

03063

15



**UNIVERSIDAD NACIONAL  
AUTONOMA DE MEXICO**

**Colegio de Ciencias y Humanidades  
Unidad Académica de los Ciclos  
Profesionales y Posgrado**

**Instituto de Investigaciones en Matemáticas  
Aplicadas y Sistemas**

**DISEÑO E IMPLEMENTACION DE UN EDITOR  
DE BASES DE DATOS**

**Que para obtener el grado de  
Maestro en Ciencias de la Computación**

**P r e s e n t a**

**S H U P I N G      R A N**

**México, D. F.**

2002



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 esta 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.

## AGRADECIMIENTO

---

Deseo expresar mi agradecimiento con toda sinceridad al Dr. Federico Martin Polo, M.C. Eduardo Aguirre Illing, M.C. Ralph Meixner Rojas por la dirección y la revisión de la tesis; al Instituto de Investigaciones Eléctricas por haberme aceptado para realizar la tesis; al Dr. Renato Barrera Rivera y M.C. Luis Hugo Peñarieta por la asesoría y la ayuda en mi formación de la Maestría en Ciencias de Computación, y finalmente, quisiera expresar mi agradecimiento a la Patria -- China y al pueblo chino por haberme apoyado en mis estudios.

# C O N T E N I D O

## CAPITULO 0 INTRODUCCION

## CAPITULO 1 OBJEIVOS Y CARACTERISTICAS GENERALES DEL SISTEMA

1.1	ANALISIS DEL PROBLEMA	1-1
1.2	OBJETIVOS DEL SISTEMA	1-4
1.3	CARACTERISTICAS GENERALES DEL SISTEMA	1-5

## CAPITULO 2 CONCEPTOS BASICOS DE BASES DE DATOS

2.1	SISTEMAS DE BASES DE DATOS	2-1
2.2	NECESIDADES DE BASES DE DATOS	2-3
2.3	ARQUITECTURA DE UN SISTEMA DE BASES DE DATOS	2-4
2.4	LOS MODELOS DE BASES DE DATOS	2-7
2.4.1	MODELO RELACIONAL	2-9
2.4.1.1	VISTA CONCEPTUAL DE LAS ESTRUCTURAS	2-10
2.4.1.2	VISTA EXTERNA DE LAS ESTRUCTURAS.	2-12
2.4.1.3	DESCRIPCION SIMPLIFICADA DE LA VISTA INTERNA DE LAS ESTRUCTURAS	2-15
2.4.1.4	COMO SE ESTABLECEN LAS RELACIONES ENTRE ENTIDADES.	2-16
2.4.1.5	INDICES.	2-20
2.4.1.6	REGLAS DE INTEGRIDAD DE LA BASE DE DATOS	2-21
2.4.1.7	COMO RECORRER LA BASE DE DATOS.	2-22
2.4.2	MODELO JERARQUICO	2-23
2.4.2.1	VISTA CONCEPTUAL	2-26
2.4.2.2	VISTA EXTERNA	2-28
2.4.2.3	VISTA INTERNA	2-28
2.4.3	MODELO DE REDES	2-31
2.4.3.1	VISTA CONCEPTUAL	2-33
2.4.3.2	VISTA EXTERNA	2-36
2.4.3.3	VISTA INTERNA	2-37
2.4.4	RESUMEN	2-37
2.5	LENGUAJES DE MANIPULACION DE BASE DE DATOS	2-38
2.5.1	ALGEBRA RELACIONAL	2-38
2.5.1.1	UNION	2-38
2.5.1.2	INTERSECCION	2-39
2.5.1.3	DIFERENCIA	2-39
2.5.1.4	PRODUCTO CARTESIANO	2-39
2.5.1.5	SELECCION	2-39
2.5.1.6	PROYECCION	2-40
2.5.1.7	JUNCION	2-40
2.5.1.8	DIVISION	2-40
2.5.2	CALCULO RELACIONAL	2-40
2.5.2.1	CALCULO RELACIONAL ORIENTADO A TUPLAS	2-41
2.5.2.2	CALCULO RELACIONAL ORIENTADO A DOMINIOS	2-42
2.5.3	SQL	2-42
2.5.4	QBE	2-43

CAPITULO 3 PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

3.1	HIPOTESIS DE LA IMPLEMENTACION	3-1
3.2	IMPLEMENTACION DE LAS ESTRUCTURAS	3-2
3.2.1	IMPLEMENTACION DE UNA ENTIDAD	3-2
3.2.2	IMPLEMENTACION DE INDICES	3-4
3.2.3	IMPLEMENTACION DE UNA BASE DE DATOS	3-6
3.3	IMPLEMENTACION DEL ACCESO A INSTANCIAS DE UNA ENTIDAD	3-7
3.4	IMPLEMENTACION DE LAS REGLAS DE INTEGRIDAD	3-7
3.5	EL MEDIO AMBIENTE DE APLICACION	3-8
3.5.1	SINCRONIZACION DE ACCESO	3-9
3.5.2	MANEJO DE LA MEMORIA COMPARTIDA	3-10
3.6	ESTRUCTURA DE ARCHIVOS	3-11
3.6.1	ARCHIVO DE ENTIDADES	3-12
3.6.1.1	ESTRUCTURA DEL ARCHIVO DE ENTIDADES	3-12
3.6.2	ARCHIVO INDICE	3-13
3.6.2.1	ESTRUCTURA DEL ARCHIVO INDICE	3-13

CAPITULO 4 DISEÑO E IMPLEMENTACION DEL EDITOR DE BASES DE DATOS

4.1	ESPECIFICACION DEL PROBLEMA	4-2
4.2	ESTRUCTURA MODULAR DEL MANEJADOR DE BASES DE DATOS	
4.2.1	DESCRIPCION GENERAL DE MODULOS	4-2
4.2.2	ESTRUCTURA JERARQUICA ENTRE MODULOS	4-3
4.2.3	CONCEPTO DE REGISTRO ASOCIADO	4-10
4.2.4	CRITERIOS DE ACTUALIZACION DE REGISTROS	4-11
4.2.4.1	INICIALIZACION	4-11
4.2.4.2	MODIFICACION	4-12
4.2.4.3	ACTUALIZACION	4-12
4.3	DECISIONES	4-15
4.4	FORTRAN 77 E INTERFACE FORTRAN 77 Y PASCAL	4-16
4.4.1	SELECCION DEL LENGUAJE PARA LA IMPLEMENTACION DEL SISTEMA	4-16
4.4.2	COMPARACION DE FORTRAN 77 Y PASCAL	4-17
4.4.3	INTERFACE ENTRE FORTRAN 77 Y PASCAL	4-19
4.5	ESTRUCTURA GENERAL DEL SISTEMA	4-20
4.5.1	LOS MODULOS DEL SISTEMA	4-20
4.5.2	NIVELES DE CONTROL DE SECUENCIA	4-22
4.5.3	NIVELES DE OPERACIONES SOBRE EL SISTEMA DE BASES DE DATOS	4-23
4.6	EL PROGRAMA PRINCIPAL	4-23
4.7	MODULO DE CONTROL Y CONSULTA DE LOS ESQUEMAS	4-25
4.7.1	SELECCION DE OPERACIONES	4-25
4.7.2	DEFINICION	4-25
4.7.2.1	DEFINICION DE UNA BASE DE DATOS NUEVA	4-26
4.7.2.2	DEFINICION DE ENTIDAD NUEVA	4-27
4.7.3	CONSULTA	4-27
4.7.3.1	CONSULTA DEL ESQUEMA DEL SISTEMA	4-28
4.7.3.2	CONSULTA DE ESQUEMAS DE BASES DE DATOS	4-28
4.7.3.3	CONSULTA DE ESQUEMAS DE ENTIDADES	4-28
4.7.3.4	CONSULTA DE ESQUEMAS DE ATRIBUTOS	4-29

4.7.4	DICCIONARIO	4-30
4.7.5	APERTURA	4-31
4.7.5.1	APERTURA DE UNA BASE DE DATOS	4-32
4.7.5.2	APERTURA DE UNA ENTIDAD	4-33
4.7.5.3	APERTURA DE UN ATRIBUTO	4-33
4.7.5.4	APERTURA DEL INDICE DE UN ATRIBUO	4-34
4.7.5.5	APERTURA DE UNA ENTIDAD ESPECIFICA CON TODOS SUS ATRIBUTOS	4-34
4.7.6	CIERRE	4-34
4.7.6.1	CIERRE DE BASE DE DATOS	4-35
4.7.6.2	CIERRE DE ENTIDAD	4-36
4.7.6.3	CIERRE DE ATRIBUTO	4-36
4.7.6.4	CIERRE DEL INDICE DE UN ATRIBUTO	4-37
4.7.7	CREACION	4-37
4.7.7.1	CREACION DE UNA ENTIDAD	4-38
4.7.7.2	CREACION DE INDICES	4-39
4.8	MODULO DE QUERY	4-39
4.8.1	STACK	4-40
4.8.2	TIPOS DE OPERACIONES	4-41
4.8.3	SELECCION DE LAS OPCIONES DEL MODULO QUERY	4-43
4.8.4	CONDICION	4-43
4.8.5	NOT	4-45
4.8.6	OR	4-46
4.8.7	AND	4-46
4.8.8	DIFERENCIA	4-47
4.8.9	ORDENA	4-47
4.8.10	POP, PUSH, RESET	4-48
4.8.11	DESPLIEGA	4-48
4.8.12	REPORTA	4-49
4.8.13	BORRA	4-50
4.8.14	MODIFICA	4-51
4.8.15	INSERTA	4-51
4.8.16	PANTALLA	4-52
4.9	MODULO DE DEFINICION DE FORMATO DE UNA PANTALLA	4-52
4.9.1	DEFINIR UNA NUEVA PANTALLA O MODIFICAR UNA PANTALLA EXISTENTES :	4-52
4.9.2	SUBROUTINAS PARA EL MANEJO DE LAS PANTALLAS	4-54
4.9.3	ESTRUCTURA DE DATOS	4-55
4.9.3.1	LA LISTA LINEAL	4-55
4.9.3.2	EL ARBOL	4-57
4.9.3.3	ESTRUCTURA TOTAL	4-59
4.9.4	PROGRAMA PARA LA DEFINICION DE FORMATO DE PANTALLAS	4-59
4.9.5	FUNCIONES DISPONIBLES PARA EL USUARIO	4-60
4.9.5.1	LOAD	4-61
4.9.5.2	UNLOAD	4-61
4.9.5.3	INICIA	4-61
4.9.5.4	ESCRIBE	4-62
4.9.5.5	LEE	4-62
4.9.5.6	REFRESCA	4-62
4.9.5.7	REPORTA	4-63
4.9.5.8	EXTRAE	4-63
4.9.5.9	DEPOSITA	4-63

4.9.5.10	OPERADOR	4-64
4.9.5.11	DEFCTL	4-64
4.10	SELECCION DE LAS OPERACIONES	4-64
4.11	DIAGNOSTICO CON EL USUARIO	4-65
4.12	RESUMEN DE LOS MODULOS DEL EDITOR	4-65

## CAPITULO 5 CONCLUSIONES

5.1	COMBINACION DE DDL Y DML	5-1
5.2	DIVERSOS TIPOS DE OPERACIONES	5-2
5.3	FACILIDAD PARA EL USUARIO	5-3
5.4	REPRESENTACION DE UNA ENTIDAD	5-3
5.5	TRABAJO FUTURO	5-3

## APENDICE A MANUAL DE USUARIOS DE EDITOR DE BASES DE DATOS

A.1	INVOCANDO EL EDITOR DE BASES DE DATOS	A-1
A.2	DESCRIPCION GENERAL DEL MANEJO DEL EDITOR	A-1
A.3	MENU PRINCIPAL	A-2
A.4	CONSULTA O CONTROL DE ESQUEMAS	A-2
A.4.1	DEFINICION	A-3
A.4.1.1	DEFINICION DE UNA NUEVA BASE DE DATOS	A-3
A.4.1.2	DEFINICION DE UNA NUEVA ENTIDAD PARA UNA BASE DE DATOS EXISTENTE	A-4
A.4.2	CONSULTA DE ESQUEMAS	A-6
A.4.2.1	CONSULTA DE ESQUEMA DE BASE DE DATOS	A-6
A.4.2.2	CONSULTA DE ESQUEMAS DE ENTIDADES	A-7
A.4.2.3	CONSULTA DE ESQUEMAS DE ATRIBUTOS	A-9
A.4.2.4	CONSULTA DE ESQUEMA DEL SISTEMA	A-10
A.4.3	APERTURA	A-11
A.4.4	CIERRE	A-12
A.4.5	CREACION	A-12
A.5	CONSULTA O MODIFICACION DE UNA BASE DE DATOS	A-13
A.5.1	CONDICION	A-13
A.5.2	NOT	A-14
A.5.3	OR	A-15
A.5.4	AND	A-15
A.5.5	DIFERENCIA	A-16
A.5.6	DESPLIEGA	A-16
A.5.7	REPORTA	A-17
A.5.8	ORDENA	A-17
A.5.9	BORRA	A-18
A.5.10	MODIFICA	A-19
A.5.11	POP	A-19
A.5.12	RESET	A-20
A.5.13	INSERTA	A-20
A.5.14	PANTALLA	A-20

APENDICE B DEFINICION DE LA PANTALLA DEL EDITOR

APENDICE C MANUAL DE USUARIOS : DEFINICION DE FORMATOS DE PANTALLA

C..1	INVOCANDO EL PAQUETE . . . . .	C-1
C..2	DESCRIPCION DE CADA OPERACION . . . . .	C-2
C..2.1	INSERTA . . . . .	C-2
C..2.2	MODIFICA . . . . .	C-4
C..2.3	CANCELA . . . . .	C-6
C..2.4	ESCRIBE . . . . .	C-6
C..2.5	TERMINA . . . . .	C-6
C..2.6	ARCHIVO . . . . .	C-7
C..3	MANEJO DE LAS RUTINAS DISPONIBLES AL USUARIO . . . . .	C-8
C..3.1	LOAD (NOMARCH, NOMDIR : ALFA) : SCREEN . . . . .	C-8
C..3.2	ESCRIBE (SCR : SCREEN) : CHAR . . . . .	C-8
C..3.3	REFRESCA (SCR : SCREEN) : CHAR . . . . .	C-9
C..3.4	LEE (SCR : SCREEN; OPERACION : OPERACIONES) : STATUS . . . . .	C-10
C..3.5	DEPOSITA (S : SCREEN; N : ALFA . . . . .	C-10
C..3.6	EXTRAE (S : SCREEN; N : ALFA . . . . .	C-11
C..3.7	INICIA (S : SCREEN) : STATUS . . . . .	C-11
C..3.8	REPORTA(S : SCREEN) . . . . .	C-12
C..3.9	UNLOAD(S : SCREEN) . . . . .	C-12
C..3.10	DEFCTL(C : CHAR; INI : BOOLEAN) . . . . .	C-13
C..3.11	OPERADOR(S : SCREEN; N : ALFA: OP : VAR UN OPERADOR) : STATUS . . . . .	C-13
C..4	INVOCANDO RUTINAS DESDE UN PROGRAMA . . . . .	C-14
C..4.1	A TRAVES DE UN PROGRAMA ESCRITO EN PASCAL: . . . . .	C-14
C..4.2	A TRAVES DE UN PROGRAMA ESCRITO EN FORTRAN: . . . . .	C-15
C..5	RESUMEN DE LOS TIPOS DE VARIABLES EQUIVALENTES . . . . .	C-15

APENDICE D REFERENCIAS Y BIBLIOGRAFIAS



## CAPITULO 0

### INTRODUCCION

Estamos en un mundo lleno de diversas informaciones. Todos los días manejamos información, nadie puede vivir sin conducir alguna información. El ser humano recuerda, piensa, y toma decisiones, éste es un proceso que involucra un complejo manejo de la información. Cualquier empresa necesita información de los recursos que utiliza para alcanzar sus objetivos. Sin la posibilidad de manejar esta información, no sería capaz de funcionar. En la misma medida en que una empresa crece en complejidad, crece también su necesidad de un control efectivo de la información.

