas, sino basarse en alguna forma abstracta de ver los datos.

Dentro de los sistemas de bases de datos existen varios enfoques para representar a nivel externo, es decir, al usuario, una visión estructurada de los datos almacenados.

Los enfoques principales como ya se mencionó son el relacional, el jerárquico y el de red. Los enfoques jerárquico y de red no son tratados a detalle ya que el desarrollo actual de los sistemas manejadores de bases de datos está basado en el enfoque relacional, por las ventajas presentadas en la introducción.

El enfoque relacional, cuya parte manipulativa es el objeto de estudio del presente trabajo, es el que presenta los datos organizados en tablas formadas de renglones y columnas cuyas intersecciones son las posiciones ocupadas por los datos. Bajo esta estructura los atributos de las entidades de las que se quiere registrar datos están representados por las columnas, mientras que los renglones representan a los registros.

Un sistema relacional es aquel en el cual el usuario percibe los datos en forma de tablas; asimismo los operadores para la manipulación de los datos, por ejemplo para la recuperación de información, producen nuevas tablas a partir de las originales. En el sistema relacional, los valores de las tablas deben tener la característica de ser atómicos, esto es, que en cada posición de la tabla haya

Únicamente un valor no un conjunto de valores. Otra característica que deben cumplir las tablas del modelo relacional es que la información de la base de datos es representada por medio de valores explícitos y no hace uso de ligas o apuntadores que conecten a una tabla con otra. Si existiera alguna relación entre dos tablas, ésta podría estar representada por otra tabla, por ejemplo, considérese la tabla A de tarjetahabientes y la tabla B de tipos de tarjeta.

A

NUM-TH	NOM-TH
100	TERESA LOPEZ
105	LUIS RODRIGUEZ
153	JORGE SANCHEZ
201	CARMEN BARAJAS
320	LETICIA PEREZ

B

TIPO-TAR	NOM-TAR
1	NACIONAL
2	INTERNACIONAL
3	FINANCIERA

Para representar la relación " qué tipo de tarjeta tiene cada tarjetahabiente" se emplearía la tabla C de cuentas.

C

TIPO-TAR	NUM-TH	NOM-TH	LIM-CRE	SALDO
1	153	JORGE SANCHEZ	750 000	0
1	320	LETICIA PEREZ	320 000	300 000
2	153	JORGE SANCHEZ	250 000	150 000
3	100	TERESA LOPEZ	500 000	250 000
3	105	LUIS RODRIGUEZ	1000 000	0
3	201	CARMEN BARAJAS	1000 000	150 000

En la tabla A el número de tarjetahabientes (NUM-TH) es único y se puede usar como identificador, al igual que en la tabla B el tipo de tarjeta.

Como identificador único de los renglones de la tabla C se considera la combinación TIPO-TAR / NUM-TH.

Las bases de datos relacionales son llamadas así porque están basadas en el modelo relacional que es una teoría que usa el término matemático de relación para nombrar a las tablas. Esta teoría se conoce con el nombre de "Modelo Relacional", es decir, que cuando un sistema es reconocido

como relacional significa que este sistema está construido bajo los principios básicos del modelo relacional.

Como el modelo relacional presenta una forma de ver y manipular los datos, está relacionado con tres aspectos de éstos: su estructura, su integridad y su manipulación, además de hacer uso de una terminología formal.

Los terminos que usa el modelo relacional se pueden equiparar informalmente con los ya mencionados:

- * Relación, que corresponde a lo que hasta el momento se ha mencionado como tabla.
- * Eneada, que corresponde a los renglones de la tabla.
- * Atributo, que corresponde a las columnas de la tabla.
- * Llave primaria, que equivale al identificador único de la tabla (columna o combinación de columnas con la propiedad de que dos renglones de la tabla no pueden tener el mismo valor en el identificador).

* Dominio, es el conjunto de valores que pueden tomar uno o más atributos (Conjunto de valores atómicos del mismo tipo).

En este trabajo se asumirá que si dos valores se pueden comparar entre sí, estos pertenecen al mismo dominio.

Una definición más formal de relación es la siguiente: Una relación sobre los dominios D_1, D_2, \dots, D_n (no todos distintos necesariamente) está compuesta por un encabezado, formado a su vez de un conjunto de atributos A_1, A_2, \dots, A_n , de tal manera que cada atributo A_i corresponde exactamente a uno de los dominios D_i ($i = 1, 2, \dots, n$) y un cuerpo constituido de eneadas donde cada eneada está formada de un conjunto de pares (atributo : valor), $(A_i : v_i)$ donde $i=1, 2, 3, \dots, n$, es decir, un par por cada atributo en el encabezado. (1)

El número de atributos de una relación determina el grado de ésta, así como la cardinalidad de una relación está determinada por el número de eneadas. La cardinalidad puede

1) DATE, C.J., "An introduction to Database Systems", Vol.I, da. ed., Addison wesley, E.U., 1986, p. 239.

cambiar con el tiempo (si se agregan o se eliminan eneadas).

A continuación se hará mención de algunas propiedades de las relaciones:

- * En una relación no existen eneadas duplicadas. Esto se deriva del hecho de que el cuerpo de la relación, es decir, el conjunto de eneadas, es un conjunto en el sentido matemático que por definición no puede contener elementos duplicados.

- * No importa el orden en que aparezcan las eneadas en una relación. Esto es, si se cambia el orden de las eneadas, la relación seguirá siendo la misma. Esta, al igual que la propiedad anterior, se deriva de que las eneadas son un conjunto y en este caso particular se aplicaría el hecho de que un conjunto no está ordenado.

- * Los atributos no están ordenados en una relación. Los atributos son vistos como un conjunto también, por lo que no tienen orden.

* Los valores de los atributos son atómicos. Esto tiene validez en el contexto relacional ya que una relación puede no tener valores atómicos.

Que los valores de los atributos sean atómicos se traduce en que la relación no contiene grupos repetitivos. Cuando esto sucede se dice que la relación está normalizada. En el enfoque relacional sólo se admiten relaciones normalizadas.

† Las asociaciones entre eneadas se representan únicamente por valores de datos en columnas sacadas de un dominio común.

Cuando sucede que dentro de una relación específica existe un atributo cuyos valores son únicos y, por tanto, se pueden usar para identificar las eneadas de una relación, se dice que ese campo es llave primaria para dicha relación.

No toda relación tendrá una llave primaria de un solo atributo; sin embargo cada relación tendrá alguna combinación de atributos que, tomados en conjunto, tienen la propiedad de la identificación única.

Se puede encontrar a veces una relación donde hay más de una combinación de atributos que poseen la propiedad de identificación única, es decir, más de una llave candidata. Una llave candidata que no es la llave primaria recibe el nombre de llave alterna.

REGLA DE INTEGRIDAD 1.

Como las eneadas representan entidades del mundo real, la llave primaria que identifica a esas eneadas en realidad es el identificador único de las entidades representadas, por lo que ningún componente de un valor de una llave primaria puede ser nulo. Un identificador que fuera nulo totalmente, equivaldría a decir que hubo una entidad que no tuvo ninguna identificación única.

Un dominio se designa como primario si y sólo si existe alguna llave primaria de un solo atributo definida sobre ese dominio.

REGLA DE INTEGRIDAD 2.

Cualquier relación que incluya un atributo que se defina sobre un dominio primario debe obedecer la siguiente

restricción; Sea D un dominio primario, y sea R_1 una relación con un atributo A que se define sobre D . Entonces en cualquier instante dado, cada valor de A en R_1 debe ser nulo, o bien igual a V , donde V es el valor de la llave primaria de alguna eneada de alguna relación R_2 (R_1 y R_2 no necesariamente distintas) con la llave primaria definida sobre D . A es denominada llave foránea.

Las llaves primarias y foráneas sirven para representar asociaciones entre eneadas, ya que con frecuencia una relación A hace referencia a una relación B . El campo en A con el cual se asocia a la relación B es la llave foránea en A y la llave primaria en B ; es así como se establecen las relaciones entre eneadas por medio de las llaves foráneas y primarias. Si el valor no nulo de la llave foránea A_n en A para alguna eneada t_1 , hace referencia a la eneada t_2 de la relación B por medio de su llave B_m , entonces, esta eneada t_2 debe existir en la relación B .

Ejemplo:

Sea D el dominio primario de las claves de delegación del Distrito Federal, esto es, existe una relación R' , cuya llave primaria se define sobre D . Sea R' la relación DELEGACIONES.

DELEGACIONES

CVE-DELEGACION	DESCRIPCION
1	Alvaro Obregon
2	Miguel Hidalgo
3	Coyoacan
4	Xochimilco
.	.
.	.
.	.

Y sea CAJEROS la relación de cajeros automáticos en el D.F., donde CVE-DELEGACION es llave foránea.

CAJEROS

CVE-CAJERO	TIPO	CALLE	COLONIA	CVE-DELEGACION
Apr0	Ncr	Tulipan	Las Flores	1
Int4	Dibo	24	Doctores	3
Sfn3	Ncr	Colina	Campestre	
Cmx8	Ncr	Azul	Nueva	4

La relación CAJEROS hace referencia a la relación DELEGACIONES con el atributo CVE-DELEGACION. Este atributo puede tener un valor nulo si se desconoce la delegación a la cual pertenece el cajero.

Si el valor es no nulo debe existir una eneada en la relación DELEGACIONES cuya llave primaria tenga ese valor. En otras palabras las asociaciones entre eneadas se representan únicamente por valores de datos en atributos con un dominio común, y tanto las entidades como las asociaciones se representan en forma de tablas, lo que conduce a una uniformidad en el conjunto de operadores para manipular los datos.

1.2 NORMALIZACION

Las normalizaciones son transformaciones que sufren las relaciones, para que a partir de estas se obtengan relaciones más fáciles de actualizar y que además sean equivalentes a las relaciones iniciales.

A excepción de la primera forma normal, la normalización no forma parte del modelo relacional, sino que es un conjunto de reglas o convenciones útiles que pueden seguir las personas encargadas del diseño de bases de datos; sin embargo, es importante tratar el tema dado que en general el diseño de una base de datos es una tarea muy compleja que debe llevarse a cabo de manera óptima para poder aprovechar las ventajas que presenta el modelo relacional. De lo contrario estas ventajas podrían verse disminuidas al tener que operar sobre una base de datos que no refleja fielmente la realidad que intenta representar o que no controla la redundancia.

La idea básica es que se comience con una relación dada y un conjunto de restricciones y se reduzca en forma sistemática esa relación a un conjunto de relaciones

equivalentes a la original, pero en alguna forma preferible a ella, usando las restricciones para guiarse en el proceso de reducción. El objetivo general de dicho proceso es reducir la redundancia y se puede entender como una disciplina con la que el diseñador de bases de datos refleja una parte de la semántica del mundo real.

En este punto es pertinente decir que las nociones inherentes a la normalización atañen al significado de los datos, mientras que, como se apreciará en el capítulo 2, el cálculo y el álgebra relacional conciernen únicamente a los valores reales de los datos, es decir, que estos lenguajes no requieren que las relaciones sobre las que operan se encuentren en alguna forma normal adicional.

Se hace referencia a operaciones de manipulación presentadas en el apartado del álgebra relacional del capítulo 2. Estas operaciones son la proyección que, descrita brevemente es la selección de un subconjunto de atributos de una relación R dada para producir otra relación R' ; la reunión de dos relaciones es la concatenación de las eneadas de cada una, donde los atributos sobre los cuales se efectúa la operación cumplen con cierta condición. En

particular, para la reunión efectuada en las normalizaciones se debe cumplir la condición de igualdad de los atributos sobre los cuales se efectúa la reunión.

La otra operación es la unión de dos relaciones, que es el conjunto de encadas que pertenecen a alguna de las relaciones o a ambas.

Las transformaciones se efectúan siguiendo la idea de descomponer una relación en dos de sus proyecciones, para que esta relación esté normalizada, es decir, que los dominios de la relación contengan valores atómicos únicamente.

De manera similar el modelo relacional tiene como ventaja que al hacer cambios en la representación de los datos por la introducción de nuevos tipos de éstos, dichos cambios no afectan al usuario.

Un concepto básico en la teoría de la normalización es el de formas normales. Se dice que una relación está en una forma normal particular si satisface las siguientes condiciones: Los valores de los atributos deben ser

atómicos; la base de datos debe representar las restricciones del mundo real; no se deben mezclar distintos tipos de información en una relación; se deben efectuar actualizaciones a una relación sin tener que afectar a otra relación dentro de la misma base de datos. Como ya se mencionó anteriormente cada una de las formas normales cumple con ciertas restricciones. Existen algunos conceptos esenciales en los que se sustentan las relaciones mencionadas.

El primero de los conceptos es el de dependencia funcional, que dice que dada una relación R, el atributo Y es funcionalmente dependiente del atributo X de R si y sólo si, siempre que dos eneadas de R coincidan en sus valores de X, también coincidan en sus valores de Y (los atributos X y Y pueden ser atributos compuestos). (2)

En símbolos se tiene $R.X \rightarrow R.Y$, que se lee como el atributo R.Y es funcionalmente dependiente del atributo R.X.

Una dependencia funcional es una forma de restricción

2) DATE, C.J., "An introduction to Database Systems", Vol. I, 4a. ed., Addison Wesley, E.U., 1986, p.364.

de integridad ya que cuando se dice que la relación F satisface la $R.X \rightarrow R.Y$ se da a entender que cada eneada de esa relación satisface esa restricción.

Para determinar las dependencias funcionales es necesaria la comprensión del significado de los datos, es decir, que una dependencia funcional es una restricción en el mundo real que la base de datos representa, de tal manera que en la definición de la base de datos se debe especificar esta restricción.

Otra idea básica para la definición de las formas normales es el concepto de dependencia funcional completa. Se dice que el atributo Y es funcionalmente dependiente en forma completa del atributo X , si es funcionalmente dependiente de X y no depende funcionalmente de ningún subconjunto propio de X .

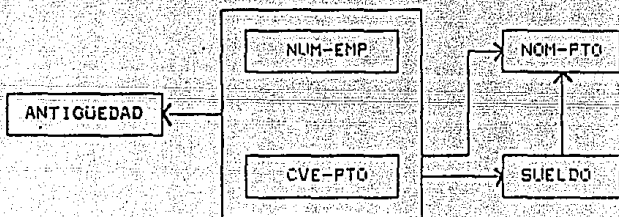
A continuación se presenta la tabulación de una relación en la que se ejemplifican ciertas dependencias no deseables. Sea EMPLEADO la relación que contiene la información

de los empleados de una empresa así como los puestos que ocupan.

NUM-EMPLEADO	CVE-PTO	ANTIGUEDAD	NOM-PUERTO	SUELDO
526	AP-1	2	ANALISTA PROG. JR.	100
526	AP-2	1	ANALISTA PROG. SR.	150
531	AP-1	1	ANALISTA PROG. JR.	100
653	AP-2	3	ANALISTA PROG. SR.	150
781	LF-0	1	LIDER DE PROYECTO	200

Se supone en este ejemplo que el sueldo es funcionalmente dependiente del atributo NOM-PTO, es decir, que el sueldo se determina por el puesto. Por ejemplo, todos los empleados que son líderes de proyecto perciben un sueldo de 200.

La llave primaria de la relación EMPLEADO está formada por el número de empleado y la clave de puesto, ya que un empleado puede ocupar más de un puesto. La siguiente figura muestra las dependencias funcionales:



El nombre del puesto y el sueldo no dependen completamente de la llave primaria (NUM-EMP, CVE-PTO), además de que no son independientes entre sí, lo que produce anomalías en las operaciones básicas de actualización ya que por ejemplo para la operación de inserción se tiene que no es permitido incluir eneadas con valor nulo en toda o parte de la llave primaria. Esto implicaría que esta relación no podría reflejar que el puesto con clave CO-0 corresponde al de 'Coordinador' hasta que se registre algún empleado con esa clave de puesto.

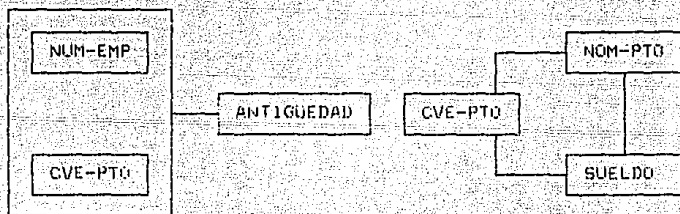
Para la operación de eliminación de eneadas, si se desea eliminar la eneada (781, LP-0, 1, Líder de Proyecto, 200), no sólo se elimina el registro de dicho empleado con la correspondiente clave de puesto, sino que se elimina también la información de que el nombre de puesto para la clave LP-0 es 'Líder de Proyecto'.

Por otro lado si se cambia el nombre del puesto para la clave AP-2, de 'Analista Programador Sr.' a 'Analista Programador A', se tendrían que buscar todas las eneadas que relacionan a la clave AP-2 y el nombre del puesto 'Analista

Programador Sr.' y cambiar al nuevo nombre; también se podría presentar la inconsistencia de que algunas eneadas con clave de puesto AP-2 tuvieran como nombre de puesto 'Analista Programador Sr.' y otras 'Analista Programador A'.

Estas dependencias pueden eliminarse sustituyendo la relación EMPLEADO por dos de sus proyecciones. Sean estas las relaciones EMPLEADO-2 (NUM-EMP, CVE-PTO, ANTIGUEDAD) y PUESTOS (CVE-PTO, NOM-PTO, SUELDO).

