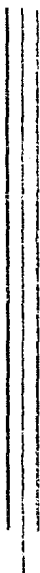


17
29



Universidad Nacional Autónoma de México

FACULTAD DE CIENCIAS



"UN ADMINISTRADOR DE BASES
DE DATOS PARA LA
INFORMACION CENSAL"

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 N
ROSA MARTHA GUERRERO MORA Y
LUZ DE GUADALUPE TORRES NAVARRO



TEJIS CON
FALLA DE ORIGEN

MEXICO. D. F.

JULIO 1990



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.

I N D I C E

Introducción.....	I
I. La problemática de la información censal.....	1
I.1. Información censal.....	2
I.2. Marco geoestadístico del Sistema Nacional de Información.....	4
I.3. Niveles de información.....	5
I.4. Visión de la información censal	5
I.5. Problemas con el procesamiento de la información..	8
I.6. Super-Base como una solución.....	10
II. Introducción a las bases de datos.....	11
II.1. Términos y conceptos básicos.....	12
II.2. Características de la tecnología de bancos de datos.....	13
II.2.1. Independencia de los datos.....	13
II.2.2. Irredundancia y habilidad para compartir datos.....	14
II.2.3. Versatilidad para representar relaciones.	15
II.2.4. Integridad.....	16
II.2.5. Flexibilidad de acceso.....	16
II.2.6. Seguridad.....	17
II.2.7. Rendimiento y eficiencia.....	18
II.2.8. Administración y control.....	19
II.2.9. Costo mínimo.....	19
III. Organizaciones lógicas de archivos.....	20
III.1. Sistemas básicos de organización de archivos...	20
III.1.1. Secuencial.....	20
III.1.2. Indizado.....	22
III.1.3. Directo.....	24
III.1.4. Cadenas y multianillos.....	25
III.2. Técnicas de acceso.....	27
III.2.1. Exploración o barrido del archivo.....	27
III.2.2. Búsqueda por bloque.....	28
III.2.3. Búsqueda binaria.....	28
III.2.4. Búsqueda secuencial indizada.....	29
III.2.5. Direccionamiento clave-igual-dirección.	29
III.2.6. Algoritmo para conversión de clave.....	29
III.2.7. Función de dispersión.....	30
III.3. Estructuras de organizaciones lógicas de archivos.....	30
III.3.1. Archivos planos.....	30
III.3.2. Estructuras de árbol.....	31
III.3.3. Archivos Jerárquicos.....	32
IV. Características de un administrador de bases de datos..	35
IV.1. Modelos de bases de datos.....	35
IV.1.1. Modelo Relacional.....	36
IV.1.2. Modelo Jerárquico.....	37

IV.1.3. Modelo Red.....	38
IV.2. Lenguaje de descripción de esquema.....	39
IV.3. Lenguaje de manipulación de datos.....	40
IV.4. Algebra Relacional.....	41
IV.5. Cálculo Relacional.....	43
V. El sistema administrador de bases de datos Super-Base...	45
V.1. El Modelo Relacional-Jerárquico de Super-Base....	45
V.2. Lenguaje para la descripción de esquemas.....	47
V.3. El lenguaje de consulta.....	48
V.4. Cálculo relacional.....	49
V.5. Lenguaje de manipulación de jerarquía.....	50
V.6. Relaciones entre variables.....	51
V.7. Operaciones conjuntas Micromap-Superbase.....	51
V.8. Facilidades del sistema.....	51
VI. La estructura del sistema administrador.....	52
VI.1. Modelo.....	52
VI.2. Estructuras de definición.....	56
VI.2.1. Esquema.....	57
VI.2.2. Diccionarios.....	57
VI.3. Estructuras de almacenamiento de datos.....	59
VI.3.1. Control de la tabla bidimensional.....	60
VI.3.2. Direccionamiento de árboles.....	61
VI.4. Estructuras de conexión con Micromap.....	62
VI.4.1. Ligas de la base de datos con Micromap...	64
VI.4.2. Ligas de Micromap con la base de datos....	65
VI.5. Estructuras de apoyo.....	66
VI.5.1. Almacenamiento temporal de instrucciones.	66
VI.5.2. Ayuda.....	68
VI.5.3. Macroinstrucciones.....	70
VI.5.4. Selección.....	70
VI.5.5. Exportar.....	71
VI.6. Variables importantes del sistema.....	71
VI.6.1. Variables auxiliares para el manejo de la jerarquía.....	72
VI.6.2. Variables para el almacenamiento temporal de datos.....	72
VI.6.3. Variables de entrada y salida.....	72
VI.6.4. Variables de estado general.....	72
VII. Descripción funcional de Super-Base.....	73
VII.1. Módulo de control de la sesión.....	73
VII.2. Subsistemas principales.....	73
VII.3. Definición de la estructura.....	74
VII.4. Subsistema de soporte de la estructura Relacional.....	75
VII.4.1. Altas.....	76
VII.4.2. Bajas.....	78
VII.4.3. Modificaciones.....	80
VII.5. Subsistema de soporte de la estructura Jerárquica.....	81
VII.5.1. Cambio de nivel.....	82
VII.5.2. Funciones Jerárquicas.....	83

VII.5.3. Importación de datos de formato estandar de datos.....	83
VII.5.4. Exportación de datos a formato estandar de datos.....	84
VII.6. Subsistema de soporte del cálculo relacional...	85
VII.7. Subsistema de soporte al usuario.....	87
VII.7.1. Funciones del MS-DOS.....	87
VII.7.2. Editor de Macroinstrucciones.....	88
VII.8. Subsistema de enlace MICROMAP-SUPERBASE.....	88
VII.8.1. Altas.....	88
VII.8.2. Bajas.....	89
VII.8.3. Consulta.....	89
VIII. Instrucciones de Super-Base.....	90
VIII.1. Introducción.....	90
VIII.2. Lenguaje de esquema.....	91
VIII.3. Comandos Internos de Super-Base.....	94
VIII.4. Comandos Externos de Super-base.....	96
VIII.5. Operaciones entre niveles que soporta la base de datos.....	97
VIII.6. Módulo de ayuda.....	98
IX. Diversas Aplicaciones.....	101
IX.1. Aplicaciones con Información Censal.....	101
IX.2. Aplicaciones en Demografía.....	107
IX.3. Aplicación de Super-Base con Micromap a cartografía censal.....	110
Conclusión.....	115
Apéndice A.....	118
Apéndice B.....	139
Bibliografía.....	152

INTRODUCCION

En muchos problemas importantes del gobierno y la industria, las decisiones orientadas a la acción administrativa están basadas en el estudio de las estadísticas vitales y de población. La constitución de los Estados Unidos Mexicanos estipula un censo de población cada 10 años, de manera que esta información sirva para la planeación de los servicios públicos, políticas nacionales, administración de los recursos, etc..

Para la presentación de los resultados del censo es necesario llevar a cabo una gran cantidad de cálculos, por lo que se hace indispensable, la utilización de herramientas de cómputo.

La información censal usualmente es capturada a nivel básico en archivos secuenciales o indizados de forma relacional. Los resultados son generados a nivel municipio, estado y país. Para la obtención de éstos es necesario implementar programas específicos, quedando, los resultados finales, integrados en diferentes archivos sin conservar su relación original. La corrección de un dato a nivel básico no genera la corrección automática a los siguientes niveles, por lo que se incurre en errores de consistencia, así como de duplicación de información.

Con el fin de poder realizar el procesamiento de la información censal más ágil y eficientemente se desarrolló un sistema administrador de bases de datos que permite el manejo de la información censal tomando en cuenta sus características. Para tales efectos este sistema crea una estructura en donde se pueden almacenar datos en forma relacional y jerárquica.

El sistema, a través de sus diversas instrucciones, permitirá la actualización, reestructuración, obtención de la información almacenada en la computadora y la generación de la mayor parte de los reportes requeridos evitando, de esta forma, la duplicación de información, la pérdida de tiempo y recursos, facilitando, además, la congruencia y consistencia de los datos. Esto agilizará la tarea de quienes se ocupan de la toma de decisiones.

El objetivo de este trabajo es describir el sistema administrador de bases de datos "SUPER-BASE".

El contenido de este trabajo está organizado de tal forma que muestra la problemática que origina y fundamenta al sistema, introduce los conceptos que se manejan a lo largo del texto y describe el modelo, estructura y funcionamiento del sistema administrador.

En el capítulo I, a manera de introducción, se describe la problemática de la información censal.

En los capítulos II, III y IV se presenta información relativa a las bases de datos, estructuras óptimas y características deseables de un administrador de bases de datos.

En el capítulo V se describe el modelo, medio ambiente y características de "SUPER-BASE".

El capítulo VI describe el modelo y sus estructuras.

El capítulo VII está dedicado a explicar el funcionamiento de "SUPER-BASE", para lo cual, se describen los subsistemas principales y los módulos que los componen.

El capítulo VIII muestra cada uno de los comandos con sus reglas sintácticas y ejemplos, además de explicar el uso de la ayuda del sistema.

En el capítulo IX se exponen algunas aplicaciones del sistema, que muestran su utilidad en diversos casos que pueden presentarse en el análisis de información censal. Asimismo se presentan aplicaciones de la base de datos trabajando en conjunto con el sistema cartográfico MICROMAP, para resolver algunos problemas interesantes de cartografía censal.

CAPITULO I

PROBLEMATICA DE LA INFORMACION CENSAL

La mayor extensión y complejidad de las sociedades modernas se debe al gran intercambio cultural a través de los diversos medios de información, que cada día impulsan el crecimiento y necesidad de una mayor comunicación. No es de extrañar que en las últimas décadas se haya desarrollado gran cantidad de trabajos relacionados con la información desde varios puntos de vista: social, científico e informático.

Dada la necesidad cada vez mayor de manejar grandes volúmenes de información, el desarrollo de las bases de datos, dentro del campo de la informática, será sin duda una de las actividades más importantes durante los próximos años. Los datos se contemplarán cada vez con mayor interés, como un recurso vital para los diversos organismos ante la necesidad de organizarse y procesar información para la toma de decisiones, con el fin de crecer, planear y mejorar. Por otra parte, los servicios de información capacitados para el acopio, organización y venta de datos tienen una gran demanda por lo que requieren manejar grandes cantidades de información de forma eficiente, oportuna y confiable.

Es impresionante observar como crecen en volumen e importancia los archivos de datos que utilizan las computadoras. Esto puede observarse claramente en la vida diaria y en todas las áreas de la industria. Este gran crecimiento es debido a que, las personas y empresas están concientes de que el acceso a la información y el almacenamiento en bases de datos ampliarán en gran medida sus posibilidades y perspectivas de acción. Por otra parte el costo de los equipos para el almacenamiento electrónico de datos decrece más rápidamente que cualquier otro en el campo de la computación y muy pronto resultará más barato almacenar datos en medios magnéticos y ópticos que sobre papel. Además los tipos de información que pueden almacenarse son cada vez más variados como: dibujos, fotografías, voces humanas, aplicaciones a diversas áreas de la industria, tomografía computarizada en medicina, etc.

Algunos de los principales problemas a los que se enfrenta la informática son: el crecimiento del volumen de datos y la complejidad de su organización y procesamiento. Para el adecuado aprovechamiento de los datos se exige del desarrollo de sistemas que satisfagan las necesidades de los usuarios (accesibles, flexibles, compatibles y consistentes) y tengan la posibilidad de transmitir datos.

Como puede observarse existe gran cantidad de áreas en las que se requiere del manejo de información, en particular, la presente tesis, tiene como objetivo resolver el problema del manejo adecuado de la información a nivel socio-económico-demográfico.

I.1 INFORMACION CENSAL.

Entre los renglones del conocimiento humano en que es imprescindible contar con medios que permitan obtener la información relacionada con un punto de interés, se encuentra sin duda alguna, la información censal puesto que sirve como base para la planeación socio-económica de un país.

Durante mucho tiempo el procesamiento y consulta de la información sólo se efectuó a través de métodos manuales. La aparición de las computadoras agilizó de gran manera la velocidad del procesamiento de datos.

El primer censo en México fué levantado en 1880 y se ha estado realizando cada 10 años hasta la fecha, excepto el que debía llevarse a cabo en 1910. La realización del censo tiene como fundamento legal los artículos respectivos de la Constitución Política de los Estados Unidos Mexicanos, de la ley Federal de Estadística y su reglamento, así como el decreto presidencial publicado en el Diario Oficial del 29 de agosto de 1979.

A grandes rasgos, el proceso para llevar a cabo el censo implica los siguientes pasos:

A) PLANEACION Y RECOPIACION.

Los preparativos y planes previos a un censo dependerán de la información que se desea obtener y de los fondos disponibles.

Primero, es necesario fijar una fecha en la cual la información ha de ser recolectada. En la planeación de los detalles debe tomarse en cuenta la forma de la cédula del censo, el mejor procedimiento de recolección de datos, la forma más eficiente de tabulación del contenido de las cédulas y el establecimiento de los medios de control para asegurar la cobertura nacional.

El territorio será dividido en áreas geoestadísticas básicas (AGEB), el tamaño de la mayoría de éstas es tal que el trabajo puede ser llevado a cabo por el empadronador dentro de un periodo predeterminado. Los AGEB se diseñan de manera tal que no crucen las delimitaciones de áreas políticas o estadísticas para las cuales se ha de publicar la información censal.

B) REUNION DE LOS HECHOS.

El procedimiento para la recopilación de información es el método de enumeración, en el que los enumeradores entrenados y remunerados van de vivienda en vivienda dentro de un área específica y recaban la información preguntando a cada individuo o jefe de familia. Esta práctica de entrevista directa tiene la ventaja de que los agentes censales pueden ser entrenados para asegurar un llenado de las cédulas más preciso, se pueden formular preguntas más elaboradas y los resultados tendrán mayor uniformidad. Sin embargo el entrenamiento necesario es costoso, y la exactitud y el detalle del censo pueden depender de la eficiencia y la paga de los enumeradores.

Los delegados y subdelegados censales estatales en cada entidad federativa verificarán e inspeccionarán las cédulas para corregir lo más pronto posible errores obvios. Las técnicas muestrales son empleadas en todos los pasos desde la crítica y la tabulación hasta el control de calidad del trabajo de rutina.

C) RECOPIACION, EVALUACION Y PRESENTACION DE LOS RESULTADOS.

Una vez terminada la etapa de recolección, las delegaciones censales enviarán la información a los centros regionales de codificación y captura previamente establecidos, donde se analizará que las anotaciones hayan sido hechas correctamente, que no haya respuestas inconsistentes y codificarán las respuestas expresadas verbalmente, de tal manera que la información de las cédulas pueda ser capturada.

Los procesos serán elaborados de acuerdo a los puntos de interés. La evaluación de la información será en gran medida efectuada en la computadora. La computadora será programada para detectar y corregir, de acuerdo con reglas específicas, ciertos tipos de inconsistencias en los resultados individuales. Por otra parte, la computadora almacenará el número de inconsistencias y falta de respuestas, para poder efectuar una estimación de la posible deformación de los resultados.

Las tabulaciones son enviadas a expertos en varios campos y estas personas llevarán a cabo comparaciones con tabulaciones censales previas para poder dar una evaluación del censo.

Finalmente la información será publicada para el uso general y se acompañará de textos explicativos.

I.2 MARCO GEOESTADISTICO DEL SISTEMA NACIONAL DE INFORMACION.

Para llevar a cabo el X Censo General de Población y Vivienda de 1980, el VI Censo Agrícola Ganadero y Ejidal y los censos económicos se elaboró un programa de apoyo cartográfico "MARCO GEOESTADISTICO" como una de sus partes principales.

Los resultados censales tradicionalmente y hasta la fecha son difundidos en publicaciones en las cuales la información aparece resumida y condensada a nivel localidad, municipio y entidad federativa.

A través del tiempo los límites municipales y las ciudades se modifican. Estos cambios no pueden ser captados por levantamientos censales si no se posee un marco espacial comparable. La carencia de este crea una incertidumbre que entorpece e imposibilita el conocimiento del alcance de las acciones y obras emprendidas por los diferentes organismos federales, estatales y municipales.

Conciente de la problemática anterior, la Coordinación General de los Servicios Nacionales de Estadística, Geografía e Informática, en su calidad de responsable de los trabajos censales ha desarrollado e implementado el sistema denominado Marco Geoestadístico del Sistema Nacional de Información.

"El Marco Geoestadístico del Sistema Nacional de Información es un mecanismo diseñado para relacionar la información estadística con la información geográfica y otras informaciones producidas por organismos públicos y privados, incluyendo la generada por la propia Coordinación General de los Servicios Nacionales de Estadística, Geografía e Informática, sobre un marco geográfico único de carácter nacional, altamente confiable en su estructura, destinado a los encargados de producir, manejar e interpretar la información." <1>

El área geoestadística básica "AGEB" es la unidad mínima del marco geoestadístico que garantiza el cubrimiento de todos los municipios y entidades federativas, así como el cubrimiento del Territorio Nacional. Estos fueron concebidos para facilitar la parte operativa del levantamiento del censo y también para manejar, procesar y resumir la información captada. Los AGEB han sido delimitados de tal modo que son congruentes con la división política administrativa del país.

1 X Censo General de Población y Vivienda, 1980.
Cartografía Geoestadística del Estado de Baja California Sur.

I.3 NIVELES DE INFORMACION.

Los niveles de información han sido estructurados de tal manera que proporcionan una referencia geográfica. Estos son:

NACIONAL

ESTATAL

MUNICIPAL

LOCALIDAD

AGEB

Donde éstos cumplen con la siguientes funciones:

- a) Permitir la captación de la información de un área geográfica determinada y referirla al marco geoestadístico.
- b) Normar y uniformar la exposición de datos ya sean gráficos o tabulares.
- c) Permitir la difusión masiva con el objeto de que pueda ser utilizada por los usuarios de la información geográfica y estadística.

I.4 VISION DE LA INFORMACION CENSAL.

Los resultados publicados de la información censal están organizados en diferentes cuadros y de acuerdo al marco geoestadístico, como se muestra en las figuras 1.1. y 1.2. La figura 1.1 corresponde a la población total por edad y sexo en la Republica Mexicana para 1980. La figura 1.2 muestra a la población de 12 años y más por municipio y sexo según estado civil.

Cada uno de estos cuadros puede verse como una tabla bidimensional en donde la primera columna contiene el identificador del objeto y las subsiguientes sus características. Puede contemplarse como una organización de tipo relacional ya que es evidente que existe una relación entre el identificador y sus características.

Por otra parte, la información básica del censo también puede ser vista como una organización de tipo relacional, donde a cada individuo le corresponden cierta características, como puede verse en la figura 1.3.

EDAD	TOTAL	HOMBRES	MUJERES
TOTAL	66 846 833	33 039 307	33 807 526
0	1 711 865	862 929	848 936
1	1 681 778	847 488	834 290
2	1 904 481	956 821	947 660
3	2 003 797	1 003 389	1 000 408
4	2 045 947	1 027 885	1 018 062
0-4	9 347 868	4 698 512	4 649 356
5	2 083 159	1 051 887	1 031 272
6	2 192 014	1 098 923	1 093 091
7	2 047 573	1 034 480	1 013 093
8	2 104 835	1 058 395	1 046 440
9	1 856 374	929 238	927 136
5-9	10 283 955	5 172 923	5 111 032
10	1 991 855	1 010 345	981 510
11	1 682 913	846 610	836 303
12	1 941 033	991 035	949 998
13	1 723 852	860 847	863 005
14	1 754 698	865 838	888 860
10-14	9 094 351	4 574 675	4 519 676
15	1 633 560	808 804	824 756
16	1 569 089	770 875	798 214
17	1 559 619	772 749	786 870
18	1 597 979	793 713	804 266
19	1 296 292	620 547	675 745
15-19	7 656 539	3 766 688	3 889 851
20	1 471 339	699 358	771 981
21	1 074 823	523 337	551 486
22	1 275 464	622 291	653 173
23	1 207 518	585 987	621 531
24	1 125 383	541 201	584 182
20-24	6 154 527	2 972 174	3 182 353
25	1 177 079	564 497	612 582
26	960 499	463 397	497 102
...			

FIGURA 1.1. Cuadro #1 del censo nacional de poblacion de 1980. Información resumen de todo el país.

MUNICIPIO Y SEXO	POBLACION DE 12 AÑOS Y MAS	SOLTEROS	ESTADO CIVIL								
			EN SU FAMILIA				VIUDES	SEPARADOS	VIUDES	OTROS CASOS	OTROS
			TOTAL	CIVIL	RELIGIOSO	CIVIL Y RELIGIOSO					
TOTAL	129 149	17 226	96 262	17 667	1 263	9 121	9 422	1 651	1 187	664	11
HOMBRES	71 181	12 174	52 685	6 716	612	24 356	4 689	693	1 218	265	26
MUJERES	57 968	5 052	43 577	6 951	651	23 775	4 733	1 968	2 666	507	51
COMONDU	38 868	14 376	17 610	4 613	309	12 699	2 208	625	627	132	17
HOMBRES	26 419	8 123	8 811	2 271	216	6 291	1 056	123	298	66	7
MUJERES	17 789	6 253	9 802	2 342	193	6 408	1 152	492	329	66	10
MULEDO	27 721	7 028	8 944	2 509	190	6 245	1 120	254	527	66	10
HOMBRES	6 108	2 814	4 483	1 247	98	3 121	576	54	103	26	6
MUJERES	6 123	4 214	4 461	1 262	92	3 124	544	190	394	40	4
PAZ, LA	66 420	31 682	36 289	10 645	556	28 164	6 104	1 022	2 431	142	10
HOMBRES	33 826	20 137	19 704	5 200	260	13 946	2 693	282	627	100	13
MUJERES	32 594	11 545	16 585	5 445	296	14 218	3 411	740	2 124	42	6

FIGURA 1.2. Cuadro # 2 del censo nacional de población de 1980. Correspondiente a el Estado de Baja California Sur.

PERSONA VARIABLES DE ESTUDIO

NOMBRE	DOMICILIO	EDAD	SEXO
LUIS	PRIETO 14	56	01	.
PEDRO	MARINA 16	35	01	.
MARIA	TORRES 3	45	02	.
MONICA	COPILO 4	25	02	.
GERARDO	J.GALLO 1	13	01	.
.

FIGURA 1.3. Tabla bidimensional de la información básica.

Por otra parte, los datos a nivel nacional no son datos aislados, sino que tienen una organización JERARQUICA intrínseca. Los datos son almacenados y procesados para obtener los resultados a diferentes niveles resumen: totales del país, subtotales por estado, subtotales por municipio, etc., en la figura 1.4 se muestra la organización jerárquica de la información censal nacional.

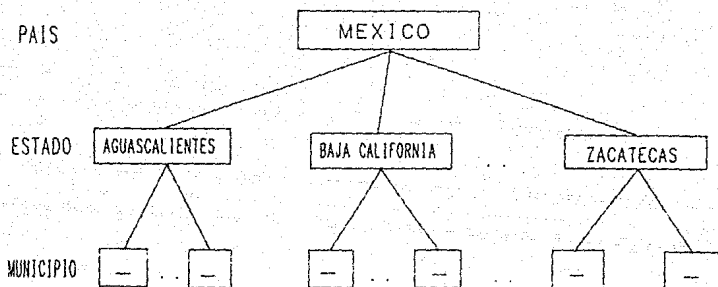


FIGURA 1.4. Representación Jerárquica de la información censal.

I.5 PROBLEMAS CON EL PROCESAMIENTO DE LA INFORMACION.

Usualmente toda la información se captura en archivos secuenciales o indizados de forma relacional.

Una de las formas comunes para el análisis de información sobre la población es la captura a nivel básico, generando posteriormente, datos a nivel municipio, estado y país. Para generarlos es necesario implementar programas específicos, quedando los resultados finales, integrados en diferentes archivos sin conservar su relación original. La corrección de un dato a nivel básico no genera la corrección automática a los siguientes niveles, por lo que se incurre en errores de consistencia, así como de duplicación de información.

Por otra parte, no existe una gran flexibilidad para la obtención de información con objetivos específicos propiciando pérdida de tiempo y falta de información para quienes la requieren. Por ejemplo, sería muy tardado y costoso: obtener listados de las localidades que tengan un determinado nivel educacional en relación con su situación económica, o listados por localidad, municipio y estado cruzando información sobre su nivel económico y su porcentaje dentro de las diferentes ramas de trabajo.

El almacenamiento de la información censal en una base de datos relacional representa serios problemas de redundancia ya que si se desea almacenar, por ejemplo, la información del número de establecimientos económicos por manzana deberán tenerse algunos campos que indiquen el estado, municipio, localidad, azeb, manzana y número de establecimientos económicos.

Entonces la información puede almacenarse como sigue:

ESTADO	MUNICIPIO	LOCALIDAD	AGEB	MANZANA	NUMERO DE ESTABLECIMIENTOS ECONOMICOS
01	COMONDU	SAN JUAN	001-3	1	0
01	COMONDU	SAN JUAN	001-3	2	3
01	COMONDU	SAN JUAN	001-3	3	5
01	COMONDU	SAN JUAN	001-3	4	2
01	COMONDU	SAN JUAN	001-3	5	2
01	COMONDU	SAN JUAN	001-3	6	1
01	COMONDU	SAN JUAN	001-3	7	0
01	COMONDU	SAN JUAN	001-3	8	1
01	COMONDU	SAN JUAN	001-3	9	0
01	COMONDU	SAN JUAN	001-3	10	0
01	COMONDU	SAN JUAN	001-3	11	1
01	COMONDU	SAN JUAN	001-3	12	1
...
01	COMONDU	SANTIAGO	005-1	1	0
01	COMONDU	SANTIAGO	005-1	2	2
01	COMONDU	SANTIAGO	005-1	3	4
01	COMONDU	SANTIAGO	005-1	4	3
01	COMONDU	SANTIAGO	005-1	5	2
01	COMONDU	SANTIAGO	005-1	6	5
01	COMONDU	SANTIAGO	005-1	7	1
01	COMONDU	SANTIAGO	005-1	8	0
01	COMONDU	SANTIAGO	005-1	9	0
01	COMONDU	SANTIAGO	005-1	10	2
01	COMONDU	SANTIAGO	005-1	11	4
01	COMONDU	SANTIAGO	005-1	12	1
...

Es evidente que el estado, municipio y localidad se repetirán una gran cantidad de veces, lo cual propicia mayor consumo de memoria y de tiempo de acceso. Ahora, supongamos que se desean conocer las manzanas que tienen de 2 a 5 establecimientos económicos, entonces tendría que aplicarse una serie de instrucciones o hacer un programa específico. Si el punto de interés cambiara tendría que modificarse el programa o incluso hacer uno nuevo.

Las bases de datos comerciales en general son de tipo relacional por lo que presentan las desventajas y problemas que mencionamos en los párrafos anteriores.

Por otra parte, existen en el mercado bases de datos jerárquicas, en las que es posible almacenar únicamente los nombres de los nodos, por lo que no son de utilidad para el almacenamiento de la información censal, además de presentar otros problemas importantes como la dificultad de manejo, que las hace poco amigables desde el punto de vista de los usuarios no muy familiarizados con el manejo de bases de datos.

I.6 SUPER-BASE COMO UNA SOLUCION.

De lo anterior se deduce la necesidad de la utilización de un administrador de bases de datos que, a través de sus diversas instrucciones, permita la fácil actualización, reestructuración, obtención de la información almacenada en la computadora y la generación de la mayor parte de los reportes requeridos, evitando, de esta forma la duplicación de información, y la pérdida de tiempo y recursos, facilitando además la congruencia y consistencia de los datos. Esto agilizará la tarea de quienes se ocupan de la toma de decisiones.

Se pensó entonces, para tales efectos, en la creación de un administrador de bases de datos llamado Super-Base. De tal manera que permita crear una estructura en la cual se permita el almacenamiento de datos de manera relacional y jerárquica.

CAPITULO II

INTRODUCCION A LAS BASES DE DATOS

Desde tiempo inmemorial el ser humano se ha preocupado por contar. En un principio contaban los animales y utensilios que poseían mediante el uso de guijarros (cálculos <2>). Un ejemplo lo constituyen las piedras de Salisbury, que se cree que servían para calcular la posición de los astros.

Con el paso del tiempo, el hombre se esforzó por tratar de fabricar máquinas que fueran capaces de efectuar operaciones en forma más rápida y eficiente, algunos ejemplos que podríamos citar son: la máquina de Antiquitera, la calculadora de Pascal, la máquina analítica de Babagge entre muchos otros.

En el año de 1890, Hollerith utilizó la primera máquina para el procesamiento de los datos del censo de Estados Unidos. La máquina de Hollerith se basaba en el principio de tarjetas perforadas, como aquel utilizado originalmente para los telares de Jaquard, sólo que las tarjetas de Hollerith almacenaban datos demográficos en vez de figuras para las telas y tenían el tamaño de un billete de un dólar.

Entre las primeras máquinas que pueden merecer el nombre de computadora están aquellas de registro unitario que utilizaban relevadores para llevar a cabo las operaciones. A finales de la década de los 40's se construyó la primera computadora que utilizaba bulbos en lugar de relevadores llamada ENIAC.