Hay muchas formas de almacenar y manejar la información. Cuando la cantidad de información no es muy grande, podemos utilizar nuestra memoria. Cuando la cantidad es grande, es necesario utilizar otros medios para hacerlo. Uno de ellos son las tablas manuales de información. Conforme crece la cantidad de información, llegará un momento en que será imposible manejar la información usando esas tablas, cada vez se vuelve más evidente la necesidad de un manejo y almacenamiento eficiente de la información, los sistemas computacionales conocidos como bases de datos han sido desarrollados con la idea de atacar este problema.

Actualmente, existen muchos modelos de bases de datos. Entre ellos, el modelo de datos relacional que propuso por primera vez CODD [1, 2], y que ha llegado a ser aceptado ampliamente por su simplicidad conceptual, independencia de datos y semántica completa. En este modelo, la vista conceptual de los datos para el usuario es el conjunto de tablas llamado relaciones, sobre las cuales, uno puede realizar las operaciones.

## INTRODUCCION

Un problema relacionado con el manejo de las Bases de Datos es : cómo se entregan los datos a los usuarios como ellos lo especifican. Este no es un problema fácil, porque cada usuario tiene una visión conceptual diferente de la estructura de la base de datos. Esto implica : cómo acepta las preguntas que los usuarios hacen, cómo se transforman éstas al manejador de la base de datos, y finalmente en qué forma se responde al usuario. De todo este trabajo se encarga el QUERY. La relación entre el usuario, el Query, el sistema de manejador de base de datos (BDMS) y la Base de datos se muestra en la figura 0-1.

Hay muchos lenguajes propuestos basados en Algebra Relacional [3] ó en Cálculo de predicados, como propuso CODD [2]. Ambos enfoques tienen algunas desventajas : los Lenguajes basados en Algebra relacional tienden a ser de tipo de procedimientos y las expresiones de Cálculo de predicados siempre son incómodos, aún para preguntas simples [5].

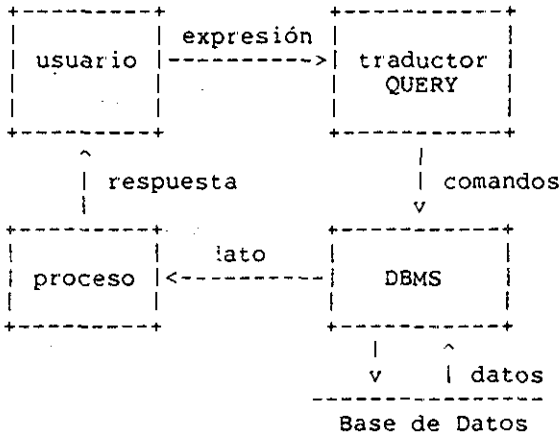


Figura 0-1. Diagrama de flujo para QUERY de base de datos para Usuarios de terminal.

## INTRODUCCION

El presente trabajo consiste en un Editor interactivo para Bases de Datos. Su propósito es, como en todos los sistemas de Query, poder aceptar las preguntas de los usuarios sobre la información de la base de datos y poder entregar los datos como ellos lo especifican. Pero a diferencia de la mayoría de los sistemas de Query, en este sistema, los usuarios no tienen que formar sus preguntas en ningún lenguaje intermedio que esté basado en Algebra relacional, en Cálculo de predicados o en algún otro, el sistema acepta las preguntas en forma de despligue de tablas como nos acostubramos ver una base de datos y el sistema puede reportar los resultados en el formato definido por el usuario, tanto por la terminal de video como por la impresora. Además, el sistema permite a los usuarios llevar un control sobre las bases de datos, como: consultar o modificar las estructuras de las bases de datos así como consultar o modificar las estructuras de las relaciones. El sistema permite a los usuarios consultar, modificar, borrar, insertar una instancia o un conjunto de instancias de las relaciones, y permite a los usuarios ordenar los conjuntos resultantes de las preguntas por el valor de algún atributo.

El sistema ha sido desarrollado en una máquina GOULD S. E. L. en el departamento de simulación del Instituto de Investigaciones Eléctricas (IIE) y el sistema está escrito en Fortran 77.

El objetivo del presente trabajo es contar con un método eficiente de consulta y actualización de las bases de datos para el desarrollo y la operación de los proyectos de :

1. simulador de la planta Núcleo Eléctrica de Laguna Verde;
2. sistema de adquisición de datos de la termoeléctrica de Manzanillo;
3. simulador del Puesto Central de Control del Metro.

## CAPÍTULO 1

### OBJETIVOS Y CARACTERISTICAS GENERALES DEL SISTEMA

#### 1.1 ANALISIS DEL PROBLEMA

Por qué es difícil para los usuarios inexpertos hacer uso de los sistemas de cómputo ? En esta sección, se analizará este problema y se tratará de descomponerlo en subproblemas. Una vez entendido el problema, ya se puede formular una solución para él.

Vamos a discutir el diagrama de flujo de datos que se muestra en la figura 0-1, el cual define las funciones individuales que deben ser realizadas en el contexto de Query de las bases de datos para los usuarios de terminal. El diagrama empieza con la formulación de una pregunta por el usuario. Cómo se formula las preguntas, depende mucho de los siguientes importantes factores :

1. La conceptualización de los datos que tiene el usuario;
2. El lenguaje en el cual el usuario debe expresar su Query.

Una vez que el usuario haya formado la pregunta, ésta tiene que ser trasladada a una secuencia de comandos del Sistema de Manejador de Bases de Datos ( Data Base Management System -- DBMS ). Y después, los comandos de DBMS deben ser eficientemente

## OBJETIVOS Y CARACTERISIIICAS GENERALES DEL SISTEMA

ejecutados para recuperar los datos apropiados de la Base de Datos.

Entonces, dónde está el problema ? Actualmente, cada paso de este procedimiento es un problema difícil y complicado. De hecho, el paso más conocido en este proceso completo es la función de DBMS, y todos nosotros estamos enterados de la complejidad de este problema. Las otras funciones que no han sido formalizadas tan bien como la función de DBMS, no podrán ser efectivamente resueltas hasta que sean muy bien conocidas.

Vamos a concentrarnos en la parte izquierda del diagrama de flujo de datos; desde la conceptualización de los datos que tienen los usuarios hasta la función de DBMS. Tenemos en la parte izquierda un proceso incompleto e informal : el usuario y su conceptualización de datos. En la parte derecha, tenemos una secuencia de comandos de DBMS muy bien formada, completa y formal. Cómo se lleva a cabo esta transformación crítica ? Cuál función es la responsable para este mejoramiento en precisión y exactitud ?

Si nuestro diagrama es completo, esta función tiene que ser el usuario mismo o el traductor Query, o ambos. Pocos pueden creer que el usuario mismo podría generar la secuencia actual de comandos de DBMS, porque éstos son de bajo nivel y siempre se muy ocultos. Vamos a considerar las otras dos posibilidades :

- a) el usuario y el traductor Query comparten la responsabilidad;
- b) el traductor Query mismo lo hace.

El primer caso es típico de muchos sistemas de hoy. Un lenguaje intermedio orientado a usuarios es definido, el traductor Query puede rutinariamente mapear a los comandos de DBMS. El usuario comparte en resolver el problema formulando sus peticiones en este lenguaje intermedio. Esto suena como una solución excelente al problema, el usuario y el DBMS se encuentran a medio camino de entenderse uno al otro. Por qué ésto no trabaja muy bien ?

Esto no trabaja muy bien, porque el lenguaje intermedio está demasiado aproximado a los comandos de DBMS y a la estructura de las bases de datos. En otras palabras, es que la realidad física actual de la estructura de las bases de datos y la arquitectura de DBMS son todavía sentidas por los usuarios, cuando los usuarios tratan de expresar su pregunta en este lenguaje intermedio. El lenguaje intermedio simplemente permite al usuario no seguir una sintaxis rígida y permite el uso de palabras orientadas al usuario para referir a la base de datos. Estas sutilezas son cierta y absolutamente críticas si el usuario debe ser capaz de usar el sistema, pero ésto no ataca el corazón

## OBJETIVOS Y CARACTERISTICAS GENERALES DEL SISTEMA

del problema.

El corazón del problema es que la conceptualización de datos que tienen los usuarios y la estructura actual de datos son diferentes. Hay muchas razones para esto: la base de datos fué diseñada para resolver las necesidades de todos los usuarios, mientras que al usuario individual sólo le preocupa su propio problema. Por el otro lado, las personas organizan sus conocimientos de una manera muy distinta a la que esos conocimientos están organizados en la base de datos. Los usuarios rara vez científicamente formalizan las cosas. Un buen ejemplo de la manera en que la gente realmente estructuran sus ideas es la estructura de un diccionario ordinario. Cada palabra que es una parte de una definición en el diccionario, está definida en algún lugar del mismo diccionario. Todas las definiciones son circulares, pero no se parecen entre sí, sin embargo ésto no limita la utilidad del diccionario. El punto es: este tipo de estructuras es significativo y útil para los usuarios, pero no lo es del tipo de estructuras que un usuario se espera a encontrar en una base de datos.

Dado que el usuario sólo tiene este entendimiento vago de la estructura actual de la base de datos, cómo podemos esperar que el usuario exprese sus preguntas en un lenguaje que requiere mucho conocimiento de cómo están estructurados los datos actualmente? Claro que no podemos. No podemos esperar que alguien trate algo que no entiende completamente. Entrenando a los usuarios para que entiendan estas estructuras es una tarea desesperada, porque aún los expertos tienen problemas, y estas estructuras están sujetas a cambios todavía.

Por lo tanto, el problema que parece estar resuelto por tener el encuentro en el medio camino entre el usuario y los comandos de DBMS, realmente es mucha carga para el usuario. Pero, cómo será nuestra otra alternativa: que el traductor realice todo el trabajo?

Al contrario de nuestra previa discusión, la cual pareció fácil al principio, tener el traductor Query haciendo todo el trabajo, parece extremadamente difícil al principio. Decimos que es difícil, porque estamos permitiendo al usuario expresarse en su manera usual e informal basada en su imprecisa conceptualización de datos. Estamos todavía obligados a generar una secuencia precisa de comandos de DBMS.

Esto es lo distinto de este intento comparando con las otras alternativas, este complejo proceso de engranar la conceptualización del usuario con los datos actuales es digno de ser un módulo en el análisis del problema, es decir, la traducción se hace sólo una vez, y todo por el traductor Query, ninguna inexactitud generada por el usuario será seguida a través del traductor. Similarmente, el usuario no sentirá las

## OBJETIVOS Y CARACTERISTICAS GENERALES DEL SISTEMA

estructuras físicas de la base de datos. El traductor Query realiza un enlace para ambos: el DBMS y el usuario, siendo seguro que los detalles de cada uno no son sentidos por el otro.

Este aislamiento funcional del proceso de traducción parece muy deseable, pero es técnicamente factible ?

La respuesta es Si. Actualmente, existen muchos sistemas que tratan de resolver este problema, pero en diferentes metodologías. El sistema QMF ( Query Management Facility ) de IBM es un sistema Query para bases de datos relacionales, que puede consultar, modificar y reportar informaciones de la base de datos usando este sistema. El sistema ROBOT del departamento de matemáticas de Dartmouth College es un sistema que trata de aceptar las preguntas de los usuarios en enguaje inglés natural.

### 1.2 OBJETIVOS DEL SISTEMA

Como se vio en la sección anterior, el usuario tiene una conceptualización de la estructura de datos diferente de la estructura de la base de datos, por lo tanto, es muy difícil para el usuario formar las preguntas con mucha formalidad, o sea, que el usuario se encargue de formar las preguntas con tanta formalidad para que las pueda aceptar el DBMS, o que se establezca un lenguaje intermedio, para que el usuario y el traductor Query se encuentren en el medio camino, el usuario se encarga de la mitad de la responsabilidad de la formalidad, y el traductor de Query se encarga de la otra mitad.

En el presente sistema, se trata de encargar toda la formalidad de los comandos para DBMS, para que los usuarios con diferentes conceptualizaciones de la base de datos puedan acceder la información con mayor facilidad.

El sistema también trata de facilitar su uso a los usuarios no familiarizados con las bases de datos, proporcionándoles una interface lo más cercana posible a la conceptualización de la estructura de datos que tiene el usuario. Muchas veces, al principio de que un usuario trata de utilizar un sistema, trata de relacionar el sistema con alguna idea que ya tiene en la mente. Por ejemplo, muchos de nosotros, nos acostumbramos a pensar en la base de datos como un conjunto de tablas donde se puede localizar la información deseada buscando en cierto orden.

## OBJETIVOS Y CARACTERISTICAS GENERALES DEL SISTEMA

Este sistema trata de facilitar al usuario utilizar el sistema, y que no tenga la dificultad de adaptar la interface al sistema, teniendo la idea común de la base de datos en la mente. En otras palabras, el sistema trata de ser amigable con el usuario para que no sienta que el sistema es algo misterioso, difícil de utilizar para lograr sus objetivos.

### 1.3 CARACTERISTICAS GENERALES DEL SISTEMA

El sistema tiene tres componentes principales :

#### 1. El subsistema de definición de los formatos de los reportes

El subsistema permite al usuario definir el formato de reporte tanto para las terminales de video como para la impresora. Así como definir la forma de representar las relaciones en base a sus atributos.

Con este subsistema, el usuario puede definir cinco tipos de datos : entero, real, lógico, cuerda de caracteres y fechas. Para formar una pantalla ( definir el formato de un reporte ), el usuario puede posicionar el texto de cada campo en lugares deseados de la pantalla y en intensidad deseada ( hay dos niveles de intensidades : nivel alto y nivel bajo ); así como para la definición del valor del campo. Se puede definir un texto sin definir su valor. Para los textos que tienen valor, se define el valor que se toma por definición, el rango del valor, o sea el valor mínimo y el máximo para valores de tipo entero o real; se define el tamaño mínimo para una cuerda de caracteres, etc., también se define la longitud del campo del valor para cada uno de los tipos de datos. Se pueden definir hasta cuatro tipos de operaciones diferentes para una pantalla combinándolas con entrada, salida o ambos. Esto es para el control de las operaciones de la definición de la pantalla, captura de datos o el despliegue de la pantalla ( para mayor detalle, referirse al apéndice C).

#### 2. El control de las bases de datos, relaciones y atributos, y la consulta de los esquemas.



## OBJETIVOS Y CARACTERISTIICAS GENERALES DEL SISTEMA

Este subsistema permite al usuario tener un control sobre las bases de datos, como abrir o cerrar una base de datos, una entidad, un atributo o un índice de un atributo; también permite al usuario definir un esquema de una base de datos nueva, o de una nueva relación en una base de datos existente. Se permite al usuario crear una nueva relación, o sea formar el archivo físico de la relación ( para mayor detalles, referirse al capítulo 3 ). También se puede crear un índice nuevo de un atributo.

Este subsistema también permite al usuario consultar los esquemas de las bases de datos existentes, o sea la definición de las estructuras de las bases de datos. Se pueden consultar los esquemas de las relaciones y los atributos, o el sistema : consultar las características de las bases de datos, relaciones o atributos.

## OBJETIVOS Y CARACTERISTICAS GENERALES DEL SISTEMA

### 3. El subsistema Query.

Este subsistema permite a los usuarios hacer preguntas sobre la base de datos en una forma interactiva, o sea, permite al usuario visualizar la base de datos o mejor dicho las relaciones en la forma en que acostumbra pensar -- la forma de tabla. El usuario no tiene que formar sus preguntas en ningún lenguaje, el sistema acepta las preguntas de los usuarios desplegando las pantallas que representan las relaciones en base de sus atributos en forma tabular, capturando los valores de los atributos para formar las condiciones.

