



885248  
UNIVERSIDAD AMERICANA DE ACAPULCO

EXCELENCIA PARA EL DESARROLLO

---

FACULTA DE CONTADURÍA ADMINISTRACIÓN E  
INFORMATICA

INCORPORADA A LA UNIVERSIDAD  
NACIONAL AUTONOMA DE MÉXICO

“LA ENSEÑANZA DE LAS BASES DE DATOS EN  
LA LICENCIATURA DE INFORMATICA EN LA  
UNIVERSIDAD AMERICANA DE ACAPULCO”.

## TESIS

QUE PARA OBTENER EL TITULO DE:  
LICENCIADA EN INFORMATICA

PRESENTA:  
CRISTINA ORDÓÑEZ BETANZOS

DIRECTOR DE TESIS:  
ING. GONZALO TRINIDAD GARRIDO



ACAPULCO, GRO.

ENERO 2001.



Universidad Nacional  
Autónoma de México

Dirección General de Bibliotecas de la UNAM

**Biblioteca Central**



**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.

## *AGRADECIMIENTOS:*

Es importante el tener una preparación profesional para forjar un mejor futuro para nuestros hijos y por ello le agradezco, a mi Universidad por el apoyo indefinido que me brindo durante mi preparación, a mi facultad por estar presente en mis inquietudes como profesionista y a los directivos y personal que la conforman, mis más sinceros agradecimientos y como olvidar, a esas personas encargadas de impartir el conocimiento que con toda paciencia siempre me contestaron, a mis preguntas muchas gracias profesores, por eso y su amistad mil gracias.

# DEDICATORIA

## **A DIOS:**

Eternamente, gracias Dios por todo lo que me has dado, por haberme permitido dar el último paso en mi vida profesional y por permitirme tener sanos a mis seres queridos, pero sobre todo porque tú eres el mejor amigo que siempre he tenido, y gracias por permitirme obtener mi título como Licenciada en Informática y ayudar a mi prójimo, gracias hoy y siempre.

## **A MIS PADRES:**

### **A TI MADRE:**

Por ser una mujer y amiga, y saber respetar mis decisiones, gracias, por que siempre me has apoyado y sobre todo por que en este día tu niña, vuelve sus sueños en realidad y se convierte en una profesionista, y por que me has sabido guiar en la vida a ti madre mi cariño, admiración y respeto por siempre.

### **A TI PADRE:**

Gracias, por tus consejos, y te dedico mi tesis con todo mi cariño, admiración y respeto, ya que gracias a sus sacrificios tanto de mi mamá como tuyos, es que estudie una carrera, gracias hoy y siempre.

### **A MIS TIAS:**

Gracias, tías por ser siempre como hasta hoy, les agradezco su eterno cariño, apoyo en todas mi decisiones, y sobre todo en mi carrera, universitaria, les prometo, nunca defraudarlas, las quiero mucho.

### **A MI HERMANO:**

Como siempre, hermanito acordándome de ti y agradeciéndote a ti por tu apoyo y cariño, comprensión que siempre me has demostrado, un eterno cariño, amor, respeto y admiración.

### **A MIS PROFESORES:**

Principalmente a mi asesor de tesis, por su paciencia y por saberme guiar en la realización de mi tesis, ya que gracias, a su apoyo se llevo a cabo la culminación de mi tesis, mi admiración y respeto para usted Ingeniero Gonzalo Trinidad Garrido.

### **A MI FAMILIA:**

Gracias, por su apoyo, en lo que ha mi tesis, corresponde, cuentan con mi cariño y respeto por siempre.

### **A MIS AMIGAS Y AMIGOS:**

Por su apoyo y comprensión, y sobre todo por su amistad y confianza de la cual siempre he gozado, y espero siempre contar con ella, y sobre todo por todos los momentos que hemos pasado, buenos y malos, por esos hoy y siempre tienen mi cariño y respeto, sobre todo Mayra, Audina, Roció, Lupita, Lidia, y todas las que siempre han estado cuando más triste o contenta estoy, y claro, también a el primo de una persona especial mi cariño y respeto por siempre, pasa lo que pase.

### **A ESA PERSONA ESPECIAL:**

Aunque sé que no hay ningún hombre ideal, pero a ti mi príncipe azul, a ti mi corazón, te dedico mi tesis, por el tiempo que debimos estar juntos, por las lagrimas, por todo el amor, por todo eso, que hace tan especial, a un hombre, hoy y siempre serás especial y nadie ocupará tu lugar hasta que muera, por siempre te amaré.

# INDICE

NUMERO	CONCEPTO	PAGINA
	INTRODUCCION.....	1
	OBJETIVOS.....	4
	HIPOTESIS.....	5
	<b>CAPITULO I</b>	
1.	PLATAFORMA TEORICO CONCEPTUAL.....	6-7
1.1	Niveles de estudio de la información.....	8
1.1.1	Definición de Información .....	8-9
1.1.2	Administración como recurso.....	9
1.1.3	Administración de la información por computadora.....	11
1.1.4	Usuarios: Trabajadores de la información.....	12
1.1.-	DATOS.....	13
	A) 1.1.1.- FUENTES DE DATOS.....	13
	1.1.1.1 Valor de los datos .....	13
	1.1.1.2 Medios de los archivos de datos.....	14
	1.1.1.3 Cualidades de la información .....	14-17
	1.1.1.4 Tipos de información .....	17
	1.1.1.5 Información de apoyo .....	19
	1.1.1.6 Información de situación .....	20
	1.1.1.7 Información de advertencia .....	20
	1.1.1.8 Información de planeación .....	20
	1.1.1.9 Información de operación internas .....	20
	1.1.1.10 Información confidencial .....	20
	1.1.1.11 Teoría de la Información.....	20
	B)1.1.2.- MODELADO DE DATOS, RELACIONES, ENTIDADES Y ATRIBUTOS.....	21-22
	1.1.2.1 El modelo de datos relacional y el álgebra relacional..	22-23
	1.1.2.2 Conceptos del Modelo Relacional .....	24
	1.1.2.3 Dominio, tuplas, atributos y relaciones .....	25
	1.1.2.4 Características .....	25
	1.1.2.5 Definición de relaciones .....	25
	1.1.2.6 Aspectos estáticos .....	26
	1.1.2.7 Definición .....	26-27
	1.1.2.8 Modelado Relacional .....	28
	1.1.2.9 Relación de uno a uno .....	28
	1.1.2.10 Relación de uno a muchos .....	29
	1.1.2.11 Relación de muchos a muchos .....	30