A continuación se muestran las dependencias funcionales de cada relación en los diagramas.



En los diagramas se aprecia que se eliminaron las dependencias funcionales no completas, lo que se tradujo en

la eliminación de anomalías en la actualización. Lo que se eliminó fue la mezcla de dos tipos de información, porque NOM-PTO y SUELDO no describen el puesto desempeñado por un empleado dado, sino que describen las características del puesto.

Con la descomposición de la relación EMPLEADO, los atributos NOM-PTO y SUELDO sí describen la entidad identificada por la llave primaria CVE-PTO.

En general, las entidades identificadas por los valores de las llaves primarias son las entidades fundamentales acerca de las cuales se registran datos en la base de datos.

EMPLEADO-2

NUN-EMP	CVE-PTO	ANTIGÜEDAD
526	AP-1	2
526	AP-2	1
531	AP-1	1
653	AP-2	3
781	LP-0	1

PUESTO

CVE-PTO	NOM-PTO	SUELDO
AP-1	ANALISTA PROG. JR.	100
AP-2	ANALISTA PROG. SR.	150
LP-0	LIDER DE PROYECTO	200
CO-0	COORDINADOR	300

Con la operación de inserción en la relación PUESTO se puede almacenar la información de que a la clave de puesto CO-0 le corresponde el nombre de puesto 'Coordinador' y un sueldo de 300, aunque hasta ahora no haya ningún empleado que desempeñe ese puesto.

En la operación de eliminación de eneadas, si se elimina en la relación EMPLEADO-2 la eneada (781, LP-0, 1), no se perderá la información de que a la clave LP-0 le corresponde el nombre de puesto 'Lider de Proyecto' y el sueldo 200.

Si se desea cambiar el nombre de puesto para una clave dada esto se efectúa directamente en la eneada correspondiente de la relación PUESTO y no en todas las eneadas de la relación EMPLEADO en que esté involucrada la llave.

Una relación está en primera forma normal (1FN) si y sólo si satisface la condición de contener únicamente valores atómicos.

Por último se define que un atributo es no primo si no es componente de la llave primaria.

En este punto se cuenta ya con las nociones necesarias para definir la segunda forma normal (2FN).

Una relación R está en 2FN si y sólo si está en 1FN y cada atributo no primo es completamente dependiente de la llave primaria (3).

Una relación que esté en 1FN y no en 2FN siempre se puede reducir a un conjunto equivalente de relaciones en 2FN.

Dicha reducción consiste en reemplazar las relaciones por proyecciones adecuadas.

3) Ibidem. p. 370.

Estas proyecciones son equivalentes a la relación original porque esta se puede recuperar tomando la reunión natural de estas proyecciones, de manera que ninguna información se pierda.

En general, dada una relación R con los posibles atributos A , B y C que satisfagan a la DF $R.A \rightarrow R.B$, R se descompone, sin pérdidas, en sus proyecciones $R_1(A,B)$ y $R_2(A,C)$.

Cualquier información que se pueda derivar de la estructura original también se puede derivar de la nueva estructura. Por otra parte la nueva estructura puede contener información que no se puede representar en la estructura original, es decir, que la nueva estructura es un mejor reflejo del mundo real.

Se dice que una relación está en tercera forma normal (3FN) si y solo si está en 2FN, y todo atributo no primo es dependiente no transitivamente de la llave primaria. (4)

4) Ibidem, p. 373.

Una relación que esté en 2FN y no en 3FN se puede reducir siempre a un conjunto equivalente de relaciones en 3FN.

Para llegar a la tercera forma normal (3FN) se puede seguir un proceso análogo al de la 2FN, sólo que el objetivo de reducir una relación a su equivalente 3FN es el de eliminar la dependencia transitiva que, en nuestro ejemplo corresponde a la dependencia que existe entre el atributo SUELDO y NOM-PTO.

En la teoría de la normalización se maneja una definición más fuerte que la de 3FN ya que ésta no satisface el caso de una relación que posea dos o más llaves candidatas compuestas y traslapadas.

Dos llaves se traslapan si comprenden dos o más atributos cada una y si tienen algún atributo en común.

Esta es la definición de la forma normal de Boyce y Cood (FNBC) y aunque es más restrictiva que la 3FN, cualquier relación se puede descomponer, sin pérdidas, en un conjunto equivalente de relaciones en FNBC.

En la definición de la FNBC se llama determinante (funcional) a un atributo que puede ser compuesto y del cual depende funcionalmente en forma completa algún otro atributo.

Entonces, se puede definir la FNBC como sigue: Una relación R está en forma normal de Boyce/Codd (FNBC) si y sólo si cada determinante es una llave candidata. (5)

En el proceso de conversión de una relación a una forma más deseable, se puede presentar el caso de que esta se puede descomponer en varias formas diferentes.

Para seleccionar la descomposición más conveniente se debe elegir la descomposición cuyas proyecciones sean independientes entre sí, es decir, que se puedan hacer actualizaciones a cualquiera de las proyecciones sin afectar a la otra.

Esto siempre que la actualización no viole la restricción de DF que se aplica a esta proyección.

5) Ibidem. p. 37d.

Dos proyecciones R_1 y R_2 de una relación son independientes si y sólo si toda DF en F se puede deducir lógicamente de las de R_1 y R_2 y si los atributos comunes a R_1 y R_2 forman una llave candidata, al menos para una de las dos relaciones. (6)

Se conocen como atómicas las relaciones que no pueden descomponerse en componentes independientes.

De lo anterior, se desprende que en ocasiones pueda estar en conflicto el descomponer una relación en componentes independientes y el descomponerla en componentes en FNBC.

Cuando se tiene el caso de relaciones que están en FNBC pero que contienen mucha redundancia se aplica un proceso para reducir las a la siguiente forma normal.

La redundancia sin control puede provocar inconsistencia, es decir que los valores de los atributos donde se registra la información duplicada pueden no

6) Cfr. ibidem. p. 380.

concordar en cada instante, además de que se ocupa más espacio de almacenamiento.

Por lo anterior en la mayoría de los casos es deseable eliminar la redundancia.

Estas relaciones pueden ser el resultado de transformar relaciones no normalizadas, es decir, relaciones que contienen grupos de repetición como valores de atributo, en relaciones con una estructura plana.

Por ejemplo, sea R la relación formada por el atributo atómico A y por los atributos B y C que son grupos de repetición.

Se muestra una tabulación de la relación R:

A	B	C
a1	b11 b12	c11
a2	b21 b22 b23	c21 c22 c23 c24

Sea R' la relación transformada. También se ejemplifica una tabulación de dicha relación.

R'

A	B	C
a1	b11	c11
a1	b12	c11
a2	b21	c21
a2	b21	c22
a2	b21	c23
a2	b21	c24
a2	b22	c21
a2	b22	c22
a2	b22	c23
a2	b22	c24
a2	b23	c21
a2	b23	c22
a2	b23	c23
a2	b23	c24

Ahora la relación R' equivalente a la relación R es una relación normalizada aunque con mucha redundancia como se puede observar. Se aplicó esta transformación ya que se supone que no existen dependencias funcionales, por lo que no se puede descomponer la estructura de la relación original R en proyecciones, y así se tiene una relación en 4NBC: R' donde los atributos B y C son independientes entre sí.

Como la relación R' está en 4NBC se desprende que la llave primaria está compuesta por los atributos A , B y C y

que no existen otros determinantes funcionales.

El siguiente paso es substituir la relación R' por sus dos proyecciones R1(A, B) y R2(A, C). Estas relaciones también están en FNBC. A continuación se presentan sus tabulaciones correspondientes:

R1		R2	
A	B	A	C
a1	b11	a1	c11
a1	b12	a2	c21
a2	b21	a2	c22
a2	b22	a2	c23
a2	b23	a2	c24

Esta descomposición se efectúa sobre la base de un nuevo tipo de dependencia ya que no se puede hacer sobre las dependencias funcionales.

El nuevo tipo de dependencia es una generalización de las dependencias funcionales y se conoce como dependencia multivaluada (DMV).

Dada una relación R con atributos A, B y C la dependencia multivaluada R.A \twoheadrightarrow R.B se cumple en R si y sólo si el conjunto de valores de B que corresponden a un

par (valor de A, valor de C) dado en R, depende tan sólo del valor de A y es independiente del valor de C. A, B y C pueden ser compuestos. (7) Como la dependencia multivaluada $R.A \twoheadrightarrow R.B$ se cumple si y sólo si también se cumple la DMV $R.A \twoheadrightarrow R.C$, es común expresarlas usando la notación $R.A \twoheadrightarrow R.B/R.C$.

Una DF es una DMV donde el conjunto de valores dependientes se compone de un solo valor. El problema de la relación R con grupos repetitivos en los valores de sus atributos consiste en que la relación comprende DMV's que no son de DF's.

Al descomponer la relación R en las proyecciones R1 y R2 ninguna de estas relaciones contiene una DMV como la mencionada, por eso representan una mejora con respecto a la relación original.

La sustitución de la relación original R por sus proyecciones R1(A, B) y R2(A, C) siempre se puede llevar a cabo sin pérdidas si y sólo si la DMV $A \twoheadrightarrow B/C$ se

7) Ibidem. p. 384.

cumple en R. En seguida se define la cuarta forma normal basada en los conceptos anteriores.

Una relación R está en cuarta forma normal (4FN) si y sólo si, siempre que exista una DMV en R, todos los atributos de R también son funcionalmente dependientes de A, es decir, que las únicas dependencias en R son de la forma siguiente: $K \twoheadrightarrow X$ donde K es una llave candidata de la relación y X es algún atributo de R. (8)

El reemplazo de una relación por dos de sus proyecciones no es la única operación aplicable en el proceso de descomposición. Existen relaciones que no se pueden descomponer sin pérdidas en dos proyecciones, pero que se pueden descomponer sin pérdidas en tres o más.

Para el estudio de las relaciones mencionadas se introduce la noción de dependencia de reunión (DR) que es una restricción, tal como lo es una DF o una DMV.

Se dice que la relación R satisface la $DR*(X, Y, \dots, Z)$

8) Idem. p. 385.

si y sólo si es la reunión de sus proyecciones sobre X, Y, \dots, Z , donde X, Y, \dots, Z son subconjuntos del conjunto de atributos de R . (9)

Es equivalente afirmar que $R(A, B, C)$ se puede descomponer sin pérdidas en $R_1(A, B)$ y $R_2(A, C)$ si y sólo si $A \twoheadrightarrow B/C$ se cumple en R , a afirmar que $R(A, B, C)$ satisface la $DR^*(AB, AC)$ si y sólo si satisface la DMV $A \twoheadrightarrow B/C$. (9) De lo anterior se deduce que las DR 's son una generalización de las DMV 's.

A las relaciones que se encuentran en 4FN pero que contienen una DR se les puede descomponer en sus proyecciones especificadas por la dependencia de reunión.

Este proceso de descomposición se puede repetir hasta que todas las proyecciones se encuentren en quinta forma normal (5FN) que a continuación se define.

Una relación R está en quinta forma normal (5FN) o forma normal de proyección-reunión (FN/PR), si y sólo si,

9) Cfr. ibidem. p. 388.

toda dependencia de reunión en R está implicada por las llaves candidatas de R. (10)

Determinar las DR de una relación no es una tarea fácil ya que a diferencia de las DF's y las DMV's no tienen una interpretación directa en el mundo real. Se puede decir que la SFN es la última forma normal con respecto a la proyección y a la reunión, porque las únicas descomposiciones válidas para una relación que está en SFN son las que se basan en las llaves candidatas y estas descomposiciones no proporcionan ninguna ventaja.

Se presenta el ejemplo de una relación y el proceso de reducción que sufre hasta tener un conjunto de relaciones en SFN, equivalentes a la relación original. Sea A la relación que representa el avance de las distintas etapas de desarrollo de los programas en el departamento de nuevos proyectos. Los atributos son PROGRAMA, MÓDULO, SISTEMA, RESPONSABLE, CUE-EMP, ETAPA Y AVANCE y la llave primaria es (PROGRAMA, ETAPA). Se ejemplifica el proceso de reducción por medio de tabulaciones de las relaciones.

10) Idem. p. 389.

A

PROGRAMA	MODULO	SISTEMA	RESPONSABLE	CVE-EMP	ETAPA	AVANCE
AA001	AA	A	Juan Avalos	JA10	Análisis	100
	AA	A	Gerardo Sam	GS12	Diseño	100
	AA	A	Gerardo Sam	GS12	Programación	100
	AA	A	Javier Lear	JL49	Pruebas	80
AA002	AA	A	Gerardo Sam	GS12	Análisis	100
	AA	A	Javier Lear	JL49	Diseño	100
	AA	A	Gerardo Sam	GS12	Programación	80
	AA	A	Javier Lear	JL49	Pruebas	0
AB001	AB	A	Javier Lear	JL49	Análisis	100
	AB	A	Javier Lear	JL49	Diseño	100
	AB	A	Javier Lear	JL49	Programación	100
	AB	A	Javier Lear	JL49	Pruebas	100

Se aprecia en la tabulación de la relación A que contiene grupos de repetición como valor de los atributos, excepto PROGRAMA. Para transformar la relación en una que esté en 1FN se eliminan los grupos repetitivos, es: es, se fuerza a que la relación tenga únicamente valores atómicos en sus atributos. Sea R la relación transformada.

A

PROGRAMA	MODULO	SISTEMA	RESPONSABLE	CVE-EMP	ETAPA	AVANCE
AA001	AA	A	Juan Avalos	JA10	Análisis	100
AA001	AA	A	Gerardo Sam	GS12	Diseño	100
AA001	AA	A	Gerardo Sam	GS12	Programación	100
AA001	AA	A	Javier Lear	JL49	Pruebas	80
AA002	AA	A	Gerardo Sam	GS12	Análisis	100
AA002	AA	A	Javier Lear	JL49	Diseño	100
AA002	AA	A	Gerardo Sam	GS12	Programación	80
AA002	AA	A	Javier Lear	JL49	Pruebas	0
AB001	AB	A	Javier Lear	JL49	Análisis	100
AB001	AB	A	Javier Lear	JL49	Diseño	100
AB001	AB	A	Javier Lear	JL49	Programación	100
AB001	AB	A	Javier Lear	JL49	Pruebas	100

La llave primaria de la relación B es la misma de la relación A y los atributos MODULO y SISTEMA no son completamente dependientes de la misma ya que se determinan únicamente por el atributo PROGRAMA. Esto quiere decir que la relación no se encuentra en 2FN, por lo que se descompone en la relación C (PROGRAMA MODULO SISTEMA) Y D (PROGRAMA ETAPA RESPONSABLE CVE-EMP AVANCE) con PROGRAMA y PROGRAMA, ETAPA como llaves primarias en cada relación.

C

PROGRAMA MODULO SISTEMA

AA001	AA	A
AA002	AA	A
AB001	AB	A

D

PROGRAMA	ETAPA	RESPONSABLE	CVE-EMP	AVANCE
AA001	Análisis	Juan Avalos	JA10	100
AA001	Diseño	Gerardo Sam	GS12	100
AA001	Programación	Gerardo Sam	GS12	100
AA001	Pruebas	Javier Lear	JL49	60
AA002	Análisis	Gerardo Sam	GS12	100
AA002	Diseño	Javier Lear	JL49	100
AA002	Programación	Gerardo Sam	GS12	80
AA002	Pruebas	Javier Lear	JL49	0
AB001	Análisis	Javier Lear	JL49	100
AB001	Diseño	Javier Lear	JL49	100
AB001	Programación	Javier Lear	JL49	100
AB001	Pruebas	Javier Lear	JL49	100

El siguiente paso para tener las relaciones en 3FN es eliminar las dependencias transitivas de la llave primaria

en cada caso. En la relación C se supone que el atributo SISTEMA depende transitivamente de la llave a través del atributo MODULO y en la relación D el atributo RESPONSABLE depende transitivamente de la llave primaria por medio del atributo CVE-EMP.

Esto se resuelve descomponiendo las relaciones C en sus proyecciones E (PROGRAMA MODULO) F (MODULO SISTEMA) y D en sus proyecciones G (PROGRAMA ETAPA CVE-EMP AVANCE) y H (CVE-EMP RESPONSABLE).

E

PROGRAMA	MODULO
AA001	AA
AA002	AA
AB001	AB

F

MODULO	SISTEMA
AA	A
AB	A

G

CVE-EMP	RESPONSABLE
---------	-------------

H

PROGRAMA	ETAPA	AVANCE
AA001	Análisis	100
AA001	Diseño	100
AA001	Programación	100
AA001	Pruebas	80
AA002	Análisis	100
AA002	Diseño	100
AA002	Programación	80
AA002	Pruebas	0
AB001	Análisis	100
AB001	Diseño	100
AB001	Programación	100
AB001	Pruebas	100

Al observar las cuatro relaciones tenemos que están en 4NF porque la relación G que es la única que tiene un determinante distinto a la llave primaria, suponiendo que el nombre del responsable es único, y este determinante también es llave candidata. Las cuatro relaciones se encuentran en 4FN y 5FN ya que no existen dependencias multivaluadas ni dependencias de reunión.

Para finalizar el presente apartado se mencionan otras formas normales que no se basan en la proyección y reunión. La primera trata de reducir una relación dividiéndola horizontalmente, de manera que la relación original se pueda recuperar por medio de la operación de unión (que se verá en el capítulo 2) y se denomina FN(3,3).