Se permite al usuario realizar las operaciones sobre conjuntos de instancias que fueron obtenidos a través de las preguntas, es decir, sobre conjuntos de instancias que cumplieron con ciertas condiciones especificadas por el usuario.

Se permite al usuario desplegar los valores de los atributos de las instancias de las relaciones, o sacar un reporte del conjunto de instancias en el formato que fué definido por el usuario a través del subsistema de definición de formatos.

Todo el sistema está basado en menú, todas las operaciones se realizan en una forma muy interactiva. El usuario no necesita tener conocimientos de la estructura interna de las bases de datos, puede acceder la información deseada sin ningún problema.

## CAPITULO 2

### CONCEPTOS BASICOS DE BASES DE DATOS

#### 2.1 SISTEMAS DE BASES DE DATOS

Un sistema de bases de datos es un sistema cuyo objetivo es almacenar y mantener información. La información puede ser cualquier cosa que sirva para el propósito de construir el sistema. Un sistema de bases de datos puede ser dividido en varias bases de datos. Se puede ver un sistema de bases de datos como se muestra en la figura 2-1 sistemáticamente. En donde :

Una base de datos está compuesta en dos partes principales : **Hardware** -- el dispositivo físico donde se almacena la información; el **software** -- el sistema de manejo de la base de datos (Data Base Management System -- DBMS) que se comporta como la interface entre el usuario y la base de datos física. Todas las peticiones de usuarios son manejadas por DBMS.

Los programas de aplicación, son los programas escritos en algún lenguaje para acceder la base de datos.

CONCEPTOS BASICOS DE BASES DE DATOS

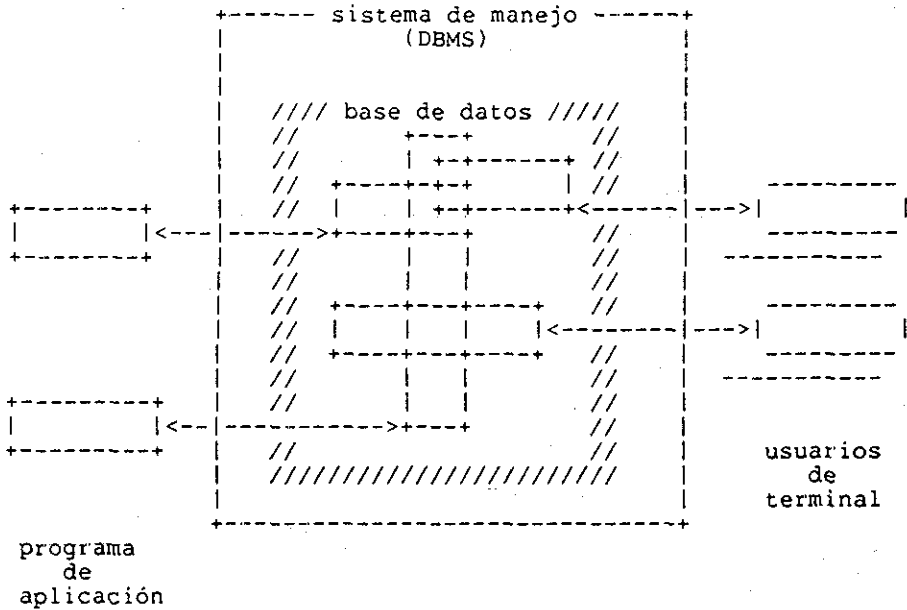


Figura 2-1. Un sistema de base de datos simplificado.

Los usuarios de terminales son aquellos que accesan la base de datos desde alguna terminal interactiva, usando un lenguaje de preguntas ( Query Language ) que es una parte del DBMS.

## CONCEPTOS BASICOS DE BASES DE DATOS

### 2.2 NECESIDADES DE BASES DE DATOS

Cualquier empresa necesita información acerca de los recursos que utiliza para alcanzar sus objetivos. Sin la posibilidad de manejar esta información, no sería capaz de funcionar. En la misma medida en la que una empresa crece en complejidad, crece también la necesidad de un control efectivo de la información.

En la actualidad, la mayoría de las grandes empresas tienen varios sistemas computacionales para manejar su información. En forma típica, la información en la computadora sirve como entrada a los diversos programas de aplicación. Cada aplicación maneja sus propios archivos, provocando que la información se encuentre muy dispersa, no estructurada y por lo tanto difícil de controlar. La estructura de la información es determinada por el programa de aplicación.

Cada vez se vuelve más evidente la necesidad de un manejo y almacenamiento eficiente de la información. Los sistemas computacionales conocidos como Bases de Datos han sido desarrollados con la idea de atacar este problema.

Se puede pensar en una base de datos como una agrupación homogénea de la información que típicamente se localizaría en diferentes archivos de datos. Los usuarios de la Base de Datos comparten la información en el sentido de que todos ellos pueden acceder los mismos datos. Una Base de Datos se estructura de una manera específica, sin embargo los usuarios pueden tener diferentes vistas de la misma información.

Algunos de los beneficios derivados del control centralizado de la información son:

1. La redundancia puede ser evitada. En un sistema normal, la misma información es almacenada en diferentes archivos privados, provocando una gran redundancia en la información. En una Base de Datos puede existir redundancia pero controlada.
2. Se pueden evitar inconsistencias, y mantener la integridad de la información.
3. La información puede ser compartida por diferentes aplicaciones.

## CONCEPTOS BASICOS DE BASES DE DATOS

4. Se pueden enforzar estandares.
5. Se pueden aplicar restricciones de seguridad.

Una de las mayores ventajas de una Base de Datos es la independencia de la información con respecto a las aplicaciones. De manera inversa las aplicaciones son independientes de la forma en la que la información está estructurada y/o almacenada.

### 2.3 ARQUITECTURA DE UN SISTEMA DE BASES DE DATOS

La arquitectura de un sistema de bases de datos puede ser dividido en tres niveles principales: externo, conceptual e interno, como se muestra en la figura 2-2.

El nivel externo es el nivel más cercano a los usuarios, en el cual, se concierne con la manera en la cual el contenido de la base de datos es visto por los usuarios individuales. También se le llama vista externa. Una vista externa consiste en múltiples tipos de registros externos. Un registro externo no necesariamente es el mismo como el registro de almacenamiento. La vista externa está definida por el esquema externo, el cual básicamente consiste en definiciones de cada uno de los tipos de registros externos. El esquema externo normalmente está escrito usando una porción del lenguaje de definición de datos (DDL -- Data Definition Language), el cual proporciona una definición de los objetos de una base de datos, a veces se le conoce como DDL externo.

CONCEPTOS BASICOS DE BASES DE DATOS

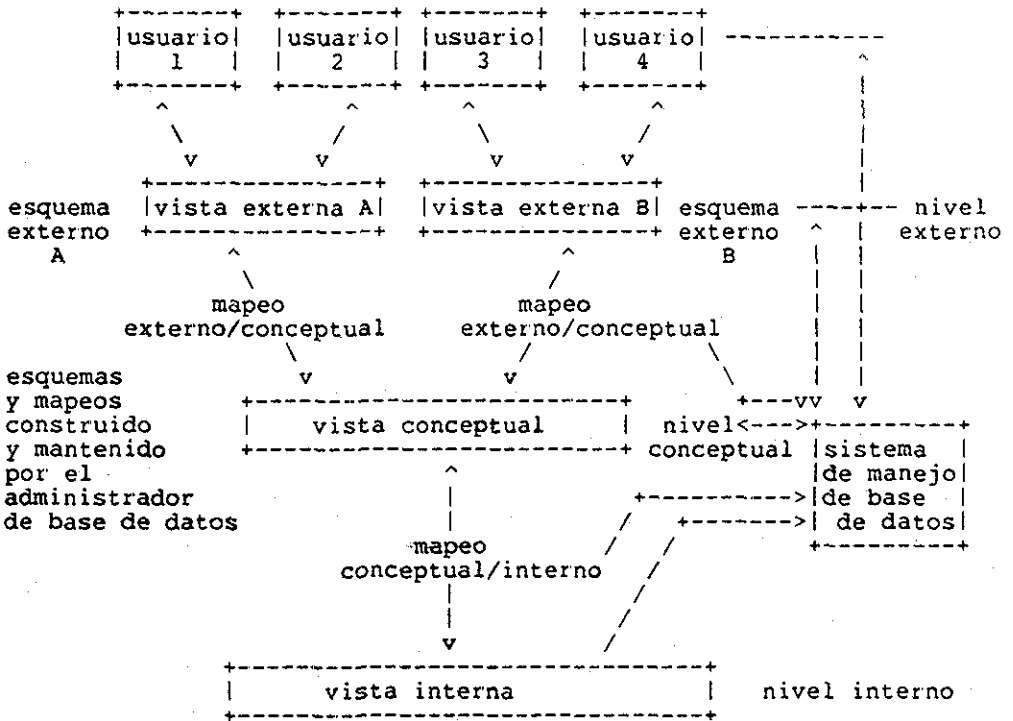


Figura 2-2. Arquitectura de la base de datos.

El nivel conceptual es un nivel de indirección del nivel externo al interno. Si el nivel externo incluye las vistas individuales de los usuarios, el nivel conceptual incluye la vista común de todos los usuarios. Se dice también que es una

## CONCEPIOS BASICOS DE BASES DE DATOS

**vista conceptual.** La vista conceptual es una representación del contenido de la información completa de la base de datos, consiste de múltiples ocurrencias de múltiples tipos de registros conceptuales. No es necesario que un registro conceptual que sea el mismo que el registro externo ni que el interno. La vista conceptual está definida por el esquema conceptual, el cual incluye las definiciones de cada uno de los tipos de los registros conceptuales. El esquema conceptual está escrito normalmente usando otro lenguaje de definición de datos -- DDL conceptual, en éste no hay que considerar estructuras de almacenamiento o estrategias de acceso solamente. Debe tener definiciones del contenido de la información. Además, se puede incluir las definiciones de control de autoridad del uso de la base de datos, cómo deben ser usados los datos, y cómo fluyen los datos de un punto a otro, etc.

El nivel interno es el nivel que se aproxima más al del almacenamiento físico de los datos, está relacionado con la manera en la cual los datos están almacenados actualmente. Es una vista interna de la base de datos, que representa el nivel más bajo de la base de datos, la cual consiste de múltiples ocurrencias de múltiples tipos de registros internos, o sea, los registros de almacenamiento. Se describe la vista interna por el **esquema interno**, el cual no sólo define los tipos de registros de almacenamiento, sino también especifica qué índices existen, cómo están representados los campos de almacenamiento, en qué secuencia física están los registros, etc. El esquema interno está escrito en DDL interno.

Como se ve en la figura 2-2, existen un mapeo entre el nivel externo y el nivel conceptual, y otro mapeo entre el nivel conceptual y el nivel interno.

El mapeo que existe entre los niveles externo y conceptual define las correspondencias entre una vista externa particular y la vista conceptual común.

El mapeo que existe entre los niveles conceptual e interno define la vista conceptual y la base de datos almacenada. Especifica cómo se mapean los registros y los campos conceptuales a su contraparte que son los registros y campos internos. Si la estructura de la base de datos sufre un cambio, se tiene que cambiar este mapeo. Sin embargo no es necesario cambiar el esquema conceptual.

En la figura 2-2, hay otro bloque importante que es el sistema de manejo de la base de datos (DBMS), el cual es el software que maneja todo el acceso a la base de datos.

El procedimiento de acceso a una base de datos es como sigue:



## CONCEPTOS BASICOS DE BASES DE DATOS

Un usuario hace su petición al DBMS en algún lenguaje de manipulación de datos particular; el DBMS interpreta la petición e inspecciona el mapeo entre los primeros dos niveles mapeando el esquema externo al esquema conceptual, y el segundo nivel al tercer nivel (esquema conceptual al esquema interno). El DBMS realiza las operaciones necesarias sobre la base de datos física, recupera las ocurrencias de los registros de almacenamiento, construye todas las ocurrencias de los registros conceptuales requeridas, y construye todas las ocurrencias de registros externos requeridas. Claro que en cada paso se necesita una conversión.

Otra parte importante de un sistema de bases de datos es el administrador de bases de datos (DBA -- data Base Administor), el cual es una persona o un grupo de personas responsables del control completo del sistema de las bases de datos. El DBA básicamente realiza las siguientes actividades :

1. decide el contenido de la información de la base de datos;
2. decide la estructura de almacenamiento y estrategia de acceso;
3. comunica con los usuarios, para ver qué requerimientos nuevos hay y trata de cumplirlos;
4. define la estrategia de respaldo de la base de datos y la estrategia de recuperación de la base de datos en caso de falla;
5. dependiendo del funcionamiento de la base de datos y los requerimientos nuevos, realiza los ajustes nuevos a la base de datos.

### 2.4 LOS MODELOS DE BASES DE DATOS

Para describir una base de datos o sea la estructura de la base de datos y los requerimientos de procesamientos sin usar ningún término físico, se necesita un vocabulario. Este vocabulario incluye las palabras para la composición de los registros de la base de datos, la relación entre ellos y la descripción de cómo se puede procesar los registros. Ninguna de

## CONCEPTOS BASICOS DE BASES DE DATOS

las palabras se refieren a términos físicos. Tal vocabulario es conocido como modelo de base de datos.

Los modelos de las bases de datos están divididos en dos partes. Una parte, a veces se la refiere como lenguaje de definición de datos ( Data Definition language -- DDL ), describe la estructura de la base de datos. La otra parte, a veces se la refiere como Lenguaje de manipulación de Datos ( Data Manipulation Language -- DML ), describe la manera en la que se procesa la información en la base de datos.

Las funciones de DDL básicamente son las siguientes :

1. Describe el nombre y tipo de cada campo;
2. Describe nombre y composición de cada registro;
3. Indica las llaves primarias y secundarias;
4. Representan las relaciones entre los registros ( para tipos de modelos jerárquico y redes );
5. Conserva la independencia lógica y física;
6. Describe subesquemas y esquemas;
7. Especifica restricciones de seguridad.

donde esquema es la representación de la estructura de la base de datos total (vista conceptual) y subesquema es la porción del esquema visto por un usuario particular (vista externa )..

Las funciones básicas de DML son las siguientes :

Describe la técnica de recuperación, reemplazamiento, inserción y borrado de los registros, incluye procesamiento directo usando llave o indirecto via relaciones entre los registros de la base de datos, permite al usuario manejar la base de datos por nombre lógico o símbolo, ésto implica una independencia de los programas desde la representación lógica de la base de datos. Para permitir un rango amplio de programas para usar la base de datos, DML soporta tantos lenguajes como posible, ésto significa que DML no debe estar estructurado como ningún lenguaje en particular.

Normalmente, las estructuras de datos soportados a nivel de usuario, Debido a su similitud, muchos sistemas no distinguen en el nivel externo y el conceptual) afectan muchos componentes del sistema. La pregunta central es : qué estructuras de datos y qué operadores asociados podría soportar el sistema ? Para responder la pregunta, veremos tres tipos de estructuras de datos.

## CONCEPTOS BASICOS DE BASES DE DATOS

muy conocidas y una comparación breve entre ellas..

- 1.. modelo relacional;
- 2.. modelo jerárquico;
- 3.. modelo de redes..

Como la base de datos implementada es de tipo relacional, pondremos más énfasis en el modelo relacional..