En fechas anteriores existían grandes volúmenes de datos que debían procesarse para obtener información sobre diferentes actividades cotidianas como cuentas bancarias, la nómina de una empresa, datos de producción en las industrias. Es decir, los datos y su procesamiento ya existían, pero se llevaba a cabo de una forma tediosa e ineficiente, ya que todas las operaciones matemáticas se hacían a mano.

Los primeros dispositivos electrónicos programables, que podían realizar velozmente operaciones matemáticas comenzaron a utilizarse no solamente para efectuar operaciones en forma rápida sino también para almacenar datos y a través de su procesamiento convertirse en información útil en la toma de decisiones.

2 Nótese que la palabra cálculo procede de "calculus" (latín) guijarro, piedra.

Sin embargo, el concepto de "Base de Datos" no había aparecido aún como se conoce ahora, ya que todo se limitaba a almacenar y procesar datos, pero sin ninguna estructura preconcebida.

Entre 1950 y 1960, las computadoras se estaban aplicando al procesamiento de grandes cantidades de información para los negocios. Así surge la necesidad de almacenar datos en una estructura predefinida aunque fuera rudimentaria. Uno de los primeros intentos fué desarrollado en COBOL en 1960, la estructura era un archivo que consistía en una colección de registros cada uno de los cuales estaba formado por una serie de campos que contenían el formato de los datos.

En principio, todos los datos necesarios para cualquier aplicación podían ser almacenados en una colección de archivos.

Pero éstos estaban muy limitados. Una de las principales restricciones que poseían era que los archivos solamente podían ser accedidos de forma secuencial o directa lo que hacía más lento el proceso de búsqueda en el primer caso y complejo en el segundo.

Debido al gran incremento en el número de aplicaciones y su complejidad, no era posible satisfacer las necesidades de los usuarios. Esto propició la búsqueda de nuevos conceptos como el de métodos de acceso, permitiendo nuevas estructuras de archivos.

II.1 TERMINOS Y CONCEPTOS BASICOS.

A partir de la década de los 50's el concepto de "Sistema de Base de Datos" comienza a aparecer paulatinamente. Sin embargo, existe todavía falta de consenso en cuanto al concepto, lo que produce que no exista una definición precisa de lo que es una "Base de Datos"; mientras unos se refieren a ella como los archivos donde los datos son almacenados, otros la identifican con el programa que los maneja. Para no incurrir en ambigüedades daremos a continuación algunas definiciones:

■ BASE DE DATOS

Es una colección de datos interrelacionados, almacenados en conjunto, sin redundancias perjudiciales o innecesarias, con el fin de servir a una o más aplicaciones de la mejor manera posible, dicho en otras palabras, es un conjunto de archivos en los que los registros se encuentran interrelacionados mediante una relación específica.

■ SISTEMA ADMINISTRADOR DE BASES DE DATOS

Es un conjunto de programas (software) capaz de dar soporte técnico a la Base de Datos como: organizar, almacenar, recopilar, manipular y mantener la estructura lógica global de los datos.

■ SISTEMA DE BASES DE DATOS

Es un conjunto de Bases de Datos independientes.

II.2 CARACTERISTICAS DE LA TECNOLOGIA DE BASES DE DATOS.

Una herramienta valiosa y esencial para el desarrollo y organización de sistemas modernos de información es la tecnología de bases de datos cuyas características indispensables son:

- 1) Independencia de los datos,
- 2) Irredundancia y habilidad de compartir datos,
- 3) Versatilidad para la representación de relaciones,
- 4) Integridad,
- 5) Flexibilidad de acceso,
- 6) Seguridad,
- 7) Rendimiento y eficiencia,
- 8) Control y administración,
- 9) Costo mínimo.

Estos conceptos se explicarán en las siguientes páginas

II.2.1 INDEPENDENCIA DE LOS DATOS.

La independencia se entenderá como el aislamiento o independencia entre los programas de aplicación y los usuarios.

Esta se hizo cada vez más necesaria, al usar las bases de datos, con el fin de proteger la información de los cambios que pudieran darse en la organización específica de los datos a nivel lógico o físico, así como de un posible cambio en los criterios relativos al almacenamiento de datos en la base.

La independencia física, es una característica de la base de datos, que permite modificar la distribución y la organización física de los archivos empleada para almacenarlos sin afectar la estructura lógica general ni a los programas de aplicación.

La independencia lógica es la capacidad de aislar las aplicaciones de los cambios que se hacen en la organización lógica de la base de datos.

Cuando los datos no son independientes y la estructura de la Base es modificada, pueden generarse diversos problemas como son: falta de información oportuna, pérdida de tiempo y alto costo. La elaboración de software es muy costosa y resulta aún más caro reprogramar y modificar programas ya existentes. La magnitud de los problemas que se presentan cuando hay modificaciones puede ser catastrófica en el caso de sistemas de información interconectados y basados en múltiples archivos.

La importancia de lograr un alto grado de independencia es evidente, aunque debe reconocerse que no existe una independencia total de los datos ya que no puede protegerse a un programa de aplicación de todos los cambios posibles. Por otra parte, la continua búsqueda de independencia ha sido un ingrediente clave en la evolución modular y rapidez de los sistemas de información modernos.

II.2.2 IRREDUNDANCIA Y HABILIDAD PARA COMPARTIR DATOS.

Es la propiedad que permite que las diversas aplicaciones tengan acceso a una base de datos central donde se encuentra toda la información requerida y, en lo posible, eliminar la necesidad de almacenar datos en forma redundante.

Las aplicaciones al compartir una misma base requerirán de ciertas facilidades como son:

- Control sobre el acceso,
- Control sobre las interacciones de programas independientes,
- Facilidad para tener visualizaciones concurrentes de los mismos datos,
- El acceso eficiente a los diferentes subconjuntos de datos,
- Mecanismos eficientes para diferentes subconjuntos de archivos y campos, etc.

La habilidad para compartir datos y la redundancia tienen una relación estrecha, por ejemplo :

En una Universidad tanto el departamento de servicios escolares como el de finanzas deben tener información relativa a los alumnos inscritos: su dirección, teléfono, etc. Cada uno puede tener su propia información causando un nivel alto de redundancia, pero si ambos departamentos compartieran sus datos entonces ésta disminuiría.

Un punto importante al compartir archivos es la habilidad de poseer un acceso eficiente de subconjuntos de interés, pues conforme aumentan el tamaño de la base de datos y el número de usuarios con diversas aplicaciones, se incrementa la tarea de recuperación que se hace cada vez más lenta.

Los inconvenientes de la redundancia no controlada son:

- El costo adicional de las copias múltiples de los mismos datos,
- La necesidad de múltiples operaciones para la actualización de por lo menos una parte de las copias redundantes.

Por otra parte la confiabilidad disminuye porque puede existir inconsistencia de la información, el formato o los nombres.

Sin embargo, existe la necesidad de armonizar el grado de redundancia con otras características deseables de la base como: rapidez de acceso, conveniencia de visión lógica, simplificación, metodología de direccionamiento, etc.

Por lo que el mejor criterio de diseño es tener una redundancia controlada o mínima que no sea perjudicial ni innecesaria.

II.2.3 VERSATILIDAD PARA REPRESENTAR RELACIONES.

Un objetivo importante de una base de datos es la capacidad de definir relaciones entre los diferentes registros lógicos, que deberán ser tan identificables como cualquier otro registro o atributo para poder definirse y manejarse sin ambigüedades. La forma esquemática de representar las relaciones que ligan o encadenan a estas entidades es mediante el uso de flechas sobre el diagrama de la base de datos. Entre las conveniencias de representar en forma de esquema las relaciones, se encuentra la fácil visualización de las interconexiones entre los diversos campos que conforman la base de datos. Por ejemplo:

Supongamos que deseamos una base de datos que contenga datos relativos a inventario, pedidos, proveedores y cuentas por pagar. Es evidente que existe una relación entre cada uno de ellos, ya que al solicitar un pedido también se afectará la cuenta de proveedores, cuentas por pagar e inventario al recibir la mercancía, etc. (ver figura 2.1)

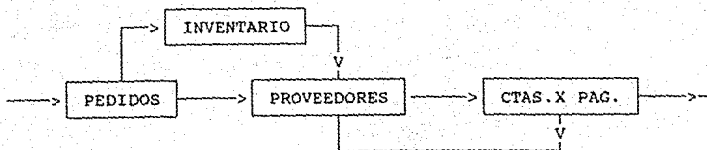


FIGURA 2.1. Diagrama relativo a movimientos contables.

II.2.4 INTEGRIDAD.

La integridad de una base de datos indica la ausencia de datos inconsistentes. La consistencia de ésta puede estar alterada momentáneamente mientras se está realizando una transacción o en forma definitiva, por daños en la base debido a diferentes causas. La base de datos, según se dijo, debe poder proporcionar información a diversos usuarios, pero debido a esto es imposible que cada uno de ellos sea responsable de la consistencia de los valores de la base y del mantenimiento de las relaciones entre los datos del usuario y el resto. Por esta razón una de las tareas principales de un sistema administrador de bases de datos es mantener la integridad realizando la coordinación global de la base que requiere de varias tareas:

- Actualización de los valores en las copias y valores dependientes,
- Coordinación del acceso de los datos por los diferentes programas de aplicación,
- La vigilancia de la consistencia, validez y calidad de los datos,
- Control de la redundancia y habilidad para compartir datos,
- Mecanismos de seguridad,
- Procedimientos de verificación y corrección, etc.

II.2.5 FLEXIBILIDAD DE ACCESO.

La flexibilidad de acceso se refiere a la capacidad de poder obtener los datos fácil y eficientemente de diversas maneras. Las propiedades que debe poseer son las siguientes:

- Habilidad para explorar y obtener información sin que haya sido previsto en la época del diseño,

- Habilidad para tener acceso a los datos en función a cualquier llave de acceso,
- Posibilidad de utilizar un lenguaje de consulta orientado a los no programadores para curiosear en el base de datos de manera inmediata,
- Capacidad para que un programa en algún lenguaje convencional de programación use la base y tenga acceso de manera eficiente a cualquier subconjunto de los datos,
- Habilidad para utilizar diversos mecanismos de control de acceso y administración de la base, etc.

II.2.6 SEGURIDAD.

Un elemento vital de cualquier empresa es la información almacenada. Mientras más importante es la información, mayor es la necesidad de protegerla contra su pérdida o robo, fallas de hardware o de software, catástrofes, acceso accidental o intencional por personas no autorizadas, la indebida destrucción o alteración, etc.

El tema de la protección de datos abarca dos aspectos:

- 1) Protección de privacidad y mantenimiento de la seguridad, y
- 2) Garantía del acceso a las personas que ejercen adecuadamente su privilegio.

La protección de privacidad y mantenimiento de la seguridad se entiende como la negación de acceso a las personas que no tengan derecho a ello.

Los tres tipos de elementos que se combinan para formar el sistema de protección son:

- 1 - Los usuarios con acceso a la base de datos,
- 2 - El tipo de acceso deseado y
- 3 - Los elementos a los que se realizará el acceso.

Los requisitos principales para mantener la seguridad de una base de datos son los siguientes:

- i) La base de datos debe estar protegida contra siniestros, como el fuego, robo, temblor y otras formas de destrucción, mediante acciones diversas como el mantenimiento de respaldos en cintas, discos, etc., en diferentes lugares.

ii) Los datos deben poderse reconstruir contra fallas de software y hardware. Esto se refiere a mantener una copia anterior de la base, una bitácora de modificaciones y rutinas adecuadas para la reconstrucción.

iii) La Base de datos debe permitir auditorias con el fin de evitar los fraudes o delitos.

iv) La seguridad de la base debe garantizar el acceso al usuario adecuado y negarlo a los que no están autorizados. Los programadores, por ingeniosos que sean, no deberían poder violar las medidas de seguridad.

v) Los sistemas deben tener un proceso de identificación positiva para permitir el acceso a la información, evitando las intrusiones malintencionadas.

vi) Una Base de datos debe tener un control sobre los privilegios de cada usuario, con el fin de verificar si la acción que desea realizar está autorizada.

vii) El sistema debe efectuar la supervisión de las acciones de los usuarios para detectar movimientos indebidos o erróneos.

II.2.7 RENDIMIENTO Y EFICIENCIA.

El rendimiento y la eficiencia se encuentran entre los requisitos importantes para el manejo de bases de datos de gran tamaño.

Las bases de datos que están diseñadas para trabajar en línea deben asegurar un tiempo de respuesta adecuado. Sin embargo, éste no tiene gran importancia para el procesamiento en lotes o fuera de línea.

El método de organización física de almacenamiento se elegirá teniendo en cuenta la mayor eficiencia de acuerdo al tipo de procesamiento.

La eficiencia en la búsqueda de información está estrechamente relacionada con los métodos para el direccionamiento de datos (forma en que el programa localiza un registro). Algunos lenguajes o sistemas operativos permiten el direccionamiento de forma indizada, otros solamente dan acceso de forma aleatoria o secuencial, esta última forma resulta lenta para la mayoría de los propósitos del procesamiento de datos, por lo que se recurre a métodos de direccionamiento más adecuados.

Las técnicas de direccionamiento conocidas son:

- Exploración o barrido del archivo (secuencial),
- Búsqueda por bloque,
- Búsqueda binaria,
- Archivos secuenciales indizados,
- Archivos no secuenciales indizados,
- Direccionamiento clave-igual-dirección,
- Algoritmos para conversión de clave,
- Dispersión (Hashing),
- Combinaciones de las anteriores, etc.

II.2.8 ADMINISTRACION Y CONTROL.

El administrador de base de datos, tiene a su cargo varias funciones como el control, la definición de la declaración y descripción de los datos, las guías para los usuarios, la determinación de los niveles de rendimiento y eficiencia que se consideren adecuados.

Para llevar a cabo estas funciones, el sistema administrador se vale de la utilización de descripciones especializadas de datos, lenguajes de control, lenguajes de control de dispositivos, lenguajes de descripción de esquemas, etc.

II.2.9 COSTO MINIMO.

A partir de los años 60, en los centros de cómputo la mayor parte del presupuesto se ha destinado al mantenimiento de programas, a expensas del desarrollo de capacidades adicionales para los sistemas existentes y de la implementación de otros. Por esto uno de los objetivos principales debe ser el mantenimiento de un bajo costo.

En consecuencia, antes de diseñar una base de datos, es necesario elaborar varios modelos con la finalidad de elegir la opción más apropiada, que proporcione un costo mínimo, una mayor eficiencia, irredundancia, consistencia, tiempo óptimo de respuesta, etc.

CAPITULO III

ORGANIZACIONES LOGICAS DE ARCHIVOS

III.1 SISTEMAS BASICOS DE ORGANIZACION DE ARCHIVOS.

El diseño adecuado de la organización de archivos está directamente relacionado con el tiempo de recuperación, la facilidad para efectuar actualizaciones y la economía del almacenamiento.

El análisis de la predicción es indispensable para la organización óptima del sistema de archivos.

Muchas técnicas mejoran la capacidad de recuperación de datos aumentando la redundancia, pero esto a su vez incrementa la cantidad necesaria de espacio y dificulta aun más las actualizaciones. Sin embargo, se puede encontrar un punto de equilibrio entre la facilidad para actualizar y el tiempo de recuperación.

Existe una gran cantidad de alternativas posibles de organización de archivos, ya que pueden obtenerse varios diseños básicos, técnicas de acceso y organizaciones híbridas de archivos.

Así, la elección del método de organización debe quedar determinada por la prioridad de los diversos criterios y los objetivos principales de la base.

Para llevar a cabo con éxito la elección es necesario conocer las diversas opciones. Por lo tanto en esta sección se expondrán brevemente los diseños básicos de las organizaciones de archivos.

Las organizaciones básicas de archivos que serán consideradas son las utilizadas con mayor frecuencia en la actualidad, y son: secuencial, indizado, directo y archivos multianillo.

III.1.1 SECUENCIAL

La característica principal de esta estructura es que los registros de datos están fijos y ordenados en una secuencia específica. Además todos los registros continen todos los valores de los atributos con el mismo orden, posición y tamaño. Por lo que los nombres de los atributos de los datos solamente se tienen que especificar una vez en la descripción del archivo. Así como definir una llave para cada uno de los registros con la finalidad

de proporcionar una secuencia específica. En caso de no definir una llave específica, la secuencia quedara determinada por el orden en que fueron almacenados. En la figura 3.1 se muestra un archivo secuencial en donde se observa que todos los registros contienen los mismos atributos y la clave les proporciona una secuencia específica.

CLAVE	NOMBRE	HORAS_TRAB	SUELDO
1	MARCO ANTONIO PEREZ	32	403102
2	LUIS MARTINEZ	40	570000
3	JOSE LOPEZ	20	243780
.			
.			
.			

FIGURA 3.1. Archivo Secuencial.

VENTAJAS :

- Eficiente para almacenar datos bien estructurados,
- Adecuado para procesamiento exhaustivo eficiente o por lotes,
- Efectivo para procesos ciclicos como nómina, etc.
- Es fácil generar archivos fijos a través de programas de procesamiento.
- Los programas para análisis de datos son fáciles de crear dado que la información se encuentra siempre en el mismo lugar.

DESVENTAJAS :

- Inflexible en su estructura,
- Difícil de actualizar,
- Consume demasiado tiempo en la búsqueda de registros únicos,
- Restringido por un conjunto limitado y predeterminado de atributos,
- Si se requiere agregar un nuevo atributo a un registro es necesario reorganizar todo el archivo, algunas veces para evitar esto lo que se hace es dejar un espacio disponible en la descripción del registro.
- Son difíciles de combinar con otros datos para generar cierta información.

En fin, los archivos secuenciales varían de excelentes a casi imposibles de manejar, dependiendo de la operaciones deseadas.

III.1.2 INDIZADO.

El archivo indizado, intenta superar algunos de los problemas de acceso que poseen los archivos secuenciales. Para lo cual, se auxilia de un archivo de índices de acuerdo a una cierta llave, que proporciona un acceso aleatorio al archivo, y un área de derrama, que permite manejar de forma eficiente adiciones al archivo.

La llave, un campo especial, constituye una manera de identificar a cada uno de los registros. Es un campo fijo reservado en todos los registros constituido por un conjunto de caracteres, alfanuméricos o numéricos, identificado en forma única. En caso de que un sólo atributo no pueda identificar de forma única al objeto, entonces será necesario especificar atributos secundarios o de orden inferior que lo determinen por completo, constituyendo así la llave. Generalmente al atributo primario de la llave se le denomina llave-primaria y a los otros se les conoce como llaves-secundarias.

CLAVE	NOMBRE	HORAS_TRAB	SUELDO
013-47	MARCO ANTONIO PEREZ	32	403102
128-15	LUIS MARTINEZ	40	570000
034-36	JOSE LOPEZ	20	243780
.			
.			
.			

Archivo de datos

CLAVE	APUNTADOR
013-47	1
034-36	3
128-15	2
.	
.	

Indice maestro

FIGURA 3.2. Archivo indizado

En la figura 3.2 se muestra un ejemplo de archivo indizado que está constituido de un índice maestro que contiene todas las claves ordenadas y un apuntador al archivo de datos, en el cual se encuentran todos los atributos relacionados con esa clave.

En algunos casos, aparte del archivo de índices y del archivo de datos, se reserva un área de derrama, un ejemplo de su funcionamiento se muestra en la figura 3.3.

	CLAVE	NOMBRE	HORAS_TRAB	SUELDO	APUNTADOR DERRAME
1	013-47	MARCO PEREZ	32	403102	1.1
2	128-15	LUIS MARTINEZ	40	570000	0
3	034-36	JOSE LOPEZ	20	243780	2.1
	.				
	.				

Archivo de datos

CLAVE	APUNTADOR
013-47	1
034-36	3
128-15	2
.	
.	

Índice maestro

	CLAVE	NOMBRE	HORAS_TRAB	SUELDO	APUNTADOR DERRAME
1.1	015-31	ALFREDO RIVERA	23	224102	3.1
2.1	036-13	JORGE ALVAREZ	32	345600	0
3.1	023-38	ANGELICA LUNA	33	456120	0
	.				
	.				

Área de derrama

FIGURA 3.3. Archivo indizado

En este caso el archivo de índices direcciona al archivo de datos y éste al área de derrama, la cual tiene como función agilizar el proceso de modificación e inserción.

Es importante notar que la forma de manejar el archivo de índices de los datos varía de acuerdo al software disponible, o en todo caso, al programador.

VENTAJAS :

- Conserva muchas de las ventajas descritas para los archivos secuenciales.
- Es factible insertar y recuperar registros individuales de manera que es necesario un número limitado de acceso de bloques, este tipo de archivo es adecuado para el acceso en línea.
- El acceso secuencial es relativamente simple y razonablemente eficiente.
- Se utiliza comunmente para efectuar consulta, con la restricción de especificar el atributo llave.

DESVENTAJAS :

- Para evitar la restricción de conocer el atributo llave para hacer una consulta se encuentran copias de los archivos pero con diferentes llaves por lo que el costo de realización y el espacio utilizado se incrementará.
- El archivo de índices ocupa espacio, sin embargo, cuando el archivo es pequeño casi no se tendrá pérdida de recursos en cuanto a espacio se refiere.
- Cuando las áreas de derrama son muy grandes se requiere una reorganización del archivo ya que los tiempos de recuperación o de procesamiento serial se vuelven excesivos.

III.1.3 DIRECTO

El archivo directo es diferente de los anteriormente expuestos. El acceso a los registros de este tipo de archivo es casi inmediato, mediante la utilización de una llave para localizarlo dentro del archivo que recibe el nombre de direccionamiento directo. Este tipo de acceso fue utilizado por primera vez en las máquinas electromecánicas de contabilidad que utilizaban un número perforado en una tarjeta con el fin de determinar el lugar donde debería archivarse el resto del contenido de la tarjeta. Actualmente los métodos de acceso directo transforman la llave mediante un algoritmo o función antes de utilizarla como dirección, a este tipo de métodos se les denomina funciones de Hashing.

VENTAJAS :

- El tiempo de acceso es mínimo, El archivo directo utiliza un cálculo para proporcionar la dirección del registro mientras que el indizado busca la llave en un archivo de índices, para determinar el registro correspondiente.
- El desempeño promedio de las transformaciones convenientes es tal que se obtienen resultados perfectamente buenos con métodos no tan perfectos.
- El acceso simple de recuperación es a través de una sola llave y no en serie. La organización del archivo directo es adecuada para el caso de registros pequeños y fijos, donde el acceso rápido es esencial y de forma simple.
- La adición, actualización y recuperación es muy fácil, ya que dada una llave inmediatamente se proporciona la localización del registro para actualizar, recuperar o añadir, sólo existirán problemas cuando se tengan colisiones.

DESVENTAJAS :

- El acceso está en función de un sólo atributo, como en el archivo secuencial indizado.
- El archivo de datos mantiene muchos registros vacíos, todo depende de la función o algoritmo utilizado.
- Una vez accesado un registro no se tiene la posibilidad de conocer el registro predecesor o sucesor.
- Los números de identificación de una persona u objeto se pueden necesitar en varios archivos. Si se utilizan diferentes algoritmos, pueden tener diferentes números llave provocando cierta confusión.
- Pueden existir colisiones, es decir, que diferentes objetos a través del algoritmo tengan asignado el mismo número de registro.
- La longitud de los registros es fija.
- El algoritmo de acceso directo depende del número de registros que serán almacenados. Si el número se incrementa el número de colisiones también aumentará por lo que se procederá a la reorganización del archivo.

III.1.4 CADENAS Y MULTIANILLOS

La organización de los archivos de cadena o multianillo está dirigida al procesamiento eficiente de subconjuntos de registros. Un subconjunto se define como un grupo de registros a todos aquellos que contengan algún valor común de atributo. Los subconjuntos de registros se encadenan mediante el uso de apuntadores que definen algún orden para los miembros del subconjunto.

Un registro puede ser miembro de muchos subconjuntos. Cada subconjunto tiene un registro denominado "Cabeza" que es el origen de la cadena. Un registro cabeza contendrá información referente a todos sus registros subordinados. Este registro

cabeza para conjuntos de subconjuntos también puede eslabonarse en una cadena. Un tipo particular de cadena es el anillo, cadena en la que el campo apuntador del último miembro se utiliza para apuntar al registro que encabeza la cadena.

La cadena consiste en un conjunto de registros relacionados por una serie de apuntadores. El registro M tiene un apuntador hacia el L y éste, a su vez, apunta a el P y así sucesivamente, como se muestra en la figura 3.4.

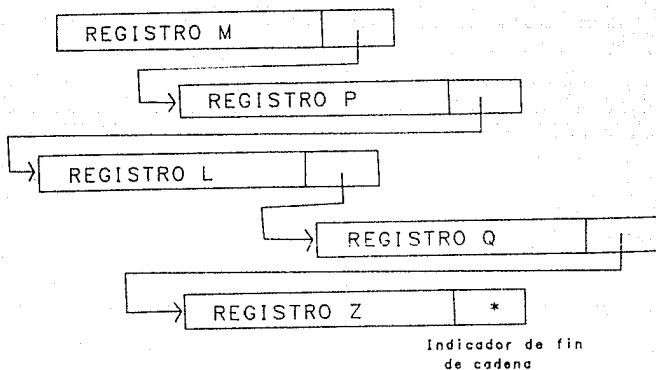


FIGURA 3.4. Registros encadenados.

Existen una gran variedad de tipos de cadena como : la cadena simple unidireccional, cadena unidireccional con apuntador de cola, cadena bidireccional, anillo simple unidireccional y multianillos.

A continuación se enunciarán las ventajas y desventajas de las cadenas y anillos de una forma general.

VENTAJAS :

- La recuperación de registros sucesores es más económica que con los archivos indizados.
- La recuperación de elementos que pertenecen a una cierta categoría tiene una gran efectividad en este tipo de archivos.

DESVENTAJAS :

- El costo de seguir las cadenas aumenta linealmente de acuerdo con su tamaño.
- En la búsqueda de algún elemento en particular, el número de registros leídos es mucho mayor que en los indizados.
- Para el diseño de un sistema de archivos multianillos es necesario conocer de antemano los datos y el uso que se les dará.
- El costo para la búsqueda exhaustiva es alto debido a que se deben de seguir los enlaces dados en los registros.
- Conforme se incrementa el número de anillos la complejidad del archivo aumenta.
- Para el seguimiento de cadenas largas se requiere de una gran cantidad de operaciones de lectura.
- Para la exploración es necesario recorrer toda la cadena cuya lectura de los registros es superflua.

Existen otras estructuras parecidas que suelen denominarse listas en árbol o multilistas y que pueden implementarse con anillos o cadenas simples.

En conclusión, la efectividad de un sistema multianillo dependerá de su adecuada asignación de atributos a los anillos.

III.2 TECNICAS DE ACCESO

Según la naturaleza de la aplicación, un programa podrá efectuar todas o sólo algunas de las siguientes funciones: consulta, modificación, inserción y borrado; sobre un archivo al que realice el acceso.

El método de acceso es la forma en que se recuperan los datos en base a una cierta organización específica de archivo.

III.2.1 EXPLORACION O BARRIDO DEL ARCHIVO.

El enfoque común para recuperar un registro de un archivo secuencial consiste en una búsqueda secuencial a través del archivo. Este método es lento para la mayoría de los propósitos del procesamiento de datos.

Debido a que la búsqueda de un registro específico es de forma secuencial el tiempo de recuperación de un registro en el mejor de los casos sería de 1 lectura o en el peor de los casos se tendrían que leer todos los N registros. Es por esto, que el tiempo promedio para recuperar un determinado registro, en donde N es el número total de registros, sería aproximadamente proporcional a

III.2.2 BUSQUEDA POR BLOQUE

Cuando un archivo está organizado secuencialmente de acuerdo a una clave. Los registros se agrupan en bloques y se hace una búsqueda secuencial del bloque en donde se debe encontrar el registro buscado. Posteriormente en este bloque se llevará a cabo una búsqueda secuencial.

Para hacer una estimación del tiempo promedio de búsqueda, supóngase que el número total de datos es N y que el tamaño de los bloques es raíz de N, entonces para conocer el tiempo promedio deberá sumarse el tiempo promedio necesario para localizar el bloque adecuado y el tiempo promedio para localizar el registro especificado dentro del bloque.

Como el número total de registros es N y el tamaño de cada bloque es raíz de N, entonces el número de bloques es N entre raíz de N, de aquí que el tiempo de búsqueda para localizar el bloque adecuado es

$$\frac{N}{2\sqrt{N}}$$

Por otra parte, el número total de registros en un bloque es raíz de N, entonces el tiempo promedio de búsqueda para localizar el registro adecuado dentro de un bloque es

$$\frac{\sqrt{N}}{2}$$

Por lo tanto, si el número total de datos es N y el tamaño del bloque es raíz de N, entonces el tiempo total promedio de búsqueda por bloque es

$$\frac{N}{2\sqrt{N}} + \frac{\sqrt{N}}{2} = \sqrt{N}$$

búsqueda por
bloque

búsqueda
secuencial

tiempo total de
localización

III.2.3 BUSQUEDA BINARIA

La búsqueda binaria comienza con un acceso al centro del archivo y compara la clave de éste con la del registro buscado. De esta manera según el resultado el área de búsqueda quedará dividida por 2 partes y el proceso se repetirá tantas veces como sea necesario.