La segunda es la forma normal de llaves y dominios (FN/LD) que se cumple si y sólo si toda restricción sobre la relación es una consecuencia lógica de las restricciones de llave y de dominio, que son afirmaciones de que un cierto atributo o una combinación de atributos es una llave candidata y de que los valores de un cierto atributo están dentro de algún conjunto prescrito de valores.

2. MANIPULACION DEL MODELO RELACIONAL

2.1 LENGUAJES DE MANIPULACION

La estructura de datos soportada por cualquier modelo de bases de datos a nivel usuario, determina la parte manipuladora de dicho modelo.

La manipulación de datos en el modelo relacional permite manejar o procesar las relaciones o entidades definidas en la base de datos.

Esta manipulación se efectúa mediante lenguajes de manipulación de alto nivel basados en el álgebra y el cálculo relacionales, es decir, que no existe un único lenguaje manipulador en el modelo relacional, sino dos planteamientos que permiten realizar dicha función.

En el primero de ellos (álgebra relacional), el usuario especifica la secuencia de operaciones que se tendrían que efectuar para obtener el resultado deseado.

El segundo planteamiento consiste en simplemente especificar lo que se desea como resultado (cálculo relacional) y el sistema se encarga de determinar las

operaciones necesarias para producir el resultado requerido.

La completud relacional es una medida básica del poder selectivo de un lenguaje. Se dice que un lenguaje es relacionalmente completo, si cualquier relación derivable de las relaciones establecidas (es decir, la base de datos), se puede recuperar por medio de una expresión del cálculo relacional usando ese lenguaje.

Se suele suponer que el término "completud relacional" incluye la restricción adicional, a saber, que cualquier relación derivable sea recuperable por medio de una sola proposición.

Uno de los propósitos del Álgebra es proveer un conjunto de operaciones sobre las relaciones para la recuperación de datos de la base de datos relacional, entiéndase por recuperación, la creación de una nueva relación a partir de las relaciones ya existentes.

El objetivo fundamental del Álgebra relacional, que incluye la recuperación de datos, es el permitir la escritura de expresiones (estas expresiones representan

relaciones o relaciones virtuales) que tienen como propósitos definir los datos a ser modificados o eliminados como resultado de una operación de actualización, definir los datos para que se presenten en forma de vista, definir los datos a los que se tiene autorizado el acceso, definir los datos sobre los que se llevará a cabo alguna operación de control de concurrencia; y definir reglas adicionales de integridad, es decir que las expresiones son la representación simbólica de las necesidades del usuario.

Adicional al álgebra y cálculo relacional se requiere de un operador para la asignación relacional, que asigne el resultado de evaluar alguna expresión a algún valor de una relación.

2.1.1 ALGEBRA RELACIONAL

Se denomina álgebra relacional al conjunto de operadores que actúan sobre las relaciones definidas en la base de datos.

Los operadores se clasifican en dos grupos: los operadores tradicionales y los operadores especiales.

Dentro de los operadores tradicionales se encuentran la unión, intersección, diferencia y producto cartesiano; estos operadores son, en esencia, los definidos por la teoría de conjuntos; se modifican ya que sus operandos son relaciones y no conjuntos arbitrarios.

El grupo de los operadores especiales está constituido por la selección, proyección, reunión y división.

Para los operadores unión, intersección y diferencia las relaciones utilizadas como operando deben ser compatibles con respecto a la unión, es decir, deben tener el mismo grado y los j -ésimos atributos de las relaciones

deben tener el mismo dominio (si el grado de las relaciones es n , entonces $j = 1, \dots, n$). A continuación se explicará a detalle cada uno de los operadores mencionados.

OPERADORES TRADICIONALES

UNION. La unión de dos relaciones (compatibles respecto a la unión) R y S , $(R \cup S)$, es el conjunto de todas las eneadas t que pertenecen a R o a S o a ambas relaciones.

INTERSECCION. La intersección de dos relaciones (compatibles con respecto a la unión) R y S , $(R \cap S)$, es el conjunto de todas las eneadas t que pertenecen a R y a S .

DIFERENCIA. La diferencia entre dos relaciones (compatibles con respecto a la unión) R y S (el orden es importante), $(R - S)$, es el conjunto de todas las eneadas que pertenecen a R y no a S .

PRODUCTO CARTESIANO. El producto cartesiano de dos relaciones R y S , $R \times S$, es el conjunto de todas las

eneadas t tales que t es la concatenación de una eneada r que pertenece a R y una eneada s que pertenece a S . La concatenación de una eneada $r=(r_1, \dots, r_n)$ y una eneada $s=(s_{n+1}, \dots, s_{m+n})$ (en ese orden) es la eneada $t=(r_1, \dots, r_n, s_{n+1}, \dots, s_{m+n})$.

OPERADORES ESPECIALES

SELECCION (σ). Este operador produce un subconjunto de las eneadas de la relación dada para la cual se cumple un predicado específico.

El predicado es una combinación booleana, donde cada término es una comparación simple, que se puede establecer como verdadera o falsa para una eneada dada inspeccionando esa eneada por separado. Si un término comprende una comparación entre los valores de los atributos dentro de las eneadas, esos atributos deben estar definidos sobre el mismo dominio.

PROYECCION (π). El operador de proyección produce un subconjunto al seleccionar los atributos específicos:

de izquierda a derecha y eliminando finalmente las eneadas duplicadas en los atributos seleccionados.

REUNION (\cup). La reunión de la relación R sobre el atributo X con la relación S sobre el atributo Y, es el conjunto de todas las eneadas t tales que t es la concatenación de una tupla r perteneciente a R y una tupla s perteneciente a S, para el cual se cumple el predicado "r.X O s.Y" donde O representa cualquier operador de comparación escalar ($=, \neq, >, \geq, <, \leq$).

DIVISION (\div). El operador división tiene como operandos una relación R de grado m+n y una relación S de grado n. El resultado es una relación de grado m.

Se debe cumplir que el m+1-ésimo atributo de R y el i-ésimo atributo de S ($i=1, \dots, n$) estén definidos sobre el mismo dominio.

Sean los primeros m atributos de R como un atributo compuesto X y los últimos n como otro atributo compuesto Y, entonces, R es un conjunto de pares de valores (x,y) y S un conjunto de valores (y).

El resultado de R / S es el conjunto de los valores " tales que el par (x,y) aparece en R para todos los valores y que aparecen en S .

Sea AUTOMOVILES la relación de automóviles en existencia y sea COLORES la relación de los colores de automóviles previstos para el siguiente año.

AUTOMOVILES		COLORES
MARCA	COLOR	DESCRIPCION
Vw	Negro	Rojo
Vw	Rojo	Azul
Vw	Azul	Amarillo
Ford	Rojo	
Ford	Negro	
Ford	Amarillo	
Nissan	Rojo	
Nissan	Verde	
Nissan	Amarillo	
Nissan	Azul	

Y sea RESIDUO la relación que resulta de dividir AUTOMOVILES / COLORES.

RESIDUO	
MARCA	COLOR
Nissan	Rojo
Nissan	Amarillo
Nissan	Azul

Como se puede observar, el resultado de cada operación algebraica es otra relación, es decir, que las relaciones son un sistema cerrado bajo el álgebra.

Como el resultado de una operación del álgebra relacional es una relación, ésta a su vez, se puede someter a operaciones algebraicas adicionales. Los operandos de cualquier operación dada se pueden especificar de esta manera como nombres simples de relaciones o como expresiones que se evalúan en relaciones, es decir, las expresiones del álgebra relacional pueden anidarse a cualquier profundidad.

De los ocho operadores presentados sólo los operadores primitivos unión, diferencia, producto cartesiano, selección y proyección constituyen el conjunto mínimo básico de operadores. Los otros tres (reunión, división e intersección) pueden ser definidos en términos de los operadores primitivos.

La unión, diferencia, producto cartesiano, proyección y selección forman un subconjunto de operadores algebraicos relacionalmente completo, de los cuales como ya se mencionó

los demás operadores pueden derivarse.

2.1.2 CALCULO RELACIONAL

Existen dos divisiones dentro del cálculo relacional: el cálculo relacional orientado a las eneadas y el cálculo relacional orientado a los dominios.

El cálculo relacional orientado a las eneadas es llamado así porque el aspecto fundamental de este cálculo es la noción de variable de eneada, que es una variable cuyos únicos valores permitidos son eneadas de una relación específica con nombre. Si T es la variable de eneada y R es la relación, se dice que T varía sobre R y entonces, en cualquier instante dado, T representa alguna eneada individual de R .

En el cálculo de eneadas, como en el álgebra relacional, se puede definir una relación en términos de un conjunto de relaciones. A esta definición se le denomina expresión del cálculo de eneadas. Las expresiones del cálculo de eneadas están constituidas por variables de eneada, por condiciones y por fórmulas bien formadas, conceptos que a continuación son explicados.

VARIABLES DE ENEADA. Como ya se mencionó, cada variable de eneada se restringe a variar sobre alguna relación con nombre. Si la variable de eneada T representa a la eneada individual t en algún instante dado, la expresión $T.A$ representa al componente A de t en ese instante, donde A es un atributo de la relación sobre la cual varía T .

De una manera más formal se define una variable de eneada T como sigue:

RANGO DE T ES $T_1; T_2; \dots; T_n$

donde T_1, T_2, \dots, T_n son expresiones del cálculo de eneadas, que representan a las relaciones R_1, R_2, \dots, R_n . Estas relaciones deben ser compatibles con respecto a la unión y sus atributos correspondientes deben tener la misma denominación en cada una de las relaciones mencionadas.

CONDICIONES. Estas condiciones toman la forma $x \dagger y$, donde \dagger es cualquiera de los símbolos $=, \neq, <, \leq, > \text{ o } \geq$ y al menos una de las expresiones x o y es una expresión de la forma $T.A$, y la otra es una expresión semejante o una constante.

FORMULAS BIEN FORMADAS (FBF). Se construyen a partir de condiciones, operadores booleanos (\wedge , \vee , \neg , \Rightarrow) y cuantificadores (\forall , \exists) de acuerdo con las reglas que se presentan a continuación:

1. Toda condición es una FBF.
2. Si f es una FBF, también lo son (f) y $\neg(f)$.
3. Si f y g son FBF, también lo son $(f \vee g)$ y $(f \wedge g)$.
4. Si f es una FBF en la cual T aparece como una variable libre, entonces $\forall T(f)$ y $\exists T(f)$ son FBF.
5. SI FBF ENTONCES FBF.

Una ocurrencia de una variable de eneadada es la presencia del nombre de dicha variable en la expresión que constituye la FBF y ocurre en el contexto de una expresión de la forma $T.A$ o como variable que sigue a uno de los símbolos cuantificadores \forall y \exists .

Cada ocurrencia de una variable de eneadada dentro de una FBF es libre o acotada como se indica a continuación.

- Dentro de una condición, todas la ocurrencias de las variables de eneadada son libres.

- Las ocurrencias de las variables de eneada en las FBFs (f), NO (f) son libres o acotadas según sean libres o acotadas en f. Las ocurrencias de las variables de eneada en las FBF (f Y g) y (f NO g) son así mismo libres o acotadas según sean libres o acotadas en f o en g (cualquiera de las dos en que aparezcan).
- Las ocurrencias de T que son libres en f, son acotadas en las FBFs $\exists T(f)$ y $\forall T(f)$. Otras ocurrencias de variables de eneada en f son libres o acotadas en estas FBFs según sean libres o acotadas en f.
- La FBF (SI f ENTONCES g) es equivalente a la FBF (NO f) O g.

Con los elementos anteriores podemos definir una expresión del cálculo de eneadas de la siguiente forma:

T.A, U.B,...,V.C DONDE f. En donde T, U,...,V son variables de eneada; A, B,...,C son atributos de las relaciones asociadas, y f es una FBF que contiene exactamente a T, U,...,V como variables libres. Esta expresión representa una proyección del subconjunto del producto cartesiano $T \times U \times \dots \times V$ (donde T, U,..., V varían sobre todos sus valores posibles) para el cual f se evalúa como verdadera. Cuando no

se incluye "DONDE f" en la expresión, esta proyección sería la proyección del subconjunto del producto cartesiano completo. La proyección a la que se hace referencia, se efectúa sobre los atributos indicados en la lista T.A, U.B, ..., V.C. Es conveniente señalar que el cuantificador universal (\forall) no es esencial, ya que cualquier FBF que contenga \forall puede reemplazarse siempre por una FBF que contenga \exists . Esto se desprende de la identidad $\forall x (f) \equiv \text{NO } \exists x (\text{NO } f)$. A continuación se presenta una sintaxis concreta para el cálculo de eneadas, que resume los conceptos anteriores (11).

rel-def

::= DEF RELACION rel-nom [lista-atrib-nom]

var-def

::= RANGO DE var-nom ES lista-expr

expr

::= lista-componentes | lista-componentes DONDE FBF

componente

::= var-nom.atrib-nom

11) Date, C.J., "An introduction to Database Systems". Vol. I. 1a. ed., editado por Addison Wesley, E.U., 1981, p. 281.

::= comparación

- | (FBF)
- | NO FBF
- | FBF Y FBF
- | FBF O FBF
- | EXISTE var-nom (FBF)
- | PARA TODA var-nom (FBF)
- | SI FBF ENTONCES (FBF)

Notas a la gramática presentada

1. Si "xyz" es una categoría sintáctica, entonces, "lista-xyz" es una categoría sintáctica constituida de una lista de una o más "xyz", los cuales están separados por una coma.
2. Las categorías rel-nom, var-nom y atrib-nom se definen como identificadores.
3. La categoría comparación representa una comparación entre los valores escalares (constantes o valores de atributo).
4. No es permitido que los componentes sean expresiones operacionales ni que estas expresiones operacionales (por ejemplo T1.A1 + T2.A2) aparezcan como

comparandos de una comparación, simplemente por claridad.

En el cálculo relacional orientado a los dominios las variables varían sobre dominios, por lo que las expresiones del cálculo de dominios se construyen con los elementos siguientes:

VARIABLES DE DOMINIO. Cada variable de dominio se restringe a variar sobre algún dominio especificado.

CONDICIONES. Dichas condiciones son de la forma $x * y$, donde $*$ es cualquiera de los símbolos =, \neq , $<$, $<=$, $>$ o $>=$ y al menos una de x , y son variables de dominio (la otra puede ser una constante); o de la forma $R(\text{término}, \text{término}, \dots)$ -condiciones de pertenencia-, donde R es una relación y cada término es un par $A:V$, en el cual A es un atributo de R y V es una variable de dominio o una constante, es decir, una condición de pertenencia se evalúa como verdadera si y sólo si existe una enada de la relación especificada que tenga los valores especificados para los atributos especificados.

FORMULAS BIEN FORMADAS. Análogas a las definidas en vi

cálculo de eneadas, tomando en cuenta que los operandos de las condiciones en el cálculo orientado a los dominios son variables de dominio.

De igual manera se aplican las reglas a las variables libres y acotadas que se definieron para el cálculo orientado a las eneadas, haciendo la conversión requerida de variables de eneada a variables de dominio. De esta forma se puede definir una expresión del cálculo de dominios como D, E, \dots, F DONDE f para la cual D, E, \dots, F son variables de dominio y f es una FBF en donde D, E, \dots, F están contenidas como variables libres. El valor de esta expresión es igual que en el cálculo de tuplas el subconjunto del producto cartesiano $D \times E \times \dots \times F$ (donde D, E, \dots, F varían sobre todos sus valores posibles) para el cual f se evalúa como verdadera. El producto cartesiano es completo si no se incluye "DONDE f ".

Se puede decir que la diferencia básica entre el cálculo de eneadas y el de dominios es la forma en que el usuario percibe la base de datos, ya que el cálculo de dominios estimula al usuario a pensar en términos de más tipos de entidad.

El hecho de que el cálculo de dominios permita que sean omitidos los cuantificadores en algunos casos, tiende a hacer que la construcción de expresiones en este cálculo sea más sencilla que el cálculo de eneadas, ya que una entidad dada aparece varias veces en la declaración coloquial de una consulta efectuada por medio del cálculo de dominios. es decir, que la expresión en este cálculo contendrá varias ocurrencias de una variable de dominios y la expresión realizada por medio del cálculo de eneadas contendrá ocurrencias de varias variables de eneadas distintas.

2.2 COMPARACION DEL ALGEBRA RELACIONAL Y EL CALCULO RELACIONAL

El álgebra relacional y el cálculo relacional proporcionan una base para el diseño de un lenguaje manipulador que sea relacionalmente completo, es decir, que sea tan poderoso como el cálculo relacional. Cada uno tiene distintos formalismos que son distintas formas de expresión, pero en esencia son equivalentes, ya que se puede demostrar que para cada expresión del álgebra relacional, existe una relación equivalente en el cálculo y para cada expresión del cálculo relacional existe una expresión en el álgebra relacional que es equivalente (ver referencias 1 y 7 en bibliografía).

Por lo anterior se afirma que el álgebra provee una base para el diseño de lenguajes manipuladores que sean relacionalmente completos; también, de lo anteriormente citado se sigue que para demostrar que dado un lenguaje L, este es relacionalmente completo, basta con mostrar que L incluye operadores análogos a los cinco operadores algebraicos primitivos.