### 2.4.1 MODELO RELACIONAL

El modelo relacional fué desarrollado inicialmente por CODD [1], ha sido objeto de numerosos estudios y desarrollos posteriores, y sin duda alguna juega un papel importante en la tecnología de las bases de datos, aunque los sistemas de bases de datos relacionales están muy pocos difundidos todavía. Algunos sistemas de bases de datos relacionales son los siguientes ( lo cual no significa que sean totalmente fieles a dicho modelo ) :

1. **MAGNUM** de Tymshare, es un producto disponible comercialmente, este sistema proporciona un lenguaje (record-at-a-time) incluyendo las facilidades computacionales y generación de reportes. Este sistema puede ser usado en modo de comandos desde una terminal, o en modo de programación en batch tradicional;
2. **Query by example**, también es un producto disponible comercialmente de IBM;
3. El sistema **NOMAD** de NCSS, este sistema soporta tanto modelo relacional, como el modelo jerárquico, e incluye varios operadores de algebra relacional;
4. **PRTV** (Peterlee Relational Test Vehicle), es un sistema experimental del centro científico UK de IBM en Peterlee, England;
5. **INGRES** de la Universidad de California en Berkeley, este sistema proporciona un lenguaje orientado a conjuntos;

## CONCEPTOS BASICOS DE BASES DE DATOS

6. Sistema R de IBM, este sistema proporciona un lenguaje orientado a conjuntos.

En esta sección daremos una especificación informal pero rigurosa del modelo relacional de la Base de Datos. Rigurosa en el sentido en que trataremos de definir en forma no ambigua cada uno de los componentes del modelo, con el objetivo de sentar un marco común de referencia.

Al especificar el modelo relacional de la Base de Datos, se definirán en forma abstracta (es decir sin tomar en cuenta como se implementarán) las estructuras que conforman la Base de Datos, las operaciones permitidas sobre ellas, y las reglas de integridad que gobernarán la Base de Datos a lo largo de su existencia.

En la sección 2.4.1.2 se introduce un ejemplo de una Base de Datos con el propósito de clarificar los conceptos analizados. Todos los ejemplos subsecuentes se refieren a dicha Base de Datos, la cual se va complementando conforme se avanza en la sección.

Como el EDITOR de bases de datos está implementado para las bases de datos instaladas en el departamento de simulación del Instituto de Investigaciones Eléctricas, y éstas son de tipo relacional, en esta sección se explicará el modelo relacional basando en el modelo de estas bases de datos.

### 2.4.1.1 VISTA CONCEPTUAL DE LAS ESTRUCTURAS -

Conceptualmente se puede ver a la Base de Datos como un conjunto de Entidades. Cada tipo de objetos en el mundo real, del cual deseamos tener conocimiento en la Base de Datos, se asocia con una Entidad. El nombre de la entidad nos permite diferenciarla de las otras entidades definidas en la Base de Datos, y por su carácter mnemotécnico podemos identificar fácilmente al tipo de objetos al cual está asociada.

Ahora bien, para cada tipo de objetos, nos gustaría tener registrada información acerca de ciertas características de los objetos que lo componen. Cada una de dichas características se refleja en un atributo de la entidad. Así, al describir una entidad, se establecen los atributos que tendrá asociados,

## CONCEPTOS BASICOS DE BASES DE DATOS

especificando para cada uno de ellos, el nombre con el cual se le identificará, y el conjunto de valores (tipo) que pueden tomar.

A la descripción de una entidad así dada, se le denomina esquema de la entidad.

El esquema general de la Base de Datos, es el conjunto de los descriptores de las entidades que la componen.

Podemos pensar en el esquema general de la Base de Datos, como en un cascarón que nos describe la manera en la cual está organizada la información.

En un momento dado de la vida de una Base de Datos, cada uno de los objetos de la vida real, se encuentra asociado con una instancia de la entidad que le corresponde según su tipo de objetos. Una instancia queda determinada en forma única por los valores asociados a cada uno de los atributos de la entidad.

Generalmente, para diferenciar entre sí las instancias de una entidad, no es necesario considerar los valores de todos los atributos. Comúnmente existe un atributo (o conjunto de atributos), cuyo valor es suficiente para identificar una instancia dada. Dicho atributo (o conjunto de atributos) al que llamaremos llave, se encuentra íntimamente ligado con la instancia a la cual identifica. Cuando es necesario referirnos a un objeto específico, nos referimos a la llave que identifica a la instancia.

Resumiendo, cada entidad de la Base de Datos tiene asociados dos componentes. El primero de ellos, el esquema, es de naturaleza estática, y define los atributos asociados a la entidad. El segundo de ellos, el depósito de instancias, es de naturaleza dinámica (es decir varía a lo largo de la vida de la Base de Datos) y está compuesto por el conjunto de instancias de la entidad.

De manera similar, una Base de Datos tiene asociados: el esquema general de la Base de Datos, que define las entidades asociadas a la Base de Datos y un conjunto de entidades.

## CONCEPTOS BASICOS DE BASES DE DATOS

### 2.4.1.2 VISTA EXTERNA DE LAS ESTRUCTURAS. -

Externamente el usuario puede ver una Base de Datos como un conjunto de tablas, cada una de ellas asociada a una entidad.

Cada uno de los atributos de la entidad, corresponde a una columna de la tabla, con el nombre y tipo del atributo asociada a ella. Nótese, que el orden de las columnas en la tabla es irrelevante, puesto que siempre podemos identificar una columna por su nombre.

Cada una de las instancias de una entidad, corresponde a un renglón en la tabla. Cada renglón tiene un valor (del tipo adecuado) asignado a cada una de las columnas, y queda así determinado en forma única. Es decir que no existen dos renglones idénticos en una tabla. El orden de los renglones en la tabla, al igual que en las columnas, es irrelevante.

No existen dos renglones que contengan el mismo valor (valores) en la columna (columnas) asociadas con el atributo (conjunto de atributos) llave de la entidad. A ésta columna (columnas) se le denomina identificador.

La figura 2-3 muestra el esquema, y la figura 2-4 muestra la vista externa de una instancia de la Base de Datos DEPTO EMP PROJ, la cual consta de tres entidades: DEPARTAMENTO, EMPLEADO, PROYECTO, cada una de ellas asociada a los departamentos, empleados y proyectos de una empresa ficticia.

CONCEPTOS BASICOS DE BASES DE DATOS

ENTERO : Número entero  
CAD-CARACTER : Cadena de caracteres de  
longitud variable

Definición del Esquema:

DEPARTAMENTO: ENTIDAD

No-DEPTO : ENTERO;  
NOMBRE-DEPTO : CAD-CARACTER;  
JEFE-DEPTO : ENTERO  
LLAVE ES No-DEPTO

END

EMPLEADO: ENTIDAD

No-EMP : ENTERO;  
NOMBRE-EMP : CAD-CARACTER;  
DEPTO : ENTERO  
LLAVE ES No-EMP

END

PROYECTO: ENTIDAD

No-PROYECTO : ENTERO;  
DESCRIPCION : CAD-CARACTER;  
LLAVE ES No-PROYECTO

END

Figura 2-3. Definición de tipos para la base  
de datos DEPTO\_EMP\_PROY.

CONCEPTOS BASICOS DE BASES DE DATOS

DEPARTAMENTO

No-DEPTO	NOMBRE-DEPTO	JEFE-DEPTO
ENTERO	CAD-CARACTER	ENTERO
1	Contabilidad	3087
3	Compras	1576
4	Ventas	1187

EMPLEADO

No-EMP	NOMBRE-EMP	DEPTO
ENTERO	CAD-CARACTER	ENTERO
1187	Juan Pereira M.	4
1245	Pedro Cisneros A.	3
1576	Adriana Fernandez	3
3087	Mariana Castro	1
4032	Jose Martinez	4

PROYECTO

No-PROYECTO	DESCRIPción
ENTERO	CAD-CARACTER
1111	Manejo de Inventario
1010	Desarrollo

Figura 2-4. Vista externa de la base de datos

DEPTO\_EMP\_PROY.



## CONCEPTOS BASICOS DE BASES DE DATOS

### 2.4.1.3 DESCRIPCION SIMPLIFICADA DE LA VISTA INTERNA DE LAS ESTRUCTURAS -

Internamente una Base de Datos está compuesta por un conjunto de archivos, cada uno de ellos correspondiendo a un depósito de intancias de una entidad. La descripción de los archivos, queda determinada por el esquema de la Base de Datos. Cada uno de los archivos que componen la Base de Datos cumple con las siguientes características:

1. Archivo contiene un solo tipo de registro.
2. Cada registro es único.
3. Dado el valor de la llave es posible localizar físicamente al registro correspondiente.
4. Los registros pueden venir en cualquier orden, no existe una secuencia determinada.
5. Cada registro tiene un número fijo de campos, cada uno de los cuales son explícitamente nombrados en correspondencia con los atributos de la entidad de la cual son depósitos de instancias.
6. Cada uno de los campos toman su valor de un dominio de posibles valores, el cual queda determinado por el tipo del atributo correspondiente.

En la figura 2-5, se muestra las diversas vistas de una base de datos de tipo relacional.

## CONCEPTOS BASICOS DE BASES DE DATOS

VISTA EXTERNA	VISTA CONCEPTUAL	VISTA INTERNA
Tabla	Entidad	Archivo
Reglón	Instancia	Registro
Columna	Atributo	Campo
Identificador	Llave	Campo Llave

Figura 2-5. Relación entre las diversas vistas de la base de datos.

### 2.4.1.4 COMO SE ESTABLECEN LAS RELACIONES ENTRE ENTIDADES. -

Conceptualmente, una relación asocia una instancia de una entidad, que denominaremos padre, a un conjunto (posiblemente vacío) de instancias de otra entidad (que puede ser ella misma) a la que denominaremos hija

En el modelo en cuestión, no existe una estructura específicamente diseñada para implementar una relación.

Las relaciones entre entidades son transparentes para el sistema. Es el usuario quien establece e interpreta las relaciones existentes, a través de los valores de los atributos de las instancias.

## CONCEPTOS BASICOS DE BASES DE DATOS

Por ejemplo: los valores asociados al atributo DEPTO de la entidad EMPLEADO de la Base de Datos DEPTO EMP\_PROY (figura 2-4) son llaves existentes en la entidad DEPARTAMENTO. Estos valores, definen una relación entre las entidades DEPARTAMENTO y EMPLEADO. Dicha relación (llamémosle ES\_EMPLEADO) mapea cada instancia de la entidad DEPARTAMENTO, al conjunto de instancias de la entidad EMPLEADO cuyo atributo DEPTO tenga el valor de su LLAVE (No-DEPTO). La relación así interpretada se muestra en la figura 2-6.

INSTANCIAS DE DEPARTAMENTO				INSTANCIAS DE EMPLEADO			
[No-DEPTO	NOMBRE-DEPTO	JEFE-DEPTO]		[No-EMP	NOMBRE-EMP	DEPTO]	
[1	Contabilidad	3087	]  -->	{{3087	Mariana Castro	1	}}
[3	Compras	1576	]  -->	{{1245	Pedro Cisneros A.	3	}}
				[1576	Adriana Fernandez	3	}}
[4	Ventas	1187	]  -->	{{1187	Juan Pereira M.	4	}}
				[4032	Jose Martinez	4	}}

Figura 2-6. Mapeo establecido por la relación  
ES\_EMPLEADO.

La relación ES\_EMPLEADO puede ser implementada a través de un atributo de la entidad hija, y es de tipo 1:N. Este tipo de relaciones cumplen con la siguiente característica: una instancia en la entidad hija está asociada a una y solo una instancia en la entidad padre. En nuestro ejemplo, esta característica significa que un empleado no puede estar trabajando en dos departamentos diferentes.

Otro tipo de relaciones que se pueden implementar a través de un atributo, pero en este caso de la entidad padre, son las relaciones de tipo M:1. Este tipo de relaciones asocian un conjunto de instancias de la entidad padre a una instancia de la entidad hija. Una relación de éste tipo sería la relación inversa de ES\_EMPLEADO, la cual mapearía un conjunto de empleados al departamento donde trabajan.

CONCEPTOS BASICOS DE BASES DE DATOS

INSTANCIAS EN PROYECTO		INSTANCIAS EN EMPLEADO		
[No-PROYECTO	DESCRIPCIÓN ]	[No-EMP	NOMBRE-EMP	DEPTO]
[1111	Manejo de inventario]	[1245	Pedro Cisneros A.	3 ]
		[1576	Adriana Fernandez	3 ]
		[3087	Mariana Castro	1 ]
		[4032	Jose Martinez	4 ]}
[1010	Desarrollo ]	[1187	Juan Pereira M.	4 ]
		[1576	Adriana Fernandez	3 ]
		[3087	Mariana Castro	1 ]}

Figura 2-7. Mapeo establecido por la relación

TRABAJA\_EN.

Supongamos ahora, que se desea establecer la relación TRABAJA EN que nos mapea cada instancia de PROYECTO con los empleados (instancias de EMPLEADO) que trabajan para dicho proyecto, de la manera que se muestra en la figura 2-7. Nótese que un empleado puede trabajar en uno ó más proyectos y que un proyecto tiene asociados uno ó más empleados.

Para implementar esta relación, no sería suficiente un atributo extra en la entidad EMPLEADO, puesto que tendríamos que asociar más de un valor a dicho atributo (por ejemplo en el caso del empleado 1576 que trabaja en los proyectos 1111 y 1010). Es necesario entonces, recurrir a lo que llamaremos Entidades Extendidas, que son entidades que establecen relaciones entre otras entidades, por medio de sus llaves. Por ejemplo la relación TRABAJA EN podría ser implementada por la entidad que se muestra en la figura 2-8. La relación TRABAJA EN es de tipo M:N. Este tipo de relaciones asocian una instancia de la entidad padre a un conjunto de instancias de la entidad hija. Una instancia de la entidad hija puede estar asociada con más de una instancia de la entidad padre.

## CONCEPTOS BASICOS DE BASES DE DATOS

TRABAJA\_EN

PROYECTO ENTERO	EMPLEADO ENTERO
1010	1187
1010	1576
1010	3087
1111	1245
1111	1576
1111	3087
1111	4032

Figura 2-8. Entidad extendida para implementar la relación TRABAJA\_EN.

Las entidades extendidas no se encuentran asociadas a ningún tipo de objetos en la vida real. Se les denomina extendidas pues registran información acerca de características de los objetos que no quedaron asociadas a ningún atributo de alguna entidad (por ejemplo el hecho de que un empleado trabaje para algún proyecto).

Una relación puede representarse gráficamente por dos rectángulos (cada uno de ellos asociado con una entidad) y una flecha (apuntando hacia la entidad hija) que las une. El nombre y tipo de la relación se especifica a un lado de la flecha. Por ejemplo la relación TRABAJA\_EN se visualiza gráficamente como se muestra en la figura 2-9.

Al diseñar el esquema general de la Base de Datos, resulta muy útil el representar las relaciones gráficamente. Como vimos en esta sección, una relación 1:N implicará un atributo extra en la entidad hija, una M:1, en la entidad padre, y una M:N, la existencia de una entidad extendida.

## CONCEPTOS BASICOS DE BASES DE DATOS

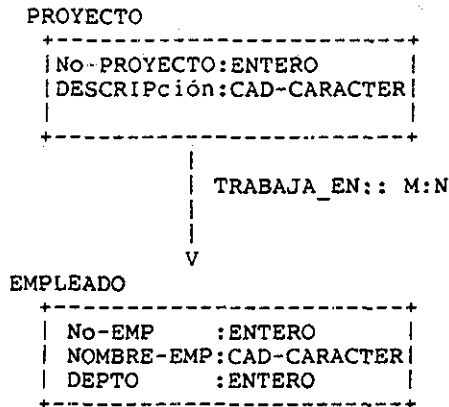


Figura 2-9. Representación gráfica de la relación TRABAJA\_EN.

### 2.4.1.5 INDICES. -

Frecuentemente se desea obtener el conjunto de instancias de una entidad que tienen asociado un valor dado a cierto atributo. La estructura índice permite obtener dicho conjunto sin necesidad de consultar cada una de las instancias de la entidad.