El número de comparaciones es, en promedio, aproximadamente el logaritmo en base dos del número total de datos. Este número es menor que en la búsqueda por bloques.

La búsqueda binaria es adecuada en el caso de búsqueda en un archivo de índices.

III.2.4 BUSQUEDA SECUENCIAL INDIZADA.

La búsqueda secuencial se efectúa en la tabla de índices la cual es más pequeña que el archivo.

Una vez que se ha encontrado el índice correcto se hace una segunda búsqueda secuencial solamente en la parte reducida de la tabla que contiene los registros.

La ventaja real de este método es que los elementos de la tabla pueden ser examinados en forma secuencial si todos los registros en el archivo deben ser accedidos. La inserción es difícil debido a que puede no existir espacio entre dos entradas en la tabla, siendo necesario mover un gran número de elementos.

III.2.5 DIRECCIONAMIENTO CLAVE - IGUAL - DIRECCION

Estos métodos consisten en la conversión directa del valor de una clave en una dirección de archivo.

La manera más simple de resolver el problema del direccionamiento consiste en incorporar a la transacción de entrada la dirección de máquina relativa del registro en cuestión.

La ventaja de este método es que permite un direccionamiento rápido sin tener que recurrir a operaciones de exploración en el índice o en el archivo.

III.2.6 ALGORITMO PARA CONVERSION DE CLAVE.

Este consiste en el empleo de un algoritmo para convertir la clave en una dirección de máquina. En algunas ocasiones es posible calcular la dirección a partir de los identificadores de entidad. La ventaja que presenta es el direccionamiento rápido. Sin embargo, rara vez se producirá un conjunto continuo de direcciones, por lo que podría haber localidades sin uso.

III.2.7 FUNCION DE DISPERSION.

La función de dispersión convierte la clave del registro en un número casi aleatorio que sirve después para determinar la localidad de almacenamiento del dato. Es una técnica útil y eficiente para la inserción y eliminación. Pero el aprovechamiento del espacio puede ser menor que con los métodos de indización. Los registros no se hallan en una secuencia adecuada para el procesamiento por lotes.

III.3 ESTRUCTURAS DE ORGANIZACIONES LOGICAS DE ARCHIVOS.

Existe una gran variedad de diseños de archivos a partir de las estructuras básicas descritas en la sección III.1. Estos diseños pueden satisfacer requerimientos que no poseen las estructuras básicas.

Los diferentes diseños también son llamados organizaciones de archivos híbridos debido a que son una combinación de las estructuras básicas. Estos son :

- Archivos Planos,
- Archivos simples,
- Estructuras multinivel de índices,
- Aplicación de un archivo secuencial indizado,
- Archivos con estructura de árbol,
- Datos estructurales jerárquicamente,
- Métodos basados en el acceso directo,
- Datos almacenados en forma de red.
- Opciones de organización de anillos complejos,
- Archivos que emplean almacenamiento virtual,
- Archivos fantasma,

En esta sección nos ocuparemos de explicar sólo los modelos que consideramos de interés para el presente trabajo, que son los archivos planos, estructuras de árbol y archivos jerárquicos.

III.3.1 ARCHIVOS PLANOS

Generalmente la forma de relacionar datos con sus atributos consiste en almacenarlos de una manera fija y predeterminada. Es necesario determinar una sola vez los nombres de los atributos que pertenecen al registro, ya que todos los registros son homogéneos.

El primer campo contiene el identificador del registro. Los siguientes campos contienen los datos de los atributos correspondientes a ese identificador.

Para definir completamente la estructura del archivo plano es necesario recurrir a una forma de codificación que represente los valores de los atributos. Los tipos más comunes de representación son :

- Caracteres alfanuméricos,
- Números Enteros,
- Números Reales, con punto fijo o punto flotante,
- Tipo lógico, verdadero o falso.

La especificación del formato de definición varía de acuerdo al lenguaje que se utilice.

A continuación se mostrará un ejemplo:

atributos : No. Cta. Nombre del alumno Edad Sexo
 Código : N 7 A 18 N 2 A 1

Archivo Plano	3478532	Eduardo Jiménez	18	M
	4598734	Yolanda López	19	F
	3957446	Laura Aristos	20	F
	3987457	Héctor Pastrana	18	M

↓
 v
 identificador

donde

- N 7 = es un número entero con 7 dígitos,
- A 18 = es de 18 caracteres alfanuméricos,
- N 2 = es un número entero con 2 dígitos,
- A 1 = es de 1 carácter alfanumérico,

FIGURA 3.5. Representación de un archivo plano.

Los nombres de los atributos y la codificación no se almacenan en el archivo, aunque deben almacenarse en alguna parte. Usualmente estas definiciones se encuentran en el diccionario de datos disponibles para consultarse.

La organización de este archivo, como puede verse, es muy parecida a la del archivo secuencial, pero no posee una secuencia específica.

III.3.2 ESTRUCTURAS DE ARBOL

Este tipo de estructura se ha utilizado por mucho tiempo para la descripción y organización tanto lógica como física de los datos. Las descripciones lógicas tienen la finalidad de describir las relaciones entre tipos de segmentos o tipos de registro. Las organizaciones físicas de datos tienen el objeto de describir conjuntos de apuntadores y relaciones entre las entradas y los índices.

Un árbol está compuesto de nodos ordenados de acuerdo a una jerarquía. La raíz es el nodo más alto de la jerarquía y es único.

Todo nodo está relacionado con otro nodo en el siguiente nivel superior llamado antecesor o padre, con excepción de la raíz. Ningún elemento puede tener más de un padre. Cada nodo puede estar relacionado con uno o varios en el nivel inmediato inferior llamados sucesores o hijos. Rama es la conexión entre el padre y el hijo. Los nodos que no tienen hijos, es decir, que se encuentran en la punta de las ramas se llaman Hojas. Esto se puede ver claramente en la siguiente figura.

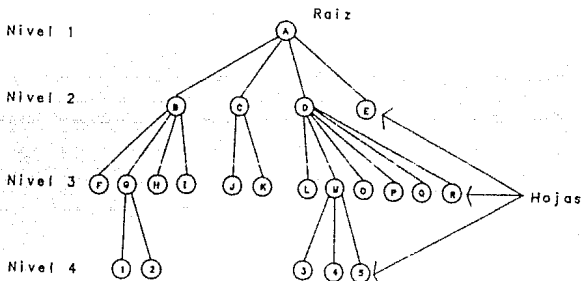


FIGURA 3.6. Representación de un árbol.

Como se mencionó, las estructuras de árbol sólo permiten que un nodo provenga de un padre. Sin embargo, una organización que permite que un nodo tenga más de un antecesor es llamada estructura de Red. Cualquier estructura de red puede transformarse en uno o varios árboles por medio de la introducción de redundancia de nodos.

III.3.3 ARCHIVOS JERARQUICOS.

Se definen como archivos jerárquicos aquellos que tienen relaciones de tipo árbol entre sus registros.

El uso de las estructuras de tipo de árbol es adecuado cuando los datos poseen una organización jerárquica natural. Ejemplos clásicos de estas estructuras son: la información censal, datos demográficos, un árbol genealógico, una jerarquía organizacional, etc.

En la figura 3.7 , se muestra el organigrama de una empresa, en donde se puede observar una jerarquía implícita, ya que todas las direcciones dependen de la dirección general.

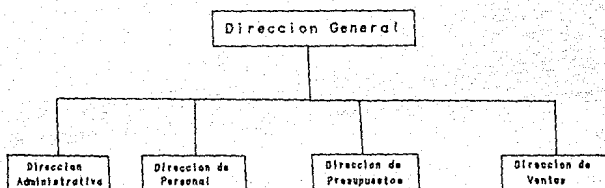
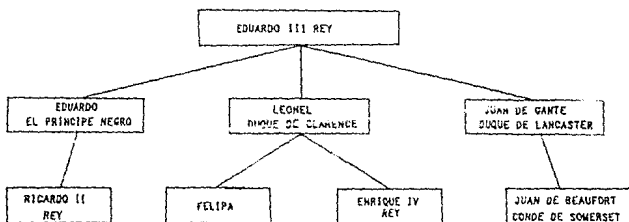


FIGURA 3.7. Arbol jerárquico.

Los archivos jerárquicos se pueden clasificar de acuerdo a su estructura: homogénea o heterogénea. A los archivos que contienen un tipo de registro diferente en cada nodo se les identifica como árboles heterogéneos, mientras que los homogéneos son aquellos que tienen el mismo tipo de registro con profundidad variable.

Un árbol genealógico contiene en cada uno de los niveles el mismo tipo de registro, que ejemplifica un árbol homogéneo, como puede observarse en la siguiente figura.



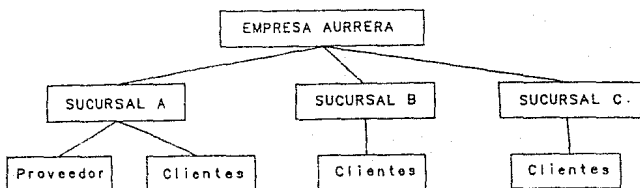
REGISTROS :

Todos contienen los mismos campos : Nombre, Fecha de Nacimiento, Lugar de nacimiento, Fecha de fallecimiento.

FIGURA 3.8. Arbol jerárquico homogéneo.

La mayoría de los programas (software) han sido diseñados para bases de datos con árboles heterogéneos de profundidad preestablecida.

En la figura 3.9 se muestra un ejemplo de árbol jerárquico correspondiente a una empresa y la información que se almacena en cada caso. Puesto que no todos los nodos contienen la misma información, éste se considera como un árbol heterogéneo.



REGISTROS :

Empresa AURRERA : Nombre, Dirección, Teléfono.

Sucursal A,B,C : Dirección, Teléfono, Gerente.

Proveedor : Tipo de producto, Nombre, Teléfono, Crédito.

Clientes : Nombre, Dirección, Teléfono, Crédito, Saldo.

FIGURA 3.9. Arbol jerárquico heterogéneo.

Es conveniente mencionar que las estructuras de árboles y redes se pueden descomponer en grupos de archivos planos con campos redundantes.

CAPITULO IV

CARACTERISTICAS DE UN ADMINISTRADOR DE BASES DE DATOS

Un sistema administrador de bases de datos es un intermediario, ya que el medio ambiente de procesamiento sirve como interface entre los archivos y las personas que buscan datos en ellos. Sin importar la forma en que un usuario solicite sus datos de interés, la solicitud y la respuesta pasarán a través del sistema administrador de bases de datos.

Un sistema administrador de bases de datos es un elemento complejo de software que especifica la forma en que los datos pueden estructurarse, controla todos los accesos a éstos y proporciona algunos otros servicios relacionados con el manejo de datos.

Por lo general, los sistemas administradores de bases de datos se encuentran bajo el control de un sistema operativo, encontrarse instalados en una computadora dedicada exclusivamente a ellos o compartir una unidad central de procesamiento con otras facilidades de software. Un administrador de bases de datos y sus programas asociados de aplicación pueden operar por lote o en un medio ambiente multiterminal en línea utilizando facilidades adicionales de comunicación de datos.

Los elementos de un administrador de bases de datos son:

- El modelo del administrador de base de datos,
- Lenguaje de descripción de esquema,
- Lenguaje de consulta,
- Algebra relacional, y
- Cálculo relacional.

IV.1 MODELOS DE BASES DE DATOS.

Cualquier sistema administrador de base de datos específico se basa en una estructura de datos y en un mecanismo de acceso a dichos datos.

A través de los años se han desarrollado diversos administradores de bases de datos para medios ambientes que van desde las grandes computadoras hasta los más pequeños microprocesadores. Sin embargo, cada uno de ellos se basa en

cinco enfoques fundamentales que son:

- Modelo Relacional,
- Modelo Jerárquico,
- Modelo de Red,
- Modelo Seudorelacional, y
- Modelo de Tablas invertidas.

Para los intereses del presente trabajo sólo se explicarán los tres primeros.

IV.1.1 MODELO RELACIONAL.

Una base de datos relacional está construida con tablas. Donde cada columna de la tabla, es un conjunto de valores de un mismo tipo que constituyen por definición un dominio. La columna j -ésima es el dominio j -ésimo (D_j) de la relación. En el lenguaje matemático, una relación se define como un subconjunto de un producto cartesiano de dominios D_1, D_2, \dots, D_n .

Por lo tanto, debido a que las tablas son básicamente relaciones, una base de datos construida con éstas es una base de datos relacional.

Se entenderá por tabla una matriz bidimensional con las siguientes características:

- a) Cada entrada de la tabla representa una llave o característica.
- b) Son homogéneas por columna.
- c) Cada columna tiene nombre propio.
- d) Todas las columnas son diferentes.

Así a una tabla como la de la Figura 4.1 la llaman relación. Una base de datos relacional, a primera vista parece que consiste en unos cuantos archivos lineales independientes, en realidad esto es cierto desde un punto de vista estructural.

Obsérvese que los campos "oficina filial", "automóvil", "empleado" y "fecha de mantenimiento" (para un determinado automóvil) son únicos, existe sólo un registro para una oficina filial específica en el archivo de filiales, uno para un cierto automóvil en el archivo de automóviles, y así sucesivamente. Esto, desde luego, significa que no hay redundancia entre los campos no llave; ya que existe sólo un registro para una oficina filial específica, todos los campos que dependen de él, como "gerente filial", "dirección filial", etc., aparecen sólo una vez.

Sin embargo, el campo "oficina filial" aparece en el archivo de automóviles y en el archivo de filiales, lo que representa un cierto grado de redundancia. Pero, en este caso, es una instancia necesaria de redundancia. Y como los tipos de registro o archivos

en la base de datos relacional no están interconectados en forma estructural como en los enfoques jerárquicos, la única manera de relacionar los archivos es repitiendo el campo llave de un archivo en el otro.

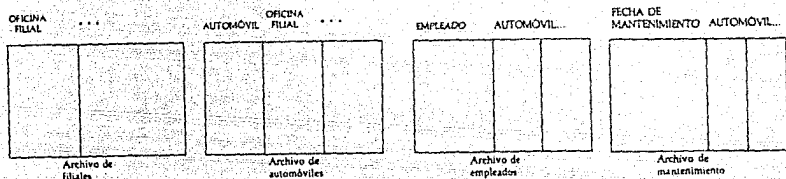


FIGURA 4.1. Modelo relacional.

IV.1.2 MODELO JERARQUICO.

Una base de datos Jerárquica consiste en un conjunto de registros que se conectan entre sí mediante enlaces, donde cada enlace representa una relación uno a muchos, en dirección descendente. Un registro es un conjunto de campos o atributos, cada uno de los cuales contiene un sólo valor.

Estos modelos tienden a ser fáciles de seguir si coinciden con el modelo del usuario.

El uso de este modelo es adecuado si las rutas entre atributos pueden determinarse sin ambigüedades.

Una estructura jerárquica con varios tipos de registros, se muestra en la figura 4.2. Los cuatro tipos de registros representan las oficinas filiales de una empresa, los automóviles asignados a cada una, los empleados autorizados a conducir un automóvil específico y las fechas en que se les dió mantenimiento.

Cada cuadro representa un conjunto de campos, que son:

- OFICINA FILIAL : número de oficina, nombre de oficina filial, ciudad de la oficina filial, etc.
- AUTOMOVIL : automóvil, fecha de adquisición, etc.
- EMPLEADO : nombre del empleado autorizado, etc.
- FECHA DE MANTENIMIENTO : fecha de mantenimiento, lugar, etc.

De esta forma la figura indicaría que cada oficina filial tiene asignado algunos automóviles, por cada uno de ellos hay varios empleados autorizados a conducirlos, y cada automóvil tiene un historial de múltiples fechas de mantenimiento.

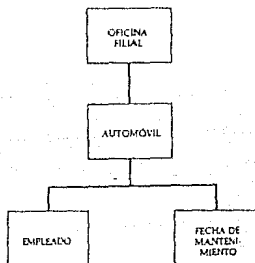


FIGURA 4.2. Modelo Jerárquico.

IV.1.3 MODELO DE RED.

Una red es un tipo muy general de estructura representada por un conjunto de puntos interconectados de alguna forma. El concepto de árbol jerárquico coincide con esta definición, pero con ciertas restricciones, es decir, un árbol es sólo un tipo limitado de red.

Una base de datos de Red consiste en un conjunto de registros que se conectan entre sí mediante enlaces, donde cada enlace representa una relación de muchos a muchos, en cualquier dirección.

La figura 4.3 muestra un ejemplo que puede considerarse como una red. Se pueden observar tres conjuntos: Universidad, equipo y estado origen, que muestran a todos los jugadores de la liga que provienen de una universidad específica, que juegan en un equipo, y que nacieron en un estado determinado, respectivamente. En esta estructura, los tres conjuntos tienen el mismo tipo de registro miembro "jugadores". Esto muestra que el registro "jugadores" tiene tres tipos de registros "padre". De esta forma la estructura no es un árbol sino una red más general.

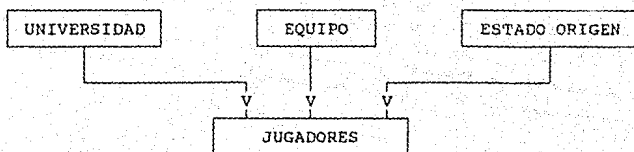


FIGURA 4.3. Un ejemplo de equipos y jugadores en forma de red.

IV.2 LENGUAJE DE DESCRIPCION DE ESQUEMA.

Para describir de una manera formal la organización de los datos, se utiliza el lenguaje de descripción de esquemas. Mediante éste se especifica la forma en que los datos deben almacenarse.

El esquema es un diagrama de los tipos de datos que se usan. Proporciona los nombres de las entidades y sus atributos, así como la especificación de las relaciones que existen entre ellos.

El resultado de la compilación de las proposiciones del lenguaje de esquema es un conjunto de tablas que se almacenan en un archivo especial llamado diccionario o directorio de datos.

El directorio de datos es un archivo que contiene metadatos, es decir, "Datos acerca de los datos". Este archivo será consultado para la lectura o modificación de los datos reales por el sistema de bases de datos.

Por ejemplo, supongamos que las relaciones o tablas bidimensionales que se desean definir son las siguientes:

CLAVE	NOMBRE	HORAS_TRAB	SUELDO
1	MARCO ANTONIO PEREZ	32	1,403,102
2	LUIS MARTINEZ	40	570,000
3	JOSE LOPEZ	20	2,243,780
.			
.			
.			

Relación de empleados

CLAVE_P	MONTO	FECHA	AÑOS
1	30,000,000	23-01-1989	10
3	45,000,000	01-03-1990	15
7	22,000,000	01-08-1987	10
.			
.			
.			

Relación de Préstamos

FIGURA 4.4. Ejemplos de relaciones.

Entonces un esquema para definir este tipo de relaciones es:

RELACION:	EMPLEADOS
CLAVE	ENTERO
NOMBRE	20 CARACTERES
HORAS_TRAB	ENTERO
SUELDO	REAL
RELACION:	PRESTAMOS
CLAVE_P	ENTERO
MONTO	REAL
FECHA	10 CARACTERES
ANOS	ENTERO

FIGURA 4.5. Ejemplo de esquema para algunas relaciones.

IV.3 LENGUAJE DE MANIPULACION DE DATOS.

Al mecanismo empleado para el manejo y recuperación de datos almacenados en una base se le denomina lenguaje de manipulación de datos.

Existen dos diferentes formas básicas :

- 1) Comunicación entre programas de aplicación y el administrador de base de datos.
- 2) Comunicación entre el usuario y el administrador de bases de datos.

PROPOSICIONES INTEGRADAS

Las proposiciones integradas constituyen la interface entre el programa de aplicación y el sistema administrador de bases de datos. En este caso, el programa de aplicación emite instrucciones al administrador para interactuar con la base de datos.

LENGUAJE DE CONSULTA

El lenguaje de consulta constituye la interface entre el usuario y el sistema administrador de bases de datos. Este actúa de forma interactiva, es decir, acepta comandos tecleados por el usuario y, tras darles respuesta, queda en espera de nuevos comandos.

Las instrucciones o comandos que normalmente se utilizan para cualquiera de estos dos casos son: abrir, cerrar, altas, búsqueda, modificar, insertar, remover, almacenar, suprimir, guardar y ordenar.

IV.4 ALGEBRA RELACIONAL.

Un lenguaje de consulta de procedimientos basado en el álgebra relacional, es una colección de operaciones que se efectúan sobre relaciones y producen relaciones como resultado.

Con el fin de obtener un resultado determinado, el usuario especificará cierta secuencia de operaciones de esta clase.

Existen 6 operaciones fundamentales en el álgebra relacional que son:

- ELEGIR relación1 valor1 comparador valor2
Produce una nueva relación a partir de la relación1 que contiene sólo a los que cumplieron con la condición.
- PROYECTAR relación1 atributo-1 ... atributo-n
Produce una nueva relación a partir de la relación1, quitando de cada registro cualquier atributo no mencionado.
- UNIR relación1 relación2 atributo1 comparador atributo2
Produce la relación consistente de todos los elementos que se encuentran en la relación1 o en la relación2 y que cumplen con la condición.
- INTERSECTAR relación1 relación2
Produce una relación que consiste de aquellos que se encuentran en ambas relaciones.

- DIFERENCIA-CONJUNTOS relación1 relación2

Son aquellos que se encuentran en relación1 pero no se encuentran en relación2.

- PRODUCTO CARTESIANO relación1 relación2

Forman una nueva relación hecha de todos los registros de relación1 y relación2.

La mayoría de estas operaciones son fáciles de entender. Sin embargo para que se puedan comprender mejor los conceptos a continuación se presenta un ejemplo del álgebra relacional.

Supongamos que utilizamos las relaciones definidas en la figura 4.5 y la información que se desea consultar es la que está indicada en la figura 4.6 .

1. "Todos los empleados con sus datos, que han trabajado tiempo completo y medio tiempo, es decir, 40 o 20 horas respectivamente."

2. "Lista todos los nombres de los empleados que han pedido préstamos menores a \$40,000,000."

FIGURA 4.6. Ejemplos de objetivos de consulta de bases de datos.

La manera en que debe utilizarse el álgebra relacional para obtener los resultados deseados es la siguiente:

1. R1 <- ELEGIR EMPLEADOS HORAS_TRAB = 40
R2 <- ELEGIR EMPLEADOS HORAS_TRAB = 20
RESPUESTA <- UNIR R1 R2
2. R1 <- UNIR EMPLEADOS PRESTAMOS CLAVE = CLAVE_P
R2 <- ELEGIR R1 MONTO < 40000
RESPUESTA <- PROYECTAR R2 NOMBRE

En resumen, la búsqueda mediante el uso del álgebra relacional es a través de una secuencia de operaciones, que sólo permiten al usuario pensar en términos de relaciones para obtener el resultado deseado.

IV.5 CALCULO RELACIONAL.

Un lenguaje de consulta basado en el calculo relacional permite la descripción formal de la información deseada sin especificar como obtenerla.

Son muchas las formas de adoptar el calculo relacional, una forma particular de este se muestra a continuación.

ENCONTRAR variable[atributo] tal que <condiciones>

Donde las posibles expresiones de las condiciones son:

<valor> = <valor>

<valor> > <valor>

<valor> < <valor>

<expresion> Y <expresion>

<expresion> O <expresion>

NO <expresion>

PARA TODA (<variables>) <expresion>

PARA ALGUNA (<variables>) <expresion>

MIEMBRO (<variables> , <relacion>)

Un lenguaje de consulta basado en el calculo relacional presenta varias ventajas sobre el lenguaje basado en el algebra relacional.

VENTAJAS:

- 1.- El usuario no declara nada acerca de como el sistema se las arreglara para obtener el resultado deseado, por lo tanto el sistema esta en libertad para optimizar el metodo.
- 2.- Requerir datos por sus propiedades es más natural para el usuario que requerirlos mediante la especificación de una serie de operaciones.

DESVENTAJAS:

- 1.- Es más difícil elaborar el software necesario para implementar el lenguaje de consulta de procedimientos basado en el calculo relacional que en el algebra relacional.

Basados en el problema planteado por las definiciones de la figura 4.5 y en los objetivos de consulta de la figura 4.6, se utilizará el cálculo relacional para resolverlo.

1. ENCONTRAR X[EMPLEADOS]
TAL QUE
PARA ALGUNA (X)
(X[HORAS_TRAB] = 40 OR
X[HORAS_TRAB] = 20)

2. ENCONTRAR X[NOMBRE]
TAL QUE
PARA ALGUNA (X,Y)
(MIEMBRO (X, EMPLEADOS) AND
MIEMBRO (Y, PRESTAMOS) AND
Y[MONTÓ] < 40000)

A diferencia del álgebra relacional, con el cálculo relacional existen muchas maneras de escribir las búsquedas, además de proporcionar mayor claridad de lo que se desea obtener.

Como el usuario no tiene que especificar la manera de obtener el resultado, al lenguaje de consulta basado en el cálculo relacional se le conoce como "un lenguaje sin procedimientos".

CAPITULO V

EL SISTEMA ADMINISTRADOR DE BASES DE DATOS SUPER-BASE

El Sistema de Super-Base fue diseñado para funcionar en computadoras personales compatibles con I.B.M. debido a su bajo costo, amplia comercialización y accesibilidad.

El lenguaje de programación de alto nivel elegido para el desarrollo del sistema fue TURBO PASCAL. Debido a que es un lenguaje estructurado que ofrece un buen ambiente de trabajo para programas grandes.

El manejador de bases de datos Super-Base es un avanzado medio ambiente de programación relacional y jerárquica para la especificación formal de diversas aplicaciones.

El sistema fue concebido originalmente para el manejo de información censal. Sin embargo, también es adecuado para las necesidades de aplicaciones en las áreas de información geográfica, económica, demográfica y todas aquellas que posean características jerárquicas.

V.1 EL MODELO RELACIONAL-JERARQUICO DE SUPER-BASE.

El sistema es un nuevo concepto de administrador de bases de datos que toma las características de los modelos relacional y jerárquico.

La representación de almacenamiento jerárquico es mediante esquemas de árboles. Es posible la existencia de múltiples árboles, es decir, de un bosque. Sin embargo, estos árboles no pueden estar entrelazados a través de sus ramas. (ver fig 5.1)

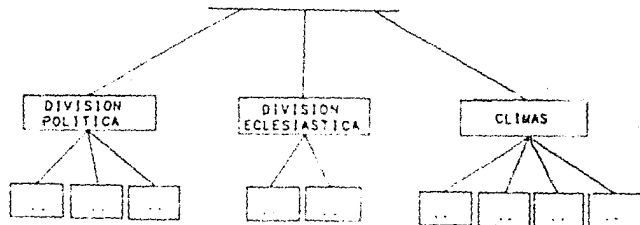


FIGURA 5.1. Representación de un bosque.

El usuario del sistema no necesita tener acceso directo a estos enlaces. El acceso a la jerarquía lógica es mediante la utilización de un comando específico que permite "navegar" a lo largo de la base de datos. La definición de rutas de acceso implicada por la estructura, significa que estas rutas no tienen que crearse durante el procesamiento de consultas o de actualización, si no que ya existen cuando se necesitan.

Los nodos, a diferencia del modelo estrictamente jerárquico, tienen una estructura diferente, ya que además de guardar el nombre de los miembros almacenan sus características. Esto es mediante tablas bidimensionales o archivos planos. En donde la organización deberá ser de tal forma que :

- La primera columna contendrá la llave de acceso, que llamaremos objeto.
- Cada una de las demás columnas representa una determinada característica del objeto, que llamaremos variable.
- Los objetos no se pueden repetir.
- Los nombres de las variables no se pueden repetir.
- Cada renglón de la tabla representa un objeto determinado con sus características.
- Todos los objetos tienen las mismas variables.

En la figura 5.2 se muestra la estructura de este modelo. Como se puede notar existe una estructura jerárquica donde cada nodo es una tabla bidimensional. Puede observarse que es semejante a hojear los libros del censo en donde el primer nivel representa un resumen de la información total del país, y cada uno de los niveles inferiores representan la información detallada de cada uno de los estados y así sucesivamente. La ventaja de este tipo de administrador de bases de datos es el hecho de poder consultar y procesar información automáticamente de acuerdo a los puntos de interés, dado que los datos conservan su relación original permitiendo, además, la consistencia de la base de datos. Por otra parte, el sistema administrador de bases de datos Super-Base permite reflejar gráficamente la información almacenada.

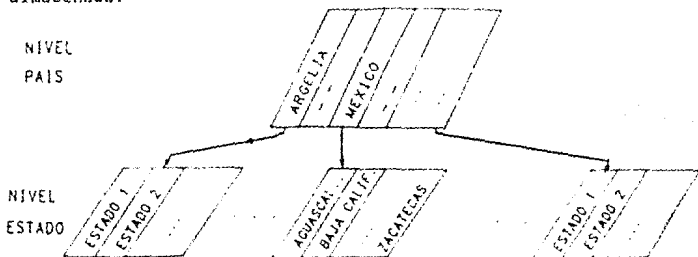


FIGURA 5.2. Esquema para un árbol jerárquico-relacional.

V.2 LENGUAJE PARA LA DESCRIPCION DE ESQUEMAS.