Así, será más fácil demostrar que un lenguaje tiene operadores algebraicos equivalentes, a probar que tiene expresiones equivalentes a las del cálculo, de lo que se desprende que la capacidad para medir el poder selectivo de un lenguaje es una de las características más importantes del Álgebra relacional.

Otra de las bondades del Álgebra sobre el cálculo relacional es que sus expresiones están libres de cuantificadores. Sin embargo, el cálculo es superior al álgebra en varios aspectos: (12)

La completud relacional representa un poder selectivo básico que en un ambiente práctico necesitaría ser mejorado, por ejemplo, el proveer de ciertas funciones de biblioteca (para ambos planteamientos, el cálculo y el álgebra relacional).

Si se analizan las expresiones concretas del cálculo relacional propuesto (sublenguaje alfa), se tiene que tales

12) Codd, E.F., "Relational Completeness of Data Base Sublanguages", en Courant Computer Science Symposia, Vol. 5, "Data Base Systems", ed. por F. Rustin, ed. Prentice-Hall, Englewood Cliffs, N.J., 1972, p. 94.

expresiones permiten la invocación potencial de funciones de biblioteca. Por ejemplo, dentro de la lista de atributos se pueden dar algunas transformaciones convenientes en la relación recuperada; se puede reemplazar un término de reunión por una función de una o más variables que se evalúe como verdadera; se puede reemplazar una variable de enreda indexada dentro de un término de reunión por una función.

Este tipo de mejoras dentro del álgebra relacional representaría la redefinición por medio de mapas de relaciones a relaciones.

La siguiente ventaja del cálculo sobre el álgebra relacional, es que el cálculo permite al usuario la definición de los datos requeridos por medio de las mismas características de los datos. en cambio, en el álgebra relacional el usuario debe establecer la secuencia de pasos a seguir para obtener los datos deseados y, a medida que los requerimientos de la base de datos son más complejos, las características de los datos se pierden entre la secuencia de operaciones dada. Además, los criterios de discriminación para la autorización en los accesos están generalmente basados en la definición de las propiedades de los datos

requeridos, no en la secuencia de operaciones establecida por el usuario para recuperarlo.

Se puede decir que el cálculo relacional está más orientado al usuario, ya que es más fácil tratar de recuperar los datos almacenados, requiriendo éstos por sus propiedades que el determinar una lista de operaciones a efectuarse para obtener el mismo resultado, lo que se traduce en una ventaja más para el cálculo relacional.

El cálculo relacional ha dado lugar a una gran variedad de lenguajes de consulta soportados por los sistemas manejadores de bases de datos mas comerciales. En particular SQL es un lenguaje no procesal que ha sido adoptado por ANSI (AMERICAN NATIONAL STANDARDS INSTITUTE) como estandar, por lo que ha tenido repercusiones en los estandares industriales.

A continuación se presenta un ejemplo de las expresiones tanto del cálculo como del álgebra relacional para efectuar una consulta a la base de datos de TARJETAHABIENTES.

La base de datos de TARJETAHABIENTES presenta la relación de CUENTAS, que contiene información del número de cuenta de cada tarjetahabiente, el tipo de tarjeta y el banco que emite cada tarjeta; así como la relación de BOLETIN con la información del número de cuenta de las tarjetas boletinadas y la fecha en que el número de cuenta ingresó al boletín.

CUENTAS

CTA-NUM	TIP-TAR	BCO-CVE
cta-1	aa	b1
cta-2	ab	b5
cta-3	aa	b5
cta-5	ac	b5
cta-9	aa	b5
cta-10	aa	b3

BOLETIN

CTA-NUM	FEC-BOL
cta-2	900224
cta-3	890518
cta-10	891230

CONSULTA

Obtener el número de cuenta y la fecha de ingreso al boletín, de las cuentas boletinadas cuyo tipo de tarjeta sea igual a 'aa' y cuya clave de banco sea 'b5'.

EXPRESION DEL ALGEBRA RELACIONAL.

Las eneadas de la relación CUENTA a considerarse son aquellas que cumplen con las condiciones (TIP-TAR = 'aa') y (BCO-CVE = 'b5'); para obtenerlas se efectúa la selección CUENTAS DONDE CUENTAS.TIP-TAR = 'aa' la cual da como resultado la siguiente relación

A

CTA-NUM	TIP-TAR	BCO-CVE
cta-1	aa	b1
cta-3	aa	b5
cta-9	aa	b5
cta-10	aa	b8

y la selección (A DONDE A.BCO-CVE = 'b5') que resulta en la siguiente relación B

B

CTA-NUM	TIP-TAR	BCO-CVE
cta-3	aa	b5
cta-9	aa	b5

El siguiente paso es construir el producto cartesiano de la relación B y la relación BOLETIN. Sea C el resultado del producto cartesiano.

C

B.CTA-NUM	B.TIP-TAR	B.BCO-CVE	BOL.CTA-NUM	BOL.FEC-BOL
cta-3	aa	b5	cta-2	900224
cta-9	aa	b5	cta-2	900224
cta-3	aa	b5	cta-3	890518
cta-9	aa	b5	cta-3	890518
cta-3	aa	b5	cta-10	891230
cta-9	aa	b5	cta-10	891230

Si a la relación C se aplica la selección:

C DONDE B.CTA-NUM = BOL.CTA-NUM

se obtendrán las encadas que cumplen las condiciones (TIP-TAR = 'aa') y (BCO-CVE = 'b5') y que se encuentran boletinadas.

Sea D el resultado de la selección anterior.

D

B.CTA-NUM	B.TIP-TAR	B.BCO-CVE	BOL.CTA-NUM	BOL.FEC-BOL
cta3	aa	b5	cta-3	890518

Finalmente para obtener únicamente el número de cuenta y la fecha de ingreso al boletín, se efectúa la proyección D(BOL.CTA-NUM, BOL.FEC-BOL) a la relación D, obteniendo así como resultado la relación E.

E

BOL.CTA-NUM	BOL.FEC-BOL
cta-3	890518

La expresión del álgebra relacional que realiza la consulta mencionada se puede resumir como se indica a continuación:

$((((CUENTA DONDE CUENTA.TIP-TAR = 'aa') DONDE CUENTA.BCO-CVE = 'b5') X BOLETIN) DONDE CUENTA.CTA-NUM = BOLETIN.CTA-NUM) BOLETIN.CTA-NUM, BOLETIN.FEC-BOL$

EXPRESION DEL CALCULO RELACIONAL

La expresión del cálculo relacional que describe la consulta propuesta es:

BOLETINX.CTA-NUM, BOLETIN.FEC-BOL DONDE EXISTE CUENTAY
(CUENTAY.CTA-NUM = BOLETINX.CTA-NUM Y CUENTAY.TIP-TAR = 'aa'
Y CUENTA.BCO-CVE = 'b5')

Esta expresión se lee: obtener el número de cuenta y la fecha de ingreso al boletín de las cuentas cuyo tipo de tarjeta sea 'aa' y cuyo banco sea 'b5'.

Para efectuar la consulta por medio del álgebra relacional se realizaron dos selecciones, un producto cartesiano, nuevamente una selección y finalmente una proyección; mientras que en el cálculo relacional únicamente se expresaron las características del resultado deseado.

3. EJEMPLO APLICATIVO

3.1 EL LENGUAJE MANIPULADOR DE DATOS NATURAL

NATURAL es el lenguaje manipulador de datos propio de el sistema manejador de bases de datos ADABAS -Adaptable Data Base System- (ver apéndice).

NATURAL permite desarrollar rápidamente aplicaciones específicas, mientras se reducen inversiones de la organización en esfuerzos de programación. Además de soportar ADABAS, NATURAL también procesa los datos almacenados en otras arquitecturas de manipulación de datos como son VSAM, DL1, IMS/DB, DB2 y SQL/DS.

NATURAL permite crear un ambiente integrado entre bases de datos tradicionales y la tecnología relacional, al utilizar el mismo potencial de manipulación de datos, dentro de un mismo programa para cualquiera de las estructuras mencionadas anteriormente,

El lenguaje NATURAL es un lenguaje de cuarta generación, es decir, que se considera a NATURAL como una

herramienta de productividad para el rápido desarrollo de aplicaciones.

Para comprender la importancia de los lenguajes de cuarta generación se describen a continuación las características de los lenguajes de primera, segunda y tercera generación.

La primera generación de lenguajes de computación estaba centrada en instrucciones en código de máquina, esto es, que las primeras computadoras fueron programadas en dígitos binarios de los códigos operativos de las máquinas. Este nivel de programación consumía mucho tiempo ya que el código de máquina está íntimamente ligado a la arquitectura particular del hardware y se requería de un conocimiento muy especializado de la máquina.

La segunda generación fue la de los lenguajes ensambladores que proporcionaban un código nemónico para cada instrucción de máquina, facilitando así el desarrollo de programas. Los lenguajes ensambladores seguían ligados a la estructura particular del hardware y al conocimiento de la máquina.

Los lenguajes de tercera generación proporcionan una interface entre el programador y las instrucciones de máquina.

Los lenguajes de cuarta generación no requieren de conocimientos tan profundos y formales como los requeridos por los lenguajes de tercera generación; éstos describen e implementan lo que se hace y no cómo se hace.

Los lenguajes de cuarta generación son usados para mejorar la productividad de los desarrolladores de aplicaciones y aún de los usuarios finales, ya que el uso de estos lenguajes es guiado por medio de menús de opciones y proporcionan ayuda básica.

Se puede decir en términos generales que estos lenguajes tienen como objetivos principales los siguientes:

- * Reducir el ciclo de desarrollo de aplicaciones.
- * Hacer más fácil la modificación de aplicaciones, al reflejar la naturaleza cambiante de los negocios y de los requisitos de los usuarios.
- * Reducir el costo del mantenimiento de las aplicaciones.

Los lenguajes manipuladores de datos de los lenguajes de cuarta generación proporcionan una interfaz no procesal a los usuarios para la recuperación de datos, es decir, que los programas no pueden ser escritos en estos lenguajes manipuladores, sino que permiten al usuario realizar preguntas sobre los datos almacenados y el sistema manejador de bases de datos determina las operaciones necesarias para satisfacer los requisitos.

Los siguientes ejemplos muestran cómo se representa la información en ADABAS, cómo se establecen las relaciones entre los objetos y la recuperación de información que puede realizarse.

CLIENTES

NUM-CLTE	NUM-CLTE	CIUDAD	BCO-COD
12758	ABC	Londres	B643123
12600	DEF	Nueva York	B688436
14811	GHI	Paris	N642841

ARTICULOS

NUM-ART	DES-ART	PRECIO	CANTIDAD
K388	Radio	500	200
R023	T.V.	300	600
R035C	Grabadora	500	375

PEDIDOS

NUM-ORD	NUM-CLTE	NUM-ART	FEC-PEDIDO	FEC-ENTREGA
521567	12600	R023	850922	851002
522321	14811	M166	851115	851221
632863	12600	R035	860222	860228

Las relaciones lógicas entre las tablas pueden establecerse usando cualquier valor. Por ejemplo, todos los pedidos de un cliente pueden ser recuperados seleccionando los pedidos para ese número de cliente:

FIND PEDIDOS WITH NUM-CLTE = '12600'

El campo NUM-ART de los renglones seleccionados puede ser usado como valor de búsqueda del correspondiente renglón de ARTICULO:

FIND ARTICULO WITH NUM-ART = (NUM-ART del registro
seleccionado)

Esta cadena de información puede extenderse fácilmente. Por ejemplo, otra tabla que contenga información detallada del banco del cliente puede ser accesada por BCO-COD.

BANCO

BCO-COD	NOM-BCO	BCO-TEL
---------	---------	---------

B643123	Bco-1	5-40-23
---------	-------	---------

B688476	Bco-2	2-54-20
---------	-------	---------

N642841	Bco-3	5-2032
---------	-------	--------

En NATURAL pueden efectuarse búsquedas complejas en las cuales son usados diferentes criterios de selección:

+ FIND ARTICULO WITH PRECIO > 30 AND CANTIDAD > 10
+ FIND CLIENTE WITH CIUDAD = 'Londres' AND
BCO-COD = 'B600000' THRU 'B699999' BUT NOT
'B650000' THRU 'B659999'

Algunas veces es necesario realizar búsquedas multi-tabla para resolver alguna consulta. Una búsqueda multi-tabla puede realizarse usando múltiples comandos FIND de los cuales el valor recuperado en el primer comando es usado como valor de búsqueda para el segundo comando.

BUSQUEDA1. FIND CLIENTE WITH NOM-CLTE = 'ABC'

BUSQUEDA2. FIND PEDIDOS WITH NUM-CLTE = BUSQUEDA1.NUM-CLTE

(Resulta en la selección de todos los pedidos del cliente 'ABC')

Este proceso no se restringe a dos tablas. Por ejemplo, se puede usar una tercera tabla para localizar los artículos ordenados por el cliente 'ABC':

BUSQUEDA1. FIND CLIENTE WITH NOM-CLTE = 'ABC'

BUSQUEDA2. FIND PEDIDOS WITH NUM-CLTE = BUSQUEDA1.NUM-CLTE

BUSQUEDA3. FIND ARTICULOS WITH NUM-ART = BUSQUEDA2.NUM-ART

(Resulta en la selección de artículos ordenados por 'ABC')

En los siguientes apartados se presentan ejemplos relativos a un caso estudio sobre contabilidad general y aplicaciones sobre un catálogo de grupos financieros y ampliación del presupuesto autorizado, en los que se analizarán consultas y se estudiarán las equivalencias de expresiones en cálculo relacional de estas consultas.

3.2 EL SISTEMA DE CONTABILIDAD GENERAL

El sistema de Contabilidad General es un sistema computarizado que se encarga del registro y aplicación de los movimientos contables, así como de la explotación de la información que se deriva de estos.

Para los fines del presente trabajo, se limita la ejemplificación de las búsquedas en el sistema de Contabilidad General, a las consultas de los movimientos contables registrados, ya que dichas consultas son básicas en la realización de procesos más sofisticados de explotación de los datos.

A continuación se hace una breve descripción de la información almacenada en la base de datos.

AGRUPACION DE CUENTAS CONTABLES: Es un catálogo que contiene información referente a la manera en que están agrupadas las cuentas contables, tales como cuentas de activo, cuentas de pasivo, etc.

CUENTAS CONTABLES: Las cuentas contables definen la estructura contable que el sistema está posibilitado para manejar en el ordenamiento y registro de la información del usuario y contiene información referente a la clave de la cuenta, de la subcuenta y descripción de cada una de ellas.

CENTROS DE COSTOS: El catálogo de centros de costos permite al sistema el control de las cuentas de resultados por Áreas de responsabilidad, esto es, llevar el control de costos, gastos y/o ingresos para las diversas Áreas que integran la empresa. La información que contiene este catálogo se refiere a la clave y descripción del centro de costo.

CONCEPTO ESTANDAR: Este catálogo contiene conceptos de uso repetitivo los cuales se usan para identificar, a nivel de toda la póliza, el tipo de movimientos contables que contienen, así como también incluye conceptos de uso generalizado que identifican partidas específicas en las pólizas para caracterizar el tipo de asiento. La información que contiene este catálogo se refiere a la clave y descripción del concepto estándar.

Las características anteriores fueron las más

importantes de los catálogos que contienen información fundamental para la operación del sistema. Los datos referentes a las pólizas son los datos generales como tipo de póliza, fecha, número y descripción; y el detalle de las partidas de la póliza.

3.3 OPERACIONES DE CONSULTA

En este apartado son presentadas todas las relaciones que constituyen la base de datos para el sistema de Contabilidad General.

Relación 1: CENTRO-COSTOS

CEN-CVE CEN-NCM

Relación 2: CONCEPTO-ESTANDAR

CON-CVE CON-DES

Relación 3: CUENTAS

CTA-CVE CTA-DES CTA-CEN CTA-CON

Relación 4: POLIZAS

POL-TIPO POL-FECHA POL-FOLIO POL-CTA
POL-DEBE POL-HABER

Se ejemplificarán las consultas básicas hechas en el sistema de Contabilidad General, primero cómo se realizarían con las expresiones del cálculo relacional vistas en el apartado 2.4 y en seguida cómo se realizarían usando

NATURAL.

CONSULTA 1

Obtener la descripción de los conceptos estándar mayores a
'concepto-10'

Expresión del Cálculo Relacional

```
CONCEPTO-ESTANDARX.CON-DES DONDE  
CON-CVE > 'concepto-10'
```

Expresión NATURAL

```
FIND CONCEPTO-ESTANDAR WITH CON-CVE > 'concepto-10'  
DISPLAY CON-DES
```

("DISPLAY" es la instrucción en NATURAL que permite desplegar en pantalla la información solicitada)

En la expresión NATURAL de la consulta, la variable de encaje es CONCEPTO-ESTANDAR (CON-CVE, CON-DES) y la fórmula bien formada es CON-CVE > 'concepto-10', lo que nos lleva a la expresión del cálculo relacional previamente vista.

CONSULTA 2

Obtener todas las cuentas cuyo centro de costo sea igual a 'centro-1'.

Expresión del Cálculo Relacional

CUENTASX.CTA-CVE DONDE CUENTASX.CTA-CEN = 'centro-1'

Expresión NATURAL

FIND CUENTAS WITH CTA-CEN = 'centro-1'

Al igual que en la consulta 1, en esta consulta se obtendrán los renglones referidos por la correspondiente fórmula bien formada.