Conceptualmente, un índice sobre un atributo X, es una función cuyo dominio queda definido por el conjunto de valores que toma el atributo X en cualquiera de las instancias de la entidad, y cuyo contradominio es el conjunto de instancias de la entidad (identificadas por su llave). Dicha función, asocia a un valor Y, el conjunto de instancias cuyo atributo X toma el valor

## CONCEPTOS BASICOS DE BASES DE DATOS

Y.

Se puede pensar en un índice como una clasificación total de las instancias de una entidad, de acuerdo a los valores del atributo asociado con el índice. Ninguna instancia está clasificada bajo dos valores diferentes.

VALOR	INSTANCIAS
	(Identificadas por su llave)
1	--> {3087}
3	--> {1245, 1576}
4	--> {1187, 4032}

Figura 2-10. Índice sobre el atributo DEPTO.

Un índice sobre un atributo que implementa una relación, define el mapeo de dicha relación. Por ejemplo, el índice sobre el atributo DEPTO de la entidad EMPLEADO sería como se muestra en la figura 2-10. Que es equivalente al mapeo establecido por la relación ES\_EMPLEADO.

### 2.4.1.6 REGLAS DE INTEGRIDAD DE LA BASE DE DATOS -

Las reglas de integridad de la Base de Datos definen los estados válidos de las instancias de una Base de Datos. El modelo en cuestión se rige por las siguientes reglas :

## CONCEPTOS BASICOS DE BASES DE DATOS

En todo momento de la vida de la Base de Datos, se cumple que,

### R1- Verificación del tipo de los Atributos.

Para todas las instancias existentes en una entidad, el valor asociado a cada uno de los atributos, es del tipo adecuado.

### R2- Completez y Actualidad de los índices.

#### Completez.-

El dominio de un índice sobre un atributo dado es el conjunto de valores asociados a dicho atributo en cualquiera de la instancias de la entidad. La unión de las imágenes de un índice, es la totalidad de las instancias de la entidad sobre la cual está definido. Es decir, todas las instancias de la entidad deben estar asociadas con algún valor a través del índice.

#### Actualidad.-

Cada una de las instancias pertenecientes a la imagen de un valor X, tienen asociado el valor X al atributo sobre el cual está definido el índice.

### R3- Unicidad de Llaves.

No existen dos instancias de una entidad con la misma Llave. Esta regla garantiza que el valor de la llave es suficiente para determinar en forma única una instancia de la entidad.

### R4- Integridad de la Relaciones.

Las llaves externas utilizadas para implementar relaciones entre entidades, corresponden a la llave de una instancia de alguna otra entidad.

## 2.4.1.7 COMO RECORRER LA BASE DE DATOS. -



## CONCEPTOS BASICOS DE BASES DE DATOS

La selección de un conjunto de instancias que satisfacen cierta característica, se realiza mediante consultas a los índices y mediante operaciones entre los conjuntos así obtenidos. Si el índice implementa una relación, nos permite viajar entre las entidades relacionadas.

Por ejemplo, supongamos que nos gustaría obtener el conjunto de instancias de la entidad EMPLEADO, que son empleados del departamento 3 y que trabajan en el proyecto 1010.

Para seleccionar dicho conjunto, primeramente obtendríamos, consultando el índice sobre DEPTO en la entidad EMPLEADO, al conjunto {1245,1576}, posteriormente, consultaríamos el índice sobre PROYECTO en la entidad extendida TRABAJA\_EN y obtendríamos al conjunto {1187,1576,3087}. Finalmente realizaríamos la intersección de ambos conjuntos para obtener el conjunto deseado {1576}.

La estructura relacional es muy fácil de entender, se aproxima más a la realidad, y la manera en que se acostumbra pensar. El lenguaje de manipulación de los datos para los modelos relacionales que es un conjunto de operadores proporcionados para manipular los datos, está representado en la forma relacional, es muy natural para los usuarios. Aunque la simplicidad de la representación de los datos no es nuestro propósito, pero por la uniformidad de la representación de la información, se puede realizar las operaciones sobre la base de datos en una manera uniforme, o sea, sólo se necesita tener un operador para cada una de las funciones básicas.

### 2.4.2 MODELO JERARQUICO

Los sistemas de bases de datos jerárquicos muy conocidos son :

1. el sistema IMS/DB (Information Management System) de IBM, este sistema tiene un lenguaje DML (record-at-a-time) conocido como DL/I;
2. Informatic's Mark IV;

CONCEPTOS BASICOS DE BASES DE DATOS

3. El sistema 2000 ( el sistema de MRI ), este sistema proporsiona un lenguaje orientado a conjuntos;
4. TDMS de SDC (Time-Shared Data Management System).

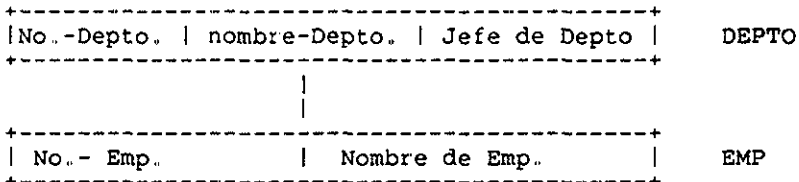


Figura 2-11. Tipo de LDBR para la base de datos DEPTO\_EMP.

Una posible representación en modelo jerárquico del ejemplo de bases de datos DEPTO\_EMP se muestra en la figura 2-11 y la figura 2-12. donde la figura 2-11 es una representación de tipo de registro lógico de base de datos (LDBR Logical Data Base Register) para la base de datos DEPTO\_EMP, y la figura 2-12 es la representación de las ocurrencias de ésta.

CONCEPTOS BASICOS DE BASES DE DATOS

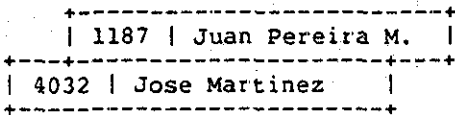
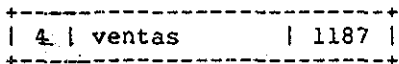
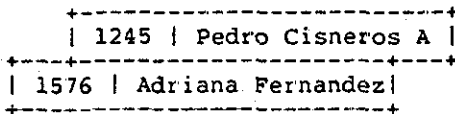
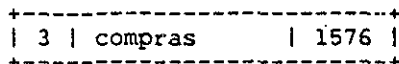
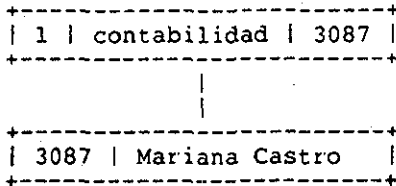


Figura 2-12. Representación jerárquica de la base de datos DEPTO\_EMP.

## CONCEPTOS BASICOS DE BASES DE DATOS

Para describir unos conceptos básicos acerca a los modelos jerárquicos, se usarán los términos de IMS.

### 2.4.2.1 VISTA CONCEPTUAL -

La vista conceptual del modelo jerárquico consiste en una colección de bases de datos físicas. Una base de datos física es un conjunto ordenado cuyos elementos son todas las ocurrencias de un tipo de registros de la base de datos física. Una ocurrencia de un registro de una base de datos física consiste en ocurrencias de segmentos de longitud fija o variable; y una ocurrencia de un segmento consiste en un conjunto de ocurrencias de campos de longitud fija.

Los registros de una base de datos física tiene la forma como se representa en la figura 2-12, cada ocurrencia de un registro de la base de datos física tiene forma de un árbol. Cada árbol consiste en una raíz que es el elemento del tope del árbol. La raíz puede tener cualquier número de tipos de segmentos -- Hijos, inclusive cero hijos. Cada hijo de la raíz puede tener cualquier número de tipos de hijos, así sucesivamente. Teóricamente se puede tener de cero a un número infinito de hijos en cada nivel del árbol, y se puede tener un número infinito de niveles en un árbol. Puede tener un número infinito de ocurrencias de segmentos en el nivel de su hijo. Ninguna ocurrencia del segmento hijo puede existir sin que exista su padre -- el nivel superior de segmento hijo. Esto establece la filosofía de la jerarquía.

Cada base de datos física está definida por una descripción de base de datos junto con su mapeo al almacenamiento físico. En esta descripción, está especificada desde el nivel superior al nivel inferior. Para cada nivel, se define el nombre del segmento, el nombre del padre del segmento, la longitud en bytes del segmento y sus campos; para cada campo, se define el nombre del campo, la longitud del campo, y la posición donde empieza el campo.

## CONCEPTOS BASICOS DE BASES DE DATOS

```
1   DBD           NAME=DEP_EMP
2   SEGM          NAME=DEPTO, BYTES=256
3   FIELD         NAME=(No.-DEPTO, SEQ), BYTES=3, START=1
4   FIELD         NAME=NOMBRE-DEPTO, BYTES=253, START=4
5   SEGM          NAME=EMP, PARENT=DEPTO, BYTES=300
6   FIELD         NAME=(No.-EMP, SEQ), BYTES=4, START=1
7   FIELD         NAME=NOMBRE-EMP, BYTES=296, START=5
```

Figura 2-13. Descripción de la base de datos  
DEPTO\_EMP (parte del esquema  
conceptual).

En la figura 2-13 se muestra la descripción de la base de datos de nuestro ejemplo, la base de datos DEPTO\_EMP.

Donde

en la línea 1, se asigna el nombre DEPTO\_EMP a la base de datos del ejemplo;

en la línea 2, se define el tipo del segmento de la raíz, le asigna el nombre DEPTO, y tiene 256 bytes de longitud;

en la línea 3-4, se define los tipos del campo de DEPTO, cada uno tiene un nombre, la longitud en bytes, y la posición inicial dentro del segmento. EL primer campo No.-DEPTO está definido como el campo de secuencia del segmento indicado por "SEQ", esto significa que dentro de la base de datos física DEPTO\_EMP, las ocurrencias de los registros de la base de datos física estarán ordenados en el orden ascendente por el

## CONCEPTOS BASICOS DE BASES DE DATOS

campo No.-DEPTO. De la igual manera, en las líneas 5-7 se define el segmento EMP y sus campos.

### 2.4.2.2 VISTA EXTERNA -

Cada vista externa particular de un usuario consiste en una colección de bases de datos lógicas, donde cada base de datos lógica es un subconjunto de bases de datos físicas correspondientes. Una base de datos lógica es un conjunto ordenado, cuyos elementos son todas las ocurrencias de un tipo de registro de bases de datos lógicas, un tipo de registros de bases de datos lógicas es un arreglo jerárquico de tipos de segmentos. Esta jerarquía está dividida desde la jerarquía de registros de bases de datos físicas de acuerdo de las siguientes reglas :

1. Cualquier tipo de segmentos de jerarquías de registros de base de datos física, junto con sus dependientes (descendientes) pueden ser omitidos desde la jerarquía del registro de la base de datos lógica.
2. Los campos de los tipos de los segmentos pueden ser un subconjunto de los tipos de segmentos de registros de base de datos física correspondiente, y puede ser reordenados dentro del tipo de segmento de registro de base de datos lógica.

Cada base de datos física está definida junto con su mapeo a la base de datos lógica correspondiente que se le llama Bloque de Comunicación de Programas ( Program Communication Block -- PCB ). El conjunto de todos los PCBs para un usuario corresponde a un esquema más su definición de mapeo asociada llamada bloque de especificación de programa ( Program Specication Block -- PSB ).

### 2.4.2.3 VISTA INTERNA -

## CONCEPTOS BASICOS DE BASES DE DATOS

Como en el modelo relacional, se puede almacenar físicamente cada relación en un archivo, aquí, en el modelo jerárquico, se puede almacenar la estructura de los árboles en un sólo archivo.

Hay muchas técnicas de almacenamiento, y varios métodos de accesos para implementar el nivel interno para el modelo jerárquico; como son :

1. métodos de acceso secuencial OS/OV ( QSAM y BSAM, conjuntamente se conoce como SAM );
2. método de acceso secuencial indexado OS/VS (ISAM);
3. método de acceso de almacenamiento virtual OS/VS (VSAM);
4. método de acceso especial de IMS llamado método de acceso secuencial de sobreflujo (overflow) (OSAM).

Para ver con más detalles cada uno de los métodos, se puede ver las referencias que están en la bibliografía [23] para SAM e ISAM, [24] para VSAM, [25] para OSAM.

La jerarquía es una manera natural para representar muchas cosas del mundo real. La base de datos DEPTO\_EMP es un buen ejemplo.

La estructura jerárquica es mucho más compleja que la estructura relacional. Los archivos de la base de datos física no son como los del modelo relacional. Se manejan archivos de varios tipos de registros, así como las ligas entre las ocurrencias del mismo tipo de registros y las de diferentes tipos (padre e hijo). Los operadores son mucho más complicados; para referirse a una ocurrencia de un tipo de registro especial, se tienen que especificar todos los niveles superiores a él.

CONCEPTOS BASICOS DE BASES DE DATOS

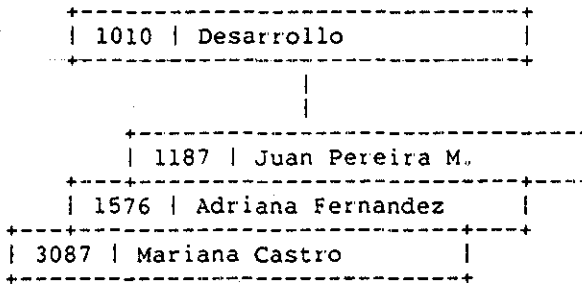
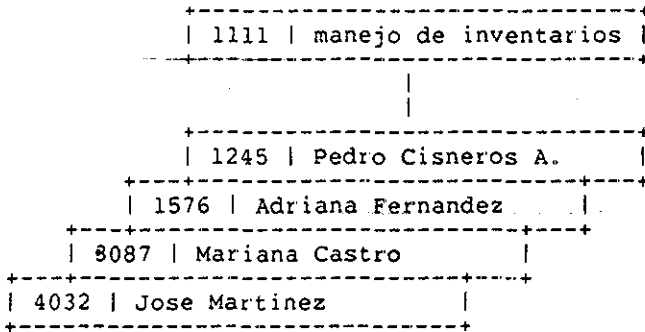


Figura 2-14. Representación de la base de datos PROY\_EMP en modelo jerárquico.



## CONCEPTOS BASICOS DE BASES DE DATOS

Para algunas operaciones, este tipo de estructuras es muy rápido y simple. Por ejemplo, la operación de consultar todas las informaciones de todas las personas que trabajan en el departamento 1, se puede localizar al registro que contiene la estructura del árbol que corresponde al departamento y recorrer todas las ocurrencias del segmento EMP. Pero para algunas operaciones, se complica mucho más este tipo de modelos. Por ejemplo, para la base de datos en cuestión, si queremos saber en qué departamento trabaja Jose Martinez, tenemos que recorrer todos los registros de todos los árboles hasta que encontremos el nombre Jose Martinez. Además se presentan muchos problemas para las operaciones de modificaciones para las bases de datos que tienen una relación de N a N. Por ejemplo, la base de datos PROY EMP de la figura 2-14; si se necesita cambiar algún segmento de EMP, se tiene que revisar todas las ocurrencias del registro que representa la estructura. Por ejemplo si Mariana Castro se cambia de departamento, tenemos que modificar tanto la ocurrencia del segmento que está en el proyecto 1111, como la del que está en el proyecto 1010.

### 2.4.3 MODELO DE REDES

Se pueden definir los sistemas de bases de datos de tipo de redes como aquellos que soportan estructuras de datos en las cuales

1. Un padre puede serlo de muchos árboles distintos.
2. Un hijo puede serlo de muchos árboles distintos.

Entre los sistemas de bases de datos de tipo de redes, podemos distinguir varios tipos, según como estén implementadas las redes :

1. De Redes Explícitas : los árboles están representados por medio de los apuntadores.
2. De Redes Implícitas : en los cuales una red se consigue haciendo corresponder registros de distintos tipos que tienen sendas claves de indexación con valores repetidos, teniendo todas ellas el mismo conjunto de valores posibles.