Es indispensable que todo administrador de bases de datos tenga la capacidad para describir sus datos con precisión, especificando sus estructuras lógicas. A tal descripción se le denomina esquema y al lenguaje empleado para describirlo se le llamara lenguaje de esquema.

Un esquema define la estructura inicial de la base de datos y la pone a la disposición del usuario.

Es necesario que el lenguaje de esquema pueda especificar los tipos de datos, su organización, sus relaciones y la llave de búsqueda. Actualmente no existe un lenguaje de esquema universal.

El lenguaje para la descripción del esquema de Super-Base es un lenguaje amigable al usuario, que provee de un medio ambiente de programación fácil para usar. El lenguaje de esquema puede incorporar los tipos elementales estandares de datos como son: numérico, cadena de caracteres, lógico, vectores numéricos, vectores de cadenas de caracteres. Además de los tipos clase, coordenada y trazo.

El tipo clase sirve para definir la estructura jerárquica de los datos dentro de la base. El tipo coordenada es una pareja de números reales donde se puede almacenar la coordenada geográfica de un determinado objeto. El tipo trazo es una colección de puntos que unidos a través de rectas delimitan un objeto dado. La creación de estos tipos fue hecha para que el sistema pudiera funcionar como una herramienta en un paquete de mapas por computadora llamado MICROMAP.

El lenguaje para la descripción de datos de Super-Base soporta los siguientes tipos :

- 1 - Clase
- 2 - Cadena de caracteres
- 3 - Numero
- 4 - Vector cadenas de caracteres
- 5 - Vector numerico
- 6 - Coordenada
- 7 - Trazo
- 8 - Lógico

Además de que se pueden especificar ciertas relaciones entre variables. (ver sección V.6)

Un ejemplo de la definición del esquema de una base de datos se muestra en la figura 5.3. El esquema define tres niveles jerárquicos "PAIS", "ESTADO" Y "MUNICIPIO". Las variables de estudio a nivel "PAIS" son : capital, población y límite del país, por otra parte, se puede observar que la población se define como la suma de la población total de cada uno de los

estados. A nivel "ESTADO" las variables son : capital, poblacion total, poblacion femenina, poblacion masculina y el limite estatal, donde tambien se define que la poblacion del estado es igual a la suma de la poblacion de los municipios. Por ultimo a nivel "MUNICIPIO" se tienen las variables : poblacion total, poblacion femenina, poblacion masculina, localizacion y el limite municipal, en donde la poblacion total esta definida como la suma de la poblacion femenina y la masculina. Por otra parte, la localizacion del municipio esta directamente relacionada con su coordenada geografica que se utiliza a través de MICROMAP, de la misma forma que las variables limite permitiendo su posterior consulta y asurado.

```

PAIS          : CLASE ; B_PAIS
CAPITAL       : STRING[20]
POBLACION     : NUMERO[9:0] = SUM(ESTADO.POBLACION_T)
LIMITE        : TRAZO ; LPS
ESTADO        : CLASE ; B_ESTADO
CAPITAL       : STRING[20]
POBLACION_T  : NUMERO[9:0] = SUM(MUNICIPIO.POBLACION_T)
POBLACION_F  : NUMERO[9:0] = SUM(MUNICIPIO.POBLACION_F)
POBLACION_M  : NUMERO[9:0] = SUM(MUNICIPIO.POBLACION_M)
LIMITE        : TRAZO ; LED
MUNICIPIO     : CLASE ; B_MUNI
POBLACION_T  : NUMERO[9:0] = POBLACION_F + POBLACION_M
POBLACION_F  : NUMERO[9:0]
POBLACION_M  : NUMERO[9:0]
LOCALIZACION : COORDENADA
LIMITE        : TRAZO ; LMN
FIN:
FIN:

```

FIGURA 5.3. Ejemplo del esquema que define una base de datos.

Para mayor informacion sobre el lenguaje de esquema consultar el capitulo VIII seccion 2.

V.3 EL LENGUAJE DE CONSULTA.

El administrador de base de datos SUPER-BASE provee de un ambiente adecuado para la recuperacion, consulta y manipulacion de los datos almacenados.

Los principales comandos definidos para la consulta son:

ABRIR - abre los archivos de una base de datos y los pone a disposicion del usuario.

CIERRA - cierra los archivos de una base de datos, esto es, los aisla del usuario.

ALTAS - permite almacenar un nuevo objeto con sus correspondientes características en el nivel actual.

MODIFICA - reemplaza la característica especificada de un determinado objeto.

BAJA - remueve de la base un objeto dado.

ORDENA - ordena respecto a un campo determinado de forma ascendente o descendente.

HOJEAR - permite el despliegue y modificación de la información.

Para ver otros comandos consultar el capítulo VIII.

V.4 CALCULO RELACIONAL.

El modelo de Super-Base permite al usuario expresar lo que desea como repuesta sin tener que especificar la forma en que debe ser calculada. Esto permite que la base de datos sea amigable al usuario.

Se considera al comando "SELECT" como parte del calculo relacional. La forma en que puede ser utilizado se muestra a continuación :

SELECT <condición sobre una característica>

Formas de condiciones :

1) <nom-var> operador <valor>

2) <valor1> operador <nom-var> operador <valor2>

la forma de acomodar los operadores debiera ser consistente.

EL OPERADOR puede ser : = , < , <= , > , >= .

Existe también un tipo de selección múltiple.

SELECT .SEL

Condición --> <condición sobre una característica>

Tipo --> <numero de tipo>

Condición --> <condición sobre una característica>

Tipo --> <numero de tipo>

.....

Esto da como resultado un archivo que contiene en cada registro el tipo de condición y el objeto que lo cumplió. El comando que le permite al usuario visualizar esta información es "DESPLIEGA".

Por otra parte, existe la posibilidad de restringir o ampliar la selección horizontal del nivel correspondiente mediante el comando de "VALIDEZ". Para poder entender su funcionamiento se mostrará a continuación un ejemplo:

Supóngase que se tiene almacenada en una base de datos todos los estados de la República Mexicana y sus municipios, con sus correspondientes producciones de trigo. Una solicitud de consulta es tratar de saber cuales son los municipios que pertenecen a Baja California, "VALIDEZ LOCAL", con producción mayor de 20 toneladas, el resultado corresponderá a los municipios 1, 2 y 4 de Baja California, los cuales pueden ser asegurados a través de MICROMAP. Sin embargo, si se desea asegurar todos los municipios de la República Mexicana con producción mayor de 20 toneladas se tendría que definir "VALIDEZ GLOBAL".

A NIVEL ESTADO

BAJA CALIFORNIA	AGUASCALIENTES BAJA CALIFORNIA ... ZACATECAS		
MUNICIPIO 1	MUNICIPIO 2	MUNICIPIO 1	... MUNICIPIO 3
MUNICIPIO 2	MUNICIPIO 3	MUNICIPIO 2	... MUNICIPIO 5
MUNICIPIO 4	MUNICIPIO 6	MUNICIPIO 4	... MUNICIPIO 6
	MUNICIPIO 9		... MUNICIPIO 7
			... MUNICIPIO 9
VALIDEZ LOCAL	VALIDEZ GLOBAL		

FIGURA 5.4. Ejemplo de Validez.

V.5 LENGUAJE DE MANIPULACION DE JERARQUIA.

Super-base provee al usuario de un lenguaje para la manipulación de la jerarquía. Se ha diseñado de tal forma que sea fácil para el usuario explorar la base de datos a través de la jerarquía definida por su esquema. El comando que permite llevar a cabo esta función es "CC" (cambio de clase).

Debido a que suponemos que el usuario está familiarizado con MS-DOS, la sintaxis del comando "CC" se diseñó de forma parecida a la del "CD" (cambio de directorio) ya que también maneja una jerarquía.

Ejemplo : CC \PAIS=MEXICO\ESTADO

En este ejemplo se transfiere el control de nivel país (MEXICO) al nivel estado. Es decir, desde ese momento los estados e información que puede consultarse pertenecerán al país MEXICO.

V.6 RELACIONES ENTRE VARIABLES.

Los tipos de relaciones entre variables se pueden agrupar de la siguiente forma:

- i) Operaciones,
- ii) Funciones jerárquicas.

i) Las operaciones se realizan entre variables, como la resta, multiplicación, división, etc. en un nivel dado.

ii) Las funciones jerárquicas son aquellas que se aplican a una variable definida a un nivel inferior, como son SUM, FRECT, etc. (ver VIII.5).

Estas formulas son de recalculacion manual, es decir, solo serán evaluadas cuando el usuario teclee el comando "CALC".

V.7 OPERACIONES CONJUNTAS MICROMAP-SUPERBASE.

MICROMAP es un sistema que provee de multiples funciones con el procesamiento de mapas por computadora que puede utilizar la informacion almacenada en la base de datos.

Lo importante de la relacion entre MICROMAP y SUPERBASE es la posibilidad de utilizar referencias a trazos en la base de datos, asi como referencias a datos en los archivos de mapas, que le permiten basicamente:

- 1) Dibujar un determinado objeto dentro de un mapa.
- 2) Consultar informacion acerca de un trazo o punto.
- 3) Cálculo de áreas.
- 4) Generacion de mapas ashurados de acuerdo a la informacion almacenada en la base de datos.

V.8 FACILIDADES DEL SISTEMA.

SUPER-BASE está diseñado en base a comandos. Se dispone de una ayuda que muestra los comandos existentes y proporciona una descripcion de ellos. Almacena las ultimas 20 instrucciones y permite editarlas.

Ademas, se dispone de comandos que realizan funciones propias del sistema operativo MS-DOS, sin tener que abandonar el sistema.

CAPITULO VI

LA ESTRUCTURA DEL SISTEMA ADMINISTRADOR

VI.1 MODELO.

El sistema administrador de bases de datos Super-Base está definido como un modelo relacional y jerárquico.

El acceso a las estructuras de una base de datos es a través de sus llaves no duplicadas y sus correspondientes ligas.

Las estructuras que se definieron contemplan las necesidades y objetivos para crear un medio ambiente adecuado que permita definir, almacenar y manipular una base de datos.

El tipo de archivos usados en el sistema son en su mayoría de acceso aleatorio.

Las estructuras han sido agrupadas de acuerdo a su naturaleza:

- 1) Estructuras de definición.
 - Esquema
 - Diccionarios
- 2) Estructuras de almacenamiento de datos.
 - Control de la tabla bidimensional
 - Direccionamiento y control de la jerarquía
- 3) Estructuras de conexión con MICROMAP.
 - Ligas de la base de datos con MICROMAP
 - Ligas de MICROMAP con la base de datos
- 4) Estructuras de apoyo.
 - Almacenamiento temporal de instrucciones
 - Ayuda
 - Macroinstrucciones
 - Seleccionar
 - Exportar
- 5) Variables importantes del sistema.

En la figura 6.1 se muestra el esquema general del sistema y cada una de sus partes es analizado con más detalle posteriormente.

La figura 6.2 muestra con mayor detalle el esquema general del sistema.

FIGURA 6.1. Estructura del sistema administrador Super-Base.

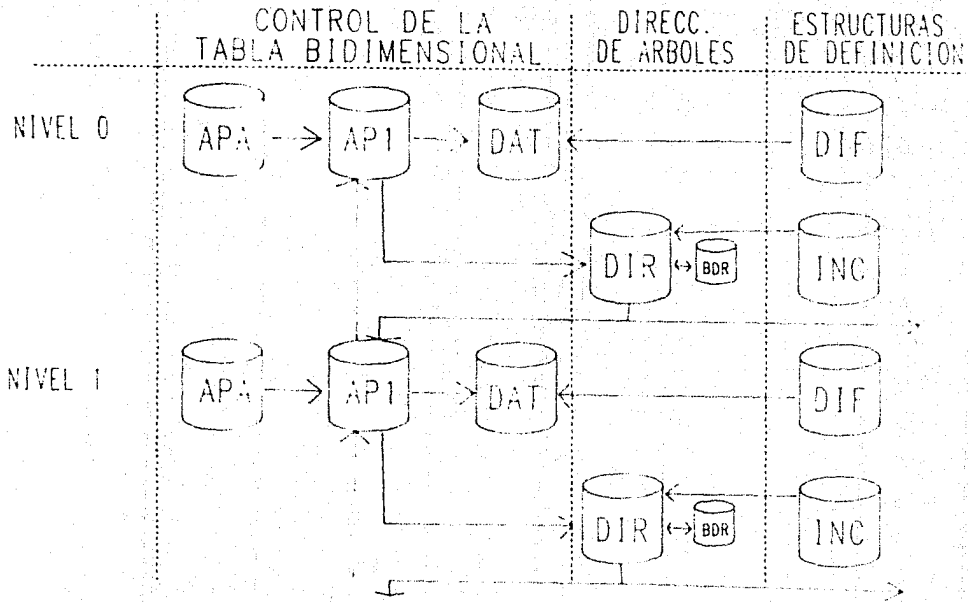
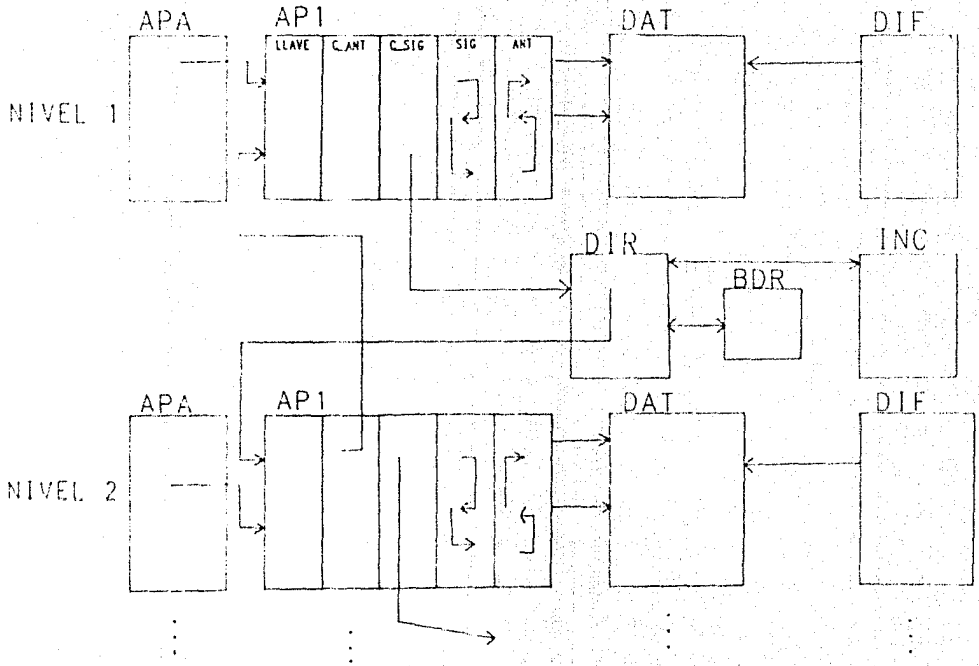


FIGURA 6-2. Estructura del sistema administrador Super-Base con mayor detalle. Las flechas en el archivo API dentro de los campos sig y ant indican que apuntan al siguiente y anterior registro.



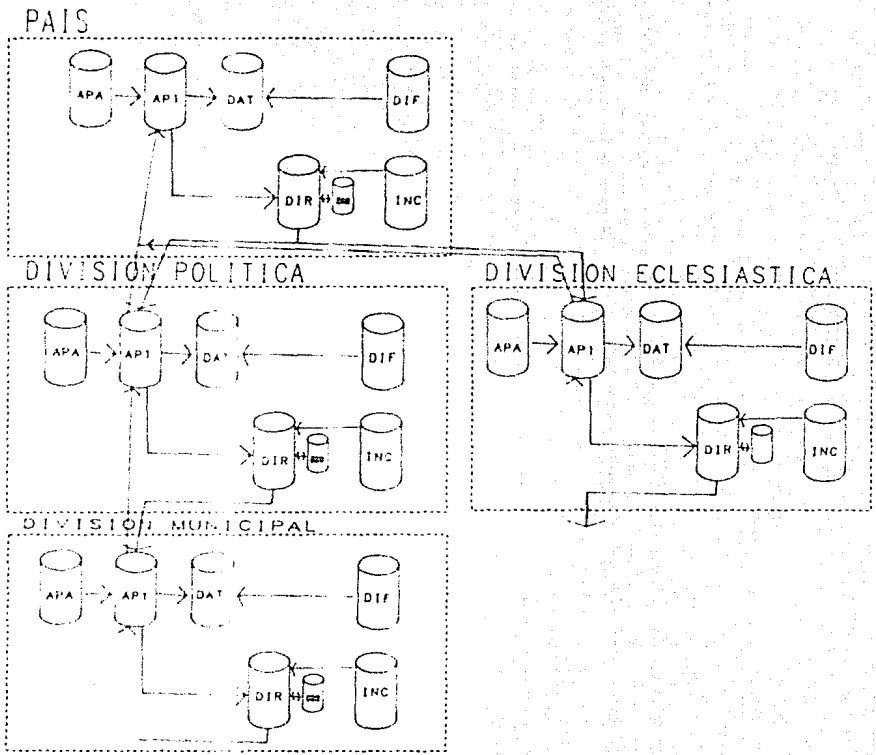
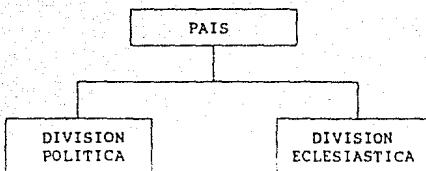


FIGURA 6.1. Estructura del sistema administrador Super-Base que muestra jerarquias multiples.

La figura 6.3 muestra el esquema una base de datos donde se tiene una jerarquía con múltiples arboles de la forma:



Así, una vez determinado el país deseado podrá elegirse consultar su división política o su división eclesiástica.

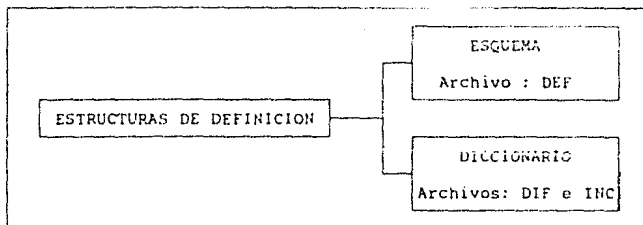
En este punto, se puede observar que SUPER-BASE es una base de datos muy general, que no solo permite la jerarquía vertical sino también "navegar" entre varias.

Aunque, la estructura de SUPER-BASE es compleja, así como los programas para su manejo, el usuario podrá fácilmente utilizarla y disfrutar de los beneficios que ofrece.

A continuación se dará una explicación más detallada de cada una de las estructuras según han sido agrupadas.

VI.2 ESTRUCTURAS DE DEFINICION.

Estas tienen la tarea de almacenar la definición de una base de datos y ponerla a disposición del administrador



VI.2.1 ESQUEMA.

La definición del esquema es almacenada en un archivo de texto con extensión DEF por el usuario.

Este archivo se define cuando se desea comenzar a utilizar una nueva base de datos. El sistema administrador solo lo utilizará la primera vez para definir el diccionario de datos y clases incluidas. Posteriormente el sistema utilizará solamente estos diccionarios para cualquier lectura y movimiento a través de la base de datos.

VI.2.2 DICCIONARIOS.

En el momento de "Definir" una base de datos, el sistema procederá a almacenar de forma permanente el esquema en dos archivos el DIF y el INC.

ARCHIVO DIF

Nombre externo : <nombre del archivo de la clase>. DIF
Nombre interno : DIF

Este archivo almacena los nombres de las variables y sus características, por lo que representa el "DICCIONARIO DE DATOS". Este archivo está formado por registros con la siguiente estructura:

REGISTRO DE TDIF :

NOMBRE	ATRIBUTO	ARCHIVO	FORMULA
80 caracteres	Arreglo de 1 a 8 de enteros	80 caracteres	80 caracteres
Nombre de la variable	Especificaciones sobre la variable	Nombre del archivo de las variables trazo	Formula

El archivo DIF contiene la siguiente informacion en atributo :

En el registro 0 :

```
atributo[1] <- longitud total de arreglote
atributo[3] <- Número de variables contenidas
```

En los demás registros :

```
atributo[1] <- tipo de variable
                1 - cadena de caracteres
                2 - número
                3 - vector de números
                4 - vector de cadenas de caracteres
                5 - coordenada
                6 - coordenadas
                7 - trazo
                8 - logico
```

```
atributo[2] <- Posicion inicial en arreglote
atributo[3] <- Longitud total del campo en arreglote
```

(informacion relativa a arreglote esta en la seccion VI.6.2)

tipo	1	2
atributo[4] <- #	cadena de caracteres	numero
atributo[5] <-	de caracteres	# de enteros
atributo[6] <-	sin uso	# decimales
atributo[7] <-	sin uso	sin uso
	s u	s u

tipo	3	4
atributo[4] <-	vector de números	vector de cadenas de carac.
atributo[5] <-	tamaño del vector	tamaño del vector
atributo[6] <-	# enteros	# caracteres
atributo[7] <-	# decimales	su
	su	su

tipo	5	6	7	8
atributo[4] <-	coordenada	coordenadas	trazo	logico
atributo[5] <-	0	0	0	1
atributo[6] <-	su	su	su	su
atributo[7] <-	su	su	su	su

```
atributo[8] <- Si esta en uso o no ( 0-no uso 1-en uso )
```

Nota :

```
El atributo[3] := 1 + atributo[4];
el primer byte sera la longitud de la cadena de caracteres.
```

ARCHIVO INC

Nombre externo : <nombre del archivo de la clase>. INC
Nombre interno : INC

Contiene los nombres de las clases incluidas que permiten el movimiento a través de la jerarquía, lo que representa un "DICCIONARIO DE CLASES".

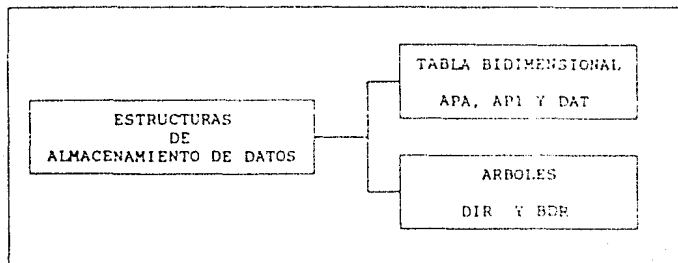
REGISTRO DE INC :

NOMBRE	ARCHIVO
el identificador de la clase	el identificador externo de la clase

VI.3 ESTRUCTURAS DE ALMACENAMIENTO DE DATOS.

Las estructuras de almacenamiento de datos son aquellas que permiten almacenar información en la base de datos, en un determinado lugar y a través de una trayectoria.

En este caso, se pueden distinguir dos tipos de estructuras: las primeras permiten el control de la tabla bidimensional y almacenamiento de las variables, y las segundas permiten el control de los árboles.



VI.3.1 CONTROL DE LA TABLA BIDIMENSIONAL.

El acceso a la base de datos en un cierto nivel es a través de APA y API. Así como, la información de cada objeto es almacenada en el archivo DAT.

ARCHIVO APA

Nombre externo : <nombre del archivo de la clase>. APA
Nombre interno : APA

Contiene las direcciones de los objetos ordenados, de tal forma que permite minimizar el tiempo de búsqueda.

REGISTRO DE APA :

APUNTA
entero de doble precisión
apuntador al archivo API y DAT

ARCHIVO API

Nombre externo : <nombre del archivo de la clase>. API
Nombre interno : API

Contiene las ligas entre los niveles y dentro de una misma clase.

REGISTRO DE API :

LLAVE	C_ANT	C_SIG	ANT	SIG
40 caracteres	entero de doble precisión	entero de doble precisión	entero de doble precisión	entero de doble precisión
nombre del objeto	apunta al objeto del nivel superior (padre)	apunta al DIR para accesar objetos a nivel inferior (hijos)	apunta al objeto anterior del mismo nivel	apunta al siguiente objeto del mismo nivel

El archivo API es direccionado por APA o DIR.

El archivo API, además, permite administrar los espacios borrados. Con el objetivo de aprovechar al máximo el espacio en disco.

En el registro 0 en la variable C_SIG se tiene almacenada la dirección del último borrado, el cual a su vez tiene almacenada la dirección del penúltimo borrado y así sucesivamente hasta el primer borrado que contendrá el valor 0 en esta variable.

ARCHIVO DAT

Nombre externo : <nombre del archivo de la clase>. DAT
Nombre interno : DAT

En cada registro se tienen los datos que pertenecen a un objeto. El archivo DAT es direccionado por APA y mantiene el mismo orden que API.

DAT es un archivo especial ya que es de longitud variable, lo cual provee al sistema de la flexibilidad para que el usuario defina las variables que desea almacenar.

Para observar mas de cerca su funcionamiento consultar capitulo VII seccion 4.1 .

REGISTRO DAT:

es un arreglo de caracteres y su longitud queda determinada por una variable al efectuar su correspondiente reset o rewrite

VI.3.2 DIRECCIONAMIENTO DE ARBOLES.

Este tipo de estructuras permiten los cambios de nivel a traves de la jerarquia definida por el esquema.

Los archivos que dan este soporte son DIR, BDH y API.

A continuacion se hablara de DIR y BDR, pero API ya ha sido explicado anteriormente.

ARCHIVO DIR

Nombre externo : <nombre del archivo de la clase>. DIR
Nombre interno : DIR

Un determinado objeto en un nivel dado puede direccionar a otros objetos a nivel inferior. Con la posibilidad de apuntar a algún árbol a nivel inferior.

ARCHIVO BDR

Nombre externo : <nombre del archivo de la clase>. BDR
Nombre interno : BDR

Almacena las direcciones de los registros borrados en DIR para su posterior uso. Su función también es aprovechar al máximo el espacio en disco.

REGISTRO BDR:

BDR_DIR
entero de 24 bits precisión
apunta a los registros borrados de DIR

VI.4 ESTRUCTURAS DE CONEXION CON MICROMAP.

Las estructuras que se exponen en esta sección permiten la conexión con otro programa llamado MICROMAP. Este sistema tiene como objetivo la digitalización, edición, almacenamiento y graficación de mapas por computadora.

Es interesante pensar que la información almacenada en una base de datos, dada de alta por Super-Base, pueda relacionarse con MICROMAP.

FIGURA 6.4. Las estructuras de conexión SUPER-BASE <--> MICROMAP.

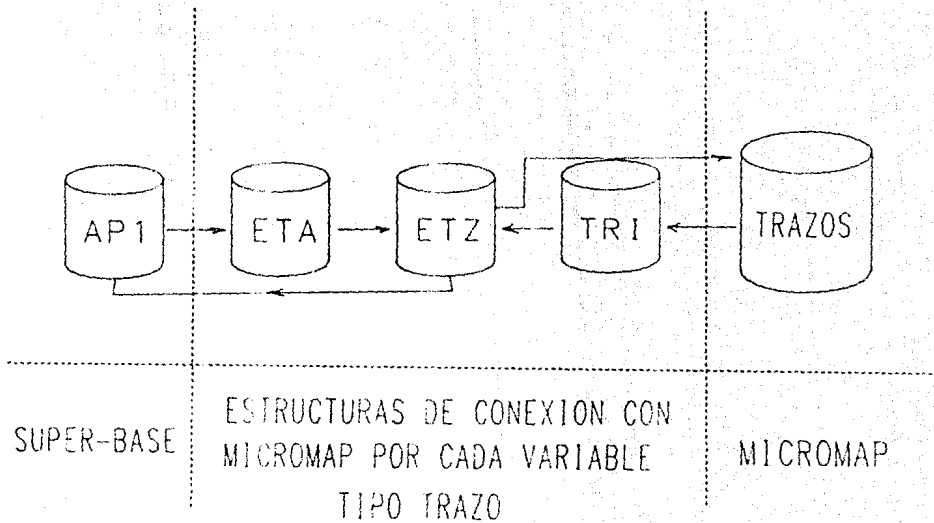
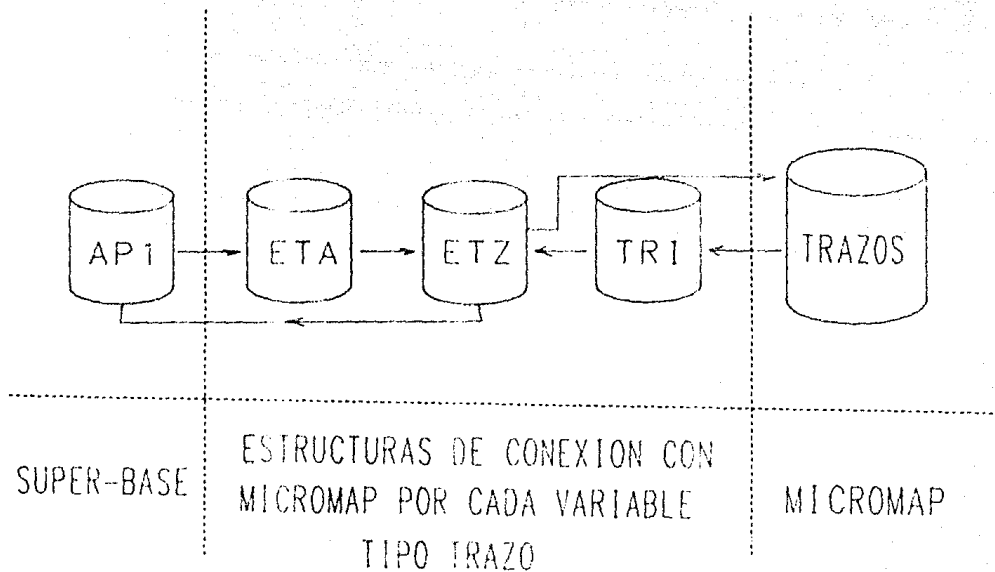
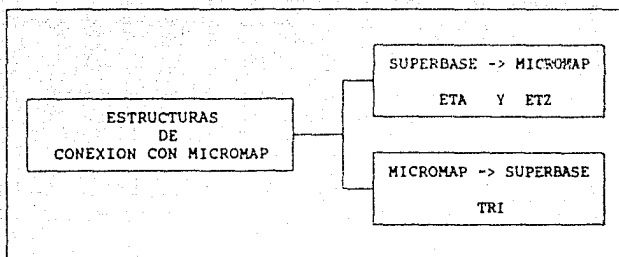


FIGURA 6.4. Las estructuras de conexión SUPER-BASE: <--> MICROMAP.