CONSULTA 3

Obtener la descripción del centro de costo de la cuenta cuya clave es 2000 0010

Expresión del Cálculo Relacional

CENTRO-COSTOSX.CEN-DES DONDE EXISTE CUENTAS
(CUENTASY.CTA-CEN = CENTRO-COSTOX.CEN-CVE + CUENTASY.CTA-CVE
= 2000 0010)

Expresión NATURAL

FIND CUENTAS WITH CTA-CVE = 2000 0010

FIND CENTRO-COSTOS WITH CEN-CVE = CTA-CVE

DISPLAY CEN-DES

En este caso se efectúa el acceso a la tabla CUENTAS por el campo CTA-CVE con el valor igual a 2000 0010; de esta búsqueda se obtiene el valor de la clave del centro de costo correspondiente a la cuenta mencionada; con este valor se recupera de la tabla CENTRO-COSTOS la descripción del centro, haciéndose esta segunda búsqueda por medio del campo CEN-CVE.

CONSULTA 4

Obtener la descripción del concepto estándar que es afectado en cada uno de los asientos de las pólizas de ingresos (POL-TIPO = 'I').

Expresión del Cálculo Relacional

CONCEPTO-ESTANDARX.CEN-DES DONDE EXISTE CUENTASY
(EXISTE POLIZASZ (POLIZASZ.POL-CTA = CUENTASY.CTA-CVE Y
POLIZASZ.POL-TIPO = 'I'))

Expresión NATURAL

FIND POLIZAS WITH POL-TIPO = 'I'

FIND CUENTAS WITH CTA-CVE = POL-CTA

FIND CONCEPTO-ESTANDAR WITH CON-CVE = CTA-CON

DISPLAY CON-DES

La consulta 4 es similar a la consulta 3. En la tabla POLIZAS no existe algún campo que sirva para relacionar esta tabla con la de CONCEPTO-ESTANDAR, pero si existe el campo POL-CTA, que puede ser usado para relacionar con la tabla CUENTAS donde se obtiene la clave del correspondiente concepto estándar. Con esta clave se puede efectuar la recuperación de la descripción en el catálogo de CONCEPTO-ESTANDAR.

CONSULTA 5

Obtener todos los asientos de las pólizas de diario (tipo de póliza = 'D') del día 26 de agosto de 1989.

Expresión del Cálculo Relacional

POLIZASX.POL-CTA, POLIZASX.POL-DEBE, POLIZASX.POL-HABER

DONDE POLIZASX.POL-TIPO = 'D' Y POLIZASX.POL-FECHA = 890826

Expresión de NATURAL

```
FIND POLIZAS WITH POL-TIPO = 'D' AND POL-FECHA = 820826  
DISPLAY POL-CTA POL-DEBE POL-HABER
```

Esta consulta consiste en una búsqueda a una sola tabla (POLIZAS), pero con un criterio de selección complejo.

CONSULTA 6

Obtener el tipo y número de póliza de todos lo asientos de pólizas agrupadas bajo el concepto estándar '10'.

Expresión del Cálculo Relacional

```
POLIZASX.POL-TIPO, POLIZASX.POL-NUM DONDE EXISTE  
CUENTASY (CUENTASY.CTA-CEN = '10' Y CUENTASY.CTA-CVE =  
POLIZASX.POL-CTA)
```

Expresión NATURAL.

```
FIND CUENTAS WITH CTA-CEN = '10'  
FIND POLIZAS WITH POL-CTA = CTA-CVE  
DISPLAY POL-TIPO POL-NUM
```


Se relacionan las tablas POLIZAS y CUENTAS por medio del campo CTA-CVE y las tablas CUENTAS y CENTRO-COSTOS por medio del campo CEN-CVE. Además, se realiza una selección adicional en la primera tabla, con el criterio de selección en el que intervienen los campos POL-TIPO y POL-FECHA.

CONSULTA 7

Obtener la descripción de los centros de costo que son afectados por las pólizas de egresos (POL-TIPO = 'E') del mes de abril de 1990.

Expresión del Cálculo Relacional

```
CENTRO-COSTOX.CEN-DES DONDE EXISTE CUENTASY
(CUENTASY.CUENTA-CEN = CENTRO-COSTOX.CEN-CVE Y EXISTE
POLIZASZ. (POLIZASZ.POL-CTA = CUENTASY.CTA-CVE Y
POLIZASZ.POL-TIPO = 'E' Y POLIZASZ.POL-FECHA >= 900401 Y
POLIZASZ.POL-FECHA <= 900430))
```

Expresión NATURAL

```
FIND POLIZAS WITH POL-TIPO = 'E' AND POL-FECHA = 900401
THRU 900430
FIND CUENTAS WITH CTA-CVE = POLIZAS.POL-CTA
FIND CENTRO-COSTOS WITH CEN-CVE = CUENTAS.CTA-CVE
DISPLAY CEN-DES
```

CENTRO-COSTOSX.CEN-DES son las descripciones de los centros de costos que se ven afectados por las cuentas (CUENTASX.CTA-CVE) en las pólizas de egreso (POLIZASZ.POL-TIPO = 'E') del mes de abril de 1990 (900401 <= POLIZASZ.POL-FECHA <= 900430).

CONSULTA 8

Obtener la descripción y la clave de las cuentas contables cuya clave sea mayor que 2000 0010 y cuyo centro de costo sea 300.

Expresión del Cálculo Relacional

CUENTASX.CTA-DES DONDE EXISTE CENTRO-COSTOSY
(CUENTASX.CTA-CEN = CENTRO-COSTOSY.CEN-CVE Y
CUENTASX.CTA-CVE = 2000 0010)

Expresión NATURAL

FIND CUENTAS WITH CTA-CVE > 2000 0010

WHERE CTA-CEN = 300

DISPLAY CTA-CVE CTA-DES

En esta búsqueda se hizo uso de la cláusula 'WHERE' que introduce el criterio adicional (CTA-CEN = 300) después de

que los renglones de la tabla han sido seleccionados por la llave primaria CTA-CVE.

En las consultas anteriores se muestra la característica no procesal del cálculo relacional y de NATURAL. Esta característica, como ya fue mencionado, conduce a una mayor facilidad de recuperación de la información para las personas no avezadas en el desarrollo de aplicaciones, pero que si conocen cual información es manejada en cada tabla y las relaciones que existen entre las tablas.

A continuación se presentarán dos aplicaciones construidas con NATURAL.

3.4 EJEMPLOS DE APLICACIONES CONSTRUIDAS CON NATURAL

Con el objeto de dar una ejemplificación real de la manipulación de datos, se muestran dos aplicaciones NATURAL. Estas aplicaciones se presentarán de la siguiente manera: primeramente se dará una descripción general de la aplicación acompañada de un diagrama jerárquico y un diagrama de entradas y salidas. A continuación se presentará la base de datos definida y se explicará el código de la aplicación. Finalmente se darán los resultados que emite cada aplicación.

ACTUALIZACION Y CONSULTA DEL CATALOGO DE GRUPOS FINANCIEROS.

Esta aplicación actualiza y consulta el catálogo de grupos financieros, ejemplificando consultas a la base de datos con criterios de selección sencillos, que tienen como finalidad verificar la validez de los movimientos que se deseen efectuar (alta, baja, cambio o consulta), dependiendo de los datos proporcionados en la pantalla de captura, o bien recuperar información en el caso de la baja, cambio o consulta.

DIAGRAMA DE ENTRADAS/SALIDAS

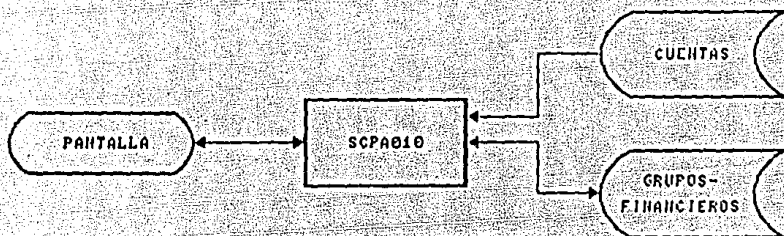
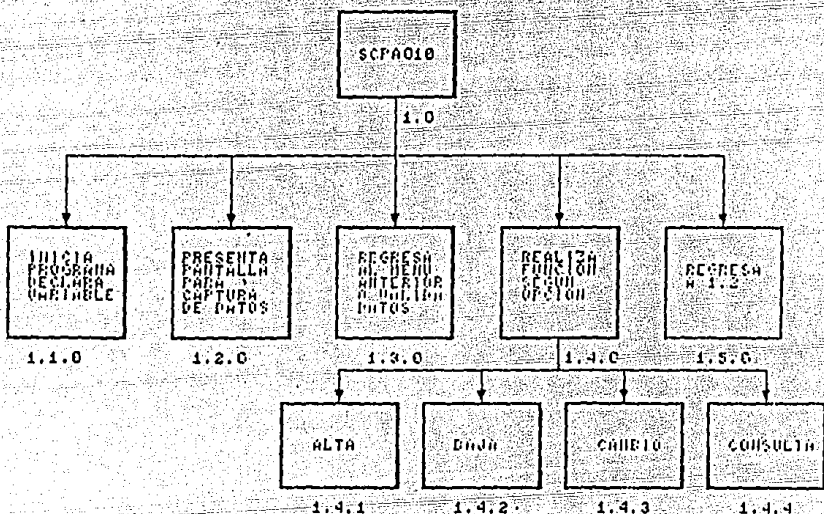


DIAGRAMA JERARQUICO



TYL	DB	NAME	F	LENG	S	D	REMARKS
1	AJ	GPO-FIN-CVE	M	3.0	M	D	CLAVE DEL GRUPO FINA
•		CLAVE DEL GRUPO FINANCIERO AL					
•		QUE ESTA ASIGNADA LA CUENTA					
1	AC	GPO-DES	A	40	M		DESCRIPCION DEL GRUP
•		DESCRIPCION DEL GRUPO					
•		FINANCIERO					
1	BX	GPO-SUB-GPO	M	2.0	M		SUBGRUPO FINANCIERO
•		SUBGRUPO FINANCIERO					
1	BY	GPO-IND-PAR	M	1.0	M		INDICADOR DE PARCIAL
•		INDICADOR DE PARCIALIDAD					
1	BZ	GPO-TIP-OPE	M	1.0	M		TIPO DE OPERACION
•		TIPO DE OPERACION					

DB: 227 File: 211 CUENTA

Default Sequence:

TYL	DB	NAME	F	LENG	S	D	REMARKS
1	AJ	CTA-CVE	M	16.0	M	D	IDENTIFICA A CADA CU
•		IDENTIFICA A CADA CUENTA A					
•		CUALQUIER NIVEL					
1	AO	CTA-DIG-VER	M	1.0	M		DIGITO VERIFICADOR C
•		DIGITO VERIFICADOR CALCULADO					
•		CON EL ALGORITMO X					
1	AP	CTA-DES	A	40	M		DESCRIPCION DE LA CU
•		DESCRIPCION DE LA CUENTA					
1	AS	CTA-NAT	A	1	M		INDICA LA NATURALEZA
•		INDICA LA NATURALEZA DE LA					
•		CUENTA					
1	AQ	CTA-CON	M	1.0	M		INDICA SI LA CUENTA
•		INDICA SI LA CUENTA ES					
•		CONTROLADORA					
1	AR	CTA-IND-PRE	A	1	M		INDICA SI LA CUENTA
•		INDICA SI LA CUENTA TIENE					
•		PRESUPUESTO ASIGNADO					
1	AU	GPO-FIN-CVE	M	3.0	M	D	CLAVE DEL GRUPO FINA
•		CLAVE DEL GRUPO FINANCIERO AL					
•		QUE ESTA ASIGNADA LA CUENTA					
1	CS	GPP-CVE	M	4.0	M	D	CLAVE PARA LA AGRUPA
•		CLAVE PARA LA AGRUPACION					
•		PRESUPUESTAL					
1	AT	CTA-STA	M	2.0	M		INDICA EL ESTADO DE
•		INDICA EL ESTADO DE LA CUENTA					

```

17:02:52          P          SPCAD10 L          CONTARES          97-04-02
0010 .....
0020 ** SISTEMA : SISTEMA DE CONTABILIDAD FROSA **
0030 ** MODULO : MENU DE CATALOGOS **
0040 ** PROGRAMA : SPCAD10 **
0050 ** OBJETIVO : MANTENIMIENTO AL ARCHIVO DE GRUPOS FINANCIEROS **
0060 ** FUNCION : ALTA, BAJA, CAMBIO, CONSULTA AL ARCHIVO DE GRUPOS **
0070 ** FINANCIEROS **
0080 ** PROGRAMA : VERONICA GARCIA MORALS **
0090 .....
0100 DEFINE DATA
0110 GLOBAL USING SCDG005
0120 LOCAL /**** DECLARA VAR'S. LOCALES
0130 1 GRUPO-FINANCIERO(N3)
0140 1 CTRL-DESC(C)
0150 1 CTRL-GPO(C)
0160 1 CTRL-CONF(C)
0170 1 DESCRIPCION(A40)
0180 1 SUB-GPO(N2)
0190 1 PARCIAL-SALDO(N1)
0200 1 TIPO-OPERACION(N1)
0210 1 CONFIRMACION(A1)
0220 1 CONFIRMACION-TEXTO1(A45)
0230 1 MENSAJE(A20)
0240 1 PROGRAM (AA)
0250 ***** DECLARA VISTAS DE USUARIO
0260 1 GPO VIEW OF GRUPO-FINANCIERO
0270 2 GPO-FIN-CVE
0280 2 GPO-DES
0290 2 GPO-SUB-GPO
0300 2 GPO-IND-PAR
0310 2 GPO-TIP-OPE
0320 *
0330 1 CTA VIEW OF CUENTA
0340 2 CTA-CVE
0350 2 GPO-FIN-CVE
0360 *
0370 END-DEFINE
0380

```

El Área global que usa el programa recibe el valor del parámetro FIMQVTO que fue actualizado en el programa anterior a éste y que determina la función a efectuarse (alta, baja, cambio o consulta). El Área local define las variables que se utilizarán en la pantalla de captura y los campos de las vistas de usuario que serán utilizadas en el programa.

```

0390 MOVE *PROGRAM TO RPROGRAM
0400 ..... ACTIVA PF'S
0410 SET KEY PF1 NAMED '---' PF2 NAMED '---'
0420 SET KEY PF3 NAMED '---' PF4 NAMED '---'
0430 SET KEY PF5 NAMED '---' PF6 NAMED '---'
0440 SET KEY PF7 NAMED '---' PF8 NAMED '---'
0450 SET KEY PF9 NAMED '---' PF10 NAMED '---'
0460 SET KEY PF11 NAMED '---' PF12 NAMED '---'
0470 SET KEY PF13 NAMED '---' PF14 NAMED '---'
0480 SET KEY PF15 NAMED '---' PF16 NAMED '---'
0490 SET KEY PF17 NAMED '---' PF18 NAMED '---'
0500 SET KEY PF19 NAMED '---' PF20 NAMED '---'
0510 SET KEY PF21 NAMED '---' PF22 NAMED '---'
0520 SET KEY PF23 NAMED '---' PF24 NAMED '---'
0530 SET KEY PF3 = 'SCPN010' NAMED 'FIN'
0540 PF5 = 'SCPA010' NAMED 'CANC'
0550 *

```

Se activan las teclas de función y se programan la tecla PF3 para regresar al menú anterior y la tecla PF5 para cancelar la función que se este realizando.

```

0560 REPEAT
0570 ..... LIMPIA VARIABLES
0580 RESET
0590 RGRUPO-FINANCIERO
0600 RSUB-GPO
0610 RPARCIAL-SALDO
0620 RTIPO-OPERACION
0630 RCTRL-DESC
0640 RCTRL-GPO
0650 RCTRL-CONF
0660 RCONFIRMACION
0670 RCONFIRMACION-TEXTO1
0680 RDESCRIPCION
0690 *
0700 MOVE (AD=I) TO RCTRL-GPO
0710 IF RMOVTO = 1 THEN MOVE (AD=I) TO RCTRL-DESC
0720 ELSE MOVE (AD=P) TO RCTRL-DESC
0730 END-IF
0740 --MOVE (AD=PN) TO RCTRL-CONF
0750 ..... PRESENTA PANTALLA PARA CAPTURA DE DATOS
0760 INPUT USING MAP 'SCMA010'
0770 PERFORM SCUS008 /** VALIDA PF'S ACTIVAS

```

Este bloque de instrucciones inicia un ciclo en el que se limpian las variables utilizadas en la pantalla de captura antes de presentar esta. Se convierte a la variable RGRUPO-FINANCIERO en una variable de entrada y si RMOVTO es igual a 1 (alta), también las variables RDESCRIPCION,

NSUB-GPO, RPARCIAL-SALDO y NTIPI-OPERACION se utilizarán como variables de entrada. En esta primera pantalla no se desplegará el texto de confirmación y la variable para confirmar la transacción (NCONFIRMACION) será una variable de salida.

```

0780 ***** VALIDA DATOS
0790 IF RGRUPO-FINANCIERO = 0
0800 REINPUT'PROPORCIONE GRUPO FINANCIERO' MARK*RGRUPO-FINANCIERO
0810 END-IF
0820 *
0830 FIND NUMBER GPO WITH GPO-FIN-CVE = RGRUPO-FINANCIERO
0840 IF *NUMBER(0830) = 0
0850 THEN IF RMOVTO NE 1
0860 REINPUT'GRUPO FINANCIERO INEXISTENTE'
0870 MARK*RGRUPO-FINANCIERO
0880 END-IF
0890 ELSE IF RMOVTO EQ 1
0900 REINPUT'GRUPO FINANCIERO YA EXISTENTE'
0910 MARK*RGRUPO-FINANCIERO
0920 END-IF
0930 *
0940 ***** REALIZA FUNCION SEGUN RMOVTO
0950 DECIDE ON FIRST VALUE OF RMOVTO
0960 VALUE 1 PERFORM ALTA
0970 VALUE 2 PERFORM BAJA
0980 VALUE 3 PERFORM CAMBIO
0990 VALUE 4 PERFORM CONSULTA
1000 NONE VALUE IGNORE
1010 END-DECIDE
1020 *
1030 END-REPEAT

```

Para la validación de datos se obtiene el número de registros de la vista GPO con llave GPO-FIN-CVE igual al grupo financiero capturado previamente.