2.- ORGANIZACIÓN DE ARCHIVOS.....	32
2.1.1 Archivos secuenciales.....	32
2.1.2 Archivos relativos .....	32
3.- BASE DE DATOS	
1.3 Definición de un sistema de bases de datos .....	37
A)CONCEPTO	37
1.1.1 Bases de datos .....	37
1.1.2 Concepto .....	37
B)DICCIONARIO DE DATOS.....	38
1.1.1.1 Diccionario de datos .....	38
C)SISTEMAS MANEJADORES DE BASES DE DATOS( DBMS).....	39
1.1.1.1Sistema manejadores de bases de datos.....	39
a) Tipos.....	39
b) Lenguaje de definición y manipulación de datos.....	40-44
D)INTRODUCCION AL DISEÑO DE BASE DE DATOS	
a) Modelo de Red .....	45
b) Modelo Jerárquico .....	45
c) Modelo Relacional .....	46
E) ADMINISTRACION DE BASE DE DATOS.....	47-51

## CAPITULO II

2.1 DISEÑO DE LAS BASES DE DATOS .....	52
2.1 Introducción .....	52
2.2 Evolución de la gestión de datos .....	53-57
2.2.1 Avances en las bases de datos .....	58
2.1.1 Modelo conceptual .....	59
A) Análisis de datos .....	60-61
B) Representación gráfica .....	62-63
2.1.2 Modelo lógico .....	64
A) Mapeo del diseño .....	64
2.1.3 Modelo físico .....	65
A) Interfases entre usuario y Base de Datos .....	65-66
B) Modelo Interno.....	66
C) Modelo Externo.....	67-70
D)Sistema de manejo de base de datos (red, jerárquico,relacional).....	70-76

### CAPITULO III

3.ASPECTOS DE DESEMPEÑO .....	77
3.1 Consultas: SQL .....	77
3.1.1 Definición de datos en SQL .....	78
3.1.2 SQL Un lenguaje de bases de datos relacionales ...	78
3.1.2.1 Conceptos de esquemas y catálogo en SQL2 .....	78
3.1.3 SQL incorporado .....	79
3.1.4 SQL .....	80-91
3.2 Implantación de las bases de datos.....	92
3.3 Operaciones de mantenimiento .....	92
3.3.1 Operaciones de entrada y salida (E/S).....	93
A) Respaldo.....	94
3.3.2 Transferencia de datos entre la memoria principal y la unidad de almacenamiento secundario.....	94
3.3.3 Procesador de entrada y salida y aditamento de unidad de control .....	95
B) Recuperación.....	96
C) Reorganización y reestructuración.....	97-99
D) Monitoreo y ajustes del desempeño.....	100-101
E) Seguridad y Privacidad .....	102-105

### CAPITULO IV

4. TOPICOS AVANZADOS DE LAS BASES DE DATOS .....	106
4.1 Bases de datos distribuidas .....	107-108
4.1.2 Construcción de una bases de datos distribuida.....	109
4.1.3 Orientación en la investigación sobre bases de datos distribuidos.....	110
4.1.4 Relaciones Jerárquicas .....	110-111
4.1.5 Relaciones Interclases .....	112-114
4.2 Bases de datos orientadas a objetos .....	114
4.2.1 Representación de Objetos .....	115
4.2.2 Relaciones entre objetos.....	115
A) Conceptos de bases de datos orientado a objetos .....	115-116
4.3.1 Definición de objeto.....	116
4.3.2 Definición de clases .....	117
B) Paradigma de bases de datos orientada a objetos .....	118
4.4.1 Encapsulamiento.....	118
4.4.2 Polimorfismo .....	118
4.4.3 Herencia .....	119
C) Lenguajes de bases de datos orientada a objetos .....	120-124
B) Aplicaciones .....	124-129

## CAPITULO V

5. CONCLUSION Y RECOMENDACIONES.....	130-131
GLOSARIO.....	132-134
BIBLIOGRAFÍA .....	135-137
ANEXOS: .....	137-139

## INDICE DE FIGURAS Y TABLAS

No. FIGURA	DESCRIPCIÓN	PAGINA
1	Sistema de Información.....	7
2	Los papeles del analista y el ciclo de desarrollo de sistemas.....	11
3	Servicios y manejo de Información.....	12
4	Empresas y fuentes de datos.....	14
5	Tipos de Información.....	17
6	Como fluye la información dentro de las organizaciones.....	19
7	Tabla de Clientes.....	23
8	Tabla de Ordenamiento.....	24
9	Orden de las tuplas en una relación.....	26
10	Relación uno a uno.....	29
11	Relación uno a muchos.....	30
12	Relación de muchos a muchos.....	31
13	Jerarquía Lógica D1/1 para el esquema externo.....	41
14	Posibles arquitecturas de los SI apoyados en diccionarios y directorios.....	42
15	SDRI y SGBD independientes.....	43
16	Estructura de Red.....	44
17	Estructura Jerárquica.....	45
18	Ejemplo de modelo de red, jerárquico y relacional.....	49-50
19	La arquitectura y los modelos de datos, elementos fuertemente interrelacionados entre sí.....	53
20	Tres puntos de vista del almacenamiento físico.....	60
21	Representación gráfica.....	62
22	Relaciones entre ESTUDIANTE, CURSO Y GRADO.....	62
23	Esquema Interno.....	66
24	Esquema Externo que se puede desarrollar a partir del esquema Interno.....	67
25	Una base de datos relacional.....	71
26	Manipulación de árboles.....	73
27	Una base de datos de red.....	74
28	Etapas de independencia de los datos.....	74
29	Ejemplo de esquema interno y externo.....	75
30	Consulta para mostrar las instrucciones básicas.....	76
31	Tabla para mostrar como se crea un consulta.....	80
32	Tabla para mostrar como se crea una consulta.....	81
33	Tabla para mostrar como se agrega una tabla.....	82
34	Tabla descriptiva.....	85
35	Hoja de datos de la tabla clientes.....	86
36	Tabla de empleados mostrando la longitud de cada campo.....	88