## CONCEPTOS BASICOS DE BASES DE DATOS

Como ejemplos de sistemas de bases de datos de tipo de redes son :

- el sistema ADABAS, de redes implícitas;
- los sistemas de bases de datos basados en las especificaciones del CODASYL, en los cuales un mismo registro puede ser a la vez padre e hijo;
- el TOTAL de Cincom, cuyos tipos de registro son o sólo padres o sólo hijos, sin adición automática;
- IDS de Honeywell (Integrated Data Store);
- DBOMP de IBM;
- DMS 1100 de UNIVAC;
- IDMS de Cullinane.

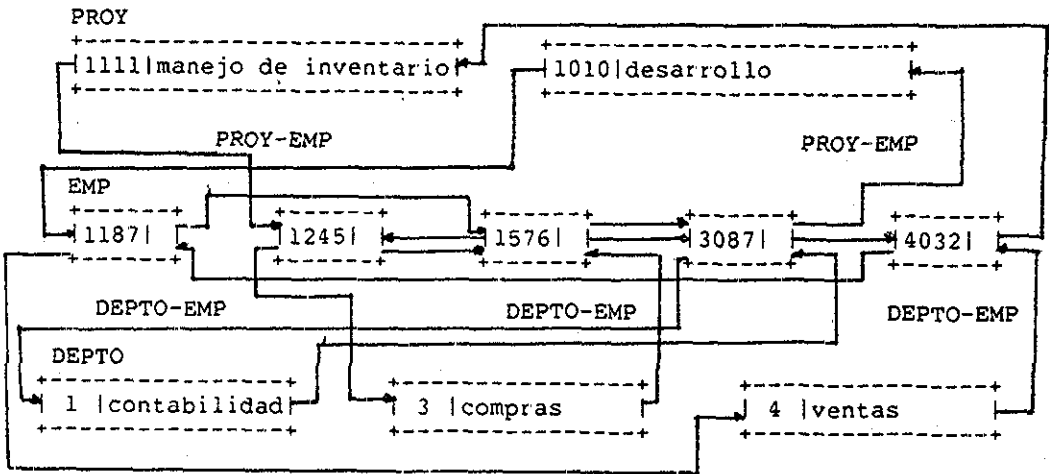


Figura 2-15. Representación de modelo de redes de la base de datos DEPTO\_EMP\_PROY.

## CONCEPTOS BASICOS DE BASES DE DATOS

Una representación en modelo de redes de la base de datos de nuestro ejemplo, se muestra en la figura 2-15.

En esta sección, se utilizarán unos conceptos del modelo de redes en términos de CODASYL (Conference on Data System and Languages) la cual es una organización de fabricantes, grandes usuarios de Informática y organismos oficiales norteamericanos.

El CODASYL desarrolló y mantiene las especificaciones del COBOL y en lo que nos concierne ha emitido las especificaciones de 1971 -- conocidas como las especificaciones del DBTG (Data Base Task Group), con sus modificaciones de 1973 y 1978. Dichas especificaciones definen tres lenguajes :

1. DDL (Data Definition Language) de esquemas, para el nivel conceptual;
2. DDL de subesquemas, para el nivel externo;
3. DML (Data Manipulation Language), lenguaje de acceso de lenguaje huésped.

En 1978 se publicó una propuesta de lenguaje para la definición de esquemas internos, llamado DSDL (Data Storage Definition Language).

Algunas implementaciones de las especificaciones del CODASYL son el IDMS, el IDS-II, el DMS-110 y el BDMS-11.

### 2.4.3.1 VISTA CONCEPTUAL -

La vista conceptual de DBTG está definida por el esquema. Este esquema consiste esencialmente en definiciones de varios tipos de registros de la base de datos, los cuales contienen conjuntos de datos llamados datos elementales. Las ocurrencias de un tipo de registro se agrupan en conjuntos. En la figura 2-16 se muestra la base de datos DEPTO\_EMP en forma de redes, en donde DEPTO y EMP son datos, y las tres ocurrencias del tipo de registro DEPTO\_EMP están agrupadas en un conjunto llamado DEPTO\_EMP; al padre se le llama propietario (owner) y a los hijos se les llaman miembros (members). En este ejemplo, DEPTO es el propietario y EMP el miembro. Cada ocurrencia de un conjunto consiste en una ocurrencia de propietario con cero o más

## CONCEPTOS BASICOS DE BASES DE DATOS

ocurrencias de miembros. En el caso del ejemplo, la situación de cero miembros puede ser que un proyecto todavía no tiene empleados asignados.

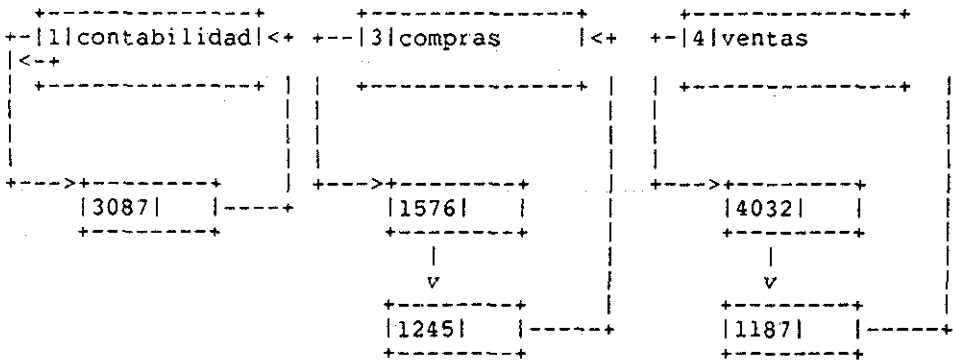


Figura 2-16. Base de datos DEPTO\_EMP.

## CONCEPTOS BASICOS DE BASES DE DATOS

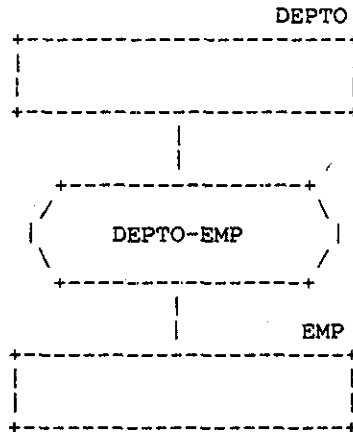


Figura 2-17. Diagrama de estructura de datos para la base de datos DEPTO\_EMP.

Otra forma de representar los tipos de conjuntos es mediante un diagrama de estructura de datos. En la figura 2-17 se representa el conjunto DEPTO\_EMP en la forma de diagrama de estructura de datos. A esta técnica para representar los conjuntos se le llaman la técnica de Buchman [9].

El modelo de redes puede representar muchos tipos de jerarquías, como son :

1. jerarquía con más de un nivel de dependencia;
2. jerarquía con más de un tipo de registros en el nivel de dependencia;
3. jerarquía con el mismo tipo de registro en más de un nivel. En la referencia [22] se describe con mucho detalle cada una de estas jerarquías y las técnicas para representarlas.

## CONCEPTOS BASICOS DE BASES DE DATOS

### 2.4.3.2 VISTA EXTERNA -

Una vista externa está definida por un subesquema. Un subesquema consiste esencialmente en una especificación de los tipos de registros en que está interesado por el usuario, sus datos y las relaciones entre estos registros. En un subesquema se puede omitir la declaración de uno o más conjuntos o más datos elementales, siempre y cuando la declaración sea consistente. Un subesquema es básicamente un subconjunto de un esquema con las siguientes diferencias importantes :

En un subesquema :

1. Se puede definir nombres propios para los conjuntos, registros, y los datos elementales;
2. Los datos elementales pueden tener diferentes tipos de datos;
3. El orden relativo dentro del registro que lo contiene puede ser cambiado.

Como se ve, los subesquemas tienen limitación de que siempre se refiere a un esquema, es decir, no puede haber subesquemas de otros subesquemas. Un programa sólo puede utilizar un subesquema, pero un mismo subesquema puede ser utilizado por muchos programas distintos.

Una red es una estructura más general que una estructura jerárquica, porque una ocurrencia de un registro dado puede tener cualquier número de inmediatos superiores, o sea, un hijo puede tener cualquier número de padres, no se limita a uno solo como el caso de estructura jerárquica. La estructura de redes nos permite tener correspondencias de N a N más directamente. Se puede comparar las figuras 2-12 y 2-15. Claro que la estructura de redes es mucho más complicada que la estructura jerárquica tanto en la estructura como en DML. En general, en la estructura de redes hay más componentes y hay más operadores para manejar los componentes.

## CONCEPTOS BASICOS DE BASES DE DATOS

### 2.4.3.3 VISTA INTERNA -

El DBAWG (Data Base Administration Working Group) del CODASYL tienen un lenguaje para la definición del esquema interno llamado Data Storage Description Language (DSDL). Por medio de DSDL se definen esquemas de almacenamiento. Un esquema de almacenamiento se puede considerar un esquema interno o bien una parte de éste, puesto que define la implementación física de un esquema conceptual o de parte de él. Sin embargo no contempla todos los aspectos ligados a la implementación como el tipo de dispositivo físico utilizado. Para mayor detalle, referirse a [26].

### 2.4.4 RESUMEN

Un sistema de bases de datos debe ser capaz de representar dos tipos de objetos : entidades y relaciones. Fundamentalmente no hay diferencias entre los dos, una relación es un tipo especial de entidades. Los tres modelos sólo difieren en la manera que permite al usuario visualizar y manipular las relaciones.

El modelo relacional permite representar las relaciones en la misma manera que las otras entidades (par de relaciones); en los modelos de redes y jerárquico, ciertas relaciones son representadas mediante ligas. Estas ligas pueden representar asociaciones de 1 a N. La diferencia entre el modelo de redes y el modelo jerárquico está en que con las ligas del modelo de redes se puede combinar para formar estructuras muy complicadas de asociaciones de N a N, y esto no es posible con el modelo jerárquico. Otra diferencia es que en el modelo de redes, las ligas tienen nombre mientras en el modelo jerárquico no.

Hemos visto unos detalles de cada uno de los modelos, y sus características, sus ventajas y sus desventajas. Para ciertas aplicaciones, puede resultar más conveniente un modelo que otro. Cuál modelo se escogería para problema real ? eso ya depende del problema, se tiene que hacer un análisis del problema, una evaluación para tomar la decisión final.

## CONCEPTOS BASICOS DE BASES DE DATOS

### 2.5 LENGUAJES DE MANIPULACION DE BASE DE DATOS

Actualmente existen muchos lenguajes de manipulación de base de datos, debido a que el tipo de estructura de datos que manejan son totalmente distintos. El lenguaje de manipulación de datos para una base de datos relacional es muy diferente que uno para base de datos de tipo redes o jerárquico. Como el EDITOR de bases de datos está orientado a una base de datos tipo relacional, vamos a discutir algunos lenguajes para bases de datos de tipo relacional como son SQL y QBE. También vamos a ver algebra relacional y cálculo relacional, por la importancia que tienen, muchos lenguajes existentes están basados en ellos.

#### 2.5.1 ALGEBRA RELACIONAL

El Algebra relacional es una colección de operaciones sobre relaciones. Cada operación toma una o más relaciones como su entrada, y produce otra relación como resultado. Codd originalmente definió un conjunto de operaciones [2]. Después ha habido varias versiones variantes (ver referencias [10], [11], [12], [13]).

##### 2.5.1.1 UNION -

La unión de dos relaciones A y B, A UNION B, es otra relación que contiene todas las tuplas (tuple) que pertenecen a A o B o ambas.



## CONCEPTOS BASICOS DE BASES DE DATOS

### 2.5.1.2 INTERSECCION -

La intersección de dos relaciones A y B,  $A \text{ INTERSECT } B$ , es otra relación que contiene las tuplas que pertenecen a A y B.

### 2.5.1.3 DIFERENCIA -

La diferencia de dos relaciones A y B,  $A \text{ MINUS } B$ , es otra relación que contiene las tuplas que pertenecen a A y no a B.

### 2.5.1.4 PRODUCTO CARTESIANO -

El producto cartesiano de dos relaciones A y B,  $A \text{ TIMES } B$ , es otra relación, la cual tiene todas las tuplas donde cada tupla es la concadenación de una tupla de A con una de B.

### 2.5.1.5 SELECCION -

El resultado de una selección sobre una relación R mediante una condición ( en general es una combinación booleana de términos, cada término puede ser una comparación simple que puede resultar verdadero y falso. Si un término involucra una comparación entre valores de dos atributos de la misma relación, estos dos deben estar definidos en el mismo dominio ) es una relación que tiene los mismos atributos de R, y contiene sólo las tuplas que cumplen la condición.

## CONCEPTOS BASICOS DE BASES DE DATOS

### 2.5.1.6 PROYECCION -

La proyección de una relación R, es otra relación que tiene un subconjunto de atributos que tiene R, con las tuplas que sólo corresponden a estos atributos eliminando las tuplas duplicadas.

### 2.5.1.7 JUNCION -

La junción (JOIN) de dos relaciones A y B, es la relación que tiene la concatenación de los atributos de A y los de B, y por lo menos un atributo de A tiene que tener el mismo dominio que uno de B. Esta relación resultante tiene las tuplas según un cierto operador (=, <>, <, >, <=, >=) se obtiene concatenando cada tupla de A con todas aquellas de B tales que los valores respectivos de A y B complan la condición expresada. Un caso particular es el llamado junción natural, en el que las dos relaciones tienen un atributo en común -- el cual por tanto aparecerá una sola vez en el conjunto de los atributos de la relación resultada.

### 2.5.1.8 DIVISION -

La división de dos relaciones A que tiene m+n atributos y B que tiene n atributos es una relación que tiene m atributos. El m+i atributo de A y el i atributo deben estar definido en el mismo dominio. La división es parecida a la división aritmética. Para mayor detalle, se puede ver en Date [22] que tiene un ejemplo muy fácil para entender.

## 2.5.2 CALCULO RELACIONAL

## CONCEPTOS BASICOS DE BASES DE DATOS

El cálculo relacional es el cálculo de predicados usado como base para un lenguaje de preguntas (query) para base de datos relacional, fué propuesto por Codd [2]. El cálculo relacional se divide en Cálculo relacional orientado a tuplas y Cálculo relacional orientado a dominios.

### 2.5.2.1 CALCULO RELACIONAL ORIENTADO A TUPLAS -

El cálculo relacional orientado a tuplas está basado en construcciones de expresiones de cálculos de tuplas. Se construye una expresión con los siguientes elementos :

1. **Variables de tuplas** : un atributo se identifica en la forma V.A donde V es el nombre de la relación que contiene A, y es la variable de tuplas, A es el nombre del atributo;
2. **Condición** : se especifica la condición en combinación booleana que tiene la forma  $x*y$  donde \* es cualquier de =, <>, <, <=, >, >=, uno de x y y tiene que ser de la forma V.A por lo menos, y el otro puede ser constante o una condición;
3. **Fórmulas bien formadas (Well-Formed Formulas -- WFFs)** : una FBF se construye por condiciones, operadores booleanos (AND, OR, NOT), y calificadores ( $\exists, \forall$ ), siguiendo las siguientes reglas :
  - cada condición es una FBF;
  - si f es una FBF, (f) y NOT (f) también son FBFs;
  - si f y g son FBFs, (f AND g) y (f OR g) también son FBFs;
  - si f es FBF en la cual T es una variable libre, T(f) y T(f) también son FBFs;
  - nada más es FBF.