El presente bloque realiza alguna de las subrutinas ALTA, BAJA, CAMBIO o CONSULTA según el valor de la variable RMOVTO y se cierra el ciclo en el que se estará iterando hasta que se decida regresar al menú anterior.

```

1040 *
1050 DEFINE SUBROUTINE ALTA
1060 PERFORM VALIDA
1070 MOVE 'ALTA EFECTUADA' TO #MENSAJE
1080 MOVE #GRUPO-FINANCIERO TO #GPO-FIN-CVE
1090 MOVE #DESCRIPCION TO #GPO-GPO-DES
1100 MOVE #SUB-GPO TO #GPO-GPO-SUB-GPO
1110 MOVE #PARCIAL-SALDO TO #GPO-GPO-IND-PAR
1120 MOVE #TIPO-OPERACION TO #GPO-GPO-TIP-OPE
1130 END OF TRANSACTION
1140 END-SUBROUTINE
1150 *

```

La subrutina ALTA realiza una validación de las variables #DESCRIPCION, #SUB-GPO, #PARCIAL-SALDO y #TIPO-OPERACION y actualiza la variable #MENSAJE que se despliega en la primera pantalla de captura, asimismo se mueve los valores de la pantalla de captura a los campos correspondientes de la vista GPO, se indica la adición del registro y se aplica la operación a la base de datos.

```

1160 DEFINE SUBROUTINE BAJA
1170 RESET #MENSAJE
1180 MOVE (AD=P) TO #CTRL-GPO
1190 #CTRL-DESC
1200 FIND NUMBER CTA WITH GPO-FIN-CVE = #GRUPO-FINANCIERO
1210 IF *NUMBER(1200) NE 0
1220 REINPUT #GRUPO FINANCIERO EXISTE EN EL ARCHIVO DE CUENTAS CONTABLES
1230 MARK*#GRIPO-FINANCIERO
1240 END-IF
1250 *
1260 FIND GPO WITH GPO-FIN-CVE = #GRUPO-FINANCIERO
1270 MOVE GPO-DES TO #DESCRIPCION
1280 MOVE GPO-SUB-GPO TO #SUB-GPO
1290 MOVE GPO-IND-PAR TO #PARCIAL-SALDO
1300 MOVE GPO-TIP-OPE TO #TIPO-OPERACION
1310 MOVE (AD=1) TO #CTRL-CONF
1320 MOVE 'REALIZA LA BAJA S/N ?' TO #CONFIRMACION-TEXTO1
1330 INPUT USING MAP 'SCMA010'
1340 PERFORM SCUS008
1350 IF *PF-KEY = 'PF5' ESCAPE TOP END-IF
1360 IF #CONFIRMACION = 'S' OR = 'N' THEN IGNORE
1370 ELSE REINPUT 'LA OPCION QUE USTED TECLEO ES INCORRECTA'
1380 MARK*#CONFIRMACION
1390 END-IF
1400 IF #CONFIRMACION = 'N' THEN MOVE 'BAJA CANCELADA' TO #MENSAJE
1410 ELSE
1420 MOVE 'BAJA EFECTUADA' TO #MENSAJE
1430 DELETE
1440 END OF TRANSACTION
1450 END-IF
1460 END-FIND
1470 END-SUBROUTINE
1480 *

```

La subrutina BAJA convierte en variables de salida a las variables ADESCRIPCION, ASUB-GPO, APARCIAL-SALDO y ATIPO-OPERACION y limpia la variable mensaje. Para validar si se puede dar de baja el registro, verifica que no exista ningún registro en la vista CTA con ese grupo financiero, de no existir dicho registro, se recupera la información de la vista GPO y se despliega la pantalla junto con el mensaje de confirmación para efectuar o no la baja. Se valida la variable NCONFIRMACION y si no se confirmo la baja se despliega en la siguiente pantalla el mensaje 'BAJA CANCELADA'. En caso contrario se indica la baja del registro y se aplica esta operación a la base dedatos.

```

1500 DEFINE SUBROUTINE CAMBIO
1510 RESET AMENSAJE
1520 MOVE (AG=9) TO ACTRL-GPO
1530 MOVE (AD=1) TO ACTRL-DESC
1540 FIND GPO WITH GPO-FIN-CVE * RGRUPO-FINANCIERO
1550 MOVE GPO-DES TO ADESCRIPCION
1560 MOVE GPO-SUB-GPO TO ASUB-GPO
1570 MOVE GPO-IND-PAR TO APARCIAL-SALDO
1580 MOVE GPO-TIP-OPE TO ATIPO-OPERACION
1590 INPUT USING MAP 'SCMA010' NO ERASE
1600 PERFORM SCUS00B
1610 IF *PF-KEY = 'PFG' ESCAPE TOP END-IF
1620 PERFORM VALIDA
1630 *
1640 MOVE 'CAMBIO EFECTUADO' TO AMENSAJE
1650 MOVE ADESCRIPCION TO GPO-DES
1660 MOVE ASUB-GPO TO GPO-SUB-GPO
1670 MOVE APARCIAL-SALDO TO GPO-IND-PAR
1680 MOVE ATIPO-OPERACION TO GPO-TIP-OPE
1690 UPDATE
1700 END OF TRANSACTION
1710 END-FIND
1720 END-SUBROUTINE
1730 *

```

La subrutina CAMBIO convierte en variable de salida la variable NGRUPO-FINANCIERO y en variables de entrada/salida las variables NDESCRIPCION, NSUB-GPO, NPARCIAL-SALDO y NTIPO-OPERACION, a continuación recupera el registro con llave GPO-FIN-CVE igual a NGRUPO-FINANCIERO y mueve las variables de la vista GPO a las variables correspondientes, despues despliega la pantalla con la información obtenida para que esta sea modificada. Se validan las teclas de función activas y la información modificada, si esta validación no reportó ningún error se actualiza la variable NMENSAJE, se actualizan los campos con la información modificada y se indica la modificación del registro para posteriormente aplicar la operación a la base dedatos.

```

1740 DEFINE SUBROUTINE CONSULTA
1750 RESET NMENSAJE
1760 MOVE (AD*P) TO NCTRL-GPO
1770 NCTRL-DESC
1780 MOVE 'YA TERMINO DE CONSULTAR * OPRIMA RETURN *'
1790 TO NCON-FIRMACION-TEXTO1
1800 FIND GPO WITH GPO-FIN-CVE = NGRUPO-FINANCIERO
1810 MOVE GPO-DES TO NDESCRIPCION
1820 MOVE GPO-SUB-GPO TO NSUB-GPO
1830 MOVE GPO-IND-PAR TO NPARCIAL-SALDO
1840 MOVE GPO-TIP-OPE TO NTIPO-OPERACION
1850 END-FIND
1860 INPUT USING MAP 'SCHAD10'
1870 PERFORM SCUS008
1880 MOVE 'CONSULTA EFECTUADA' TO NMENSAJE
1890 END-SUBROUTINE
1900

```

En la subrutina de consulta, todas las variables se convierten en variables de salida y se despliega la información recuperada de la vista GPO con el campo GPO-FIN-CVE igual a NGRUPO-FINANCIERO.

```

1810 DEFINE SUBROUTINE VALIDA
1820
1830 IF #DESCRIPCION = '
1840 REINPUT 'EL CAMPO QUE USTED TECLEO NO DEBE SER IGUAL A ESPACIOS'
1850 MARK#DESCRIPCION
1860 END-IF
1870
1880 IF #SUB-GPO = 0
1890 REINPUT 'EL SUBGRUPO DEBE SER MAYOR QUE CERO' MARK#SUB-GPO
2000 END-IF
2010
2020 IF #PARCIAL-SALDO NE 1 AND #PARCIAL-SALDO NE 2
2030 REINPUT 'DEBERA SER (1) PARCIAL (2) SALDO' MARK#PARCIAL-SALDO
2040 END-IF
2050
2060 IF #TIPO-OPERACION NE 1 AND #TIPO-OPERACION NE 2
2070 REINPUT 'DEBERA SER (1) SUMA (2) RESTA' MARK#TIPO-OPERACION
2080 END-IF
2090
2100 END-SUBROUTINE
2110 END
***** E
NEXT FIN

```

18.50.57 Mac SCHA010 Lib1 CONTARES 91-04-23

```

0010 * MAP2; PROTOTYPE
0020 * INPUT USING M-D 'XXXXXXXX'
0030 * #CONFIRMACION #CONFIRMACION-TEXT01 #CTRL-CONF #CTRL-DESC
0040 * #CTRL-GPO #DESCRIPCION #FUNCION #GRUPO-FINANCIERO #MENSAJE
0050 * #PARCIAL-SALDO #PROGRAM #SUB-GPO #TIPO-OPERACION
0060 DEFINE DATA PARAMETER
0070 | #CONFIRMACION (A01)
0080 | #CONFIRMACION-TEXT01 (A045)
0090 | #CTRL-CONF (C)
0100 | #CTRL-DESC (C)
0110 | #CTRL-GPO (C)
0120 | #DESCRIPCION (A040)
0130 | #FUNCION (A008)
0140 | #GRUPO-FINANCIERO (N03.0)
0150 | #MENSAJE (A020)
0160 | #PARCIAL-SALDO (N01.0)
0170 | #PROGRAM A008)
0180 | #SUB-GPO (N02.0)
0190 | #TIPO-OPERACION (N01.0)
0200 END-DEFINE
0210 FORMAT PS=021 LS=080 ZP=OFF SG=OFF KD=ON
0220 * MAP2; MAP PROFILES ..... 200.....
0230 * ,TTAAAMMOOD D I D I N D I D I Y ?)-81-(
0240 * 023079 NO NUCL X 01 SYSPROF VL 1
0250 .....
0260 INPUT { IP=OFF /
0270 /
0280 002T *DATE (AD=01) /*.02S008 A008
0290 023T *PROMOCION Y OPERACION. S.A. DE C.V.'(1)
0300 070T *TIMX (AD=01) /*.01S008 T
0310 /
0320 002T *SCHA010 SISTEMA DE CONTABILIDAD'(1)
0330 070T #PROGRAM (AD=01) /*.01U008 A008
0340 /
0350 028T 'ACTUALIZACION DE CATALOGOS'(1)
0360 /
0370 031T 'GRUPOS FINANCIEROS'(1)
0380 /
0390 005T #MENSAJE (AD=01) /*.01U020 A020
0400 036T #FUNCION (AD=01) /*.01U008 A008
0410 /
0420 /
0430 /

```

```

0440 004T '1. GRUPO FINANCIERO      >'
0450 029T #GRUPO-FINANCIERO (AD=MYT'_' CV=#CTRL-GPO ) /*.99U003.N03.0.
0460 /
0470 /
0480 004T '2. DESCRIPCION          >'
0490 029T #DESCRIPCION (AD=MYT'_' CV=#CTRL-DESC ) /*.99J040.A040.
0500 /
0510 /
0520 004T '3. SUB-GRUPO            >'
0530 029T #SUB-GPO (AD=MYLT'_' CV=#CTRL-DESC ) /*.99G002.N02.0.
0540 /
0550 /
0560 004T '4. PARCIAL-SALDO        >'
0570 029T #PARCIAL-SALDO (AD=MYLT'_' CV=#CTRL-DESC ) /*.99U001.N01.0.
0580 /
0590 /
0600 004T '5. TIPO DE OPERACION     >'
0610 029T #TIPO-OPERACION (AD=MYLT'_' CV=#CTRL-DESC ) /*.99U001.N01.0.
0620 /
0630 /
0640 /
0650 /
0660 /
0670 002T #CONFIRMACION-TEXTO1 (AD=01'_) /*.01U045.A045.
0680 079T #CONFIRMACION (AD=MYT'_' CV=#CTRL-CONF ) /*.99U001.A001.
0690 * MAP2: VALIDATION *****
0700 * MAP2: END OF MAP *****
0710 END
***** End of List *****
NEXT FIN

```

```

18:51:37 Subroutine : SCUS008 Lib: CONTARES 91-04-23
0010 *****
0020 ** SISTEMA : SISTEMA DE CONTABILIDAD PROSA **
0030 ** MODULO : CONTROL DEL SISTEMA **
0040 ** PROGRAMA : SCUS008 **
0050 ** OBJETIVO : VALIDACION DE PF'S **
0060 ** FUNCION : VALIDACION DE PF'S **
0070 ** **
0080 ** PROGRAMA : VERONICA GARCIA MORALES **
0090 *****
0100 *
0110 DEFINE SUBROUTINE SCUS008
0120 *
0130 IF *PF-KEY GE 'PF1'
0140 REINPUT *
0150 END-IF
0160 *
0170 END-SUBROUTINE
0180 *
0190 END
***** End of List *****
NEXT FIN

```

02/04/91
SCMH010

PROMOCION Y OPERACION, S.A. DE C.V.
SISTEMA DE CONTABILIDAD
ACTUALIZACION DE CATALOGOS
GRUPOS FINANCIEROS
MENU DEL PROGRAMA

13:07:16
SCPM010

1. ALTAS
2. BAJAS
3. CAMBIOS
4. CONSULTA

QUE OPCION DESEA USTED ?

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12--
-- -- FIN -- -- -- -- -- -- -- -- -- -- -- --

02/04/91
SCMA010

PROMOCION Y OPERACION, S.A. DE C.V.
SISTEMA DE CONTABILIDAD
ACTUALIZACION DE CATALOGOS
GRUPOS FINANCIEROS
ALTA

13:24:25
SCPA010

ALTA EFECTUADA

1. GRUPO FINANCIERO > 123
2. DESCRIPCION > DESCRIPCION DE PRUEBA _____
3. SUB-GRUPO > 1_
4. PARCIAL-SALDO > 1
5. TIPO DE OPERACION > 1

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12--
-- -- FIN -- -- CANC -- -- -- -- -- -- -- -- -- -- -- --

02/04/91
SCMA010

PROMOCION Y OPERACION, S.A. DE C.V.
SISTEMA DE CONTABILIDAD
ACTUALIZACION DE CATALOGOS
GRUPOS FINANCIEROS
BAJA

13:22:26
SCPA010

- 1. GRUPO FINANCIERO > 123
- 2. DESCRIPCION > DESCRIPCION DE PRUEBA _____
- 3. SUB-GRUPO > 1_
- 4. PARCIAL-SALDO > 2
- 5. TIPO DE OPERACION > 1

REALIZA LA BAJA S/N ?

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PFB---PF9---PF10---PF11---PF12--
-- -- FIN -- CANC -- -- -- -- -- -- --

02/04/91
SCMA010

PROMOCION Y OPERACION, S.A. DE C.V.
SISTEMA DE CONTABILIDAD
ACTUALIZACION DE CATALOGOS
GRUPOS FINANCIEROS
CONSULTA

13:16:40
SCPA010

- 1. GRUPO FINANCIERO > 900
- 2. DESCRIPCION > INETERESES COBRADOS _____
- 3. SUB-GRUPO > 66
- 4. PARCIAL-SALDO > 1
- 5. TIPO DE OPERACION > 1

YA TERMINO DE CONSULTAR * OPRIMA RETURN *

Enter-PF1---PF2---PF3---PF4---PF5---PF6---PF7---PF8---PF9---PF10---PF11---PF12--
-- -- FIN -- CANC -- -- -- -- -- -- --

AMPLIACION DEL PRESUPUESTO.

Esta aplicación actualiza el archivo PRESUPUESTO, ampliando el presupuesto autorizado para una cuenta determinada y en base a un porcentaje de aumento específico.

En este caso para la recuperación de información del archivo PRESUPUESTO se determina previamente a la consulta a la base de datos, un rango de valores para la llave primaria, ya que está formada por los campos PRE-ANO, CTA-CVE y CEN-CVE y se tienen valores para PRE-ANO y CTA-CEN, pero el campo CEN-CVE puede variar de 0 a 99999999. Para obtener la descripción del centro de costo afectado por la ampliación del presupuesto se relacionan los archivos PRESUPUESTO y CENTRO-COSTO por medio del valor del campo PRESUPUESTO.CEN-CVE.