37	Tabla de consulta con Order by.....	91
38	Tabla de consulta con Order by.....	92
39	Ejemplo de una sencilla tabla de almacenamiento.....	100
40	Ejemplo de sql .....	104
41	Sistema Distribuido de Bases de Datos.....	108
42	Tablas de almacenamiento.....	109
43	Relaciones jerárquicas.....	111
44	Representación de relaciones 1:M.....	112
45	Representación de relaciones M:N.....	113
46	Modelo OO vs. Modelo E-R.....	115
47	Estructura de un objeto.....	117
48	Representación de entidades reales con objetos.....	117
49	Jerarquía de clases: un ejemplo de herencia.....	120

## INTRODUCCION

En la actualidad han habido grandes cambios tanto tecnológicos como científicos, esto ha tenido gran repercusión en las empresas debido ha que anteriormente las operaciones tanto contables, administrativas y otras se realizaban de forma manual, todo esto ha quedado atrás, es decir obsoleto, por lo cual ha sido necesario estar a la vanguardia ya que la empresa que no cambia su sistema arriesga su permanencia en el mercado, es decir que no cuenta con toda la información al día ya que es necesario manejar la información de forma más eficaz.

Por lo cuál todas las empresas se encontraron en un grave problema, ya que el volumen de información iba cada vez en mayor aumento y con esto surge la necesidad de contar con un software que facilite el manejo y control de la misma, es con esto que surge como respuesta ante esta problemática las bases de datos, con las 12 reglas codd.

Pues, bien las bases de datos son un conjunto de tabla relacionadas entre sí, que cuenta con información relevante de una empresa y en la que se puede realizar diversas operaciones como son: consultas, eliminaciones, aumento de campos, etc.

En Acapulco al igual que en cualquier otro estado de la república tienen gran auge las bases de datos, como por ejemplo en una empresa automotriz en la cual tardaría normalmente mas de 15 días y hasta un mes en aceptar o rechazar un crédito a un cliente, con las bases de datos, se envía las solicitudes vía internet y en 2 o 3 días máximos hay respuesta.

En mi trabajo hablo en cinco capítulos en los cuales se discute la importancia de un sistema, así como lo importante que es la información. Dentro de una empresa es y el manejo y control de toda la información, así como lo importante que son los componentes básicos con que cuenta un sistema, para poder ser reconocido como tal, por lo que hay que cuidar que sus insumos sean ciertos y poder obtener un buen resultado.

Gracias a la información se pueden tomar muchas decisiones que pueden ser decisivas para una empresa y por lo cual es necesario, tomar una decisión oportuna y en el momento preciso. Esto es, que hay que tomar la importancia de la información, así como de sus cualidades para que los niveles gerenciales puedan tomar una mejor decisión y esto se logra con una herramienta tan poderosa como son las bases de datos y los resultados esperados serán aún mejores que los previstos.

Pues bien en el primer capítulo, conceptualizo términos básicos como lo que es, la información la utilidad que se puede dar a la misma, así como a las personas que les interesa que la información sea de forma eficaz, eficiente y oportuna para que la misma tenga mayor valor.

Una base de datos tiene diversos, conceptos básicos como son tuplas, dominios, atributos, relaciones, así como también mencionamos lo que es un modelo relacional y la diferencia principal que existe entre la información y un dato, así como otros elementos básicos de las bases de datos como lo son los diccionarios de datos.

Una base de datos, debe de ser manejada por un encargado de la base de datos, que no es más que el manejador de bases de datos.

Como no podemos avanzar sin antes mencionar el cómo surgen las bases de datos, en el segundo capítulo, menciono como evolucionan las bases de datos y sus doce reglas establecidas por Codd, así como los avances que ha tenido una base de datos, hasta llegar ha ser lo que hoy conocemos como bases de datos.

En las bases de datos, se utiliza el modelo relacional, que se basa de varios años de investigación, y de donde surge el modelo de bases de datos orientado a objetos, por supuesto, menciono los niveles de la arquitectura de la base de datos como son: nivel interno, nivel conceptual, nivel externo.

Luego entonces es importante mencionar que las bases de datos manejan ciertas operaciones, de las cuales se habla en el tercer capítulo, en el cual se definen conceptos básicos del SQL, también plantea como recuperar la información y la seguridad de la misma con el fin de no ser mal utilizada.

En el cuarto capítulo de mi trabajo menciono como perduraran las bases de datos orientadas a objetos, y que pasos hay que seguir para construir una buena base de datos distribuida por supuesto que las bases de datos orientadas a objetos cuentan con propiedades como son: Encapsulamiento, Poliformismo y Herencia.

Por tal motivo es relevante mencionar que el objetivo de mi trabajo es sugerir un modelo de enseñanza de las bases de datos en la Universidad Americana de Acapulco, ya que en las primeras materias que son impartidas se llega a representar ciertas dificultades en el aprendizaje, por lo difícil que es comprender ciertos términos como son dominios, tuplas, tablas, SQL, ya que la gran mayoría de los estudiantes vienen sin algún conocimiento de la misma.

Por lo cual al haber consultado varios libros, personas, revistas, encontramos que es mucho más fácil, el entender como funcionan las bases de datos y la importancia que tienen en la actualidad, ya que en las empresas la mayoría maneja lo que es una base de datos por el volumen de la información.

Y sobre todo que estos conocimientos, se puedan aplicar en diversas bases de datos como son: Progress, Access, Oracle, dBase.

## OBJETIVOS

- Proponer una base para la enseñanza para las bases de datos en la Universidad Americana de Acapulco.
- Unificar conceptos de difícil comprensión para el alumno de la carrera de la Licenciatura de Informática.
- Demostrar la importancia que tienen las bases de datos hoy en la actualidad.
- Facilitar el manejo de los estudiantes de informática en las bases de datos.
- Controlar y eficientizar el manejo de los sistemas de una organización.
- Mostrar al alumno él porque es importante el aprendizaje de las bases de datos.
- Identificar la importancia de tener un adecuado control de la información en una organización.

## **HIPOTESIS**

Ante la problemática que presenta el aprendizaje de las bases de datos en la Licenciatura en Informática, concretada en una basta bibliografía y productos informáticos, lo extenso del tema y la ausencia de estándares mínimos para aprenderlas, el presente trabajo tiene establecida la siguiente hipótesis:

Para lograr una clara concepción general de las bases de datos en la Licenciatura en Informática se propone una monografía que cubra todos los tópicos, del programa siguiendo una secuencia de ejercicios relacionados entre sí.