En la figura 6.4. muestra a mayor detalle las estructuras de conexión de SUPER-BASE con MICROMAP y su funcionamiento.

VI.4.1 LIGAS DE LA BASE DE DATOS CON MICROMAP.

La conexión entre una base de datos y MICROMAP es a través de apuntadores, en donde cada objeto puede direccionar a un conjunto de trazos. Estos trazos pueden pertenecer a diferentes mapas.

Las estructuras que dan soporte a este tipo de ligas son ETA y ETZ.

ARCHIVO ETA

Nombre externo : <nombre del archivo de la clase>. ETA

Nombre interno : ETA

Representa la liga entre los objetos y sus trazos, mediante un apuntador auxiliar de API a ETZ. Puesto que API esta directamente relacionado con los objetos y ETA tiene una relacion estrecha con los trazos.

REGISTRO ETA:

APUNTA_ETZ
entero de doble precision
apunta a los registros de ETZ

ARCHIVO ETZ

Nombre externo : <nombre del archivo de la clase>. ETZ

Nombre interno : ETZ

Guarda la información de todos los trazos de un objeto y los objetos que están asignados a un mismo trazo. Además, contiene un apuntador a los trazos y el nombre del archivo al que pertenece.

REGISTRO DE ETA :

LLAVE 1	APUNTA 1	LLAVE 2	APUNTA 2	NOM_ARCH	APUNTA
entero de doble precisión	entero de doble precisión	entero de doble precisión	entero de doble precisión	8 caracteres	entero de doble precisión
apunta al objeto en API	apunta al siguiente reg. en ETZ que pertenece al mismo objeto	apunta al objeto en API	apunta al siguiente reg. en ETZ que pertenece al mismo objeto	nombre externo del archivo al que pertenece ese trazo	apunta al archivo de trazos

VI.4.2 LIGAS DE MICROMAP CON LA BASE DE DATOS.

Esta estructura da la posibilidad de que una vez especificado un trazo en MICROMAP se pueda consultar información en la base de datos.

ARCHIVO TRI

Nombre externo : <nombre del archivo de la clase>. TRI

Nombre interno : TRI

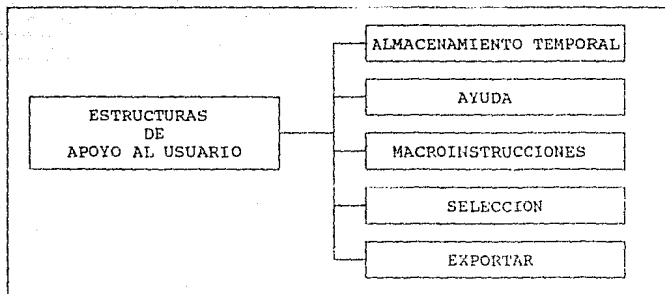
La liga entre MICROMAP y la base de datos, es decir, los trazos y ETZ, es mediante el uso del archivo TRI y el cual es creado en el momento que se asigna el primer trazo con la base de datos.

REGISTRO TRI:

APUNTA_ETZ
entero de doble precisión
apunta a los registros de ETZ

VI.5 ESTRUCTURAS DE APOYO.

Para la ejecución de otras tareas de menor importancia se tienen las estructuras de apoyo.

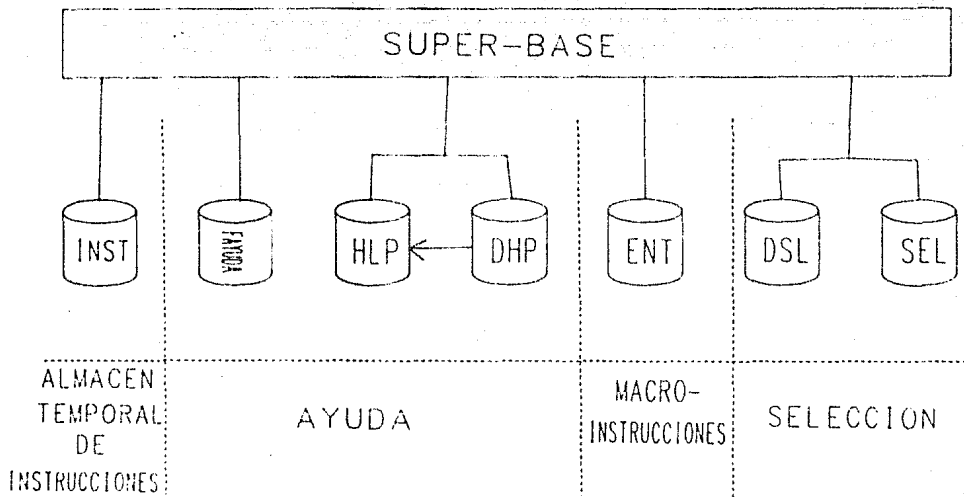


En la figura 6.5. muestra con mayor detalle las estructuras de apoyo.

VI.5.1 ALMACENAMIENTO TEMPORAL DE INSTRUCCIONES.

El almacenamiento temporal de instrucciones es un apoyo para el usuario, ya que permite la edición, repetición y consulta de las últimas 20 instrucciones. La razón de que sea un archivo en disco es debido al tamaño del sistema, de lo contrario ocuparía espacio en memoria indispensable para otras tareas.

FIGURA 6.5. Muestra las estructuras de apoyo.



ARCHIVO INST

Nombre externo : INST.AUX

Nombre interno : INST

REGISTRO INST:

INSTR
cadena de 255 caracteres
almacena una instrucción

VI.5.2 AYUDA.

El sistema provee al usuario de ayuda rápida y adecuada. Muestra al usuario una pantalla de ayuda en donde se encuentran agrupados y clasificados todos los comandos de la base de datos. Además el usuario podrá pedir información de alguno de ellos. La ayuda está diseñada de tal forma que permite explorar el manual de usuario. Las estructuras que dan soporte a esto son: Fayuda, Hlp y Dhlp.

ARCHIVO FAYUDA

Nombre externo : B_AYUDA.INC

Nombre interno : FAYUDA

Tipo de archivo : Texto

Tiene almacenadas en formato de texto, las pantallas que mostrará el sistema al ser solicitado el "HELP". En el cual se encuentran todos los comandos agrupados de acuerdo a sus funciones.

REGISTRO FAYUDA:

FAYUDA
un renglón de un archivo de texto
almacena las pantallas a mostrar

ARCHIVO HLP

Nombre externo : SUP_BASE.HLP
 Nombre interno : HLP

Tiene almacenado el manual de usuario en registros de tal forma que permite desplegado directo en la impresora. Este se encuentra almacenado en formato de registros que permite el acceso aleatorio. Debido a esto el usuario puede solicitar información de cualquier parte del manual desde la computadora y cuando lo necesita.

REGISTRO DE DHLP:

RENG	LF	CR	FF
67 caracteres	1 caracter	1 caracter	1 caracter
Contiene la información del manual de usuario	caracter de fin de línea	caracter de regreso de carro	caracter de cambio de página

Este archivo es direccionado por DHLP.

ARCHIVO DHLP

Nombre externo : SUP_BASE.DHP
 Nombre interno : DHLP

Tiene almacenados los apuntadores de direccionamiento, que indicarán al sistema cuáles son las páginas que deben de mostrarse al usuario dada una determinada solicitud de información. El sistema dará al modulo de ayuda un numero de solicitud y este recurrirá a la información almacenada en DHLP.

REGISTRO DHLP:

INI	FIN
entero	entero
Apuntador de página inicial	Apuntador de página final

VI.5.3 MACROINSTRUCCIONES.

Super-Base contiene además un intérprete de macroinstrucciones que permite al usuario escribir una colección de instrucciones que serán ejecutadas automáticamente por el sistema sin intervención del mismo, facilitando las tareas que requieran de series de instrucciones repetitivas.

El intérprete de macroinstrucciones será ampliado posteriormente para, utilizarse como un lenguaje de programación incluyendo instrucciones como FOR, WHILE, REPEAT, etc, para obtener el máximo provecho de una base de datos en particular.

ARCHIVO ENT

Nombre externo : <nombre del archivo que indique el usuario>
Nombre interno : ENT
Tipo de archivo : Texto

Cada uno de los registros es una instrucción a ejecutar.

VI.5.4 SELECCION.

Dado que Super-Base permite la selección de objetos de acuerdo a determinadas condiciones establecidas por el usuario sobre sus características. El administrador tendrá que hacer uso de estructuras auxiliares las cuales son los archivos DSL y SEL.

ARCHIVO DSL

Nombre externo : <nombre del archivo de la clase>. DSL
Nombre interno : DSL

Es una estructura que me permite almacenar las direcciones de los objetos que cumplieron con las condiciones solicitadas.

REGISTRO DSL:

TIPO	APUNTADOR
entero	entero de doble precisión
tipo de condición	apunta a API

ARCHIVO SEL

Nombre externo : <archivo indicado por el usuario>. SEL
Nombre interno : SEL

Tiene como objetivo almacenar las condiciones y el tipo asignado. SEL es un archivo de texto donde la información está guardada por pares: el primer registro es la condición y el segundo es su tipo.

REGISTROS SEL:

1 Registro

CONDICION
cadena de 80 caracteres
contiene una condicion

2 Registro

TIPO
entero
tipo de condición

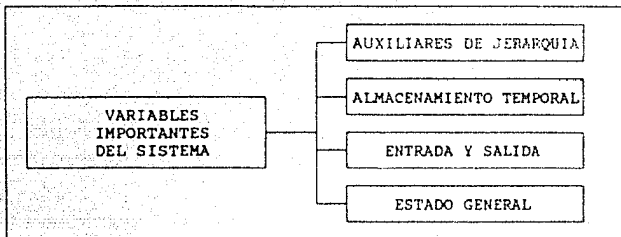
VI.5.5 EXPORTAR.

El administrador de Bases de datos Super-Base tiene diseñada una estructura que permite la exportación de información en formato SDF, debido a que la mayoría de los paquetes comerciales que manejan datos pueden tomarlos de dicho formato .

VI.6 VARIABLES IMPORTANTES DEL SISTEMA.

Las variables que se mencionan en esta sección juegan un papel importante en el sistema.

Dada su utilidad las variables las dividiremos en cuatro partes: auxiliares para el manejo de la jerarquía, almacenamiento temporal de los datos, entrada/salida y de estado general.



VI.6.1 VARIABLES AUXILIARES PARA EL MANEJO DE LA JERARQUIA.

Estas variables almacenan la información referente a la trayectoria en un momento dado.

NIV contiene el nombre de la clase, nombre del objeto y nombre del archivo de cada uno de los niveles dentro de la trayectoria actual.

DIR_OBJ almacena los apuntadores de cada uno de los objetos de los niveles anteriores correspondientes a la trayectoria actual.

VI.6.2 VARIABLES PARA ALMACENAMIENTO TEMPORAL DE DATOS.

ARREGLOTE es la variable donde se almacena temporalmente la información que va ser guardada en DAT.

VI.6.3 VARIABLES DE ENTRADA Y SALIDA.

ENTRADA_A es una variable que guarda el nombre del archivo que contiene tanto la información de una Macroinstrucción como los datos solicitados por el sistema.

N_SALIDA especifica el dispositivo de salida. Es decir, si la información va ser escrita en pantalla, archivo o impresora.

VI.6.4 VARIABLES DE ESTADO GENERAL.

Son aquellas que definen el medio ambiente con el usuario como : prompt, path, campana, retardo, digitos y auto.

CAPITULO VII

DESCRIPCION FUNCIONAL DE SUPER-BASE

VII.1 MODULO DE CONTROL DE LA SESION.

Este módulo se encarga de la interface con el usuario y tiene como objetivo la recepción y atención de los comandos enviados. Además de lo anterior, se encarga de coordinar adecuadamente la acción del resto de los procedimientos a través del control de procesos de que se dispone.

Al principio de la sesión, el sistema hace la lectura de las variables iniciales, a continuación, aparece en pantalla la señal "0:", que implica que el sistema está listo para recibir instrucciones.

Cada instrucción enviada al sistema es verificada y enviada a un procedimiento que se encarga de dividirla en dos partes; la primera, será identificada como el comando y la segunda, si existe, como sus parámetros. Si el comando es correcto, entonces pasará el control al procedimiento que se encargará de ejecutarlo. Al finalizar, el sistema, quedará listo para recibir una nueva instrucción.

El usuario tiene la facilidad de asignar como el dispositivo de entrada a un archivo, por lo que se permite la construcción de "macroinstrucciones".

El sistema provee al usuario de un almacenamiento temporal de 20 instrucciones que permite su consulta, recuperación y edición.

VII.2 SUBSISTEMAS PRINCIPALES.

Una vez concluida la etapa de análisis se procedió a una de las etapas de mayor complejidad que es la programación.

Es importante señalar que para llevar a cabo la elaboración del software, como ya se mencionó, se contemplaron diferentes factores como:

- lenguaje de programación
- tipo de máquina
- estructura
- objetivos.

A rasgos generales podemos considerar a SUPER-BASE compuesto por seis subsistemas básicos:

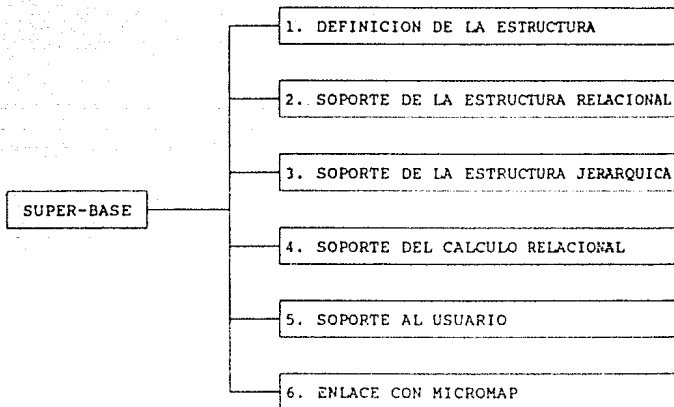


FIGURA 7.1. EL diagrama muestra los diferentes subsistemas que constituyen el sistema SUPER-BASE.

A continuación se explicarán cada uno de los subsistemas.

VII.3. DEFINICION DE LA ESTRUCTURA.

La tarea principal de este subsistema es la definición de la estructura de una base de datos de acuerdo al esquema establecido por el usuario.

El sistema se encarga de abrir el archivo .DEF que tiene almacenada la estructura de la base que se desea definir. Hace la lectura de todos sus registros. Para cada registro ejecuta la verificación sintáctica además prepara o inicializa las estructuras para la creación de la base de datos.

En cuanto a la verificación estructural, un registro es aceptado si ha sido construido conforme a las reglas de sintaxis correspondientes.

Cuando éste es rechazado los mensajes de error adecuados son desplegados.

Ademas el sistema se encarga de crear los archivos .DIF en donde se almacenan, de manera permanente, los nombres de las variables, sus caracteristicas para cada uno de los niveles y las especificaciones que tendra .DAT, de tal forma que permite el rápido acceso de los datos.

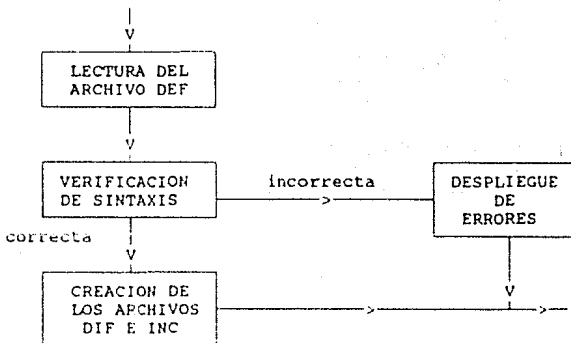


FIGURA 7.2. Definición de la estructura.

VII.4. SUBSISTEMA DE SOPORTE DE LA ESTRUCTURA RELACIONAL.

El subsistema de soporte de la estructura relacional, como su nombre lo indica, nos permite realizar altas, bajas y cambios en un nivel determinado, el cual tiene una estructura relacional.

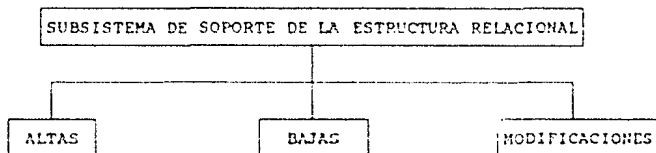


FIGURA 7.3. El subsistema de soporte de la estructura, esta constituido a su vez por 3 subsistemas.

Las funciones de estos subsistemas se especificarán a continuación.

VII.4.1. ALTAS

El subsistema de altas de SUPER-BASE tiene como función permitir el ingreso de nuevos objetos a la base de datos en el nivel actual, registrándolos adecuadamente en la estructura.

Conceptualmente se puede dividir en módulos que realizan las tareas fundamentales del proceso:

- I) Módulo de control
- II) Módulo de búsqueda
- III) Módulo de lectura de variables
- IV) Módulo de inserción.

I) Módulo de control

Coordina adecuadamente la acción del resto de los módulos, hace la lectura del objeto pasando el control al módulo de búsqueda el cual regresa una variable lógica que indica si el objeto ya fué dado de alta. Si no ha sido dado de alta se procederá al módulo de lectura de variables e inserción, de lo contrario se desplegará un mensaje de error ya que el sistema no permite la duplicación de llaves.

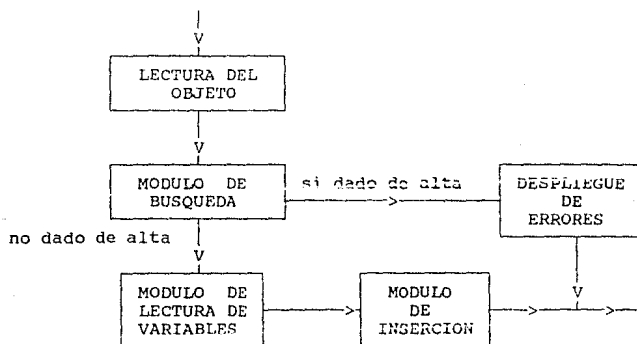


FIGURA 7.4. Módulo de control de ALTAS.

II) Módulo de búsqueda.

Realiza búsquedas binarias sobre el archivo .APA, para detectar si el objeto en cuestión fué dado de alta con anterioridad, guardando la dirección donde lo encontró o el lugar donde deberá insertarse, esta localidad y una variable lógica que indica si se encontró son regresadas al módulo de control.

III) Módulo de lectura de variables.

La tarea principal de este módulo es la lectura, verificación y almacenamiento temporal de las variables de acuerdo a la estructura.

La lectura de éstas se lleva a cabo a través del dispositivo de entrada. En caso de que un error sea detectado, el proceso de lectura se repetirá tantas veces como sea necesario.

La verificación detecta errores cometidos al realizar la lectura de las variables. Y se lleva a cabo considerando los siguientes criterios:

- Una variable de cadena de caracteres puede contener caracteres alfanuméricos, sin exceder la longitud definida previamente en la estructura.
- Una variable numérica, contendrá exclusivamente números, de acuerdo a los enteros y decimales especificados.
- En un vector de caracteres, cada entrada es a su vez una cadena de caracteres alfanuméricos, por lo que la verificación se realizará individualmente para cada una de las componentes como variable de cadena de caracteres.
- En un vector numérico, cada entrada será verificada como una variable numérica.
- El tipo coordenada es considerado como un vector numérico con dos entradas.
- Una variable lógica solo puede tener dos valores: falso o verdadero.

IV) Módulo de inserción.

Una vez realizada la lectura de las variables se procederá a introducir la información a la base de datos.

El módulo de inserción para llevar a cabo esta tarea ejecutará los pasos siguientes:

A) Detección de registros borrados.

Accesará el registro 0 del archivo API para verificar la existencia de registros borrados. En caso de existir alguno, la dirección de inserción apuntará a la localidad del último borrado

y se procederá a actualizar el registro 0 con la dirección del penúltimo borrado. En caso de no haber registros borrados, la dirección de inserción apuntará a la siguiente localidad del último registro de API.

B) Almacenamiento en APA.

Dado el apuntador de inserción por el módulo de búsqueda se procede a recorrer todos los registros de APA una posición desde esa dirección hasta el final del archivo.

C) Almacenamiento en API.

La información será almacenada en la dirección indicada por la dirección de inserción como se vio en A). Durante la inserción de los datos se establece su relación a través de apuntadores con el objeto anterior y el siguiente del mismo nivel, del objeto del que proviene en el nivel anterior, y de los objetos que pertenecen a éste en niveles inferiores. De tal forma que almacena la información necesaria para conservar el modelo relacional y jerárquico.

D) Almacenamiento de .DAT.

La información relativa a las variables se guardará de acuerdo al archivo .DIF. El número de registro en que se insertó en API, será el mismo para insertar el dato en .DAT.

E) Actualización del DIR.

La actualización se lleva a cabo sólo si el nivel es mayor que uno. Si es el primer elemento a dar de alta en una cierta trayectoria entonces se deberán almacenar los apuntadores necesarios en .DIR.

VII.4.2. BAJAS.

El subsistema de bajas tiene la tarea de dar de baja del sistema los objetos especificados, efectuando los movimientos adecuados y actualizando las ligas.

Los módulos en que se divide son:

- I) Módulo de control
- II) Módulo de búsqueda.
- III) Módulo de verificación
- IV) Módulo de borrado

I) Módulo de control.

Coordina adecuadamente la acción del resto de los módulos.

Hace la lectura del objeto a dar de baja pasando el control al módulo de búsqueda, el cual indicará si estaba dado de alta o no.

Si el objeto se encuentra dado de alta se procederá al módulo de verificación y de borrado. En caso contrario se desplegará el mensaje de error correspondiente.

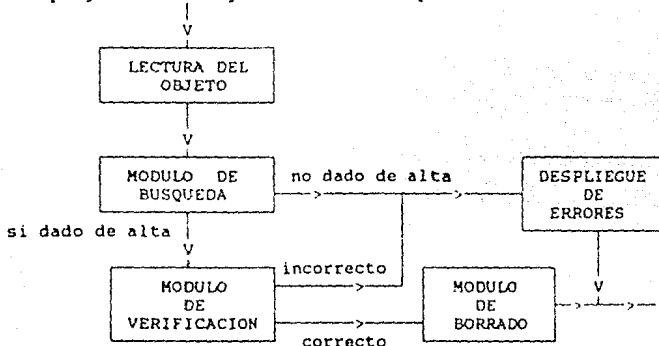


FIGURA 7.5. Módulo de control de BAJAS.

II) Módulo de Búsqueda.

Realiza la misma función que el módulo de búsqueda en el subsistema de altas.

III) Módulo de verificación.

El sistema deberá verificar si el objeto a dar de baja tiene elementos dados de alta en niveles inferiores.

En caso de tener elementos no se pasará el control al módulo de borrado y se desplegará un error.

IV) Módulo de borrado.

Una vez hecha la verificación, se procederá a realizar los siguientes pasos.

A) Actualización de API. En esta parte se llevará a cabo la actualización de las ligas que direccionan al objeto anterior y siguiente.

ESTA TESIS NO DEBE
SALIR DE LA BIBLIOTECA

B) Actualización de DIR. Si el nivel es diferente de uno, entonces se procederá a actualizar las ligas de DIR y en el caso de borrarse un registro en este se almacenará la dirección en BDR.

C) Actualización en APA. Proporcionada la dirección por el módulo de búsqueda se procederá a recorrer todos los registros en una posición con incremento -1 desde esa dirección+1 hasta el final del archivo.

D) Actualización de borrados. La dirección del objeto borrado en API será almacenada en el registro 0 de API y en la localidad del último borrado se almacenará la dirección del penúltimo borrado.

VII.4.3. MODIFICACIONES.

El objeto de este subsistema es permitir la actualización de la información almacenada en la base de datos.

Para llevar a cabo esta tarea se auxilia de los siguientes módulos.

- I) Módulo de control
- II) Módulo de búsqueda.
- III) Módulo de despliegue y lectura.
- IV) Módulo de actualización.

I) Módulo de control.

Además de coordinar los diferentes módulos, se encarga de hacer la lectura del objeto a modificar. Se pasa el control al módulo de búsqueda. Si el objeto fué encontrado, se procederá a los módulos de despliegue, lectura y actualización de la información.

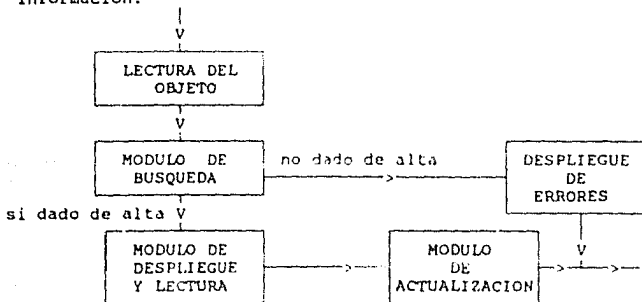


FIGURA 7.6. Módulo de control de MODIFICACIONES.

II) Módulo de búsqueda. (Referirse al módulo de búsqueda en altas).

III) Módulo de despliegue y lectura.

Se encargará de hacer las lecturas necesarias para desplegar en pantalla la información almacenada. Permittedole al usuario modificarla. Simultáneamente, el sistema, hará una verificación de esta.

IV) Módulo de actualización.

A) Actualización en APA, en caso de modificarse la llave se almacenara el nuevo valor en API en el mismo registro, conservando los demas campos, y se procedera a ordenar APA.

B) Actualización en DAT. En caso de que el usuario haya modificado alguna de las variables, se procedera a almacenar los nuevos datos en DAT en el mismo registro.

VII.5. SUBSISTEMA DE SOPORTE DE LA ESTRUCTURA JERARQUICA.

El subsistema de soporte de la estructura jerarquica, tiene como funciones el permitir a el usuario navegar a traves de la estructura jerarquica, utilizar funciones entre niveles y exportación e importación de datos en un cierto formato.

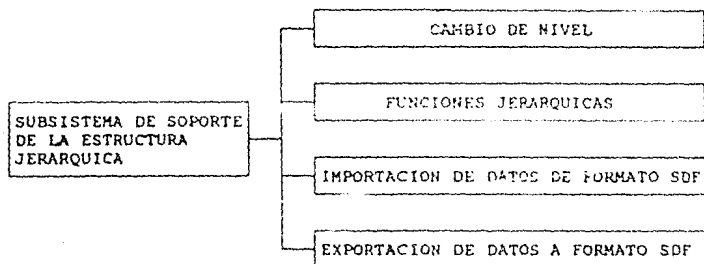


FIGURA 7.7. EL diagrama muestra los subsistemas que constituyen el sistema de soporte de la estructura jerárquica.

A continuación se explicaran las funciones de los subsistemas.

VII.5.1. CAMBIO DE NIVEL.

El subsistema de cambio de nivel fué diseñado para permitir al usuario moverse a través de la estructura jerárquica.

La forma en que puede realizarlo se divide en los siguientes módulos.

- I) Módulo de control.
- I) Módulo de desplegado de árbol.
- III) Módulo de cambio de clase.
- IV) Módulo para fijar un objeto.

I) Módulo de control.

Este recibe el parametro de la instrucción, verifica su sintaxis y lo traduce a acciones que serán llevadas a cabo por los diferentes módulos contenidos en este subsistema.

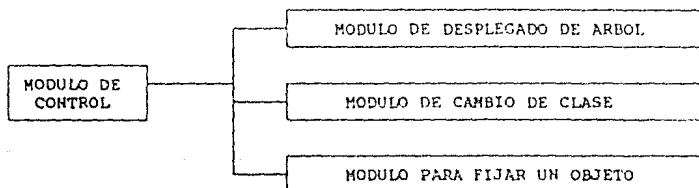


FIGURA 7.3. Módulo de control de CAMBIO DE NIVEL.

II) Módulo de Desplegado del árbol.

El módulo de desplegado tiene como función el permitir la presentación al usuario de la trayectoria actual.

III) Módulo de cambio de clase.

Tiene como tarea el cambio de clase, verificando su existencia en el archivo INC, actualizando variables temporales, cerrando las del nivel actual, y efectuando la apertura de los archivos correspondientes.

IV) Módulo de fijar objeto.

Verifica su existencia a través del modulo de búsqueda (ver modulo de búsqueda en altas) y si esta dado de alta almacenara en ciertas variables globales el objeto, su direccion en APA y API.