DIAGRAMA DE ENTRADAS/SALIDAS

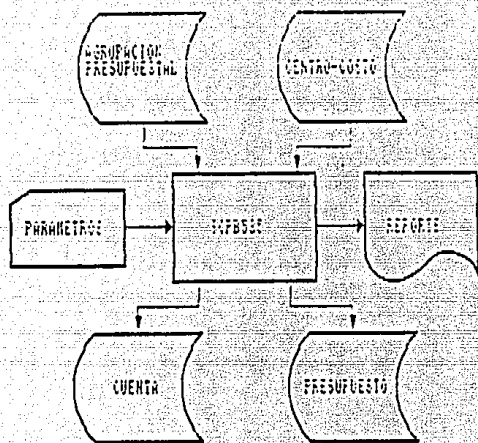
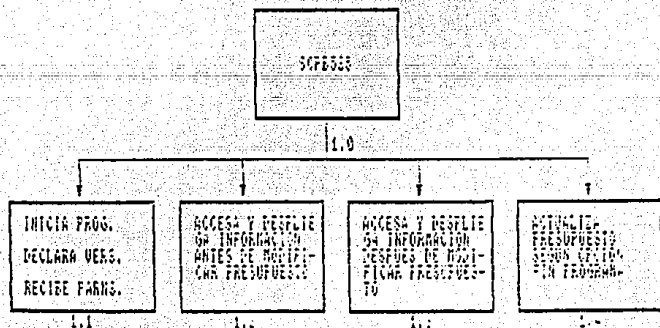


DIAGRAMA JERARQUICO



TYL	DB	NAME	F	LENG	S	D	REMARKS
1	A0	PRE-ANO	N	2.0	N	D	ANO AL QUE PERTENECE
*		ANO AL QUE PERTENECE EL PRESU-					
*		PUESTO					
1	A0	CEN-CVE	N	8.0	N		CLAVE QUE IDENTIFICA
*		CLAVE QUE IDENTIFICA A CADA					
*		CENTRO DE COSTO					
1	AJ	CTA-CVE	N	16.0	N		IDENTIFICA A CADA CU
*		IDENTIFICA A CADA CUENTA A					
*		QUALQUIER NIVEL					
1	CS	GPP-CVE	N	4.0	N	D	CLAVE PARA LA AGRUPA
*		CLAVE PARA LA AGRUPACION PRE-					
*		SUPUESTAL					
1	CZ	PRE-FEC	N	6.0	N		FECHA DE ACTUALIZACI
*		FECHA DE ACTUALIZACION					
1	C0	PRE-GEN	A	2	N		INDICADOR DE ACTUALI
*		INDICADOR DE ACTUALIZACION					
P 1	A1	PRE-MES					MES AL QUE PERTENCE
*		MES AL QUE PERTENCE EL PRESU-					
*		PUESTO					
2	C1	PRE-OR1	P	15.0	N		PRESUPUESTO ORIGINAL
*		PRESUPUESTO ORIGINAL					
2	C2	PRE-AUT	P	15.0	N		PRESUPUESTO AUTORIZA
*		PRESUPUESTO AUTORIZADO					
2	A3	PRE-EJE	P	15.0	N		PRESUPUESTO EJERCIDO
*		PRESUPUESTO EJERCIDO					
2	A2	PRE-COM	P	15.0	N		PRESUPUESTO COMPROME
*		PRESUPUESTO COMPROMETIDO					
2	A4	PRE-EJE-ANT	P	15.0	N		PRESUPUESTO EJERCIDO
*		PRESUPUESTO EJERCIDO PERIODO					
*		ANTERIOR					
1	CX	PRE-ANO-CEN-CTA	B	26	N	S	SUPERDESCRIPTOR FORM
*		SUPERDESCRIPTOR FORMADO POR					
*		PRE-ANO					
*		CEN-CVE					
*		CA7-CVE					
1	CY	PRE-ANO-C7A-CEN	B	26	N	S	SUPERDESCRIPTOR FORM
*		SUPERDESCRIPTOR FORMADO POR					
*		PRE-ANO					
*		CTA-CVE					
*		CEN-CVE					
1	C3	PRE-ANO-CEN-GPO	B	14	N	S	SUPERDESCRIPTOR FORM
*		SUPERDESCRIPTOR FORMADO POR					
*		PRE-ANO					
*		CEN-CVE					
*		GPP-CVE					

DB: 227 Files: 210 - CENTRO-COSTO

Default Sequence:

TYL	DB	NAME	F	LENG	S	D	REMARKS
1	AD	CEN-CVE	M	8.0	N	D	CLAVE QUE IDENTIFICA
		CLAVE QUE IDENTIFICA A CADA					
		CENTRO DE COSTO					
1	AE	CEN-NOM	A	40	N		NOMBRE DEL CENTRO DE
		NOMBRE DEL CENTRO DE COSTO					
1	AF	CEN-FEC-INI-VIG	M	6.0	N		FECHA DE INICIO DE O
		FECHA DE INICIO DE OPERACIONES					
		DEL CENTRO DE COSTO					
1	AG	CEN-FEC-FIN-VIG	M	6.0	N		FECHA DE TERMINACION
		FECHA DE TERMINACION DE					
		OPERACIONES DEL CENTRO DE					
		COSTOS					

DB: 227 Files: 210 - AGRUPACION-PRESUPUESTAL

Default Sequence:

TYL	DB	NAME	F	LENG	S	D	REMARKS
1	CS	GPP-CVE	M	4.0	N	D	CLAVE PARA LA AGRUPA
		CLAVE PARA LA AGRUPACION PRE-					
		SUPUESTAL					
1	CT	GPP-DES	A	40	N		DESCRIPCION PARA NOM
		DESCRIPCION PARA NOMBRAR A UN					
		GRUPO PRESUPUESTAL					

*****DOM OUTPUT TERMINATED*****

DB: 227 Files: 211 - CUENTA

Default Sequence:

TYL	DB	NAME	F	LENG	S	D	REMARKS
1	AJ	CTA-CVE	M	16.0	N	D	IDENTIFICA A CADA CU
		IDENTIFICA A CADA CUENTA A					
		CUALQUIER NIVEL					
1	AO	CTA-DIG-VER	M	1.0	N		DIGITO VERIFICADOR C
		DIGITO VERIFICADOR CALCULADO					
		CON EL ALGORITMO X					
1	AP	CTA-DES	A	40	N		DESCRIPCION DE LA CU
		DESCRIPCION DE LA CUENTA					
1	AS	CTA-NAT	A	1	N		INDICA LA NATURALEZA
		INDICA LA NATURALEZA DE LA					
		CUENTA					
1	AQ	CTA-COM	M	1.0	N		INDICA SI LA CUENTA
		INDICA SI LA CUENTA ES					
		CONTROLADORA					
1	AR	CTA-IND-PRE	A	1	N		INDICA SI LA CUENTA
		INDICA SI LA CUENTA TIENE					
		PRESUPUESTO ASIGNADO					
1	AU	GPO-FIN-CVE	M	3.0	N	D	CLAVE DEL GRUPO FINA
		CLAVE DEL GRUPO FINANCIERO AL					
		QUE ESTA ASIGNADA LA CUENTA					
1	CS	GPP-CVE	M	4.0	N	D	CLAVE PARA LA AGRUPA
		CLAVE PARA LA AGRUPACION					
		PRESUPUESTAL					
1	AT	CTA-STA	M	2.0	N		INDICA EL ESTADO DE
		INDICA EL ESTADO DE LA CUENTA					

```

0010 * .....
0020 * SISTEMA : CONTABILIDAD PROSA.
0030 * PROGRAMA : SCPB525
0040 * OBJETIVO : AMPLIACION DE PRESUPUESTO POR AÑO-CUENTA Y PORCENTAJE
0050 * FUNCION : PERMITIR AL USUARIO POR MEDIO DE UNA CORRIDA BATCH LA
0060 * AMPLIACION DEL PRESUPUESTO POR EL AÑO-CUENTA Y PORCENTAJE
0070 * EN UN RANGO DE MESES.
0080 * TIP.PROC : BATCH
0090 * PROGRAMA : VERONICA GARCIA MORALES.
0100 * FECHA : 09 DE MAYO DE 1989.
0110 * RESPON. : JUAN CARLOS RAMOS ESPINOSA
0120 * .....
0130 **
0140 DEFINE DATA LOCAL
0150 1 AÑO-E (N02)
0160 1 A CUENTA-E (N16)
0170 1 REDEFINE A CUENTA-E
0180 2 ACUE-E (N04)
0190 2 ASCU-E (N04)
0200 2 ASSC-E (N04)
0210 2 ASSS-E (N04)
0220 1 AMES-INI-E (N02)
0230 1 AMES-FIN-E (N02)
0240 1 APOR-CIENTO (N2.2)
0250 1 APROCESO (A01)
0260 *
0270 1 A LLAVE-E (A26)
0280 1 REDEFINE A LLAVE-E
0290 2 AÑO-EL (N02)
0300 2 A CUENTA-EL (N16)
0310 2 A CENTRO-EL (N08)
0320 1 A LLAVE-R (A26)
0330 1 REDEFINE A LLAVE-R
0340 2 AÑO-RL (N02)
0350 2 A CUENTA-RL (N16)
0360 2 A CENTRO-RL (N08)
0370 1 A TOTAL (N15)
0380 1 A DES-GPP-E (A30)
0390 1 A DES-CEN-E (A30)
0400 1 A DES-CTA-E (A30)
0410 1 A DES-CTA-K (A30)
0420 1 A PERC (N2.5)
0430 1 A TOT-DEC (N3.5)
0440 1 A TOT-ANT (N18)
0450 1 A TOT-DES (N18)
0460 1 A TOT-DIF (N18)
0470 1 A MONT-PERC (N15.5)
0480 1 REDEFINE A MONT-PERC
0490 2 A ENT (N15)
0500 2 A DEC (N0.5)
0510 1 A CONSEC (N06)
0520 1 A GPO-PTAL (N04)
0530 1 A I (N02)
0540 1 A CONTADOR (N04)
0550 1 A CONTADOR I (N04)
0560 1 A NENCI (A10)

```

```

17:02:28 P SPC8525-1 CONTIARLS 91-04-02
0570 1 RENC2 (A20)
0580 1 RDATN (N08)
0590 1 REDEFINE RDATN
0600 2 RFill (N02)
0610 2 RFECHA-DIA (N06)
0620 1 RCUENTA-K (N16)
0630 1 REDEFINE RCUENTA-K
0640 2 RCUE-K (N04)
0650 2 RSSS-K (N12)
0660 *
0670 1 PRE VIEW OF PRESUPUESTO
0680 2 PRE-ANO
0690 2 CEN-CVE
0700 2 CTA-CVE
0710 2 GPP-CVE
0720 2 PRE-AUT (P15/13)
0730 *
0740 1 GPP VIEW OF AGRUPACION-PRESUPUESTAL
0750 2 GPP-CVE
0760 2 GPP-DES
0770 *
0780 1 CEN VIEW OF CENTRO-COSTO
0790 2 CEN-CVE
0800 2 CEN-NOM
0810 *
0820 1 CTA VIEW OF CUENTA
0830 2 CTA-CVE
0840 2 CTA-DES
0850 2 GPP-CVE
0860 END-DEFINE

```

En el primer bloque se definen las variables que se ocuparán en el programa y las vistas de usuario.

```

0870 *
0880 ..... ENTRADA DE PARAMETRO
0890 *
0900 INPUT RANO-E RCUENTA-E RMES-INI-E RMES-FIN-E RPOR-CIENTO RPROCESO
0910 MOVE RANO-E TO RANO-EL RANO-RL
0920 MOVE RCUENTA-E TO RCUENTA-EL RCUENTA-RL
0930 MOVE 0000000 TO RCENTRO-EL
0940 MOVE 99999999 TO RCENTRO-RL
0950 COMPUTE RPERC = 1 + (RPOR-CIENTO / 100)
0960 *

```

Se reciben los parámetros RANO-E, RCUENTA-E, RMES-INI-E, RMES-FIN-E, RPOR-CIENTO y RPROCESO y se determina el rango de valores que puede tener la llave PRE-ANO-CTA-CEN (RLLAVE-E, RLLAVE-R) que es una llave compuesta por los campos PRE-ANO, CTA-CVE y CEN-CVE, además se calcula el factor de aumento del presupuesto.

```

0970 ..... DEFINICION DE FORMATO
0980 FORMAT PS=35 LS=132
0990 MOVE *DATN TO RDATN
1000 *
1010 IF APROCESO = 'P' THEN
1020 MOVE 'PREVIO' TO RENC1
1030 ELSE
1040 MOVE 'FIRME' TO RENC1
1050 END-IF
1060 RESET RDES-CEN-E AMONT-PERC RTOT-DEC RI ACUENTA-K
1070 MOVE ACUE-E TO ACUE-K
1080 *

```

Se determina el tamaño de página y línea y se actualiza la variable RENC1 que forma parte del encabezado del reporte.

```

1100 ..... OBTIENE INFORMACION DE CUENTA
1110 FIND(1) CTA WITH CTA-CVE = ACUENTA-E
1120 MOVE CTA-CVE TO ACUENTA-K
1130 MOVE CTA-DES TO RDES-CTA-E
1140 MOVE GPP-CVE TO RGPO-PTAL
1150 END-FIND
1160 *
1170 RESET R555-K
1180 *
1190 FIND(1) CTA WITH CTA-CVE = ACUENTA-K
1200 MOVE CTA-DES TO RDES-CTA-K
1210 END-FIND
1220 *
1230 FIND(1) GPP WITH GPP-CVE = RGPO-PTAL
1240 MOVE GPP-DES TO RDES-GPP-E
1250 END-FIND
1260 *
1270 MOVE 'ANTES DEL PROCESO' TO RENC2

```

En este bloque se obtiene la descripción de la cuenta y su grupo presupuestal, así como en una segunda recuperación de la vista CTA se obtiene la descripción del primer nivel de la cuenta. También se obtiene la descripción del grupo presupuestal.

```

1280 *
1290 ..... DEFINICION DE ENCABEZADO
1300 AT TOP OF PAGE
1310 WRITE NOTITLE
1320 1T *PROGRAM
1330 48T *PROMOCION Y OPERACION, S. A. DE C. V.'
1340 127T *PAGE-NUMBER (EM=ZZZ9)
1350 / 42T *DIRECCION EJECUTIVA DE ADMINISTRACION Y FINANZAS'
1360 123T *DATE
1370 // 81T *SUBDIRECCION DE CONTABILIDAD Y PRESUPUESTO'
1380 123T *TIMX
1390 / 41T *AMPLIACION AUTOMATICA DEL PRESUPUESTO POR CUENTA Y X'
1400 / 49T *PROCESO EN: RENC1 '(MILLONES DE PESOS)'
1410 // 01T *CUENTA: R' ACUE-E (EM=9999) ASCU-E (EM=9999)
1420 R' SSC-E (EM=9999) R' SSS-E (EM=9999) RDES-CTA-E
1430 76T *GRUPO PRESUPUESTAL: R' NGPO-PTAL RDES-GPP-E
1440 / 29T RDES-CTA-K
1450 / 1T *MES INICIAL: R' MES-INI-E *MES FINAL: R' MES-FIN-E
1460 76T *PORCENTAJE: R' RPOR-CIENTO
1470 / 1T '-' (131)
1480 // 3T *CENTRO DESCRIPCION
1490 / 18T *ENERO/JULIO FEBRERO/AGOSTO MARZO/SEPTIEMBRE'
1500 ABRIL/OCTUBRE MAYO/NOVIEMBRE JUNIO/DICIEMBRE'
1510 / 1T '-' (131)
1520 / 56T RENC2
1530 END-TOPPAGE

```

Se define el encabezado del programa.

```

1540 .....
1550 ..... ANTES DE ACTUALIZACION
1560 FIND PRE WITH PRE-ANO-CTA-CEN = RLLAVE-E THRU RLLAVE-R
1570 *
1580 FIND CEN WITH CEN-CVE = CEN-CVE(1560)
1590 MOVE CEN-NOM TO RDES-CEN-E
1600 END-FIND
1610 *
1620 COMPUTE PRE-AUT(*) = PRE-AUT(*) / 1000000
1630 *
1640 WRITE 1T CEN-CVE RDES-CEN-E
1650 / 8T PRE-AUT(01) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1660 28T PRE-AUT(02) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1670 48T PRE-AUT(03) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1680 68T PRE-AUT(04) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1690 88T PRE-AUT(05) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1700 108T PRE-AUT(06) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1710 / 8T PRE-AUT(07) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1720 28T PRE-AUT(08) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1730 48T PRE-AUT(09) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1740 68T PRE-AUT(10) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1750 88T PRE-AUT(11) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1760 108T PRE-AUT(12) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1770 / 20T 'TOTAL:'
1780 28T PRE-AUT(13) (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZ9)
1790 *
1800 ADD PRE-AUT(13) TO RTOT-ANT
1810 ADD 1 TO RCONTADOR
1820 END-FIND
1830 *
1840 WRITE / 20T 'TOTAL POR CUENTA : '
1850 44T RTOT-ANT (EM=ZZZ,ZZZ,ZZZ,ZZZ,ZZZ,ZZ9) //

```


Se recuperan los registros de la vista PRE que están dentro del rango previamente determinado por las variables RLLAVE-E y RLLAVE-R, para cada uno se obtiene la descripción del centro de costo de la cuenta, se convierten todos los meses del presupuesto autorizado a millones y se despliegan los montos y el total por cuenta.

```

1860 *
1870 MOVE 'DESPUES DEL PROCESO' TO RENC2
1880 ..... CALCULA PRESUPUESTO AMPLIADO
1890 FIND PRE WITH PRE-ANO-CTA-CEN = RLLAVE-E THRU RLLAVE-R
1900 *
1910 AT START OF DATA
1920 WRITE // 5BT 'DESPUES DEL PROCESO' //
1930 END-START
1940 *
1950 ADD 1 TO RCONTADOR1
1960 RESET PRE-AUT(13)
1970 FOR R1 = RRES-INI-E TO RRES-FIN-E
1980 IF PRE-AUT(R1) > 0 THEN
1990 COMPUTE RMONT-PERC = PRE-AUT(R1) * RPERC
2000 ADD RDEC TO RTOT-DEC
2010 MOVE RENT TO PRE-AUT(R1)
2020 END-IF
2030 END-FOR