# **CAPITULO**

## **I**

## CAPITULO I

### 1. PLATAFORMA TEORICO CONCEPTUAL

Los sistemas de información de cualquier organización deben ser estudiados y modificados constantemente. Por lo general toda empresa tiene un "ciclo evolutivo" que afecta sus sistemas de información; por ello deben ser diseñados en forma elástica, teniendo en cuenta su crecimiento, la introducción de nuevos métodos y cambios menores y mayores.

Para tomar decisiones adecuadas y con altas probabilidades de éxito, es necesario, además de un buen sistema, considerar los atributos y productos de la información. Los principales problemas que debe superar para desarrollar un buen sistema de información son: Ref. : [11,P.35].

1. Mal diseño de los reportes.
2. Repetición innecesaria de información.
3. Inadecuados canales de comunicación.
4. Circulación de datos innecesarios.
5. Inadecuados métodos de proceso.
6. Inexistencia de una cadena de información desde la base hasta los niveles más altos.

Es un axioma en la administración el que nunca se posee toda la información para cubrir todas éstas facetas y contingencias de la toma de una decisión; sin embargo, entre más información tenga usted para tomar una decisión, más adecuada será la selección realizada. Ahora bien, en forma práctica las organizaciones deben definir apriori sus necesidades de información para ésta toma de decisiones; las necesidades pueden ser clasificadas en:

- a) Ambientales.
- b) Competitivas.
- c) Internas.

APRIORI:Prioridad, importancia, relevancia.

Las **ambientales** incluyen los datos que se refieran a aspectos económicos, sociales o políticos del medio en el cual la empresa esta operando, etc.

Las **necesidades competitivas** se relacionan con el funcionamiento pasado y actual de la empresa, mismo que debe presentar el status de la organización en el mercado, el rendimiento de la inversión, los nuevos productos, las pautas de mercadotecnia que deben seguirse, las nuevas instalaciones, el desarrollo de las firmas competitivas, etc.

Las **necesidades internas** abarcan la identificación de aquellos puntos que sean fuertes y aquellos que sean débiles de una empresa, la inclusión de datos sobre costos, grado de productividad, recursos humanos, etc. Ref.: [11, P.35] .

Hay que aclarar la diferencia entre **información** y **datos**. Aunque muchas personas utilizan estos términos como sinónimos, existe una importante diferencia entre estos conceptos desde el punto de vista de la informática.

Es importante considerar los datos como los insumos o resultados de un fenómeno, es decir, se trata de magnitudes, cifras o relaciones por introducir o derivar de la operación de un sistema. Los datos también pueden estos no ser numéricos; por ejemplo, hechos, principios, etc. Otro concepto de estos datos es aquel que trata como elementos susceptibles de observación directa o de componentes elementales indivisibles de la información. La diferencia básica que existe entre estos datos e información consiste en que los datos no son útiles o significativos como tales, sino que son procesados y convertidos en una forma útil llamada información.

Como podemos ver relación que es establecida entre datos e información a través de un proceso. El conjunto esquematiza un sistema de información. (Figura No.1).



Esto puede considerarse que la información es el conocimiento derivado del análisis de los datos. Es necesario que se haga notar que la información obtenida en un proceso determinado puede servir como dato para otro; por ejemplo, en un sistema de nóminas se tiene como datos las horas trabajadas por cada uno de los empleados y un sueldo por hora, y la información obtenida por el sistema serán; los sueldos semanales de cada empleado. Esta misma información podrá servir como dato de otro sistema en el que se desea determinar los costos semanales por departamento.

Se indicó que puede verse como el producto de un proceso, pero puede considerarse también como un elemento integrador que permita la coordinación de diversos subsistemas, como un agregado de datos que pueden utilizarse directamente, como elemento de reducción de incertidumbre, como representación del conocimiento, como noticia, etc. Sin embargo, el concepto que distingue mayormente la información en relación con otras formas de comunicación, para los propósitos que nos ocupan, en el que se considera como una comunicación que modifica los parámetros de una toma de decisiones. Ref.: [11, P.30].

### **1.1. Niveles de estudio de la información**

C. Shannon y W. Weaver, en un trabajo ya clásico, desarrollaron el concepto de cantidad de información; no obstante, lo que miden en realidad es la cantidad de signos de un mensaje. Según Weaver, este estudio se refiere a lo que se ha denominado nivel técnico, en el cual se interesó por la capacidad de un canal para transferir signos.

Los otros niveles reconocidos por Weaver son: el semántico, que se relaciona con el estudio de la información considerando su significado, y el nivel pragmático, en el cual lo más importante es el efecto que la información tiene sobre el receptor de la misma.

#### **1.1.1. Definición de información**

La información es el significado que tiene una comunicación para un receptor en una situación dada, en relación con un problema específico.

En este sentido, un sistema de información es aquel que debe cumplir, como parte de su operación, una importante labor de comunicación. Aún más, en una organización podría decirse que un sistema de información es un sistema de comunicación formal con capacidad para transformar la información. Ref.: [11, P.33].

Este concepto de información implica no sólo la existencia de ciertos datos o signos de un mensaje y su eventual transmisión a un receptor y, por lo tanto, una modificación del comportamiento del sistema "problema-decisión".

Puede observarse que el concepto de información mencionado nos lleva a la esencia misma de los procesos de planeación, coordinación, dirección, etc., que en última instancia, son aplicaciones específicas de la toma de decisiones.

Cabe también mencionar que, bajo este concepto, la información que no es aplicable a un problema específico puede clasificarse como "ruido", aun cuando en muchas ocasiones debe generarse y almacenarse para usos especiales o futuros que no estén directamente asociados con la toma de decisiones; por ejemplo, datos históricos y datos detallados de las operaciones que pueden servir para fines fiscales, para operar modelos matemáticos o para usarse en otros procesos. Ref. : [11, P.33].

En la era industrial lo más importante era el uso del capital, dinero y recursos tangibles, para generar nuevos productos. En el presente los recursos básicos son las ideas y el uso de la información.

El empleo estratégico de la información continua creando, virtualmente en todas las industrias, nuevas oportunidades. La habilidad para hacer uso de la información, más que los recursos financieros, para que obtenga ventajas competitivas ya sean a través de nuevos productos y servicios o con un trato más eficaz hacia los clientes, Proveedores y competidores, será el factor que decida cuáles empresas tendrán éxito en el año 2000. Ref: [2, P.8].