VII.5.2. FUNCIONES JERARQUICAS.

El sistema SUPER-BASE tiene definidas ciertas funciones entre niveles. Que puede utilizar el usuario cuando define una base de datos. El sistema calculara estas funciones hasta el momento en que el usuario se lo indique (CALC).

La forma de realizar esta tarea es mediante un proceso recursivo por nivel, el cual calcula las fórmulas de cada uno de los objetos.

Durante el proceso el sistema identifica cada una de las formulas y, de acuerdo a estas, hace las operaciones necesarias y finalmente almacena el resultado.

VII.5.3 IMPORTACION DE DATOS DE FORMATO ESTANDAR DE DATOS.

Este subsistema tiene como función básica la importación de datos de formato de texto o formato estandar de datos (SDF).

Esta formado de los modulos siguientes:

- I) Módulo de lectura de campos.
- II) Módulo de proceso.

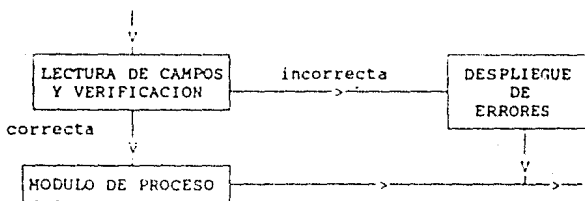


FIGURA 7.9. Funcionamiento de IMPORTACION.

I) Módulo de lectura de campos.

El módulo de lectura se ocupa de la lectura del nombre del archivo de datos y de sus campos, su posición inicial y final y de que variable se trata, verificando su existencia.

II) Modulo de proceso.

Este modulo fue diseñado para la importación de datos de formato de texto.

Todos los registros de este archivo tienen la misma estructura por lo que se podría visualizar como una tabla bidimensional. Sin embargo, el sistema durante el proceso los almacenara de acuerdo a la jerarquia especificada.

VII.5.4. EXPORTACION DE DATOS A FORMATO ESTANDAR DE DATOS.

Este subsistema esta encargado de la exportacion de datos en formato SDF, tomando en cuenta las especificaciones previamente dadas por el usuario.

Para cumplir con sus funciones este sistema se auxilia de los siguientes módulos.

- I) Módulo de control
- II) Módulo de verificación
- III) Módulo de reporte.

I) Modulo de control.

El módulo de control fué diseñado para permitir al usuario definir las variables que desea exportar y su formato de salida. A su vez este coordina la verificación y el reporte final.

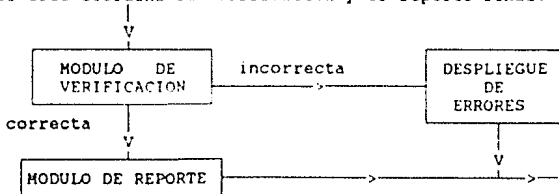


FIGURA 7.10. Módulo de control de EXPORTACION.

II) Módulo de verificación.

El proceso de verificación se lleva a cabo para cada una de las variables especificadas. Dicha verificación consiste en los pasos siguientes:

A) Comprueba la existencia de la clase dentro de la estructura jerárquica, y examina que estas respeten un orden de acuerdo al esquema de la base de datos.

B) Verifica la existencia de la variable en el archivo DIF, almacenando temporalmente las características que serán necesarias para agilizar el proceso del reporte.

III) Módulo de reporte.

El proceso necesario para llevar a cabo el reporte es recursivo. Este consiste en la extracción de los valores de cada una de las variables especificadas, que pertenecen a ese nivel, para cada uno de los objetos. La recursividad se da cuando hay un cambio de nivel. Simultáneamente se ve auxiliado por un almacenamiento temporal de salida en donde se almacenan dichos valores y realizará la impresión en el momento adecuado.

VII.6 SUBSISTEMA DE SOPORTE DEL CALCULO RELACIONAL.

El sistema administrador de base de datos da como soporte del cálculo relacional la selección de objetos de acuerdo a ciertas condiciones.

A continuación se muestran los módulos que lo conforman:

- I) Módulo de control.
- II) Módulo de verificación.
- III) Módulo de proceso y selección.

I) Módulo de control.

Realiza la lectura de las condiciones, así como el tipo que se le asigna.

Pasando a la etapa de la verificación, la cual indicara si debe o no realizarse el módulo de proceso.

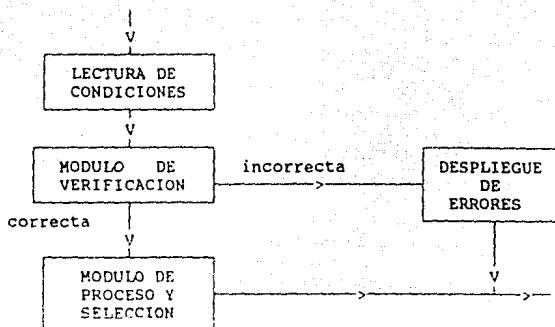


FIGURA 7.11. Módulo de control de CALCULO RELACIONAL.

II) Módulo de verificación.

Para cada una de las condiciones se identificarán las partes que la componen: los operadores, las variables y los parámetros.

Se comprobará:

- la existencia de la variable en el archivo DIF,
- la congruencia de los operadores y de sus parámetros.

III) Módulo de proceso y selección.

Para cada uno de los objetos se examinará si cumplen o no con la condición y en caso afirmativo se almacenará en el archivo SEL la siguiente información:

- el tipo de condición que cumple,
- el nombre del objeto y
- su apuntador a API.

Esto permite agilizar el despliegado, análisis y reportes de la información de los objetos seleccionados.

VII.7 SUBSISTEMA DE SOPORTE AL USUARIO.

El subsistema de soporte al usuario proporciona facilidades para realizar tareas de MS-DOS y creación de macroinstrucciones.

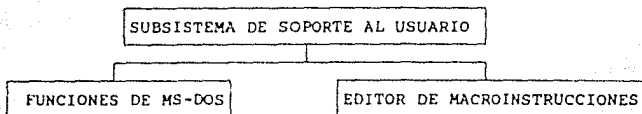


FIGURA 7.12. El subsistema de soporte al usuario.

VII.7.1. FUNCIONES DE MS-DOS.

Permite al usuario tener varias de las facilidades dadas por el sistema operativo MS-DOS. Las instrucciones existentes se presentan en la siguiente figura.

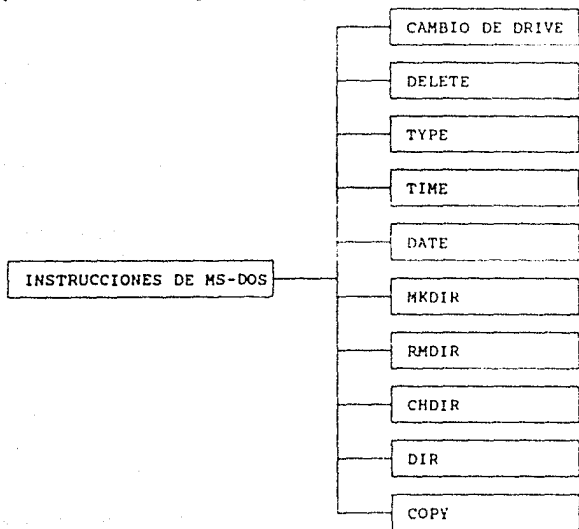


FIGURA 7.13. Instrucciones de MS-DOS implementadas en SUPER-BASE.

Para implementar este tipo de comandos en el sistema, se utilizaron las instrucciones del unit DOS de Turbo Pascal 5.5.

En cada uno de los casos se verifica la sintaxis adecuada de la instrucción y si es correcta se procederá a su ejecución.

VII.7.2. EDITOR DE MACROINSTRUCCIONES.

El editor de macroinstrucciones es un editor de línea que fué diseñado para que el usuario pueda almacenar instrucciones en un archivo de texto. Este, posteriormente, podrá asignarse como entrada del sistema que se encargará de procesar las instrucciones.

VII.8 SUBSISTEMA DE ENLACE MICROMAP-SUPERBASE

Este subsistema tiene como función principal la comunicación entre los sistemas MICROMAP y SUPER-BASE. El objetivo de esta comunicación es poder reflejar gráficamente lo almacenado en la base de datos:

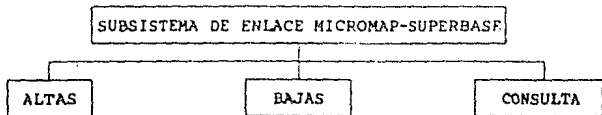


FIGURA 7.14. El subsistema de enlace MICROMAP-SUPERBASE.

VII.8.1 ALTAS.

Debido a que existe una conexión entre los dos sistemas deberán existir ligas entre objetos y trazos, por lo que el proceso para asignar un trazo a un objeto, es decir, darlo de alta en la base de datos es el siguiente:

- 1) Una vez determinado el objeto y el trazo se procederá a almacenar las ligas en ETA y ETZ.
- 2) Se almacenará también la liga entre un mapa y la base de datos en TRI.

VII.8.2. BAJAS.

Permite borrar las ligas entre el trazo y el objeto, es decir, modifica los valores ETA, ETZ y TRI.

VII.8.3. CONSULTA.

Las estructuras de ETA, ETZ y TRI dan soporte para la consulta de información. Esta puede llevarse a cabo de dos formas:

- 1) Dado un objeto se desea la información de los trazos que pertenecen a él y en qué mapa están almacenados.
- 2) Especificando un trazo, se puede saber a qué objetos pertenece, en una cierta base de datos.

CAPITULO VIII
INSTRUCCIONES DE SUPER-BASE

VIII.1 INTRODUCCION.

El administrador de bases de datos Super-Base ofrece diferentes comandos a través de los cuales el usuario puede definir la estructura de la base, editar, consultar, etc.

La pantalla está dividida en tres partes:

INFO-NIVEL	CLASE OBJETO	PAIS
COMANDOS	@:ABRIR EJEM1 @: @:	
INFO-GENERAL	DIRECTORIO : A:	BASE : EJEM1

- INFO-NIVEL : contiene el nivel o clase de la estructura jerárquica en la que se está trabajando actualmente y el nombre del objeto fijado.

- COMANDOS : es la parte de la pantalla donde se pueden escribir los comandos y sale desplegada la información solicitada.

- INFO-GENERAL : contiene la información general como el directorio actual y la base de datos que está en uso.

Definiciones generales :

NOMBRE-CLASE : es el nombre de la clase definida en la estructura de la base.

NOMBRE-OBJETO : es la llave de acceso o nombre del objeto dado de alta con anterioridad.

NOM-VAR : es el nombre de las variables de estudio en una clase dada.

Nota : cuando se haga referencia al "nivel" se entenderá por la clase.

En las siguientes secciones se da información relativa a:

- El lenguaje de esquema,
- Los comandos existentes,
- Las fórmulas entre niveles, y
- El módulo de ayuda.

Es importante señalar que las instrucciones del sistema se dividen en comandos internos y externos. Los comandos internos sólo podrán utilizarse cuando se tenga abierta una base, y los externos se pueden usar en cualquier momento, por ejemplo:

Internos: altas, bajas, cc, imprime, ordena, usar, etc.

Externos: dir, edita, cd, time, date, cls, type, mkdir, etc.

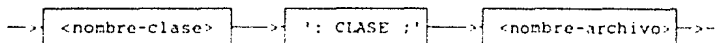
La mayoría de los comandos requieren parámetros para su operación, por esta razón deberá teclearse el comando, un espacio y a continuación el parámetro adecuado.

VIII.2 LENGUAJE DE ESQUEMA.

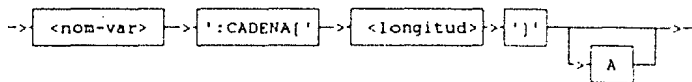
El lenguaje de esquema es el que permite al usuario definir la estructura de la base de datos que desea usar.

La sintaxis del lenguaje se mostrará a través de los siguientes diagramas de trenes:

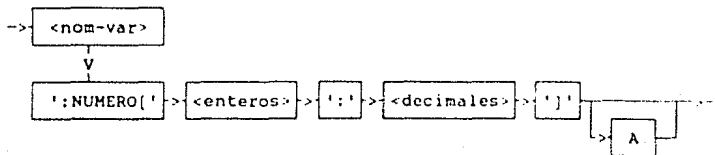
TIPO CLASE



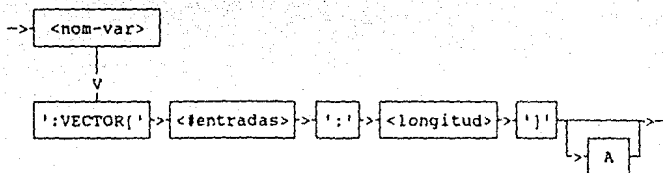
TIPO CADENA DE CARACTERES



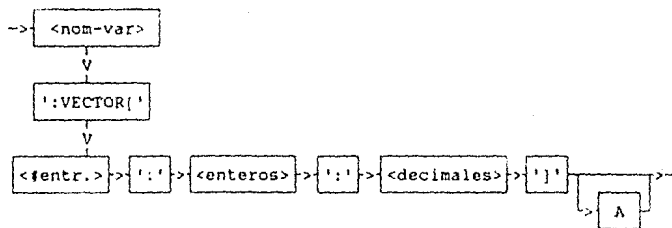
TIPO NUMERICO



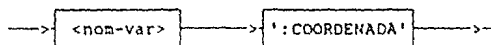
TIPO VECTOR DE CADENA DE CARACTERES



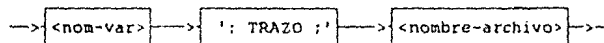
TIPO VECTOR NUMERICO



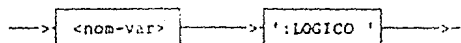
TIPO COORDENADA



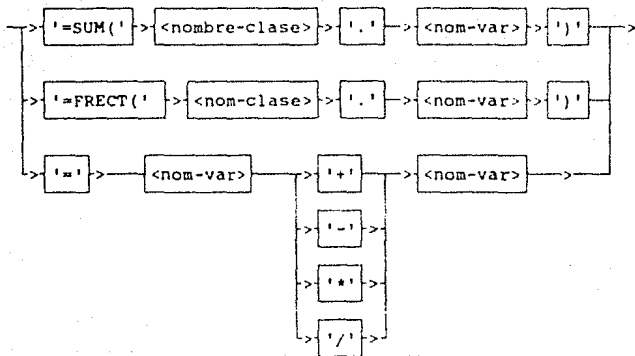
TIPO TRAZO



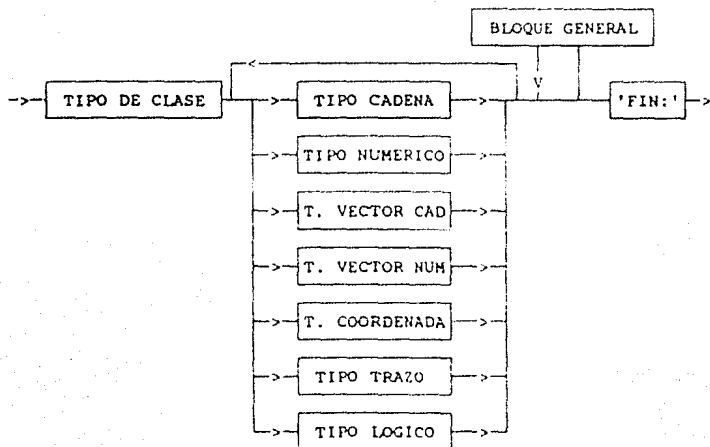
TIPO LOGICO



BLOQUE A



BLOQUE GENERAL



A continuación se mostrará un ejemplo de un esquema:

```
PAIS      : CLASE ; B_PAIS
CAPITAL   : CADENA[20]
POBLACION : NUMERO[9:0] = SUM(ESTADO.POBLACION_T)
LOCALIZACION : COORDENADA
LIMITE    : TRAZO ; PLI
ESTADO    : CLASE ; B_ESTADO
CAPITAL   : CADENA[20]
POBLACION_T : NUMERO[9:0] = POBLACION_F + POBLACION_M
POBLACION_F : NUMERO[9:0]
POBLACION_M : NUMERO[9:0]
LOCALIZACION : COORDENADA
LIMITE     : TRAZO ; ELI
FIN:
```

El esquema muestra el nivel jerárquico de la información, en donde la posición superior es el PAIS y el siguiente nivel ESTADO, es decir, que cada país (padre) podrá contener estados (hijos).

Las variables de la estructura relacional que se contemplan en cada nivel son:

- A nivel país : capital, población, localización y límite.
- A nivel estado : capital, población total, población femenina, población masculina, localización y límite.

Por otra parte se puede observar que se establecieron ciertas relaciones entre los datos de la base como son:

- La población de cada uno de los países debe ser igual a la suma de las poblaciones de cada uno de los estados que le corresponden.
- La población total de cada uno de los estados debe ser igual a la población femenina más la población masculina.

Todas estas fórmulas serán calculadas en el momento que el usuario lo indique mediante el comando "CALC".

VIII.3 COMANDOS INTERNOS DE SUPER-BASE.

Los comandos internos son aquellos que pueden ser utilizados solamente si se tiene una base de datos en uso. Debido a que están directamente relacionados con una base de datos, permiten actualizarla, desplegar su información, exportar, importar, así como definir ciertas variables importantes del sistema.

Los comandos internos de que se dispone son los siguientes:

altas	da de alta un objeto con sus características.
agrupa	agrupa variables tipo vector para su posterior despliegue.
areas	permite almacenar las áreas de variable trazo.
auto	da numeración automática a los objetos.
baja	da de baja un objeto.
calc	calcula las operaciones definidas.
cc	cambia de clase dentro de la base.
ceros	activa el switch que permite desplegar las variables de los objetos cuya información contenida sea cero.
cerrar	cierra la base activa.
digitos	establece el número de dígitos deseados para auto.
exportar	permite exportar información a formato SDF o texto.
frect	calcula la frecuencia de aparición de la variable especificada de la siguiente clase de nivel inferior.
hojear	despliega y edita la información almacenada en la clase actual.
imprime	despliega la información que contiene cada uno de los objetos o del objeto fijado.
importa	importa datos de un archivo de texto a la base.
lista	despliega los objetos contenidos en la clase.
modifica	modifica el nombre o características del objeto fijado en ese momento.
noauto	desactiva la numeración automática de los objetos.
noceros	desactiva el switch de ceros, por lo tanto no permite desplegar las variables de los objetos que contienen ceros.
renumera	renombra los objetos dándoles una numeración progresiva.
select	permite seleccionar objetos.

ordena	ordena los objetos de una clase dada.
tabla	renombra los objetos de acuerdo a una tabla de correspondencia específica.
usar	activa los campos a utilizar.
validez	permite definir el dominio local o global.

Estos comandos son descritos con detalle en el apéndice A.

VIII.4 COMANDOS EXTERNOS DE SUPER-BASE.

Los comandos externos son aquellos que pueden ser utilizados en cualquier momento con o sin base de datos en uso. Estos permiten definir el medio ambiente de trabajo, definir y abrir una base de datos, así como utilizar comandos de MS-DOS sin tener que abandonar el sistema.

Los comandos externos de que se dispone son los siguientes:

abrir	abre una base de datos.
campana	activa la campana.
car_preg	define el caracter que aparecerá en una pregunta.
cd	cambia de directorio.
cls	limpia la pantalla.
date	despliega y actualiza la fecha.
default	inicializa las variables de Super-Base.
define	inicializa los archivos de la base.
del	borra un archivo en disco.
dir	despliega los archivos de un directorio.
entrada	define la entrada de instrucciones.
edita	edita un texto.
help	despliega información de ayuda del sistema. ver sección VIII.6
memoria	despliega la cantidad de memoria disponible.
mkdir	crea un directorio.

nocampana	desactiva la campana.
prompt	cambia el prompt de Super-Base.
retardo	el tiempo de sonido de la campana.
rmdir	borra un directorio.
salida	define el dispositivo de salida.
salir	termina la ejecución de Super-Base.
time	despliega y actualiza la hora.
type	despliega el contenido de un archivo.

Estos comandos son descritos con detalle en el apéndice B.

VIII.5 OPERACIONES ENTRE NIVELES QUE SOPORTA LA BASE DE DATOS.

Las operaciones entre niveles que pueden definirse en la estructura de la base de datos son :

SUM

Es la suma de alguna de las variables definidas en la siguiente clase inferior.
Sintaxis : SUM(nom-clase.nom-var)

CTA

Indica el número de objetos que se encuentran en el siguiente nivel inferior.
Sintaxis : CTA

PROM

El promedio es la suma de los valores entre el número de objetos relativos a una variable determinada en el siguiente nivel inferior.
Sintaxis : PROM(nom-clase.nom-variable)

FREC

Indica la frecuencia de aparición de una determinada característica relativa a una cierta variable en el siguiente nivel inferior. Si se desea la tabla de frecuencias de aparición de todos los posibles valores entonces usar el comando FRECT.
Sintaxis : FREC (nom-clase.nom-vaible)

FRECR

El objetivo de esta operación es muy parecido al de FREC sólo que ésta da como resultado la frecuencia relativa, es decir, la frecuencia entre el número de objetos.
Sintaxis : FRECR (nom-clase.nom-var)

FRECRP

El objetivo de esta operación es muy parecido al de FRECR sólo que ésta da como resultado la frecuencia relativa en porcentaje, es decir, la frecuencia relativa por 100.
Sintaxis : FRECRP (nom-clase.nom-var)

Actualmente se están desarrollando otras funciones de utilidad que podrán utilizarse en versiones posteriores.

VIII.6 MODULO DE AYUDA.

El Administrador de bases de datos Super-Base provee de un módulo de ayuda, que permite la consulta rápida de información acerca del manejo del sistema de acuerdo a las necesidades del usuario.

La forma de invocar la ayuda es mediante el comando "HELP" de las siguientes formas :

1) @: HELP

a continuación se mostrará una pantalla donde están contenidas todas las instrucciones subdivididas de acuerdo a su naturaleza. El usuario tiene la posibilidad de pedir información sobre alguna en particular.

A Y U D A

COMANDOS EXTERNOS

MS-DOS

DEFINICION

AUXILIARES

A: B: C: D: E:
 CD
 CLS
 DATE
 DEL
 DIR
 MEMORIA
 MKDIR RMDIR
 RENAME
 TIME
 TYPE

ABRIR
 DEFINE
 SALIR

CAMPANA
 CARPREG
 DEFAULT
 ENTRADA
 EDITA
 HELP
 NOCAMPANA
 PROMPT
 RETARDO
 SALIDA
 UNIDADES

Teclar el Comando : _____ o <enter>

A Y U D A

COMANDOS INTERNOS

EDICION

DEFINICION

HERRAMIENTAS MICROMAP

ORDEN

AGRUPA	CC	TMPPIME	DIGITALIZA	RENUMERA
ALTAS	CERRAR	IMPORTA	DIGITALIZA_BORRA	ORDENA
AUTO	USAR	LISTA		TABLA
BAJA			<u>OPERACIONES</u>	
CEROS	<u>SELECCION</u>		CALC	
DESAGRUPA			FRECT	
DIGITOS	SELECT		AREA	
HOJEAR	DESPLIEGA			
MODIFICA	VALIDEZ			
NOAUTO				
NOCEROS				

Teclar el Comando : _____ o <enter>

2) @: HELP <comando en particular>

luego de teclear el comando anterior, se mostrará la información correspondiente al comando solicitado.

A Y U D A

ALTAS

Da de alta un objeto con sus características.

Sintaxis : ALTAS valor-numérico

Comentario : El valor-numérico representa el número de altas que se desea hacer.

Las variables de los objetos que se podrán dar de alta serán aquellas que estuvieron determinadas por el comando usar. La lectura de la información se hará del dispositivo de entrada.

PgUp

PgDn

F1/F2/Esc

14/1

Funciones especiales :

- F1 - Permite el acceso a todo el manual
- F2 - Pregunta el número de la página del manual que desea consultarse.
- Esc - Para salir de la ayuda

14/1 -> Indica que se está consultando la página 14 sección 1.

CAPITULO IX
DIVERSAS APLICACIONES

En este capítulo se presentarán algunos ejemplos de la utilización del sistema en la información referente al censo de población y vivienda de 1980, estadísticas vitales y algunos datos sobre los censos económicos.

IX.1 APLICACION CON INFORMACION CENSAL.

Para la aplicación se obtuvieron datos referentes a la cantidad de habitantes de cada estado y municipio por sexo y total desglosado por edades.

A) PREPARACION DE LA BASE DE DATOS.

En base a la información censal que se desee almacenar se procederá a definir la estructura adecuada. Este proceso está formado de los siguientes pasos:

PRIMERO. Desde el sistema operativo se ejecutará el programa "BASE" a continuación el sistema administrador presentará la pantalla que se muestra en la figura 9.1 .

e: []
DIRECTORIO : A: BASE :

FIGURA 9.1. Pantalla inicial del sistema.

SEGUNDO. Se procederá a escribirse la definición de la estructura de la base, mediante el comando "EDITA".<4>

```
@:EDITA EJEM_T1
Para terminar escribir .FIN
=>pais:clase;T1_PAIS
=>  edades:vector[21:8]
=>  hombres:vector[21:9:0]=sum(estado.hombres)
=>  mujeres:vector[21:9:0]=sum(estado.mujeres)
=>  total:vector[21:9:0]=sum(estado.total)
=>  estado:clase;T1_ESTA
=>    edades:vector[21:8]
=>    hombres:vector[21:9:0]
=>    mujeres:vector[21:9:0]
=>    total:vector[21:9:0]
=>  fin:
=>fin:
=>.FIN
@:DEFINE EJEM_T1
@:E
```

DIRECTORIO : A: BASE :

FIGURA 9.2. Definición de la estructura de la base.

TERCERO. Una vez almacenada la estructura, se procederá a la inicialización de los archivos de la base de datos, mediante el comando "DEFINE".

B) APERTURA DE LA BASE DE DATOS.

Para poder almacenar información en la base de datos es necesario abrirla, lo cual se logra utilizando el comando "ABRIR".

El sistema administrador de bases de datos indicará al usuario el nombre de la base activa y la clase inicial de trabajo, como se muestra en la figura 9.3.

4 Para mayor información sobre los comandos consultar el capítulo VIII y los apéndices A y B.

CLASE OBJETO	PAIS
@:ABRIR EJEM_T1 @: @:	
DIRECTORIO : A: BASE : EJEM_T1	

FIGURA 9.3. Forma de activar la base de datos.

C) CAPTURA DE LA INFORMACION.

Para capturar la información se utilizan diversos comandos que permiten dar de alta objetos y sus características a diferentes niveles.

PRIMERO. Antes de proceder a dar de alta deberán especificarse los campos de la estructura relacional que serán usados. Por default el sistema considera que ningún campo diferente de la llave será utilizado.

En el caso particular de este ejemplo se tecleará el comando "USAR TODO", con lo que todos los campos relacionales de ese nivel entrán en uso.

SEGUNDO. En este momento el sistema ha sido preparado para comenzar a dar de alta con las características especificadas anteriormente. Para comenzar a capturar la información se tecleará el comando "ALTAS". En la figura 9.4 se muestra un ejemplo.

CLASE OBJETO	PAIS
@:USAR TODO @:ALTAS 10	
DIRECTORIO : A: BASE : EJEM1	

CLASE OBJETO	PAIS
NOMBRE DEL OBJETO: MEXICO EDAD: [1]:0-4 [2]:5-9 . .	
DIRECTORIO : A: BASE : EJEM1	

FIGURA 9.4. Forma de dar de alta información en la base de datos.

D) CONSULTA Y MODIFICACION.

La consulta de la información puede hacerse mediante el comando "IMPRIME". En caso de tener algún error podrá modificarse mediante el comando "MODIFICA" u "HOJEAR".

Para la consulta y edición de vectores se dispone de las instrucciones "AGRUPA", "DESAGRUPA" y "HOJEAR", un ejemplo de su utilización se muestra a continuación.

CLASE OBJETO	PAIS
@:CC PAIS=MEXICO @:AGRUPA CUADRO1=EDADES,HOMBRES,MUJERES,TOTAL @:HOJEAR CUADRO1	
DIRECTORIO : A: BASE : EJEM1	

FIGURA 9.5. Forma de utilizar el comando "AGRUPA" y "HOJEAR".

PAIS : MEXICO

EDADES	HOMBRES	MUJERES	TOTAL
0-4	4698512	4649356	9347868
5-9	5172923	5111032	10283955
10-14	4574675	4519676	9094351
15-19	3766688	3889851	7656539
20-24	2972174	3182353	6154527
25-29	2325060	2479332	4804392
30-34	1885628	1952431	3827059
35-39	1664573	1742361	3406934
40-44	1359706	1385492	2745198
45-49	1134689	1180940	2315629
50-54	912884	951079	1863963
55-59	732503	733400	1465903
60-64	541862	573284	1115146
65-69	417298	458400	875698
70-74	339002	365882	704884
75-79	228660	251658	480318
80-84	132494	156872	289366
85-89	49002	65725	114727
90-94	21800	32231	54031
95-99	10711	16326	27037
100--	5709	9350	15059
NOESP	92754	100495	193249
TOTAL	33039307	33807526	66846833

FIGURA 9.6. Despliegado de la información con el comando "HOJEAR".