```

En la segunda parte del proceso se obtienen los mismos registros y para el rango de meses dado por los parámetros, se calcula el porcentaje de aumento y se suma al monto del presupuesto autorizado del mes correspondiente.

```

2040 ADD PRE-AUT(1:12) TO PRE-AUT(13)
2050 *
2060 IF RCONTADOR = RCONTADOR1
2070 COMPUTE ROUNDED PRE-AUT(13) = PRE-AUT(13) + RTOT-DEC
2080 END-IF
2090 *
2100 COMPUTE PRE-AUT(*) = PRE-AUT(*) / 1000000
2110 ADD -PRE-AUT(13) TO RTOT-DEC
2120 *
2130 WRITE IT CEN-CVE RDES-CEN-E
2140 / 8T PRE-AUT(01) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2150 28T PRE-AUT(02) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2160 48T PRE-AUT(03) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2170 68T PRE-AUT(04) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2180 88T PRE-AUT(05) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2190 108T PRE-AUT(06) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2200 / 8T PRE-AUT(07) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2210 28T PRE-AUT(08) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2220 48T PRE-AUT(09) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2230 68T PRE-AUT(10) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)
2240 88T PRE-AUT(11) (EM=Z2Z,Z2Z,Z2Z,Z2Z,Z29)

```

```

2250          9BT PRE-AUT(11) (EM=222,222,222,222,229)
2260          10BT PRE-AUT(12) (EM=222,222,222,222,229)
2270          20T 'TOTAL:'
2280          13BT PRE-AUT(13) (EM=222,222,222,222,229)
2290 *
2300 COMPUTE PRE-AUT(*) = PRE-AUT(*) * 1000000
2310 ***** ACTUALIZA PRESUPUESTO. SI EL PROCESO ES EN FIRME
2320 IF #PROCESO = 'F' THEN
2330   UPDATE RECORD (1900)
2340   END OF TRANSACTION
2350 END-IF
2360 *
2370 END-FIND

```

Se suman los meses actualizados y los no actualizados a la ocurrencia total y si es necesario se redondea este total y se convierte a millones los montos del presupuesto autorizado para todos los meses. A continuación se despliegan estos montos, se multiplican los mismos para convertirlos a pesos nuevamente y si el proceso es en firme se indica la actualización de los registros y se aplica la operación a la base de datos.

```

2380 *
2390 COMPUTE #TOT-DIF = #TOT-DES - #TOT-ANT
2400 *
2410 WRITE / 20T 'TOTAL POR CUENTA:'
2420   44T /TOT-DES (EM=222,222,222,222,229)
2430   // 20T 'D I F E R E N C I A S'
2440   44T #TOT-DIF (EM=222,222,222,222,229)
2450 *
2460 END
***** End of List *****
NEXT FIN
/* END PROGRAM

```

Al final se calcula la diferencia entre el total del presupuesto antes del proceso y el total del presupuesto después del proceso y se despliega esta diferencia.

DIRECCION EJECUTIVA DE ADMINISTRACION Y FINANZAS
 SUBDIRECCION DE CONTABILIDAD Y PRESUPUESTO
 AMPLIACION AUTOMATICA DEL PRESUPUESTO POR CUENTA Y X
 PROCESO EN PREVIO (MILLONES DE PESOS)

02/04/91
 19:17:49

CUENTA: 1101 0000 0000 0000 CAJA

GRUPO PRESUPUESTAL: 9999 SIN GRUPO PRESUPUESTAL

CAJA

MES INICIAL: 4 MES FINAL: 5

PORCENTAJE: 10.00

CENTRO	DESCRIPCION	ENERO/JULIO	FEBRERO/AGOSTO	MARZO/SEPTIEMBRE	ABRIL/OCTUBRE	MAYO/NOVIEMBRE	JUNIO/DICIEMBRE
--------	-------------	-------------	----------------	------------------	---------------	----------------	-----------------

ANTES DEL PROCESO

10000000 DIRECCION GENERAL							
	100	90	55	70	120	85	
	90	90	50	40	45	65	
	TOTAL:	900					
	TOTAL POR CUENTA:		900				

DESPUES DEL PROCESO

10000000 DIRECCION GENERAL							
	100	90	55	77	132	85	
	90	90	50	40	45	65	
	TOTAL:	919					
	TOTAL POR CUENTA:		919				

D I F E R E N C I A S

19

NEXT FIN

En las aplicaciones anteriores se aprecia la facilidad que proporciona NATURAL en las operaciones de manipulación de la información, ya que en las instrucciones de recuperación de los datos, no se manejaron criterios de selección complejos, ni se tuvo que hacer operaciones adicionales para el direccionamiento de los archivos o declaración de llaves de acceso, únicamente se determinaron en algunos casos los valores iniciales y finales de las llaves de acceso, lo que se puede efectuar si se conocen las características de la información que se desea obtener.

En el ejemplo de ampliación del presupuesto, se especificaron los valores iniciales y finales de la llave de acceso, para obtener los registros del archivo PRESUPUESTO con el año 91 y la cuenta 1101 0000 0000 0000.

CONCLUSIONES

Como parte del modelo relacional, la manipulación de datos tiene dos vertientes principales: el Álgebra y el cálculo relacional.

El cálculo relacional es una manera más automática de obtener el resultado final, porque el usuario se limita a definir el resultado que desea y deja al sistema el decidir qué operaciones se requieren para obtener ese resultado de la base de datos. El cálculo relacional es una notación para definir una relación que se puede derivar de las relaciones ya existentes en la base de datos.

Existen tres niveles de automatización en los lenguajes manipulativos.

Una en cada por vez. Es el nivel más bajo en el que el programador procesa un registro por vez.

Álgebra. En los lenguajes basados en el Álgebra, el usuario declara un conjunto de operaciones de alto nivel que deben ejecutarse sobre ciertas relaciones.

Cálculo. Es el nivel más alto de automatización en el que el usuario especifica lo que quiere y el sistema se ocupa de satisfacerlo.

La completud relacional proporciona una medida del poder expresivo de un lenguaje de manipulación. Un lenguaje es relacionalmente completo si cualquier relación definida mediante una sola expresión del cálculo, también se puede definir mediante una sola expresión de dicho lenguaje. Así, el cálculo y el álgebra relacional son una base para el diseño de lenguajes que no necesitan hacer ciclos de instrucciones.

Aunque en esencia el álgebra y el cálculo relacional son equivalentes, la ventaja del álgebra relacional sobre el cálculo relacional es que al demostrar que un lenguaje es relacionalmente completo, es más fácil demostrar que tiene operaciones equivalentes a las del álgebra relacional, que hacerlo a través de demostrar que tiene expresiones equivalentes a las del cálculo relacional; es decir, el álgebra relacional proporciona un patrón de medida del poder expresivo de un lenguaje manipulativo mucho más fácil de

seguir.

El cálculo relacional tiene como ventaja respecto al álgebra relacional que desde el punto de vista de usuario los lenguajes de manipulación basados en el cálculo relacional son más fáciles de usar porque son mucho más similares al lenguaje natural; en cambio los lenguajes basados en el álgebra relacional son más parecidos a los lenguajes de programación.

En la actualidad existen varios productos en el mercado que cumplen con la definición básica del modelo relacional. En particular, se puede afirmar que en lo que a la parte manipulativa se refiere, es difícil instrumentar un sistema que aproveche el potencial del modelo relacional en su totalidad.

APENDICE

ADABAS.

ADABAS es un sistema quasi-relacional porque los datos son presentados externamente como tablas (las relaciones entre las entidades están representadas por apuntadores de listas invertidas, que se explican más adelante) y porque los apuntadores se contruyen dinámicamente en base a las relaciones entre los datos.

Las diferencias más significativas con el modelo relacional son las siguientes:

Se considera cierto orden en las tablas de listas invertidas.

Se puede definir un orden para toda la base de datos (por ejemplo, que todos los renglones de la relación A precedan a los renglones de la relación B).

Se puede definir cualquier número de llaves de búsqueda (llave de búsqueda es un campo o combinación de campos sobre los que se construye un índice que permite el acceso directo o secuencial).

Es importante mencionar que ADABAS maneja criterios de selección complejos con un optimizador dinámico que utiliza preselección multi-llave, consiguiendo tiempos de respuesta cortos cuando se seleccionan o recuperan grandes tablas de información.

La manipulación de datos es efectuada internamente por NATURAL por medio de 2 tipos de operadores. Los operadores de búsqueda local, que establecen el direccionamiento de algún registro en la base de datos, ya sea que se efectúe una búsqueda directa del registro o que se localice el registro en términos de su posición relativa a algunas direcciones previamente establecidas. El otro tipo de operadores es el que consiste de aquéllos que actúan sobre un registro cuya dirección ha sido previamente determinada.

Un ejemplo de los operadores de búsqueda local es la consulta: "localizar el primer registro en secuencia física en la tabla T".

Se puede ejemplificar a los operadores del segundo tipo con la operación: "borrar el registro cuya dirección está dada en el Área B".

EJEMPLOS DE OPERADORES

FIND FIRST: Encuentra el primer registro de T en secuencia física y regresa su dirección.

FIND FIRST BY SEARCH KEY EQUAL: Localiza el primer registro de T en secuencia de la llave de búsqueda para la cual se puede especificar algún valor y regresa su dirección.

FIND NEXT: Encuentra el primer registro que sigue al registro identificado en A y regresa la dirección del registro encontrado en A.

FIND FIRST WITH SEARCH KEY EQUAL: Localiza el primer registro de T que sigue al registro identificado por A y que tiene el mismo valor que la llave de búsqueda; también regresa en A la dirección del registro encontrado.

FIND FIRST WITH SEARCH KEY GREATER: Localiza el primer registro de T que sigue al registro identificado por A y que tiene valores mayores que los valores de la llave de búsqueda; regresa en A la dirección del registro encontrado.

RETRIEVE: Recupera el registro identificado en el Área A.

UPDATE: Actualiza el registro identificado en el Área A.

DELETE: Borra el registro identificado en el Área A.

STORE: Adiciona un nuevo registro y regresa su dirección.

Para describir más a detalle como actúan los operadores de búsqueda local, en seguida se describe a grandes rasgos la estructura de ADABAS.

Una base de datos ADABAS consiste de dos componentes principales: el ALMACENAMIENTO DE DATOS Y EL ASOCIADOR.

ALMACENAMIENTO DE DATOS: Contiene la información actualmente almacenada en forma comprimida. El módulo de almacenamiento de ADABAS se basa en la completa separación de las estructuras de datos y de acceso, lo que permite el manejo independiente de las mismas.

ASOCIADOR: Es una unidad organizacional usada para la administración física de los datos. Esta unidad consiste de

listas invertidas para los campos llave y las tablas convertidoras de direcciones.

ADABAS mantiene una lista invertida para cada uno de los campos llave declarado en cada archivo ADABAS. La lista invertida de cada descriptor o campo llave es ordenada por valor.

Cada entrada de esta lista contiene el número de registros con ese valor en su descriptor, así como la lista de ISN's (Internal Sequence Number, que es el identificador lógico de cada registro físico) de esos registros. La siguiente figura muestra la lista invertida del descriptor CIUDAD en el archivo de CLIENTES.

Valor	No. de regs.	ISN's
Londres	27	3....
Nueva York	61	96....
Paris	31	2 6 23 76

Las tablas convertidoras de direcciones sirven para determinar físicamente la localidad del registro. Se puede apreciar gráficamente en la siguiente figura la relación entre la lista invertida, el convertidor de direcciones y el data storage.

Asociador

	<u>Lista invertida</u>			<u>Convertidor de direcciones</u>
Londres	27	3...		1
				1
Nueva York	61	96...		1
				2
Paris	31	2 6 23 76...		12
				2
				12
				2

Almacenamiento de Datos

Bloque 1

1	12758	ABC	Londres
2	12600	DEF	Nueva York
3	14811	GHI	Paris

Bloque 2

6	11643	R	Paris
96	14542	S	Nueva York
14	10002	B	Munich

Bloque 3 ...

De la lista invertida sabemos que existen 31 registros cuya ciudad es Paris (los ISN's de estos registros son 2, 6, 23, 76, ...). Para determinar, por ejemplo, la localidad física del registro con el ISN 6, ADABAS usa el ISN como un índice en el convertidor de direcciones. La sexta entrada en el convertidor de direcciones es 2. Por lo tanto el ISN 6 se encuentra físicamente en el bloque 2 del almacenamiento de datos. Este procedimiento es transparente para el usuario.

Dos tablas ADABAS se pueden acoplar físicamente ejecutando una utilería ADABAS. Los dos archivos a acoplarse

deben tener un campo descriptor común. Por ejemplo, el archivo CLIENTE y el archivo PEDIDOS se pueden acoplar por medio del campo Num-clte.

Una vez que se ha generado la lista física de acoplamiento, se pueden realizar búsquedas en las cuales puede ser usado cualquier descriptor de cualquiera de los archivos acoplados dentro del criterio de búsqueda.

Si existe una búsqueda a una sola tabla, pero con un criterio de selección complejo en el que intervienen dos lista invertidas, ADABAS selecciona los registros que están referidos en las dos listas invertidas.

Si la búsqueda anterior es realizada con frecuencia, lo más conveniente es declarar un superdescriptor formado por los dos campos, de tal manera que sólo se tenga que tener acceso a la lista invertida de este superdescriptor, evitando así el acceso a dos listas invertidas y la selección que se hace posteriormente a partir de estas dos listas invertidas (ésta sería una tarea del administrador de la base de datos no del usuario).

BIBLIOGRAFIA

1. CODD, E.F., RELATIONAL COMPLETENESS OF DATA BASE SUBLANGUAGES, EN COURANT SCIENCE SYMPOSIA, VOL. 6: DATA BASE SYSTEMS, ED. R. RUSTIN, PRENTICE-HALL, INC., ENGLEWOOD CLIFFS, N.J., 1972
2. CODD, E.F. FURTHER NORMALIZATION ON THE DATA BASE RELATIONAL MODEL, EN COURANT COMPUTER SCIENCE SYMPOSIA, VOL. 6: DATA BASE SYSTEM, ED. POR R. RUSTIN, PRENTICE-HALL, ENGLEWOOD CLIFFS, N.J., 1972
2. TER BEKKE, J.H., A DATA MANIPULATION LANGUAGE FOR RELATIONAL STRUCTURES, EN PROCEEDINGS OF THE 2ND INTERNATIONAL CONFERENCE ON VERY LARGE DATA BASES: SYSTEMS FOR LARGE DATA BASES, ED. POR P.C. LOHEMANN Y E.J. NEUHOLD, 1976
3. MARTIN, JAMES, ORGANIZACION DE LAS BASES DE DATOS, 1a. ED., ED. PRINTECE-HALL HISPANOAMERICANA, E.U. 1977
4. OLLE, T. WILLIAM, THE CODASYL APPROACH TO DATA BASE MANAGEMENT, 1a. ED., JOHN WILEY & SONS, E.U. 1977
5. DATE, C.J., A FORMAL DEFINITION OF THE RELATIONAL MODEL, ACM-SIGMOD VOL 13 NO. 1, SEPTIEMBRE 1982
6. PIROTTE, A., A PRECISE DEFINITION OF BASIC RELATIONAL NOTIONS AND OF RELATIONAL ALGEBRA, ACM-SIGMOD, VOL. 13, NO. 1, SEPTIEMBRE 1982
7. MAIER, DAVID, THE THEORY OF REALTIONAL DATABASES, ROCKVILLE, MD.: COMPUTER SCIENCE PRESS, 1983
8. NIEMI TIMO, A STRAIGHTFORWARD FORMALIZATION OF THE RELATIONAL MODEL, EN POC. 1984 ACM-SIGMOD
9. MERRET, T.H., RELATIONAL INFORMATION SYSTEMS. RESTON VA.; RESTON PUBLISHING COMPANY, 1984
10. DATE, C.J., AN INTRODUCTION TO DATABASE SYSTEMS, VOL. 1, 4a. EDICION, ED. ADDISON WESLEY, E.U. 1986
11. DEEN, S.M., FUNDAMENTOS DE LOS SISTEMAS DE BASES DE DATOS, 1a. EDICION. ED. GUSTAVO GILI, ESPAÑA 1987

12. BATE, JOSEPH ST. JOHN; VADHIA, DINESH B., FOURTH GENERATION LANGUAGES UNDER DOS AND UNIX, 1a. ED., BSP PROFESSIONAL BOOKS, GRAN BRETAÑA, 1987.
13. ADABAS INTRODUCTION MANUAL, ADA-410-000, POR SOFTWARE AG, JUNIO 1982.
14. NATURAL ADVANCED TECHNIQUES WORKSHOP, NTR-129-082, POR SOFTWARE AG, JUNIO 1984.
15. NATURAL 2 REFERENCE MANUAL, NAT-210-022, POR SOFTWARE AG, JULIO 1987.