## CONCEPTOS BASICOS DE BASES DE DATOS

### 2.5.2.2 CALCULO RELACIONAL ORIENTADO A DOMINIOS -

Se puede construir una expresión de cálculo relacional orientado a dominios con los siguientes elementos :

1. **Variables de dominio :** son las variables que representan dominios;
2. **Condiciones :** pueden ser comparaciones simples de forma  $x*y$  como cálculo de tuplas, excepto que  $x$  y  $y$  son variables de dominio o constantes; puede ser de la forma  $R(A:V, A:V, \dots, \dots)$ , donde  $R$  es una relación,  $A$  es un atributo de  $R$ , y  $V$  es una variable de dominio o constante;
3. **Fórmulas bien formadas :** como se describe en la sección anterior.

### 2.5.3 SQL

SQL (Structured Query Language) es el sublenguaje de datos del sistema R. Incluye facilidad DDL y DML. DML opera sobre nivel externo y conceptual. DDL puede ser usado para definir objetos en el nivel externo, nivel conceptual y los índices del nivel interno. SQL proporciona también la facilidad de control de datos.

Las consultas que se hace mediante SQL tiene una estructura principal de `SELECT a FROM r WHERE c`, donde  $a$  es un conjunto de nombres de atributos,  $r$  es un conjunto de relaciones y  $c$  es la condición que se especifica para la consulta. SQL entrega el resultado de una consulta al usuario en una forma tabular. Con SQL también se puede realizar operaciones de modificaciones de la base de datos, y tiene una forma similar a la de consulta. Para mayor detalle, se puede ver la referencia [14].

SQL permite al usuario usarlo tanto en una terminal de video interactivamente como en un programa. Tiene las siguientes características importantes :

1. **Simplicidad :** muchos problemas pueden ser expresados en SQL en una manera muy fácil y concisa;

## CONCEPTOS BASICOS DE BASES DE DATOS

2. **Completez** : el lenguaje es relacionalmente completo. Esto es para una pregunta muy larga, el usuario no tiene que recurrir a repeticiones (loops) ni tiene que hacer saltos;
3. **Noprocedimental** : Para hacer una pregunta, el usuario sólo especifica qué datos quiere, y no necesita especificar cómo obtenerlos;
4. **Independencia de datos** : Los usuarios no necesitan saber la estructura física, sino sólo la estructura lógica, de esta manera, se logra la independencia de los datos físicos;
5. **Soporte para lenguaje de alto nivel** : SQL proporciona un conjunto de elementos indispensables de SQL para este propósito.

### 2.5.4 QBE

QBE (Query By example) fué desarrollado por M. M. Zloof en un laboratorio de IBM. QBE tiene capacidad de aceptar las preguntas del usuario en forma tabular, y le entrega el resultado al usuario en forma tabular también. QBE sólo proporciona la facilidad para los usuarios de terminal de video. Para mayor detalle se puede ver las referencias [15]-[21].

Con QBE se puede formar preguntas especificando las condiciones en forma tabular. QBE permite al usuario especificar la condición usando los operadores relacionales =, <>, <, <=, >, >=, se puede especificar una condición sobre más de una tabla, en caso de que no se puede especificar alguna condición por la limitación de las tablas, se le proporciona otra tabla de condición extra al usuario. Se puede lograr especificar los operadores booleanos sólo variando la forma en que se llene la tabla para la pregunta. Se puede sacar el resultado de una pregunta en el orden que especifica el usuario. El QBE puede entregarle al usuario el reporte de una consulta conteniendo solamente los valores de los atributos que fueron especificados por el usuario. De esta manera, el usuario se siente más libre para trabajar con QBE, construyendo las preguntas en una manera más natural.

## CAPITULO 3

### PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

El objetivo de este capítulo es proporcionar un panorama general del modelo conceptual del sistema de las bases de datos implementado, para el cual se diseñó e implementó el EDITOR de bases de datos, así como registrar decisiones en el análisis que impactaron el diseño de la implementación.

#### 3.1 HIPOTESIS DE LA IMPLEMENTACION

Al diseñar la implementación del modelo conceptual, se manejaron varias hipótesis relacionadas con técnicas de diseño, con los recursos disponibles y con las características de las aplicaciones definidas sobre la Base de Datos.

Al considerar las estructuras de datos (lógicas y de control) que soportarían al modelo conceptual se tomaron en cuenta los siguientes factores:

1. Las estructuras de datos a implementar deberían permitir a cada programa de aplicación, acceso rápido a la información requerida.

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

2. Ser compactas y minimizar dentro de lo posible la redundancia.
3. No deberían de requerir mucho software para soportarlas. Se deseaba compatibilidad entre las estructuras para manejarlas con un conjunto mínimo de operaciones primitivas.
4. Seguridad en el manejo.
5. Identificación clara de sus funciones.

Las aplicaciones definidas sobre la Base de Datos permitían suponer de que después del periodo inicial de carga la información permanecería básicamente estática.

La seguridad y consistencia de la información fué un factor determinante.

### 3.2 IMPLEMENTACION DE LAS ESTRUCTURAS

#### 3.2.1 IMPLEMENTACION DE UNA ENTIDAD

Una entidad se implementó con un archivo estructurado llamado archivo entidad. Dicho archivo está formado por las siguientes componentes:

- + Tabla descriptora de la estructura del archivo.
- + Esquema.
- + Depósito de instancias físicas.

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

- + Tabla hash.
- + Bit Map.

La tabla descriptora de la estructura del archivo contiene la información necesaria para acceder las otras cuatro componentes del archivo. Dicha información consiste, para cada componente, de su dimensión, su dirección inicial física y la dimensión de sus registros lógicos.

La componente esquema consiste de la descripción de los atributos de la entidad. Dicha componente es cargada en memoria formando la tabla descriptora de atributos (TDAT) al abrir la entidad. Esta componente contiene información de las características físicas de cada uno de los atributos como son tipo, longitud, posición, archivo índice asociado, etc. Una de estas características es la versión del atributo que es utilizada para verificar la congruencia entre un archivo entidad y los archivos índices generados sobre dicha entidad. Las versiones de un atributo específico de la entidad y de su índice correspondiente deben ser iguales para que dicho índice pueda ser utilizado para acceder la Base de Datos.

La componente esquema es actualizada cada vez que existen modificaciones al contenido de las instancias de la entidad.

Los otros tres componentes se encuentran relacionados con la implementación del depósito de instancias, las reglas de integridad que le incumben y la eficiencia de su manejo:

El Depósito de instancias es un archivo que contiene todas las instancias de una entidad dentro de una base de datos. Cada uno de sus registros corresponde a una instancia del depósito de instancias de la entidad.

La Tabla Hash, mapea el valor de una llave a la localización física de un bloque físico del disco. La instancia correspondiente se localiza en este bloque por medio de una búsqueda binaria. Al dar de alta un registro, se consulta la Tabla para ver si existía anteriormente.

El Bit Map indica por medio del valor del bit correspondiente a la posición lógica del registro, si el espacio está disponible para insertar un nuevo registro o no. Esta estructura facilita las operaciones de insertar y borrar un registro. El Bit Map también sirve para mantener la consistencia de la Tabla de Hash con el pool de instancias. Por ejemplo, el procedimiento que se lleva para insertar una nueva instancia es el siguiente :



## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

- a) encontrar el espacio libre para insertar la instancia revisando el Bit\_Map;
- b) insertar la instancia;
- c) actualizar la tabla de Hash;
- d) actualizar el Bit\_map.

Supongamos que se cae la máquina en alguno de los pasos, la base de datos no se queda inconsistente; se puede insertar la instancia de nuevo como si no hubiera pasado nada.

### 3.2.2 IMPLEMENTACION DE INDICES

Un índice corresponde a un archivo estructurado (Tipo ISAM) llamado archivo índice, con las siguientes componentes:

1. Tabla descriptora de la estructura del archivo.
2. TABLAS. Almacena las tablas del directorio de acceso.
3. ARREGLOS. Almacena los arreglos de elementos de cada conjunto.

La tabla descriptora de la estructura del archivo contiene la información necesaria para acceder las otras dos componentes del archivo. Dicha información consiste, para cada componente, de su dimensión, su dirección inicial física y la dimensión de sus registros lógicos.

La componente TABLAS, almacena el directorio en una estructura jerárquica de dos niveles de acceso. La presente implementación se ha limitado a dos niveles con 127 valores cada tabla de ambos niveles, dando un total de 16,129 (127\*127) clases diferentes, o sea que un atributo indexado no puede tomar más de ese número de valores.

PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

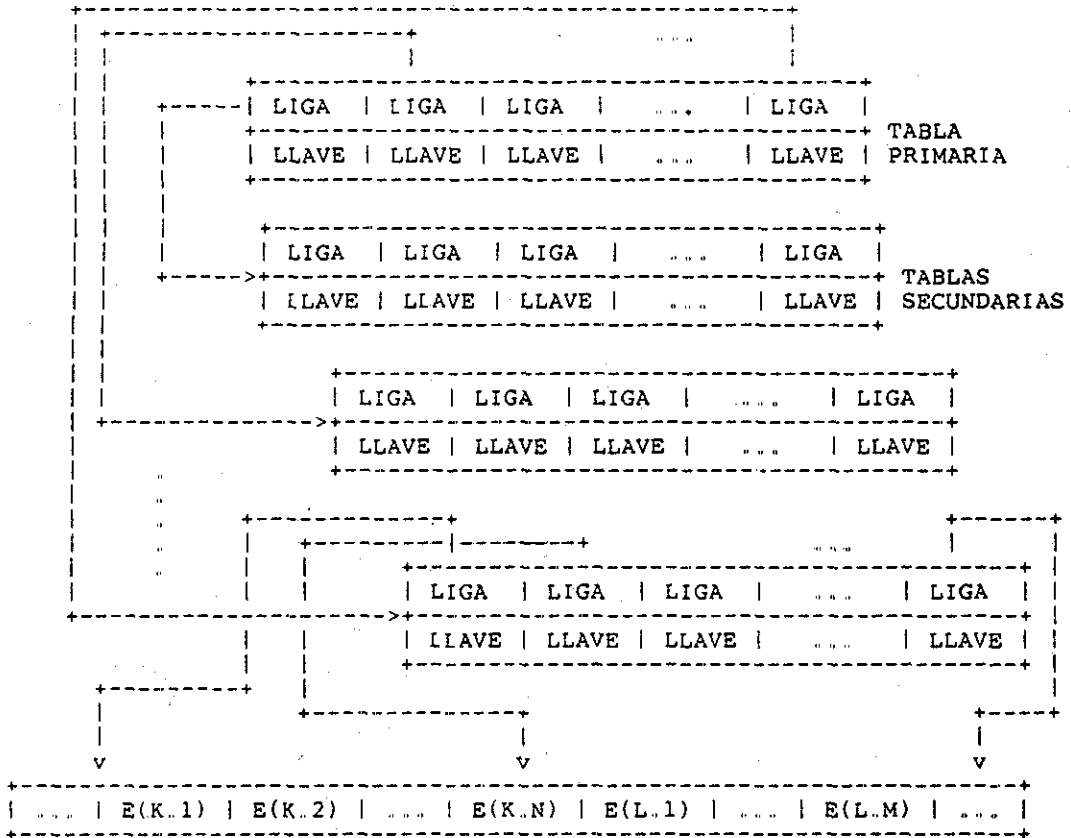


Figura 3-1. Relación entre tablas y éstas a los arreglos de elementos.

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

La componente ARREGLOS, almacena en forma empaçada los arreglos de elementos de los conjuntos. Los elementos están ordenados por los valores de algún otro atributo de la instancia de la Entidad.

El diagrama de la figura 3-1, muestra en forma esquemática las relaciones entre tablas y de estas a los arreglos de elementos:

La tabla primaria selecciona los valores que aparecen al inicio de cada tabla secundaria, mientras que cada tabla secundaria establece el inicio del arreglo de elementos. Los valores de las llaves se encuentran ordenados en forma ascendente. Las tablas en realidad tienen capacidad para 128 elementos pero para minimizar los accesos a disco el primer valor de cada tabla es igual al final de la anterior, esto con el propósito de calcular el tamaño, o la cardinalidad de cada conjunto antes de ser leído del disco.

Existen dos posibilidades en la operación de los índices:

1. Existen menos de 127 clase de atributo. En cuyo caso solo existe la tabla primaria y esta apunta directamente a los arreglos de elementos. Esto minimiza el acceso a disco puesto que en general no se requerirá ninguna lectura durante la operación de la base de datos.
2. Existen más de 127 clase de atributos. En cuyo caso existen tablas secundarias. En una búsqueda es necesario efectuar una búsqueda binaria sobre la tabla primaria posicionándose en el elemento mayor que a la vez sea menor o igual al buscado. Esto indica el índice de tabla secundaria que deberá leerse de disco y en la que deberá hacerse una segunda búsqueda binaria.

### 3.2.3 IMPLEMENTACION DE UNA BASE DE DATOS

El esquema general de la Base de Datos se encuentra almacenado en un archivo de Descriptores de Entidades llamado Archivo de Acceso Secundario.

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

Al abrir una Base de Datos, una imagen de este archivo se carga en la estructura llamada Tabla Descriptora de Entidades (TDEN). Esta tabla mantiene la información acerca de las entidades de la Base de Datos, y controla el acceso a ellas.

### 3.3 IMPLEMENTACION DEL ACCESO A INSTANCIAS DE UNA ENTIDAD

Asociado a una entidad (referenciado en la tabla TDEN) se tiene un buffer (registro asociado) en donde se almacena la información de la instancia que se desea ver. Dicha información se obtiene (usando la tabla hash) del Depósito del archivo de entidades correspondiente. Paralelamente se guarda una referencia (también en TDEN) con la identificación interna (posición en el Depósito) de la instancia que se tiene almacenada en el buffer (instancia actual).

Existen dos tipos de accesos a instancias de una entidad, el primero de ellos, se realiza a través de la tabla hash, en este caso, el usuario conoce la llave de la instancia que se desea acceder, por lo que mediante la operación LEE ENT actualiza en TDEN la instancia actual, y carga en el buffer la información.

El segundo tipo de acceso, se realiza a través de los elementos de un conjunto de instancias obtenido consultando a los índices. Para este caso, asociado a una entidad se tiene un conjunto (el cual está referenciado en TDEN). La operación NEXT permite acceder secuencialmente cada una de las instancias de dicho conjunto. Los conjuntos de instancias se implementaron en una estructura llamada SET DESEC la cual guarda una referencia a la entidad a la que pertenecen las instancias del conjunto. No se permiten operaciones entre conjuntos de instancias de entidades diferentes.

### 3.4 IMPLEMENTACION DE LAS REGLAS DE INTEGRIDAD

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

### 1. RI1. Verificación Del Tipo De Los Atributos.

Queda como responsabilidad del usuario la verificación del tipo de cada uno de los atributos en las instancias que sean dadas de alta o modificadas. No se efectúa, por el momento, ninguna verificación interna.

### 2. RI2. Completez y Actualidad De Los Indices.

La actualización de índices no se efectúa en línea en una Base de Datos que al estar abierta en el modo apropiado sufra alguna modificación en un atributo que esté indexado. Es necesario crear el índice fuera de línea una vez terminada la actualización. En la operación del sistema un control de la validez de los índices es llevado; toda vez que una instancia sufra modificaciones en atributos indexados, se anulará el índice correspondiente por medio del seguimiento de versiones, tanto del atributo como del archivo índice correspondiente.

Una Base de Datos abierta en modo LECTURA/ESCRITURA tiene bloqueado el acceso a índices, por lo que todos los servicios relacionados a éstos regresarán con un código de error. Esto se debe a que, puesto que los índices no son actualizados en línea y pueden ser modificados no hace sentido que sean utilizados mientras los atributos sufren modificaciones.

### 3. RI3. Unicidad De Llave.

La inserción de registros a través de la tabla hash garantiza esta regla de integridad.

### 4. RI4. Integridad De Las Relaciones.

Es responsabilidad del usuario el velar por el cumplimiento de esta regla de integridad.