### **1.1.2. Administración de la información como recurso**

Con el fin de lograr la máxima utilidad de la información, ésta debe administrarse de manera correcta, como ocurriría con cualquier otro de los recursos de la empresa. Los directivos deberán entender que existen costos que se asocian con la producción, distribución, seguridad, almacenamiento y recuperación de la información. Aunque la información aparentemente se encuentra siempre a nuestro alcance, su uso estratégico como un apoyo de la competitividad de nuestro negocio no debe considerarse como un **elemento gratuito**. Ref. : [2, P.1].

### **1.1.2.1 UTILIDAD DE LA INFORMACION**

Roman R. Andrus sugiere que la información se puede evaluar en términos de utilidades las cuales, a pesar de la exactitud de la información, pueden facilitar o retardar su uso. E identificando cuatro utilidades de la información:

- 1.Utilidad de la forma: En la medida en que la forma de la información se aproxima a los requerimientos del gerente, su valor se incrementa.
- 2.Utilidad de la oportunidad: La información tiene un valor mayor para el gerente si está disponible cuando se necesita.
- 3.Utilidad del lugar (accesibilidad física). La información tiene un mayor valor si se pueden acceder o se puede entregar más fácilmente. Los sistemas en línea maximizan tanto la utilidad del tiempo como la de lugar.
4. Utilidad de posesión: (posición en la organización). Quién posea la información afecta fuertemente su valor puesto que controla su diseminación a otros.

Andrus señala que la información tiene un sólo costo como también un valor asociado con ella, tanto en términos de exactitud como en la de utilitarios. Si el costo de la adquisición de información excede su valor, hay dos alternativas:

1. Incrementar su valor aumentando la exactitud y/o con el incremento de su utilidad.
2. Disminuir el costo disminuyendo la exactitud y/o reduciendo alguna de las utilidades.

### **1.1.2.2 SATISFACCION DE LA INFORMACION.**

La contribución de un elemento en particular de la información para mejorar la calidad de la toma de decisiones, es difícil o imposible de determinar en el contexto de una organización. Una medida sustituta para la utilidad de la información en la toma de decisiones es la satisfacción de la información, el grado al cual el gerente se satisface con el resultado del sistema de información formal. Suministra un enfoque para la determinación de la satisfacción de la información.

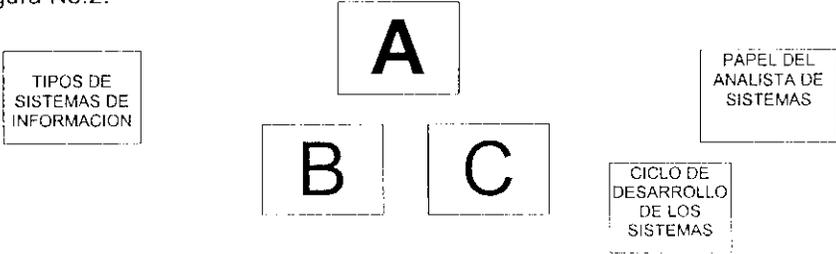
Si quien toma decisiones se da cuenta que un sistema de información formal está disponible para suministrar la información requerida, consultará en primer lugar dicha fuente. Si la información está disponible fácilmente, la satisfacción con el sistema se refuerza. Si no, deberá organizar una amplia investigación para conseguir la información requerida se acentuará la frustración o la insatisfacción con el sistema de información formal. La satisfacción de información sirve como base para evaluar los sistemas de información formal.

**1.1.3. Administración de la información por computadora**

La disponibilidad actual de las computadoras ha generado todo un incremento y una diversificación, tanto para la sociedad en general, como para los negocios en particular. La administración de la información que es generada por computadora, difiere en diversas formas de aquella que se obtiene manualmente. A menudo, se tiene una mayor cantidad de información si ésta se genera utilizando sistemas computacionales; los costos para crear y mantener la información computarizada, son aparentemente mayores; la información que genera la computadora puede llegar a multiplicarse a velocidades impresionantes. Con frecuencia la información que se genera por computadora se trata con menos escepticismo de diferentes tipos de sistemas de información, los diversos papeles del analista de sistemas y las etapas que constituyen el ciclo de desarrollo de los sistemas, tal como se muestra en la figura No. 2.

Un analista de sistemas debe comprender los diversos tipos de sistemas de información, los papeles del analista y el ciclo de desarrollo de sistemas. Ref. : [2, P.2].

Figura No.2.



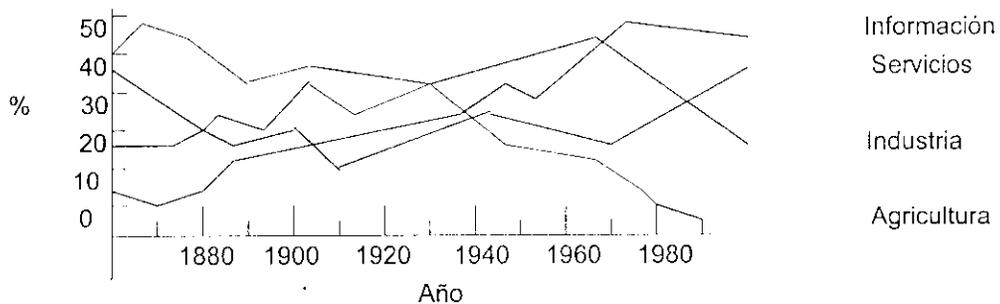
#### 1.1.4 Usuarios: trabajadores de la información

El analista John Naisbitt, autor de *Megatrends: Ten New Directions for Changing Our Lives*, es uno de los muchos que ha estudiado la transformación de la economía hacia un estado donde ésta depende, en gran medida, de los trabajadores de la información. En 1950, en el amanecer de la computación en los negocios, el 17 % de la fuerza de trabajo estaba en los empleos relacionados con la información. Para finales de la década de los ochentas este porcentaje había crecido hasta, aproximadamente, un 70%. Ref.: [2, P.8].

En la actualidad, la industria de manufactura genera sólo alrededor del 28% de los sueldos y salarios del sector privado. El resto proviene de las empresas de servicios y manejo de información ver en la figura número 3.

La fuerza de trabajo laboral en Estados Unidos está representada por Información, Servicios, Industria, Agricultura.

Figura número.3.