E) CAMBIO DE CLASE.

Una vez terminada la tarea de dar de alta y corregir los datos en un determinado nivel, se procederá a la captura del siguiente nivel. Para lo cual es necesario un cambio de clase utilizando el comando "CC", como se muestra a continuación.

CLASE OBJETO	ESTADO
@:CC PAIS=MEXICO\ESTADO @:USAR TODO @:ALTAS 10	
DIRECTORIO : A: BASE : EJEM1	

CLASE OBJETO	ESTADO
NOMBRE DEL OBJETO: CHIHUAHUA EDAD: [1]:0-4 [2]:5-9	
DIRECTORIO : A: BASE : EJEMI	

FIGURA 9.7. Cambio de clase y alta de la información.

Así se procederá sucesivamente hasta que se almacene en la base de datos toda la información deseada.

F) CALCULO DE FORMULAS.

El sistema administrador solamente efectúa el cálculo de las fórmulas definidas en el esquema de la base de datos cuando el usuario lo indique con el comando "CALC".

La modificación de un dato realizado después del cálculo de las fórmulas no afectará el resultado de estas contenido en los niveles superiores hasta el momento en que se efectúe nuevamente el cálculo.

CLASE OBJETO	PAIS
@:CALC Esperar un momento calculando las fórmulas... @:	
DIRECTORIO : A: BASE : EJEMI	

FIGURA 9.8. Cálculo de fórmulas.

G) RESULTADOS.

Los resultados pueden obtenerse de diversas formas: desplegarse en pantalla, listarse en impresora o guardarse en un archivo. Como se vió en el inciso D, el despliegue de vectores se logra mediante el comando "HOJEAR" indicando el cuadro que se desea consultar.

IX.2 APLICACION CON INFORMACION DEMOGRAFICA.

En este caso se obtuvieron datos de estadísticas vitales referentes a la mortalidad por estado de las tablas de "Defunciones generales, por causas, según la clasificación internacional de enfermedades" para el periodo de 1980. Donde la clasificación internacional de enfermedades es la siguiente:

GRUPO	ESPECIFICACION
I	Enfermedades infecciosas y parasitarias
II	Tumores
III	Enfermedades de las glándulas endocrinas, de la nutrición, del metabolismo y trastornos de la inmunidad
IV	Enfermedades de la sangre y de los órganos hematopoyéticos
V	Trastornos mentales
VI	Enfermedades del sistema nervioso y de los órganos de los sentidos.
VII	Enfermedades del aparato circulatorio
VIII	Enfermedades del aparato respiratorio
IX	Enfermedades del aparato digestivo
X	Enfermedades del aparato genitourinario
XI	Complicaciones del embarazo, del parto y del puerperio
XII	Enfermedades de la piel y del tejido celular subcutáneo
XIII	Enfermedades del sistema osteomuscular y del tejido conjuntivo
XIV	Anomalías congénitas
XV	Ciertas afecciones originadas en el periodo perinatal
XVI	Signos, síntomas y estados morbosos mal definidos
XVII	Clasificación suplementaria de causas externas de traumatismo y envenenamiento
NESP	No especificada

La estructura adecuada para esta base de datos se muestra a continuación.

```
EJEMPLO2.DEF  
  
PAIS:CLASE;T2_PAIS  
  I:NUMERO[5:0]  
  II:NUMERO[5:0]  
  III:NUMERO[5:0]  
  IV:NUMERO[5:0]  
  V:NUMERO[5:0]  
  VI:NUMERO[5:0]  
  VII:NUMERO[5:0]  
  VIII:NUMERO[5:0]  
  IX:NUMERO[5:0]  
  X:NUMERO[5:0]  
  XI:NUMERO[5:0]  
  XII:NUMERO[5:0]  
  XIII:NUMERO[5:0]  
  XIV:NUMERO[5:0]  
  XV:NUMERO[5:0]  
  XVI:NUMERO[5:0]  
  XVII:NUMERO[5:0]  
  NESP:NUMERO[5:0]  
  TOTAL:NUMERO[5:0]  
ESTADO:CLASE;T2_ESTA  
  I:NUMERO[5:0]  
  II:NUMERO[5:0]  
  III:NUMERO[5:0]  
  IV:NUMERO[5:0]  
  V:NUMERO[5:0]  
  VI:NUMERO[5:0]  
  VII:NUMERO[5:0]  
  VIII:NUMERO[5:0]  
  IX:NUMERO[5:0]  
  X:NUMERO[5:0]  
  XI:NUMERO[5:0]  
  XII:NUMERO[5:0]  
  XIII:NUMERO[5:0]  
  XIV:NUMERO[5:0]  
  XV:NUMERO[5:0]  
  XVI:NUMERO[5:0]  
  XVII:NUMERO[5:0]  
  NESP:NUMERO[5:0]  
  TOTAL:NUMERO[5:0]  
  
FIN:  
FIN:
```

FIGURA 9.9. Definición de la estructura de la base.

Una vez almacenada la definición de la base se procederá a "DEFINIR" y "ABRIR" la base de datos. Así como a dar de "ALTA" los datos, de la misma manera como se hizo en la sección anterior.

CLASE OBJETO	PAIS
@:DEFINE EJEMPLO2 @:ABRIR EJEMPLO2 @:USAR TODO @:ALTAS 1 NOMBRE DEL OBJETO: MEXICO I: 128 @:CC PAIS=MEXICO\ESTADO	
DIRECTORIO : A: BASE : EJEMPLO2	
CLASE OBJETO	ESTADO
@:USAR TODO @:ALTAS 32 NOMBRE DEL OBJETO: AGUASCALIENTES I: 340	
DIRECTORIO : A: BASE : EJEMPLO2	

FIGURA 9.10. Cambio de clase y alta de la información.

Cuando se cometen errores en la captura, podrá utilizarse el comando "HOJEAR" que permite el despliegado de los campos en forma de tabla y su edición.

Una vez teclado el comando "HOJEAR", mediante el uso de las flechas, se podrá colocar el cursor en la posición del campo y registro que desea modificarse. Al utilizar la tecla "F2" podrá editarse su contenido.

Objeto	I	II	III	IV	V	VI	VII	VIII	IX	X
AGUASCALIE	548	286	252	24	56	55	497	468	213	69
BAJA CALIF	434	287	239	27	48	56	520	452	207	58
CAMPECHE	437	274	276	21	65	56	465	389	198	59

Objeto	XI	XII	XIII	XIV	XV	XVI	XVII	NESP	TOTAL
AGUASCALIE	18	6	16	99	288	62	439	1	3397
BAJA CALIF	13	91	27	94	298	642	4594	0	3324
CAMPECHE	14	10	22	82	294	40	445	2	3149

FIGURA 9.11. Modificación y consulta con "HOJEAR".

La información puede consultarse y desplegarse en cualquier momento que sea necesario, mediante los comandos "HOJEAR" o "IMPRIME". A continuación se muestran algunos de los reportes que pueden obtenerse.

PAIS	: MEXICO
ESTADO	: AGUASCALIENTES
I :	548
II :	386
III :	252
IV :	24
V :	56
VI :	55
VII :	497
VIII :	468
IX :	213
X :	69
XI :	18
XII :	6
XIII :	16
XIV :	99
XV :	288
XVI :	62
XVII :	439
NESP :	1
TOTAL :	3397

S - para salir

FIGURA 9.12. Ejemplo del comando "IMPRIME".

IX.3 APLICACION DE SUPER-BASE CON MICROMAP A CARTOGRAFIA CENSAL.

En esta sección se presenta una aplicación de MICROMAP y SUPER-BASE. Los datos utilizados son la información estadística de la localidad Nuevo Ideal, Dgo, referente a los establecimientos económicos y población por manzana.

Objeto	XI	XII	XIII	XIV	XV	XVI	XVII	NESP	TOTAL
AGUASCALIE	18	6	16	99	288	62	439	1	3397
BAJA CALIF	13	91	27	94	298	642	4594	0	3324
CAMPECHE	14	10	22	82	294	40	445	2	3149

FIGURA 9.11. Modificación y consulta con "HOJEAR".

La información puede consultarse y desplegarse en cualquier momento que sea necesario, mediante los comandos "HOJEAR" o "IMPRIME". A continuación se muestran algunos de los reportes que pueden obtenerse.

PAIS	: MEXICO
ESTADO	: AGUASCALIENTES
I :	548
II :	286
III :	252
IV :	24
V :	56
VI :	55
VII :	497
VIII :	468
IX :	213
X :	69
XI :	18
XII :	6
XIII :	16
XIV :	99
XV :	288
XVI :	62
XVII :	439
NESP :	1
TOTAL :	3397

3 - para salir

FIGURA 9.12. Ejemplo del comando "IMPRIME".

IX.3 APLICACION DE SUPER-BASE CON MICROMAP A CARTOGRAFIA CENSAL.

En esta sección se presenta una aplicación de MICROMAP y SUPER-BASE. Los datos utilizados son la información estadística de la localidad Nuevo Ideal, Dgo, referente a los establecimientos económicos y población por manzana.

A) DEFINICION DE LA BASE DE DATOS Y ALMACENAMIENTO DE LA INFORMACION.

El nombre dado a la base de datos fué IDEAL2 que tiene la estructura siguiente :

IDEAL2.DEF

```

MUNICIPIO:CLASE;M_MUNI1
SUPERFICIE:NUMERO[9:2]=SUM(LOCALIDAD.SUPERFICIE)
POBLACION:NUMERO[9:0]=SUM(LOCALIDAD.POBLACION)
ESTABLE_ECON:NUMERO[4:0]=SUM(LOCALIDAD.ESTABLE_ECON)
LIMITE:TRAZO;LIE
LOCALIDAD:CLASE;M_LOCI
SUPERFICIE:NUMERO[9:2]=SUM(AGEB.SUPERFICIE)
POBLACION:NUMERO[9:0]=SUM(AGEB.POBLACION)
ESTABLE_ECON:NUMERO[4:0]=SUM(AGEB.ESTABLE_ECON)
LIMITE:TRAZO;LIL
AGEB:CLASE;M_AGEB1
SUPERFICIE:NUMERO[9:2]
POBLACION:NUMERO[9:0]=SUM(MANZANA.POBLACION)
ESTABLE_ECON:NUMERO[4:0]=SUM(MANZANA.ESTABLE_ECON)
LIMITE:TRAZO;LIA
MANZANA:CLASE;M_MANZ1
SUPERFICIE:NUMERO[9:2]
POBLACION:NUMERO[9:0]
ESTABLE_ECON:NUMERO[2:0]
LIMITE:TRAZO;LMZ
FIN:
FIN:
FIN:

```

La información de Nuevo Ideal se tenía originalmente en DBASE por lo que fué importada a SUPER-BASE mediante el comando "IMPORTA" especificando cada una de las variables contenidas y su posición en la jerarquía.

CLASE OBJETO	MUNICIPIO
@:DEFINE IDEAL2	
@:ABRIR IDEAL2	
@:IMPORTA DATOS.	
.	
.	
.	
DIRECTORIO :	A: BASE : IDEAL2

FIGURA 9.13. Almacenamiento de la información en la base de datos.

B) DIGITALIZACION DE MAPA CON MICROMAP.

Por otra parte, también se digitalizó el plano de Nuevo Ideal a nivel de manzanas, las cuales están numeradas dentro de los AGEBS que comprende la localidad. Donde se quiere relacionar este mapa con los datos de número de establecimientos económicos (variable ESTABLE_ECON) por manzana y asurarlos según diferentes rangos. El mapa se puede ver en la Figura 9.17.

C) ENLACE ENTRE MICROMAP Y SUPER-BASE.

Para establecer el enlace entre los objetos de la base de datos y los trazos es necesario llevar a cabo los siguientes pasos:

CLASE OBJETO	MUNICIPIO
E:CC =039\LOCALIDAD=0001\AGEB=002-8\MANZANA E:DIGITALIZA LIMITE	
DIRECTORIO : A: BASE : IDEAL2	

FIGURA 9.14. Instrucciones necesarias para llevar a cabo el enlace.

La primera instrucción significa que cambiamos de clase (CC) del municipio 039 a través de la localidad 0001, que es la clave Nuevo Ideal, a través del AGEB 002-8 y llegamos hasta el nivel MANZANA.

La segunda instrucción tiene como objetivo relacionar los trazos de las manzanas con los datos, para lo cual se teclea el comando "DIGITALIZA" <variable-trazo> que ubicará al usuario en MICROMAP en el mapa de Nuevo Ideal, que estaba digitalizando, y le permitirá realizar operaciones de ALTAS, BAJAS, CAMBIOS y CONSULTAS en las relaciones trazo-dato y dato-trazo para la <variable-trazo> especificada a nivel manzana.

D) DETERMINACION DE LA SELECCION.

Para poder asurar el mapa es necesario llevar a cabo una selección de acuerdo a un cierto criterio. Este debe indicar los grupos a formar y su identificador o tipo, por ejemplo:

TIPO	CONDICION
1	ESTABLE ECON = 1
2	2 <= ESTABLE_ECON <= 5
3	6 <= ESTABLE_ECON <= 10
4	11 <= ESTABLE_ECON

El tipo 1 de ashurado agrupará las manzanas con un establecimiento económico, el segundo agrupará las manzanas que tienen de 2 a 5 establecimientos económicos, etc.

El primer paso para llevar a cabo la selección es regresar a la base de datos y continuar con los pasos que se muestran a continuación:

CLASE OBJETO	MUNICIPIO
@:CC =039\LOCALIDAD=0001\AGEB=003-2\MANZANA	
DIRECTORIO : A: BASE : IDEAL2	

CLASE OBJETO	MANZANA
@:VALIDEZ GLOBAL @:SELECT .SEL Nombre de la seleccion : SELECT1 Generando el archivo de seleccion: SELECT1.SEL Para terminar teclear ".FIN" Condicion->ESTABLE_ECON=1 Tipo->1 Condicion->2<=ESTABLE_ECON<=5 Tipo->2 Condicion->6<=ESTABLE_ECON<=10 Tipo->3 Condicion->11<=ESTABLE_ECON Tipo->4 Condicion->.FIN @:	
DIRECTORIO : A: BASE : IDEAL2	

FIGURA 9.15. Forma de efectuar una selección.

La validez debe ser global si se quieren ashurar de la misma forma todas las manzanas de todos los AGEBS.

Para llevar a cabo la selección se tecleará el comando "SELECT" y se irán dando los datos de acuerdo a el criterio definido.

E) DESPLIEGUE DE LOS OBJETOS SELECCIONADOS.

Para ver en pantalla el resultado de la selección se utiliza el comando "DESPLIEGA".

CLASE OBJETO	MANZANA
<pre> @:DESPLIEGA Condicion->ESTABLE_ECON=1 Tipo->1 Condicion->2<=ESTABLE_ECON<=5 Tipo->2 Condicion->6<=ESTABLE_ECON<=10 Tipo->3 Condicion->11<=ESTABLE_ECON Tipo->4 TIPO MANZANA 1 01 1 02 2 03 ... @: </pre>	
DIRECTORIO : A: BASE : IDEAL2	

FIGURA 9.16. Despliega los objetos seleccionados.

F) ASHURADO DE MAPAS.

Para ashurar el mapa una vez determinada una selección se procederá a entrar a MICROMAP y solicitar su ashurado estableciendo color, inclinación y separación de las líneas.

Así un mapa resultado es como el que se muestra en la figura 9.18.

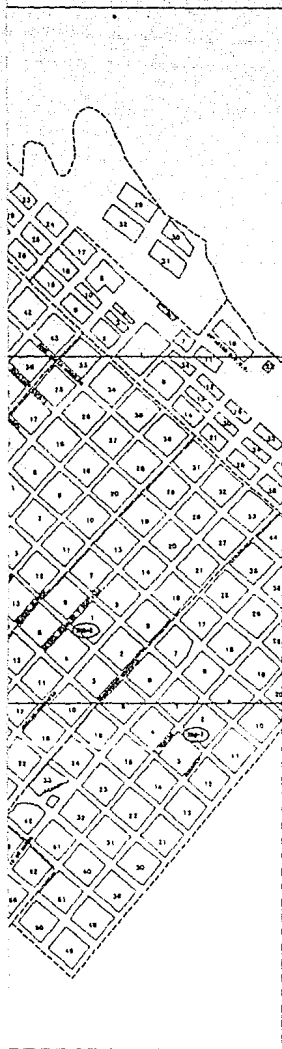
ESTADO DURANGO

CLAVE : 10

MUNICIPIO
NUEVO IDEAL

CLAVE : 039

LOCALIDAD,
NUEVO IDEAL
CLAVE : 0001



ESTADO DURANGO

CLAVE : 10

MUNICIPIO

NUEVO IDEAL

CLAVE : 039

LOCALIDAD,

NUEVO IDEAL

CLAVE : 0001

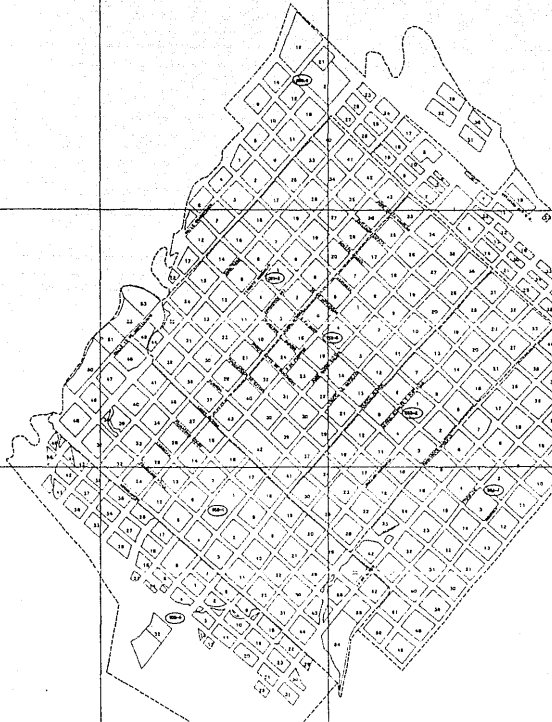


Fig. 9.17 Mapa de Nuevo Ideal

ESTADO DURANGO

CLAVE : 10

MUNICIPIO

NUEVO IDEAL

CLAVE : 039.

LOCALIDAD

NUEVO IDEAL

CLAVE : 0001

NUMERO DE
ESTABLECIMIENTOS ECONOMICOS
POR MANZANA



1



DE 2 A 5



DE 6 A 10



MAS DE 11

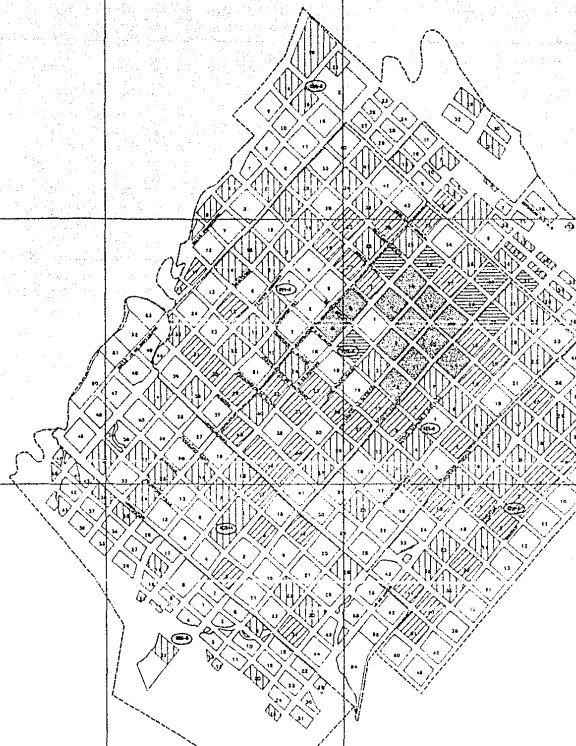


Figura 9.18 Mapa Resultado de Nuevo Ideal

CONCLUSION

A manera de conclusión quisiéramos comentar algunas ideas que nos parecen interesantes:

- 1) las expectativas de esta base de datos a futuro y
- 2) el trabajo que queda aún por hacer.

1. Expectativas a futuro.

El manejo de la información censal podría llevarse a cabo utilizando esta base de datos, así como el procesamiento de todo tipo de información que tenga una estructura jerárquica implícita.

Creemos que la implementación que hemos desarrollado es un ejemplo de la forma en que una base de datos jerárquica y relacional puede funcionar y de la implementación de la sintaxis y comandos para el manejo de los datos en ambas estructuras. Sin embargo pensamos que el concepto de SUPER-BASE va más allá de esta implementación y que su sintaxis y comandos representan un concepto general.

Lo anterior es importante pues podrán hacerse implementaciones de SUPER-BASE (manteniendo sintaxis y conceptos) en otros lenguajes y en otro tipo de máquinas. Esperamos en esta tesis haber definido el concepto general y las bases de una sintaxis que pudiera llegar a adoptarse como standard.

La sintaxis del administrador de bases de datos definida en esta tesis se asemeja en gran medida a la forma natural en que piensa el ser humano en las estructuras jerárquicas. Al hojear los libros del censo encuentra una esta estructura: las primeras páginas presentan resultados a nivel país (totales), las páginas siguientes (y en algunos casos los volúmenes siguientes) contendrán datos a un nivel inferior, como podría ser el caso de los municipios de cada estado, y así sucesivamente. Por esta razón creemos que a pesar de que la implementación de un administrador de bases de datos basado en este concepto puede ser complicada, vale la pena desde el punto de vista de claridad en el manejo de la información y en el ahorro de espacio de almacenamiento y tiempos de consulta.

Creemos que existen numerosas aplicaciones a este tipo de administradores que no están necesariamente relacionadas con la información socioeconómica o geográfica. Un ejemplo de dicha aplicación podría ser un archivo de profesores en una universidad, en donde el curriculum de cada uno se almacena, así como las materias que ha impartido en cada semestre. El esquema para este caso podría ser:

```

profesor:clase;profe
  calle:cadena[20]
  números:cadena[10]
  estudios:clase;estu
    grado:cadena[20]
    universidad:cadena[20]
    fecha:cadena[10]
  fin:
materias:clase;mate
  clave:cadena[4]
  inscritos:numero[3:0]
  fin:
fin:

```

Lo anterior muestra que un administrador basado en la sintaxis descrita puede utilizarse para un gran número de propósitos y aplicaciones.

2. Sobre el trabajo pendiente.

Ningún trabajo estará completo si no se indica lo que falta por hacer. Presentaremos en esta sección lo que creemos que está pendiente.

Desde el punto de vista de la implementación pensamos que falta revisar algunas algoritmos y hacerlos más eficientes tratando de reducir al mínimo el número de lecturas a disco pues esto consume bastante tiempo de procesamiento.

Falta terminar la definición de la sintaxis para el programa generador de reportes jerárquicos. Un generador de reportes relacionales es fácil de implementar, pero al introducir una jerarquía el problema se complica.

Una vez definida una sintaxis para el generador de reportes jerárquicos se requiere lograr una implementación adecuada para que sea de fácil uso desde el punto de vista de los usuarios.

Por otra parte, la implantación de este administrador de bases de datos para el manejo de la información censal a nivel nacional presenta ciertos problemas a considerar:

- La limitación de la capacidad de almacenamiento en una Computadora Personal,
- La consistencia de la información en los diferentes estados,
- La centralización de la información, y
- La seguridad de la información y claves de acceso según privilegios.

Una posible solución que proponemos para resolverlos es ampliar la capacidad del administrador para que pueda trabajar en red, en donde en cada estado habría al menos un nodo. El Instituto Nacional de Estadística, Geografía e Informática establecería los lineamientos internos del manejo de la información que reglamentarían a todos los estados. Cada estado sería responsable de la consistencia de su información y tendría la obligación de asignar responsables de la seguridad de su base de datos.

Para fines prácticos, creemos que lo más importante ya está hecho y esperamos que pueda ser de utilidad.

APENDICE A

COMANDOS INTERNOS DE SUPER-BASE.

Los comandos internos son aquellos que pueden ser utilizados solamente si se tiene una base de datos en uso. Debido a que están directamente relacionados con una base de datos, permiten actualizarla, desplegar su información, exportar, importar, así como definir ciertas variables importantes del sistema.

A continuación se da una explicación detallada de cada uno de los comandos.

AGRUPA

Agrupar variables de tipo cadena para su posterior edición.

Sintaxis : AGRUPA <nombre-cuadro>=<nom-var>[,<nom-var>[,...]]

Comentario : El nombre-cuadro representa el nombre del cuadro que define el usuario.

Las variables deben de ser de tipo vector, tanto tipo carácter como numérico, siempre y cuando posean la misma longitud.

Ejemplo :

@: AGRUPA CUADRO1=EDADES,TOTAL,MASCULINO,FEMENINO

@:

Ver : HOJEAR.

ALTAS

Da de alta un objeto con sus características.

Sintaxis : ALTAS valor-numérico

Comentario : El valor-numérico representa el número de altas que se desea hacer.

Las variables de los objetos que se podrán dar de alta serán aquellas que estuvieron determinadas por el comando usar. La lectura de la información se hará del dispositivo de entrada.

Ejemplo :

@: USAR nada
@: ALTAS 1
NOMBRE DEL OBJETO : PAIS1
@:
@: USAR todo
@: ALTAS 1
NOMBRE DEL OBJETO : PAIS2
POBLACION : 530
@: LISTA

OBJETOS DE LA CLASE : PAIS

PAIS1
PAIS2

@:

Ver : AUTO, ENTRADA, LISTA.

AREA

Permite calcular el área encerrada por un trazo determinado y almacenarlo en una variable de tipo numérico.

Sintaxis : AREA <nom-var tipo trazo>['>' <nom-var tipo numérico>]

Comentario : Existe dos formas de utilizar este comando. El primero solo solicita que se calcule el área del objeto que se

tiene fijado en ese momento de acuerdo a una variable trazo. El segundo almacena cada una de las áreas en la base de datos en la variable especificada para todos los objetos de ese nivel por cualquier trayectoria. Las unidades en que calcula el área esta dado por el comando "UNIDADES" por default son Hectáreas.

Ejemplos:

Tipo 1:

@: CC \PAIS=MEXICO\EXTADO=AGUASCALIENTES

@: AREA LIMITE

El área del objeto AGUASCALIENTES es:

@: UNIDADES K

@: AREA LIMITE

El área del objeto AGUASCALIENTES es:

Tipo 2:

@:

@:AREA LIMITE>SUPERFICIE

ESPERAR UN MOMENTO CALCULANDO LAS AREAS

@:

Ver : UNIDADES.

AUTO

Proporciona numeración automática a los objetos.

Sintaxis : AUTO valor-numérico

Comentario : El valor-numérico será un número apartir del cual comenzará en forma progresiva a darle un nombre numérico a los objetos que sean dados de alta, con tantos digitos como fuera

establecido por el comando DIGITOS.

Ejemplo :

@: DIGITOS 3

@: AUTO 5

@: ALTAS 2

NOMBRE DEL OBJETO : 005

<- nombre de objeto automático

...

NOMBRE DEL OBJETO : 006

<- nombre de objeto automático

...

@:

Ver : DIGITOS, ALTAS.

BAJA

Da de baja un objeto de una clase determinada.

Sintaxis : BAJA nombre-objeto

Comentario : Este comando debe ser ejecutado con precaución ya que una vez dado de baja un objeto no se podrá recuperar la información.

Nota : Un objeto no podrá ser dado de baja mientras contenga información a un nivel inferior.

Ejemplo :

@: LISTA

OBJETOS DE LA CLASE : PAIS

PAIS1 PAIS2

@: BAJA PAIS1

@: LISTA

OBJETOS DE LA CLASE : PAIS

PAIS2

@:

Ver : ALTAS.

CALC

Calcula las operaciones definidas en la base.

Sintaxis : CALC

Comentario : Para información acerca de las operaciones que pueden definirse en la base ver la sección VIII.5.

Ejemplo :

@: IMPRIME

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS2
POBLACION : 0

@: CALC

Calculando todas las fórmulas, favor de esperar un momento

@: IMPRIME

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS2
POBLACION : 320

@:

Ver : sección VIII.5 .

CC

Cambia de clase o nivel dentro de la base.

Sintaxis : CC [\][nombre-clase][=nombre-objeto]

[\nombre-clase[=nombre-objeto[.]]]

Comentario : En el momento de ejecutar el CC colocará al usuario en la trayectoria especificada y actualizará la sección de

info-nivel.

El funcionamiento de esta instrucción es muy parecido al de "CD o CHDIR" de MSDOS para el manejo de subdirectorios.

Ejemplos:

```
@: CC \ <-- cambio a la clase principal o raíz.
@: CC \PAIS=MEXICO\ESTADO <-- cambio directo a la clase ESTADOS
                                del país MEXICO.
@: CC =AGUASCALIENTES <-- El objeto fijado es "AGUASCALIENTES"
@: CC = <-- ningún objeto fijado.
@: CC LOCALIDAD <-- cambia a la siguiente clase que se
                                llama LOCALIDAD.
@: CC .. <-- cambio a la siguiente clase superior.
```

CEROS

Activa el switch que permite desplegar las variables de los objetos cuya información contenida sea cero.