## 3.5 EL MEDIO AMBIENTE DE APLICACION

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

En el medio ambiente de aplicación generalmente se tendrá más de una Base de Datos trabajando simultáneamente, y más de un usuario trabajando sobre alguna de ellas.

Para controlar al conjunto de bases de datos definidas en el sistema, y su acceso concurrente, se creó una estructura llamada Tabla Descriptora de Bases de Datos (TDBD).

La TDBD mantiene información acerca de la localización de los archivos asociados con una Base de Datos, y el número de usuarios en cada modo de acceso (LECTURA, LECTURA/ESCRITURA).

### 3.5.1 SINCRONIZACION DE ACCESO

Durante la operación normal del sistema de la Base de Datos será común encontrar más de un usuario accedendo el mismo o diferentes objetos del sistema, sean estos bases de datos, entidades, atributos, etc.; por esto es de vital importancia que exista algún mecanismo construido y vigilado por las rutinas de acceso de sincronización en un acceso concurrente.

Es fácil ver que resultados impredecibles se obtendrían de permitirse que mientras un usuario lee de la Base de Datos otro escriba en la misma; aunque es posible formular un mecanismo que garantice una correcta operación en el caso general de lectura y modificación a nivel instancia de entidad, estos por lo general son muy costosos. Es por esto que un mecanismo más sencillo de implementar se ha formulado, el cual consiste de:

- a) El modo de acceso será fijado en la apertura de la Base de Datos. Los posibles modos de apertura son: LECTURA y LECTURA/ESCRITURA.
- b) Se permitirá un número ilimitado de lectores siempre que no existan escritores.
- c) Un escritor bloqueará el acceso a cualquiera deseando abrir una Base de Datos.

En resumen no se permite que exista un usuario leyendo y otro escribiendo a nivel de toda una Base de Datos. Aunque esto parezca algo drástico, no es así para la mayoría de las

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

aplicaciones a las que se pretende destinar la Base de Datos. Esto sin embargo tiene un impacto sobre otros conceptos de la Base de Datos, como se verá más adelante.

### 3.5.2 MANEJO DE LA MEMORIA COMPARTIDA

En sistemas de uso general que a su vez son accedidos por un número no determinado de usuarios, resulta común observar una sub-utilización de la memoria. Esto se debe a que en general cada uno asigna espacio para el peor caso que por lo regular no es el más frecuente.

El sistema de la Base de Datos fué diseñado de forma que solo aquellas estructuras necesarias para la operación sean asignadas en memoria, así mismo con el objeto de optimizar las necesidades de memoria del sistema como un todo se decidió disponer de un depósito compartido de memoria.

En resumen la memoria compartida actúa como un depósito de memoria a nivel del sistema global, donde se adquieren y liberan bloques de memoria con el objeto de almacenar estructuras dinámicas del sistema.

La gran mayoría de las estructuras del sistema son de creación dinámica, por lo que estas existen únicamente mientras sean utilizadas por el usuario.

El control del espacio adquirido por una tarea es establecido utilizando la estructura de control Maestra. Su objetivo es poder liberar espacio no utilizado. Se considera necesario efectuar el siguiente control:

- a) Todo el espacio adquirido a nivel Base de Datos. Esto es, será posible establecer los segmentos de memoria adquiridos para una instancia de una Base de Datos específica.
- b) El espacio adquirido por una Tarea. Esto es, será posible establecer los segmentos de memoria adquiridos por una tarea específica; con el objeto de liberarlos si ésta terminase de ejecutar sin antes liberarlos.

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

- c) Verificar la validez de una dirección, proporcionada por el usuario, a un segmento adquirido.

Debido a los criterios de diseño antes mencionados es necesario que el manejador de la memoria compartida sea notificado toda vez que una base de datos es abierta y cerrada, pues toda la memoria adquirida para esta debe ser registrada, y liberada en forma automática al cerrar la base de datos. Durante la operación de apertura de una Base de Datos se verificará que ésta no haya sido abierta anteriormente, en cuyo caso se generará un error.

Durante las operaciones de adquisición de memoria es necesario indicar la Base de Datos que será dueña del segmento.

### 3.6 ESTRUCTURA DE ARCHIVOS

Los archivos de la base de datos pueden ser de 2 tipos: Archivos físicos de entidades y archivos físicos índice. Los archivos de entidades mantienen todos los datos relacionados con la información de las instancias de entidades así como información de control para su acceso e integridad. Los archivos índices contienen información similar relacionada con el acceso a la base de datos a través de índices. Por archivo físico se entiende un archivo en disco el cual es accesible a través del sistema operativo. En un momento dado, se subdividen éstos en secciones o archivos lógicos. Cada archivo físico tiene un archivo lógico de control con el cual se implementan o controlan los archivos lógicos de información. Cada archivo lógico se encuentra alineado por bloque físico ('Block boundary').

Los módulos usuarios del módulo de I/O accesan los archivos a nivel de registro lógico, mientras que la unidad mínima de acceso a disco del módulo de I/O es un bloque. El módulo de I/O solicita al sistema operativo la transferencia de un bloque siempre que el registro lógico deseado no se encuentre previamente en memoria (Buffer físico).



## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

### 3.6.1 ARCHIVO DE ENTIDADES

Este archivo físico tiene 5 componentes lógicas; 4 de ellas de información mas una de control.

#### 3.6.1.1 ESTRUCTURA DEL ARCHIVO DE ENTIDADES -

Las componentes del archivo de entidades son:

##### 1. Instancias de la entidad.

El archivo de instancias de la entidad está formado por una secuencia de registros lógicos que contienen la información de las instancias de una entidad. La longitud de cada registro lógico debe ser la misma.

La posición física de cada registro lógico se denomina "nombre interno" y está asignada por el módulo de I/O.

El módulo de I/O agrega a cada registro lógico 2 bytes con los cuales se forma listas ligadas de registros con igual código hash (Véase Archivo 'Hash').

La administración del espacio libre en esta componente se lleva por medio de otro archivo lógico llamado "Bit Map".

##### 2. Bit map.

El archivo lógico "Bit map" representa la ocupación de registros lógicos de instancias de la entidad. Este archivo lógico está formado por 1 registro lógico de longitud fija igual a 2 bloques físicos del disco con lo cual se tiene una capacidad máxima de 12288 instancias. Cada bit del registro indica si el registro lógico del archivo de instancias de la entidad correspondiente a la posición del bit, es válido o no.

##### 3. Archivo "hash".

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

El objetivo del archivo lógico "hash" es tener una independencia del posicionamiento físico de las instancias de la entidad. Está constituido por una secuencia de registros lógicos de longitud fija igual al tamaño de llave + 2 bytes. Lógicamente, cada registro contiene el nombre de la llave de un registro en el archivo de instancias de la entidad y su nombre interno.

El posicionamiento de estos registros lógicos se calcula por medio de un algoritmo de "Hashing" que depende del nombre de la llave y del tamaño del archivo de instancias de la entidad. Se resuelve el problema de colisión automáticamente por medio de tener todas las instancias colisionadas en el mismo bloque.

### 4. Esquema.

Este archivo está formado por una secuencia contigua de registros lógicos de igual longitud que contienen la definición de los atributos de la entidad.

### 5. Archivo de control.

Este archivo sirve para acceder los archivos lógicos de información. Está formado por un solo registro lógico.

## 3.6.2 ARCHIVO INDICE

Este archivo físico tiene 3 componentes lógicas; 2 de ellas de información mas una de control.

### 3.6.2.1 ESTRUCTURA DEL ARCHIVO INDICE -

## PANORAMA GENERAL DE LA BASE DE DATOS IMPLEMENTADA

Las componentes del archivo índice son:

### 1. Tabla Primaria.

La tabla primaria consiste de una secuencia de llaves de acceso, que corresponden a los valores del atributo indexado, existiendo un valor diferente para cada una de las tablas en la componente de las tablas secundarias.

### 2. Tablas Secundarias.

La componente de las tablas secundarias consiste de tablas que contienen un número fijo de parejas ordenadas (valor llave, índice conjunto). Los valores de la llave de acceso corresponden a las diferentes clases de instancias bajo el atributo indexado, mientras que los índices de conjuntos corresponden al inicio del conjunto de instancias que contienen el valor llave en el atributo que se indexa.

### 3. Conjuntos.

La componente de conjuntos consiste de una cuerda del tamaño del universo de la entidad, con los nombres internos (número físico de registro de cada instancia) de las instancias de la entidad. El orden de los elementos del conjunto para una clase están ordenados por un atributo seleccionado durante la creación del índice.

### 4. Archivo de Control.

Este archivo sirve para acceder los archivos lógicos de información. Está formado por un solo registro lógico.

## CAPITULO 4

### DISEÑO E IMPLEMENTACION DEL EDITOR DE BASES DE DATOS

Antes de atacar cualquier problema, se tiene que analizarlo muy bien, o sea la especificación del problema, en qué consiste, cuáles son los objetivos para resolver el problema y cuáles son los alcances de la solución del problema. Qué componentes o herramientas se necesita para resolverlo, y cuáles son los disponibles. Después viene el paso de estudiar cada una de las componentes, y finalmente es el paso de resolver el problema, utilizando las componentes existentes. En el caso del diseño e implementación del EDITOR de bases de datos, el procedimiento que se siguió fué :

1. analizar y especificar el problema, o sea, definir los objetivos y los alcances del sistema;
2. estudiar la implementación de la base de datos, o sea cómo está implementada, y cuáles son sus características;
3. estudiar los componentes o herramientas que existen analizando cuáles podrían ser necesarios para el sistema;
4. diseño e implementación del sistema, utilizando los componentes existentes.

En este capítulo, se describe con más detalle cada uno de estos pasos.

## **DISEÑO E IMPLEMENTACION DEL EDITOR DE BASES DE DATOS**

### **4.1 ESPECIFICACION DEL PROBLEMA**

Como se vió en el capítulo 3, las bases de datos del sistema para el cual está implementado el EDITOR DE BASES DE DATOS son de tipo relacional. Este sistema consiste de varias bases de datos, cada una contiene varias entidades, y a su vez cada entidad de un conjunto de atributos. Cada atributo puede ser de tipo entero (un byte, dos bytes, o cuatro bytes) o de tipo real, lógico, cuerda de caracteres, estructurado, etc. Cada atributo puede ser indexado o no. Se puede acceder la base de datos en modo de acceso directo mediante la tabla de hash por la llave ( el primer atributo en la definición del esquema de la entidad ), o por el valor de un atributo indexado, obteniendo el conjunto de instancias que tienen el mismo valor en este atributo. Una característica importante es en este sistema de bases de datos no se relaciona una entidad con otra explícitamente, o sea cada entidad es un objeto aislado.

El objetivo de tener el EDITOR DE BASES DE DATOS es para contar con un método eficiente de consulta y actualización de las bases de datos. El EDITOR debe contener dos partes: la parte de control : esta parte permite al usuario llevar un control sobre el sistema de bases de datos : consultar o modificar las estructuras de las bases de datos, así como consultar o modificar las estructuras de las entidades. La otra parte del sistema permite a los usuarios consultar, modificar, borrar, e insertar una instancia o un conjunto de instancias de entidades, también permite al usuario sacar reportes de las instancias ordenadas (o sin orden) por el valor de algún atributo en orden ascendente o descendente en el formato que define el usuario, se tiene que poder sacar los reportes tanto en la terminal de video como en la impresora.

### **4.2 ESTRUCTURA MODULAR DEL MANEJADOR DE BASES DE DATOS**

El Manejador de la Base de Datos se encuentra organizado en módulos. Cada módulo corresponde a una abstracción de una estructura de datos, en donde los procedimientos que lo forman implementan las funciones sobre dicha estructura. Un módulo

## DISEÑO E IMPLEMENTACION DEL EDITOR DE BASES DE DATOS

puede estar formado por rutinas a nivel usuario y por rutinas internas o solo por estas últimas.

### 4.2.1 DESCRIPCION GENERAL DE MODULOS

Los módulos que forman el Manejador son los siguientes:

- Manejo del Esquema
- Manejo de Conjuntos
- Manejo de Memoria Compartida
- Manejo de Indices
- Manejo de Entrada/Salida

Las funciones de cada uno de ellos son las siguientes:

#### 1. Manejo del Esquema

Contiene todas las funciones que tienen relación con el esquema de una Base de Datos y con el acceso a la información. Forma el conjunto de rutinas generales de apertura y cierre de una base de datos, de entidades y atributos, así como las operaciones de lectura y escritura por llave y las operaciones de lectura y modificación de atributos de las instancias de entidades. Este módulo a su vez, utiliza los siguientes módulos:

- Manejo de Conjuntos
- Manejo de Memoria Compartida
- Manejo de Indices

## DISEÑO E IMPLEMENTACION DEL EDITOR DE BASES DE DATOS

Las rutinas que lo forman son:

### a) Nivel usuario

ABRE\_BD : Abre una base de datos  
CIER\_BD : Cierra una base de datos  
ABRE\_ENT : Abre una entidad  
CIER\_ENT : Cierra una entidad  
CREA\_ENT : Crea la entidad dada bajo la base de  
datos especificada.  
ABRE\_ATR : Abre un atributo  
LEE\_ENT : Lee un registro dada una llave  
ESC\_ENT : Escribe el registro actual en el  
archivo de entidades  
BORRAENT : Elimina la entidad cuya llave se  
encuentre en el registro actual  
LEE\_ATR : Lee un campo del registro actual  
(en memoria)  
ESC\_ATR : Modifica un campo del registro  
actual  
NEXT : Obtiene siguiente elemento del  
conjunto actual  
ASO\_CON : Hace actual un conjunto  
DESA\_CON : Quita el conjunto de su condición  
de actual

### b) Nivel interno

CREA\_TDE : Crea en memoria la tabla IDEN

## DISEÑO E IMPLEMENTACION DEL EDITOR DE BASES DE DATOS

CREA\_TDA : Crea en memoria la tabla TDAT  
BUS\_TDEN : Busca un nombre en la tabla TDEN  
BUS\_TDAT : Busca un nombre en la tabla TDAT  
CARGARCH : Carga una tabla a partir de un archivo

### 2. Manejo de Conjuntos

Contiene las funciones de operaciones sobre conjuntos que existen en memoria. Los conjuntos son generados inicialmente por el módulo de índices y únicamente existen cuando el modo de acceso a la base de datos es de lectura (Read Only).

Utiliza los siguientes módulos:

Manejo de Memoria Compartida

Las rutinas que lo forman son:

#### a) Nivel usuario

CREA\_CNJ : Crea un conjunto de elementos de una entidad específica.  
LIBE\_CNJ : Libera un conjunto, regresando la memoria asignada al POOL.  
SIG\_ELEM : Obtiene el siguiente elemento del conjunto.  
ES\_ELEMN : Establece si la instancia actual es miembro del conjunto.  
CARD\_CNJ : Regresa la cardinalidad del conjunto.



## DISEÑO E IMPLEMENTACION DEL EDITOR DE BASES DE DATOS

COMP\_CNJ : Obtiene el complemento del conjunto..

UNION : Obtiene la union de dos conjuntos..

INTERSEC : Obtiene la interseccion de dos conjuntos..

DIFERENC : Obtiene la diferencia de dos conjuntos..

UNIVERSO : obtiene el conjunto que contiene todas las instancias de una entidad..

### b) Nivel interno

IN\_SORT : Indica si un dado elemento es o no miembro del conjunto y ordena el conjunto con el fin de optimizar futuras búsquedas..

## 3. Manejo de Memoria Compartida

Este módulo efectua la administración del espacio en memoria compartida que es solicitado dinámicamente por los demás módulos del Manejador. Todo el espacio solicitado por una tarea para una base de datos dada se mantiene ligado por este módulo. No utiliza a ningún otro módulo..

Las rutinas que lo forman son :

### a) Nivel interno

ADQMEM : Obtiene espacio del área de memoria compartida

LIBMEM : Libera espacio adquirido