Los trabajadores de la información, son aquellos que se ganan la vida al crear, utilizar, procesar, administrar o intercambiar información, en ocasiones reciben el nombre de trabajadores de cuello blanco para distinguirlos de los llamados trabajadores de cuello que prestan sus servicios en la industria y el campo. Aunque exacto, el término trabajador de la información es demasiado abstracto y normalmente no se emplea en las organizaciones e industrias de sistemas de información. El nombre más común para este tipo de trabajador es usuario, término que se refiere a aquellos que utilizan la información y los sistemas de información.

Anteriormente mencionados diferentes tipos de usuarios: banqueros, corredores de acciones y agentes de seguros o de bienes raíces. Existen muchos otros entre los que se incluyen ingenieros, asistentes administrativos, vendedores, técnicos, y gerentes de toda clase. Es importante reconocer el alto grado de dependencia que la sociedad tiene de los sistemas de información, y de las personas que los operan, para soportar sus actividades cotidianas. Sin sistemas de información eficaces muchas industrias serían inoperables. Actualmente, ¿serían capaces los bancos, la bolsa de valores o las aerolíneas de operar sin sistemas de información? La respuesta es no. Ref. : [2, P.9].

## 1.1 DATOS

### A) FUENTES DE DATOS

#### 1.1.1.1 VALOR DE LOS DATOS

El valor de los datos es la información o los datos mismos contenidos en cada campo de datos. El campo de datos "nombre del cliente" puede tomar valores tales como "profesor Higgins" y "Mark Twain". Los valores que toman los campos de datos pueden ser cuantitativos, cualitativos o descriptivos, dependiendo de la manera en que los campos de datos describan a la entidad.

La información acerca de una empresa se puede representar con la ayuda de varias entidades, y una entidad individual se puede representar por medio de varios campos de datos. Los valores que toman los campos se conocen como datos. Al conjunto de valores tomados por los campos de datos de una entidad se le llama *ocurrencia de la entidad*. Las entidades están interconectadas por ciertas relaciones.

Un modelo inherente de las entidades basado en los campos de datos que las representan, así como en las relaciones que interconectan a las entidades, se conoce como *modelo conceptual*. El modelo conceptual muestra una visión general del flujo de los datos de una empresa. Ref. : [2, P.16].

### 1.1.1.2 MEDIOS DE LOS ARCHIVOS DE DATOS

Las fuentes de datos son fundamentales en el procesamiento de información de una empresa. A continuación se enlistan algunos ejemplos de empresas y fuentes de datos.

Figura No.4.

Medio	Fuentes de datos
Agencias corredoras	Actividades bursátiles, tendencias históricas de la bolsa de valores.
Compañías industriales	Inventarios.
Lineas aéreas	Vuelos, tripulaciones, pasajeros, reservaciones, mantenimiento, inventario.
Estaciones policíacas	Patrones de criminalidad, tipos de crímenes
Hospitales	Registros médicos de pacientes, utilización de camas, facturación, inventario de suministros.
Compañías de seguros	Asegurados vitalicios, demandas, primas.
Universidades	Registros de estudiantes, cursos ofrecidos, requisitos de estudio, programas de cursos.
Bancos	Cuentas de cheques, cuentas de ahorro, préstamos a plazos, préstamos hipotecarios.

El planteamiento convencional del procesamiento de datos consiste en desarrollar uno o varios programas para cada aplicación distinta; por ejemplo, aplicaciones bancarias tales como cuentas de cheques y de ahorro, préstamos a plazos y préstamos hipotecarios. Esto tiene como resultado, ciertamente, que para cada aplicación haya uno o más archivos de datos.

### 1.1.1.3 CUALIDADES DE LA INFORMACIÓN

La explosión de la información como se llama a veces a este enorme crecimiento de las necesidades de la información y a la mayor disponibilidad de este recurso puede conducir, si no se ponen los medios para evitarlo, a una polución informativa. Fenómeno análogo a la contaminación del aire, en la que la información, al perder sus cualidades, no puede cumplir sus objetivos, llegando incluso a ser más nociva que beneficiosa para sus destinatarios. Ref. : [4,P.8].

Para evitar el peligro de la polución informativa se debe exigir a la información un conjunto de cualidades que mantengan a su valor comunicativo, ya que para hacer honor a su nombre debe ser capaz de informar, es decir, de aportar un conocimiento.

Las cualidades que debe poseer la información, y que hacen de ella un recurso fundamental de las organizaciones y de los individuos, son básicamente: precisión, oportunidad, plenitud, significado e integridad. Todas ellas en el grado que exija cada sistema concreto.

La **precisión** se puede definir como el porcentaje de información correcta sobre la información total del sistema: fichero, base de datos, etc. Ref. : [4, P.8].

Se comprende fácilmente que no se puede exigir la misma precisión a un censo que a una nómina. Los errores en el caso del censo, mientras se conserven dentro de ciertos límites, no alterarán de forma significativa los resultados, sin embargo en la nómina no es posible admitir ni el más ligero error. Tanto en este caso como en otros en los que se maneja información financiera, como cuentas corrientes, la precisión ha de ser del cien por cien.

De todas formas, usted como usuario ha de tener presente que el tratamiento por ordenador no puede mejorar la calidad de los datos elaborados, lo único que puede hacer, en algunos casos, la máquina es señalar ciertos errores o incompatibilidades, e incluso sustituir el dato detectado como imposible por otro que no tenga error aparente, pero que nunca sabremos si era o no el dato real.

En resumen, si queremos que los resultados del ordenador sean precisos, debemos también suministrarle datos precisos, no pudiendo pretender en los resultados una precisión superior a la que tenían los datos de entrada. Una precisión baja lleva a una falta de credibilidad del usuario hacia la información que se le proporciona.

La **oportunidad** se refiere al tiempo transcurrido desde el momento en que se produjo el hecho que originó el dato hasta el momento en que la información se pone a disposición del usuario. Otras veces la oportunidad se mide en función del momento en que el dato tendría que estar disponible, o bien respecto al desfase que produce el proceso por ordenador. Ref. : [4, P.9].

Al igual que ocurre con la precisión, también la oportunidad depende de cada aplicación. Por ejemplo, para un censo en el cual se manejan millones de informaciones de carácter bastante estable un tiempo de proceso de meses no le resta oportunidad a la información, en cambio esta demora en la obtención de los indicadores de coyuntura, como índices de precios al consumo, sería inadmisibles.