Sintaxis : CEROS

Ejemplo :

```
@: CEROS
@: IMPRIME
```

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS2
POBLACION : 320

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS3
POBLACION : 0

@:

Ver : NOCEROS.

CERRAR

Cierra la base activa.

Sintaxis : CERRAR

Comentario : También actualizará la sección de info-general.

Ejemplo :

@: CERRAR

Ver : ABRIR.

DESPLIEGA

Permite al usuario desplegar en pantalla cuales son los objetos que cumplieron con una determinada selección.

Sintaxis : DESPLIEGA

Comentario : Este comando muestra en pantalla cuales fueron las condiciones que se solicitaron y los objetos que lo cumplieron y la condición que cumplieron. La utilidad principal es que permite generar asurados sobre el mapa de acuerdo a la selección.

Ejemplo :

@: DESPLIEGA

@:

Ver : SELECT,VALIDEZ.

DIGITALIZA

Permite al usuario dar de alta trazos de MICROMAP en la base de datos.

Sintaxis : DIGITALIZA <nom-var de tipo trazo>

Comentario : Se pueden tener diferentes tipos de variable trazo es por esta razón que debe especificarse la variable tipo trazo que desea dar de alta en la base de datos. Esta opción sólo podrá usarse si se invoco a la base de datos por MICROMAP.

Ejemplo :

@: DIGITALIZA LIMITE

(a continuación regresara a MICROMAP y podrán darse de alta los trazos)

DIGITOS

Establece el número de dígitos deseados para el comando AUTO.

Sintaxis : DIGITOS valor numérico

Comentario : El valor numérico será la cantidad de dígitos que contendrá el nombre numérico del objeto, cuando este encendido el switch de auto.

Ejemplo :

@: DIGITOS 5

@: AUTO 20

@: ALTAS 2

NOMBRE DEL OBJETO : 00020 <- nombre de objeto automático

...
NOMBRE DEL OBJETO : 00021 <- nombre de objeto automático

...

@:

VER : AUTO, ALTAS.

EXPORTAR

Permite exportar información de la base de datos a formato de texto o SDF.

Sintaxis : EXPORTAR <nombre-de-definición>

Comentario : El comando exportar despliega la información en el dispositivo de salida. En particular, si se desea almacenar en un archivo de texto deberá asignarse el dispositivo de salida a un archivo determinado y al terminar de ejecutarse deberá definirse como PANTALLA al dispositivo de salida.

Ejemplo :

@: SALIDA DATOS3

@: EXPORTAR DATOS3

@: SALIDA PANTALLA

@:

Ejemplo del archivo de definición :

Los datos deberán estar por pares

- 1er renglón nombre de la variable con trayectoria
- 2do renglón

+ posición inicial +longitud total + número de decimales
 incluidos

Nota: aunque la variable sea alfanumérica deberán escribirse el número de decimales.

TYPE DATOS3.EXP

```
PAIS.LLAVE
10 8 0
PAIS.POBLACION_TOTAL
20 8 0
PAIS.ESTADO.LLAVE
30 10 0
PAIS.ESTADO.POBLACION_TOTAL
20 8 3
```

Ver : IMPORTAR.

FRECT

Calcula la frecuencia de aparición de la variable especificada del nivel inferior de acuerdo al objeto que se tiene fijado en ese momento.

Sintaxis : FRECT nombre-clase.nom-var

Comentario : Una vez tecleada la instrucción desplegara en el dispositivo de salida cada uno de los posibles valores de la variable indicada con sus correspondientes frecuencias de aparición.

Ejemplo : Si se encuentra en el nivel AGEb y se tiene definido el objeto 007-0.

@: FRECT manzana.tipo_comercio

C A L C U L O D E F R E C U E N C I A S

CLASE :MUNICIPIO
OBJETO:039
CLASE :LOCALIDAD
OBJETO:0001
CLASE :AGEB
OBJETO:007-0

VARIABLE DE ESTUDIO : MANZANA.ESTABLE_ECON

VAR FRECUENCIA

1 =	14
2 =	1
3 =	3
8 =	1

HOJEAR

Despliega en la pantalla una tabla bidimensional donde en cada renglón se encuentran todos los objetos dados de alta y en las columnas las variables especificadas por el comando usar o por el comando agrupa.

El usuario, en este momento, tiene la posibilidad de mover el cursor mediante las teclas de las flechas y oprimir "F2" para editar y modificar información.

Sintaxis : HOJEAR [<nombre-cuadro>]

Comentario : La llave del objeto no puede ser modificada en Hojear, para llevarlo a cabo es necesario fijar el objeto y utilizar "MODIFICA LLAVE".

En caso de especificar el nombre del cuadro que se desea hojear podrán verse los valores de los vectores especificados por

el comando agrupa y correspondientes al objeto fijado con anterioridad, este tiene la función de visualizar los vectores agrupados como una tabla.

Otra forma de imprimir la información es mediante "IMPRIME".

Otra forma de modificar los campos es mediante "MODIFICA".

Ejemplo :

@: HOJEAR

OBJETO	CAPITAL	POBLACION	SUPERFICIE
ESPAÑA	MADRID	120000	123000
FRANCIA	PARIS	80000	126000
ITALIA	ROMA	90000	80000
MEXICO	D.F.	75000	235000

Ver : MODIFICA, IMPRIME, AGRUPA.

IMPRIME

Despliega la información que contiene cada uno de los objetos o del objeto fijado.

Sintaxis : IMPRIME

Comentario : Las variables de los objetos que contengan ceros serán impresas según sea indicado por los comandos CEROS o NOCEROS.

Ejemplo :

@: CEROS

@: IMPRIME

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS2
POBLACION : 320

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS3
POBLACION : 0

@: NOCEROS
@: IMPRIME

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS2
POBLACION : 320

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS3

@:

Ver : BROWSE.

IMPORTA

Importa la información que está contenida en un archivo de texto, en la base que se tiene abierta en ese momento.

Sintaxis : IMPORTA

Comentario : Mediante esta opción se establece cuales son las variables que se desean importar y su localización. El sistema automáticamente jerarquiza la información de acuerdo a la definición dada. También permite almacenar la información de acuerdo a una trayectoria inicial.

Ejemplo :

@: IMPORTA

Nombre del archivo de datos a importar (FIN para salir) ---> UNO

Numero de campos ---> 4

Favor de teclear la trayectoria base ---> \

Campo # 1

pos. inicial ---> 1

pos. final ---> 10

trayectoria (respecto a la trayectoria anterior) --->

variable (vacio ==> llave) --->

(es la llave de MUNICIPIO)

Campo # 2

pos. inicial ---> 13

pos. final ---> 17

trayectoria (respecto a la trayectoria anterior) --->

variable (vacio ==> llave) ---> SUP

Campo # 3

pos. inicial ---> 20

pos. final ---> 25

trayectoria (resp. a la trayectoria anterior) ---> LOCALIDAD
(entonces es la llave)

Campo # 4

pos. inicial ---> 32

pos. final ---> 37

trayectoria (respecto a la trayectoria anterior) --->

variable (vacio ==> llave) ---> CLAVE

FAVOR DE ESPERAR UN MOMENTO IMPORTANDO LA INFORMACION

@:

Ver : EXPORTAR.

LISTA

Despliega los objetos contenidos en la clase.

Sintaxis : LISTA

Ejemplo :

@: LISTA

OBJETOS DE LA CLASE : PAIS

PAIS2
PAIS3
PAIS4

@:

MODIFICA

Modifica el nombre o características de un objeto.

Sintaxis : MODIFICA

Comentario : Para utilizar este comando primero se debe fijar un objeto (CC =), y al teclear MODIFICA se podrán cambiar el nombre o características del objeto fijado. Las variables que modificará serán aquellas que estén especificadas en usar.

Ejemplo :

@: CC =PAIS2

@: IMPRIME

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS2
POBLACION : 320

@: MODIFICA

POBLACION : 330

@: IMPRIME

OBJETOS DE LA CLASE : PAIS

OBJETO : PAIS2
POBLACION : 330

@:

Ver : BROWSE.

NOAUTO

Desactiva la numeración automática de los objetos.

Sintaxis : NOAUTO

Comentario : El efecto de este comando se puede ver claramente al dar de alta un objeto. Cuando se tiene AUTO el nombre del objeto se dará automáticamente como un número con tantos dígitos como se haya especificado y cuando es NOAUTO el usuario podrá dar libremente el nombre del objeto.

Ejemplo : Ver ALTAS y AUTO.

NOCEROS

Desactiva el switch de ceros, por lo tanto no permite desplegar las variables de los objetos que contienen ceros.

Sintaxis : NOCEROS

Ejemplo : Ver CEROS, IMPRIME.

RENUMERA

Renombra los objetos dandoles una numeración progresiva.

Sintaxis : RENUMERA archivo

Comentario : La renumeración de los objetos es arbitraria y el número de dígitos queda definido por el comando DIGITOS. La correspondencia anterior y actual queda almacenada en el archivo especificado.

Ejemplo :

@: LISTA

OBJETOS DE LA CLASE : PAIS

PAIS2
PAIS3
PAIS4

@: DIGITOS 3

@: RENUMERA Corres1.txt

@: LISTA

OBJETOS DE LA CLASE : PAIS

001
002
003

@: TYPE Corres1.txt

RENUMERACION DE OBJETOS DE :

CLASE : PAIS

OBJETO :

PAIS2	---> 001
PAIS3	---> 002
PAIS4	---> 003

@:

Ver : TABLA.

SELECT

Permite seleccionar objetos que cumplan con cierta condición de acuerdo a la validez.

Sintaxis : SELECT [parámetro]

Comentario : Existen diferentes formas de hacer una selección como: una selección sencilla, una selección múltiple o si se desea volver a hacer una selección que ya se hizo con anterioridad sólo deberá teclearse el nombre del archivo donde se guardo la selección.

El parámetro puede ser :

1) Directamente una condición como :

```
SELECT 10<POBLACION_TOTAL<100
```

que tendrá asignado el tipo 1 de default.

2) No se especifica una selección determinada sino que se va a generar una nueva:

```
SELECT .SEL
```

Para terminar escribir .FIN

```
Condición --> 0 <= ESTABLE_ECON <= 4
```

```
Tipo --> 1
```

```
Condición --> 5 <= ESTABLE_ECON < 7
```

```
Tipo --> 2
```

```
Condición --> 7 <= ESTABLE_ECON
```

```
Tipo --> 3
```

```
Condición --> .FIN
```

3) Un determinado nombre de archivo *.SEL que ya tiene almacenada una selección :

```
SELECT UNO.SEL
```

Ver : VALIDEZ.

ORDENA

Ordena los objetos de una clase dada.

Sintaxis : SORT [A][D]

Comentario : La forma en que ordena los objetos será :

default - (teclear solamente SORT) es ascendente,

"A" - ascendente,

"D" - descendente.

Ejemplo :

@: LISTA

OBJETOS DE LA CLASE : PAIS

CANADA MEXICO ALEMANIA

@: SORT

@: LISTA

OBJETOS DE LA CLASE : PAIS

ALEMANIA CANADA MEXICO

@:

TABLA

Renombra los objetos de acuerdo a una tabla de correspondencia especifica.

Sintaxis : TABLA archivo.ETB

Comentario : Al ejecutarse esta opción generará un archivo de salida donde indicará cual es la correspondencia que realizó y cuales fueron los errores que encontro.

Ejemplo :

@: LISTA

OBJETOS DE LA CLASE : PAIS

CANADA MEXICO ALEMANIA

@: TYPE PRUEBA1.ETB

CANADA

001

MEXICOA

002

@: TABLA PRUEBA1.ETB

@: LISTA

OBJETOS DE LA CLASE : PAIS

001
MEXICO
ALEMANIA

@: TYPE PRUEBA.STB
RENUMERACION DE OBJETOS DE :
CLASE : PAIS
OBJETO :

CANADA --> 001

MEXICO --> No se encontro ese objeto

@:

Ver : RENUMERA.

USAR

Activa los campos a utilizar.

Sintaxis : USAR TODO

USAR NADA

USAR nom-var[,nom-var[...]]

Comentario : La utilidad principal se puede observar al dar de alta, modificar o al imprimir. Los modos que tiene son : TODO pone a disposición todas las variables, NADA sólo pone a disposición el nombre del objeto y el último modo es enunciar las variables a utilizar separadas por comas.

Ejemplo :

@: USAR nombre,edad,sexo
@: IMPRIME

OBJETOS DE LA CLASE : EMPLEADOS

OBJETO : 00137
NOMBRE : JUSTINO MORALES
EDAD : 23
SEXO : MASCULINO

@: USAR NADA

@: IMPRIME

OBJETOS DE LA CLASE : EMPLEADOS

OBJETO : 00137

@: USAR nombre

@: IMPRIME

OBJETOS DE LA CLASE : EMPLEADOS

OBJETO : 00137

NOMBRE : JUSTINO MORALES

@: USAR TODO

@: IMPRIME

OBJETOS DE LA CLASE : EMPLEADOS

OBJETO : 00137

NOMBRE : JUSTINO MORALES

EDAD : 23

SEXO : MASCULINO

DOMICILIO : RETONO # 13

@:

VALIDEZ

Permite definir el rango de selección.

Sintaxis : VALIDEZ especificación

Comentario : La especificación puede ser 'LOCAL' o 'GLOBAL'.

En el caso de elegir LOCAL la selección sólo será de los objetos que se encuentran en esa trayectoria. Si se teclear 'GLOBAL' la selección será de todos los objetos al mismo nivel pero de cualquier trayectoria que cumplan con la condición.

Este comando es de gran utilidad ya que permite que un determinado ashurado en MICRO-MAP sea sólo para una determinada trayectoria o serán tomados en cuenta todos los objetos que están a ese nivel dentro de la jerarquía.

Ejemplo :

@: VALIDEZ LOCAL

@:

@: VALIDEZ GLOBAL

@:

Ver : SELECT.

APENDICE B

COMANDOS EXTERNOS DE SUPERBASE

Los comandos externos son aquellos que pueden ser utilizados en cualquier momento con o sin base de datos en uso. Estos permiten definir el medio ambiente de trabajo, definir y abrir una base de datos, así como utilizar comandos de MS-DOS sin tener que abandonar el sistema.

En las siguientes páginas se explican detalladamente los comandos externos de que dispone Super-Base.

ABRIR

Sirve para abrir o poner a disposición una base de datos.

Sintaxis : ABRIR [Nombre-base]

Comentario : En caso de tener ya abierta una base y querer abrir otra, la base automáticamente cerrará los archivos correspondientes de la base anterior y abrirá los de la nueva.

Las secciones de info-nivel y de info-general se actualizarán inmediatamente.

Ejemplo :

@:ABRIR EJEM1
DIRECTORIO : A: BASE :

CLASE OBJETO	PAIS
@:ABRIR EJEM1	
@:	
@:	
DIRECTORIO : A: BASE : EJEM1	

Ver : CERRAR.

CAMPANA

Activa la campana y cuando ocurre un error sonará una campana.

Sintaxis : CAMPANA

Ver : NOCAMPANA, RETARDO, DEFAULT.

CAR_PREGP

Define el caracter que aparecerá cuando este solicitando la información en altas.

Sintaxis : CAR_PREG [caracter]

Ejemplo :

```
@: USAR NADA
@: CAR_PREG >
@: ALTAS 1
NOMBRE DEL OBJETO > ESTADOS UNIDOS
@: CAR_PREG ?
@: ALTAS 1
NOMBRE DEL OBJETO ? GUATEMALA
@: CAR_PREG :
@: ALTAS 1
NOMBRE DEL OBJETO : GUATEMALA
@:
```

Ver : DEFAULT.

CD

El comando cambia el directorio de trabajo al directorio especificado (path). Este tambien despliega el directorio asignado cuando no se le especifica el path.

Sintaxis : CD [path]

Comentario : Cuando se ejecuta este comando también se actualiza la sección de info-general.

Ejemplo :

```
Ø: CD \hola
Ø: CD
A:\hola
Ø:
Ø: CD \
Ø: CD
A:\
Ø:
```

CLS

Limpia la sección de comandos.

Sintaxis : CLS

Ejemplo : CLS

DATE

Despliega o actualiza la fecha.

Sintaxis : DATE {dd-mm-aa}

Comentario : Con esta instrucción se actualiza la fecha del reloj interno de la computadora y se debe de recordar que los números que pueden asignarse son:

dd - del 1 al 31

mm - del 1 al 12

aa - del 80 al 99

Si solamente se teclea date mostrará la fecha actual y pedira la nueva fecha, si se desea conservar la fecha se deberá teclear <enter> en caso de lo contrario se actualizara la fecha.

Ejemplo :

```
Ø: DATE
Ø: La fecha actual es 3-12-88
Ø: La nueva fecha es (dd-mm-aa) : 4-12-88
Ø:
```

Ø: DATE 4-12-88

Ver : TIME.

DEFAULT

Inicializa las variables de Super-Base.

Sintaxis : DEFAULT

Comentario : Las variables que se inicializan son las siguientes:

Drive asignado	A
Prompt	@:
Campana	Activada
Car_preg	:
Retardo	1500
Auto	Desactivado
Digitos	7
Ceros	Desactivado
Salida	Pantalla
Entrada	Teclado
Path	' '
Unidades	H

Ejemplo :
@: Default

DEFINE

Inicializa los archivos de la base.

Sintaxis : Define [Nombre-base]

Comentario : Esta instrucción inicializa los archivos de la base para permitir abrir posteriormente la base y poder introducir información. Y borrará la información que estaba en ellos si ya estaban presentes en el disco, por lo tanto debe ejecutarse con precaución. Mientras realiza el comando indicará en la pantalla en que parte de la ejecución se encuentra.

Ejemplo :
@: Define base1
PAIS : CLASE : E_PAIS
 CAPITAL : CADENA[10]
 POBLACION : NUMERO[9:0]
FIN :
@:
Ver : ABRIR, CERRAR.

DEL

Borra el archivo especificado de disco.

Sintaxis : DEL [unidad de disco:]trayectoria_archivo

Comentario : El archivo a borrar puede ser de cualquier tipo que se encuentre en el unidad de disco correspondiente. En este caso, para seguridad del usuario, la instrucción * no tiene validez.

PRECAUCION : Una vez borrado un archivo no podrá ser recuperado.

Ejemplo :

@: DEL def1.txt

@:

DIR

Despliega los archivos de un directorio.

Sintaxis : DIR [unidad de disco:][trayectoria_archivo][w][p]

Comentario : Si sólo se teclea dir desplegará todos los archivos que se encuentren en el directorio de trabajo. Cuando se teclea el unidad de disco y la trayectoria_archivo entonces desplegará los archivos contenidos en la trayectoria especificada.

El comando dir acepta los siguientes modos:

/p - selecciona el modo de página, causando que el directorio sea desplegado con pausa por cada página.

/w - selecciona el modo de desplegado a lo ancho, sin dar otra información que el nombre de los archivos.

El comando dir lista el tamaño tiempo y fecha en que fueron creados o de su última modificación, y al final indicará los bytes libres y el total de bytes del disco.

Note que los siguientes comandos de dir son equivalentes:

COMANDO	EQUIVALENTE
dir	dir *.*
dir archivo	dir archivo.*
dir .ext	dir *.ext

Ejemplo :

Q: DIR A:
Directorio de A:*.*

EJE1.DIF	2000	2-12-1988	17:30
EJEVIII.DEF	496	6-12-1988	12:15
2 archivos		360000 bytes libres	
		362492 bytes totales	

Q: DIR/W A:
Directorio de A:*.*

EJE1.DIF	EJE2.DEF
2 archivos	360000 bytes libres
	362492 bytes totales

Q: DIR *.def
Directorio de A:*.def

EJE2.DEF	
1 archivos	360000 bytes libres
	362492 bytes totales

Q: DIR B:\hola*.def
Directorio de B:\hola*.def

0 archivos	73728 bytes libres
	362496 bytes totales

ENTRADA

Define la entrada de instrucciones, que pueden ser: pantalla o el nombre de un archivo como una macro.

Sintaxis : ENTRADA {Nombre-entrada}

Comentario : Si se tecldea solamente entrada se le asignará el teclado. Cuando se le da un nombre de entrada Super-Base procederá a leer las instrucciones del archivo [Nombre-entrada] que corresponde a una macro. Cuando el archivo de entrada de las macroinstrucciones se termina, entonces automáticamente la

entrada será asignada a PANTALLA.

Ejemplo :

@: ENTRADA macrol

....

@: ENTRADA

@: CLS

Ver : SALIDA.

EDIT

Es un editor sencillo de línea, con aplicación principal para editar la definición de una base o una macro.

Sintaxis : EDIT [archivo]

Ejemplo :

@: EDIT MACROL

Para terminar escribir .FIN

-->SALIDA IMPRESORA

-->IMPRIME

-->LISTA

-->SALIDA

-->.FIN

@:

MEMORIA

Despliega la cantidad de memoria disponible.

Sintaxis : MEMORIA

Ejemplo :

@:MEMORIA

LA MEMORIA DISPONIBLE ES : 71440

@:

MKDIR

Crea un directorio.

Sintaxis : MKDIR [unidad de disco:]path

Comentario : Este comando permite crear una estructura de

multiniveles de directorios. Recuerde que los directorios creados con mkdir siempre serán subdirectorios del directorio actual de trabajo a menos que especifique explícitamente un path diferente.

Ejemplo :

```
@: MKDIR \TRABAJO
@: MKDIR \TRABAJO\TRABAJO1
```

Ver : RMDIR.

NOCAMPANA

Desactiva la campana, por lo tanto cuando hay un error no suena la campana.

Sintaxis : NOCAMPANA

Ejemplo :

```
@: NOCAMPANA
```

Ver : CAMPANA, DEFAULT, RETARDO.

PROMPT

Cambia el prompt de Super-Base por una cadena de un máximo de 10 caracteres.

Sintaxis : PROMPT [cadena]

Ejemplo :

```
@: PROMPT A>
A> PROMPT @:
```

Ver : DEFAULT.

RETARDO

Es el tiempo que suena la campana cuando hay un error.

Sintaxis : RETARDO valor-numérico

Comentario : El valor numérico tiene que ser un valor entero positivo.

Ejemplo :

⓪: RETARDO 10
⓪: RETARDO 1500

Ver : CAMPANA, NOCAMPANA, DEFAULT.

RENAME

Cambia de nombre a un archivo en disco.

Sintaxis : Rename <nombre_viejo> <nombre_nuevo>

Ejemplo :

⓪: RENAME HOLA.DEF-NUEVO.TXT
⓪:

RMDIR

Borra un directorio de la estructura multinivel de directorio.

Sintaxis : RMDIR [unidad de disco:]path

Comentario : Rmdir borrará un directorio siempre y cuando no contenga archivos ni subdirectorios. Para poder remover un directorio por lo tanto deben de borrarse todos los archivos y subdirectorios.

Nota : No podrá removerse un directorio que contiene archivos ocultos.

Ejemplo :

⓪: RMDIR \hola
Ver : MKDIR.

SALIDA

Define el dispositivo de salida.

Sintaxis : SALIDA [dispositivo-salida]

Comentario : Los dispositivos de salida pueden ser los siguientes: Pantalla, Impresora o el nombre de un archivo. En caso de escribir solamente salida Super-Base asignará por default el dispositivo de pantalla.

Ejemplo :

@: SALIDA impresora

@: DIR

(A continuación el desplegado de directorio se imprimirá en la impresora)

@: SALIDA sall

@: DIR

(A continuación el desplegado de directorio se escribirá en un archivo llamado sall.)

@: SALIDA pantalla

@: DIR

(A continuación el desplegado de directorio se escribirá en pantalla)

@:

Ver : ENTRADA.

SALIR

Termina la ejecución de Super-Base.

Sintaxis : SALIR

Comentario : Automáticamente al salir de Super-Base los archivos de la base actual serán cerrados.

Ejemplo :

@: SALIR

Gracias por usar SUPER-BASE

(A continuación usted se encontrará en MICROMAP)

TIME

Despliega o actualiza la hora.

Sintaxis : TIME [horas:minutos[:segundos[.décimas]]]

Comentario : Con esta instrucción se actualiza la hora del reloj interno de la computadora y se debe de recordar que los números que pueden asignarse son:

horas - de 0 a 23

minutos - de 0 a 59

segundos - de 0 a 59

décimas - de 0 a 99

Si solamente se teclea time mostrará la hora actual y pedirá la nueva hora, si se desea conservar la hora se deberá teclear <enter> en caso de lo contrario se actualizará la hora.

Ejemplo :

@: TIME

@: La hora actual es 12:54:27.14

@: La nueva hora es (hh:mm:ss.dd) : 12:59

@:

@: TIME 12:59

@:

Ver : DATE.

TYPE

Despliega el contenido de un archivo.

Sintaxis : TYPE [unidad de disco:]archivo

Comentario : Con el comando type se puede desplegar el contenido de un archivo en disco.

Ejemplo :

```
@: TYPE base1.def
pais:clase;pais
  poblacion:numero[9:0]=sum(estados.pobla)
  estados:clase;estados
    pobla:numero[9:0]
  fin:
fin:
@:
```

UNIDADES

Permite al usuario definir en que unidades desea que se calcule y almacene el área.

Sintaxis : UNIDADES <especificación>

Comentario : La especificación puede ser 'M' o 'H' o 'K' que corresponden a Metros', Hectáreas y Kilometros' repectivamente. Esta instrucción esta directamente relacionada con el comando AREA.

Ejemplo :

```
@: UNIDADES M
@:
```

Ver : AREA.

BIBLIOGRAFIA

- 1) Título . Sistemas de administración de bancos de datos.
Autor . Alfnoso F. Cardenas
País . México
Editorial . Limusa
Copyright . 1983
Edición . Primera
Fecha . 1983
Volumen . ---
No. Pág. . 618
- 2) Título . Diseño de bases de datos.
Autor . Gio Wiederhold
País . México
Editorial . Mc Graw Hill
Copyright . 1985
Edición . Segunda
Fecha . 1986
Volumen . ---
No. Pág. . 921
- 3) Título . Organización de las bases de datos
Autor . James Martin
País . México
Editorial . Prentice Hall Hispanoamericana
Copyright . 1977
Edición . Traducción de la primera versión en Inglés
Fecha . 1987
Volumen . ---
No. Pag. . 544
- 4) Título . An introduction to Data Base Systems
Autor . C. J. Date
País . Estados Unidos de América
Editorial . Addison Wesley Publishing Company
Copyright . 1976
Edición . Segunda
Fecha . 1976
Volumen . ---
No. Pág. . 366
- 5) Título . Introduction to Modern Information Retrieval
Autor . Gerard Salton
País . Republica de Singapur
Editorial . Mc Graw Hill International Book CO.
Copyright . 1983
Edición . Segunda
Fecha . 1984
Volumen . ---
No. Pág. . 448

- 6) Titulo . Introducción a las Bases de Datos
 Autor . Mark L. Gillenson
 País . México
 Editorial . Mc Graw Hill
 Copyright . 1988
 Edición . Primera
 Fecha . 1987
 Volumen . ---
 No. Pág. . 391
- 7) Titulo . Fundamentos de Bases de Datos
 Autor . Henry F. Korth
 País . México
 Editorial . Mc Graw Hill
 Copyright . 1988
 Edición . Primera
 Fecha . 1989
 Volumen . ---
 No. Pág. . 525
- 8) Titulo . Data Base Managment Systems
 Autor . Michael J. Freiling
 País . Estado Unidos de America
 Editorial . Alfred Publishing CO., INC.
 Copyright . 1982
 Edición . Segunda
 Fecha . 1982
 Volumen . ---
 No. Pág. . 63
- 9) Titulo . Introducción a la Demografia.
 Autor . Mortimer Spiegelman
 País . México D.F.
 Editorial . Fondo de Cultura Económica
 Copyright . 1972
 Edición . Segunda Reimpresión
 Fecha . 1985
 Volumen . ---
 No. Pág. . 492
- 10) Titulo . X Censo general de población y vivienda, 1980.
 Subtitulo . Estado de Baja California Sur
 Volumen . I, Tomo 3. México, 1982.
 Autor . Secretaria de Programación y Presupuesto
 País . México, D.F.
 Editorial . Talleres Gráficos de la Nación - México
 Fecha . 1982
 No. Pág. . 92

- 11) Titulo . X Censo general de población y vivienda, 1980.
 Subtitulo . Cartografía Geoestadística del Estado de Baja California Sur
 Volumen . I, Tomo 3. México, 1981.
 Autor . Secretaria de Programación y Presupuesto
 País . México, D.F.
 Editorial . Talleres Gráficos de la Nación - México
 Fecha . 1981
 No. Pág. . 18
- 12) Titulo . ALGRES : a system for the specification and prototyping of complex databases.
 Autor . S. Ceri, S. Crespi-Reghezzi, G. Lamperti, L. Lavazza, R. Zicari.
 País . Milano, Italia
 Editorial . Departamento de matemáticas de la Universidad Moderna y Depto. de electrónica del Politécnico de Milano.
 No. Pág. . 18
- 13) Titulo . TURBO PASCAL
 Subtitulo . Reference Guide
 Autor . BORLAND
 País . Estados Unidos
 Editorial . Borland International
 Copyright . 1989
 Versión . 5.5
 Fecha . 1989
 No. Pág. . 468