En general, el valor de la información va disminuyendo con el transcurso del tiempo\*, e incluso, después de cierto momento, puede llegar a perder totalmente la relevancia que pudiera tener.

Otra cualidad que ha de tener la información es la **plenitud**, lo que significa que ha de ser completa para poder cumplir sus fines. Por ejemplo, un informe que se emite con el objeto de que un directivo tome una decisión ha de contener todos los elementos informativos necesarios para apoyar dicha decisión. La plenitud absoluta es imposible de conseguir, y lo que se suele pretender en los sistemas de información es alcanzar un nivel que se considere suficiente, el cual dependerá de dos factores: de los datos existentes en el sistema de información y de los que el sistema sea capaz de localizar durante una consulta concreta. En este segundo factor influirá la flexibilidad e idoneidad del lenguaje de recuperación y el acierto en la formulación de la consulta. Así pues, la plenitud no es sólo función de la información en sí misma, sino también de otros factores externos, tanto técnicos como humanos. Ref.: [4, P.9].

La información que se suministra al usuario tiene que tener también **significado**; es decir, ha de tener el máximo contenido semántico posible, ya que sin él no sería verdadera información. Esto lleva a que ha de ser comprensible e interesante, lo que supone no proporcionar a los usuarios grandes masas de información que por su volumen no pueden ser asimiladas. Un volumen de información justa es condición indispensable para que ésta sea significativa. Cuando se realiza el diseño de un sistema es preciso tener en cuenta que la información que se suministra ha de ser, fácil de interpretar, sólo la necesaria y suficiente para que se cumplan los fines esperados. Ref. : [4,P.10].

Asimismo, toda la información contenida en el sistema debe ser **coherente** consigo misma, siendo esta una característica fundamental del sistema de información que permitirá obtener resultados concordantes.

Esta coherencia interna ha de ir unida a una consistencia respecto a las reglas semánticas propias del mundo real; es decir, la información, además de ser consistente en sí misma, ha de representar lo más fielmente posible el mundo real, cualidad que en las bases de datos se suele conocer con el nombre de integridad, y que coincide en gran parte con el concepto de precisión. Date (1983) considera sinónimos ambos conceptos. Ref. : [4, P.11].

Es preciso también atender a la seguridad de la información, ya que ésta ha de ser protegida tanto frente a su deterioro por causas físicas o lógicas como frente a accesos no autorizados.

Todos estos requisitos de la información es necesario tenerlos muy presentes cuando se están haciendo los estudios que llevarán a la implantación de un sistema de información. Hay que buscar el punto de equilibrio necesario para alcanzar los objetivos del sistema a un costo aceptable, ya que cuantas más cualidades reúna la información más se incrementa su precio; por ejemplo, si con un plazo de tiempo algo mayor también se pueden cumplir los objetivos, podría no estar justificada una considerable elevación de los costes para aumentar la oportunidad.

Por otro lado, las cualidades pueden resultar incompatibles con otras; así pretender una gran precisión lleva consigo generalmente una pérdida de oportunidad. Por ello, es necesario llegar a una solución, ese punto de equilibrio entre las diversas cualidades de la información dentro de sus objetivos y sus costos.

#### 1.1.1.4 TIPOS DE INFORMACION

A fin de tener un marco de referencia adecuado, clasificaremos la información de acuerdo con las siguientes dicotomías:

Tabla No.5

<b>activa</b>	-	<b>inactiva</b>
<b>recurrente</b>	-	<b>no recurrente</b>
<b>documentada</b>	-	<b>oral</b>
<b>interna</b>	-	<b>externa</b>
<b>histórica</b>	-	<b>proyectada a futuro</b>

La información activa es la que implica la acción inmediata de quien la recibe; por ejemplo, las órdenes de compra recibidas de los clientes. La acción producida por la información que contienen será procesar el pedido para que pueda ser surtido. Para que una información pueda considerarse como activa debe ser precisa y oportuna.

Por lo contrario, la información inactiva es la que no requiere que se inicie una acción al recibirla; por ejemplo, el recibir la comunicación de que se pagó la nómina implica una información inactiva; generalmente nos indica hechos pasados. La información inactiva debe tratar de ser eliminada en un sistema eficiente. Ref. : [11, P.34].

La información recurrente es la que se genera a intervalos regulares; algunos ejemplos de este tipo son la información contable y los informes de ventas. Por otra parte, la información no recurrente es formulando ocasionalmente para auxiliarse en decisiones como la construcción de una planta o la expansión de una línea de productos. Es evidente que la información no recurrente es la más difícil de manejar y generalmente su manejo resulta más complicado de automatizar que el de la información recurrente.

La información documentada es la de tipo formal que se registra por escrito o bien se conserva en forma codificada; por ejemplo las tarjetas perforadas, la memoria de una computadora y una cinta magnética.

La información oral no se registra por escrito y se pierde al terminar la comunicación. La información de tipo oral no es controlable mientras que la información documentada puede serlo en cierta medida.

La información interna es aquella que se genera dentro de la empresa; por ejemplo, los estados financieros y los reportes presupuestales. La información externa, por su parte se genera en el ambiente de la empresa; por ejemplo, los planes de los competidores, la demanda de un producto por el público consumidor y el ingreso per cápita. Ref: [11, P.34].

La información histórica sirve de base para elaborar proyecciones futuras. La información contable, por ejemplo, es de tipo histórico; se basa en hechos pasados sobre los cuales ya no se tiene ningún control.

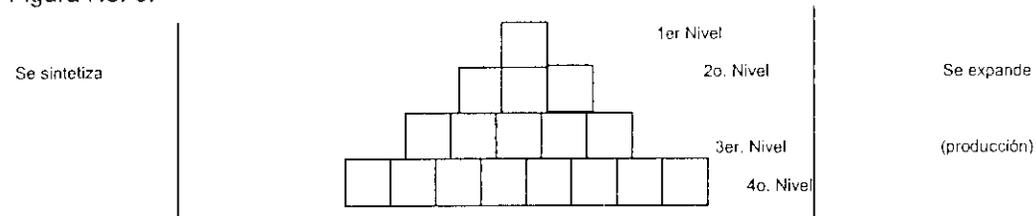
Las proyecciones a futuro nos indican cuál podrá ser el estado de cierta información en un tiempo posterior al actual; por ejemplo, tendencia de mercado y estados financieros proyectados.

Es pertinente indicar que las informaciones activa, recurrente, interna e histórica resulta las más adecuadas para automatizarlas.

La información externa y la información proyectada a futuro son importantes para los niveles directivos de las organizaciones, debido a que éstos dedican la mayor parte de su tiempo a la planeación. Los niveles inferiores de decisión utilizan un mayor porcentaje de tiempo en funciones de control, mientras que la sección superior utiliza la mayor parte de su tiempo en función de la planeación.

Consideremos ahora cómo fluye la información dentro de las organizaciones en el sentido vertical. En la figura No. 6 se ve que la información, a medida que baja de niveles en forma de órdenes, sugerencias, políticas, etc., se va ampliando y aumentando en cantidad y en detalle. En cambio, la información que fluye desde los niveles inferiores hacia los superiores, se va concentrando y resumiendo, es decir, se sintetiza; esta síntesis consiste en informes, reportes respuestas. Podemos deducir, por lo tanto, que la información requerida para las decisiones de alto nivel debe ser concisa y sintética para que los niveles superiores no se vean abrumados por un excesivo número de informes demasiado detallados que les quita mucho tiempo para considerarlos. Ref. : [11, P.35] .

Figura No. 6.



Existen otros tipos de información en las que se clasifican de acuerdo a las personas que la utilizan y son:

#### 1.1.1.5 Información de Apoyo

Mantiene informados a los administradores en relación con situaciones actuales o niveles de logros. Ejemplo: Los volúmenes de ventas del día de ayer.

#### **1.1.1.6 Información de Situación**

Mantiene a los administradores al tanto de los problemas presentes y de la crisis. Ejemplo: El avance en el día de ayer en las negociaciones de los contratos laborales.

#### **1.1.1.7 Información de Advertencia.**

Señala que están ocurriendo cambios, ya sea en la forma de oportunidades que se presentan, o bien, presagios de problemas futuros que afectarían el éxito de la empresa. Ejemplo; Fluctuaciones significativas en los precios de materias primas como el acero, o bien la energía.

#### **1.1.1.8 Información de Planeación**

Descripción de los principales desarrollos y programas que deben iniciarse en un futuro. Ejemplo: Como se espera que se amplíe o se contraiga el mercado de la industria.

#### **1.1.1.9 Información de Operaciones Internas.**

Indicadores clave de cómo la organización o las personas se están desempeñando; útiles para presentar informes sobre la salud general de una organización. Ejemplo: Rendimiento acumulado sobre el capital patrimonial.

#### **1.1.1.10 Información Confidencial.**

Informes, rumores y opiniones respecto a las actividades en el entorno de la organización. Ejemplo: Las demandas de los clientes por nuevos productos o servicios.

#### **1.1.1.11 Teoría de la Información**

El término teoría de información se utiliza con frecuencia para referirse a la teoría matemática de la comunicación. La teoría matemática tiene su aplicación directa en los sistemas de información gerencial, pero suministra comprensión acerca de la naturaleza de la información. Ref. : [12, P.22].

Los problemas de comunicación de información en los sistemas de información se pueden considerar en términos de tres niveles:

1. **Nivel técnico.** ¿Cómo puede transmitirse la información de una manera exacta?
2. **Nivel semántico.** (Presentación). ¿Cómo pueden converger de manera precisa los símbolos transmitidos para el significado deseado?
3. **Nivel de efectividad (calidad)** ¿De qué manera es apropiado el mensaje como un motivador de la acción humana?

La teoría matemática de la comunicación tiene que ver con el nivel técnico. Las siguientes dos secciones tratan con aspectos de los niveles semánticos y de efectividad.

## **B) 1.1.2 EL MODELADO DE DATOS, RELACIONES, ENTIDADES Y ATRIBUTOS**

### **1.1.2.1 El modelo de datos relacional y el álgebra relacional**

El modelo relacional de los datos fue introducido por Codd (1970). El cual se basa en una estructura de datos simple y uniforme - la relación - y tiene fundamentos teóricos sólidos.

El modelo relacional se está estableciendo firmemente en el mundo de las aplicaciones de bases de datos, y existen en el mercado muchos paquetes de SISTEMA DE GESTIÓN DE BASE DE DATOS (SGBD) relacionales.

El modelo E-R hace uso de relaciones o en su defecto de entidades relación para relacionar dos entidades diferentes pero pertenecientes a la base de datos.

En el modelo relacional, las relaciones entre tablas (implementación de conjunto de entidades) se realiza por medio de campos comunes.

Las operaciones básicas del álgebra relacional, o de relaciones son: Unión, intersección y diferencia de relaciones de coincidencia ( $\cup, \cap, -$ ), Proyección por dominios ( $\Pi$ ), Selección de relaciones ( $\sigma$ ), Unión\_j de relaciones arbitrarias ( $\bowtie$ ).

Estas operaciones, junto con los operadores de comparación y boléanos utilizados en las proposiciones de calificaciones del cálculo relacional, proporcionan las herramientas para los sistemas basados en el álgebra relacional.

Estos sistemas pueden utilizarse para dar apoyo a un cálculo relacional o pueden estar directamente disponibles para los usuarios. Su naturaleza es inherentemente de procedimiento y estos sistemas son comparables con aquellos que utilizan archivos múltiples no jerárquicos no enlazados y operaciones convencionales de procesamiento de datos. Ref:[6, P.554]

### **1.1.2.2 CONCEPTOS DEL MODELO RELACIONAL**

El modelo relacional representa la base de datos como una colección de relaciones.

Si damos un vistazo una relación como una **tabla** de valores, cada fila de la tabla representa una colección de valores de datos relacionados entre sí. Dichos valores se pueden interpretar como hechos que describen una entidad o un vínculo entre entidades del mundo real.

El nombre de la tabla y los nombres de las columnas ayudan a interpretar el significado de los valores que están en cada fila de la tabla.

Los nombres de las columnas - Nombre, NúmEstudiante, Grado, Carrera - especifican cómo interpretar los valores de datos de cada fila, con base en la columna en la que se encuentra cada valor. Todos los valores de una columna tienen el mismo tipo de datos. El tipo de datos que describen los tipos de valores que pueden aparecer en cada columna se llama dominio. Ref: [ 16, P.15 ]

### **1.1.2.3 DOMINIO, TUPLAS, ATRIBUTOS Y RELACIONES.**

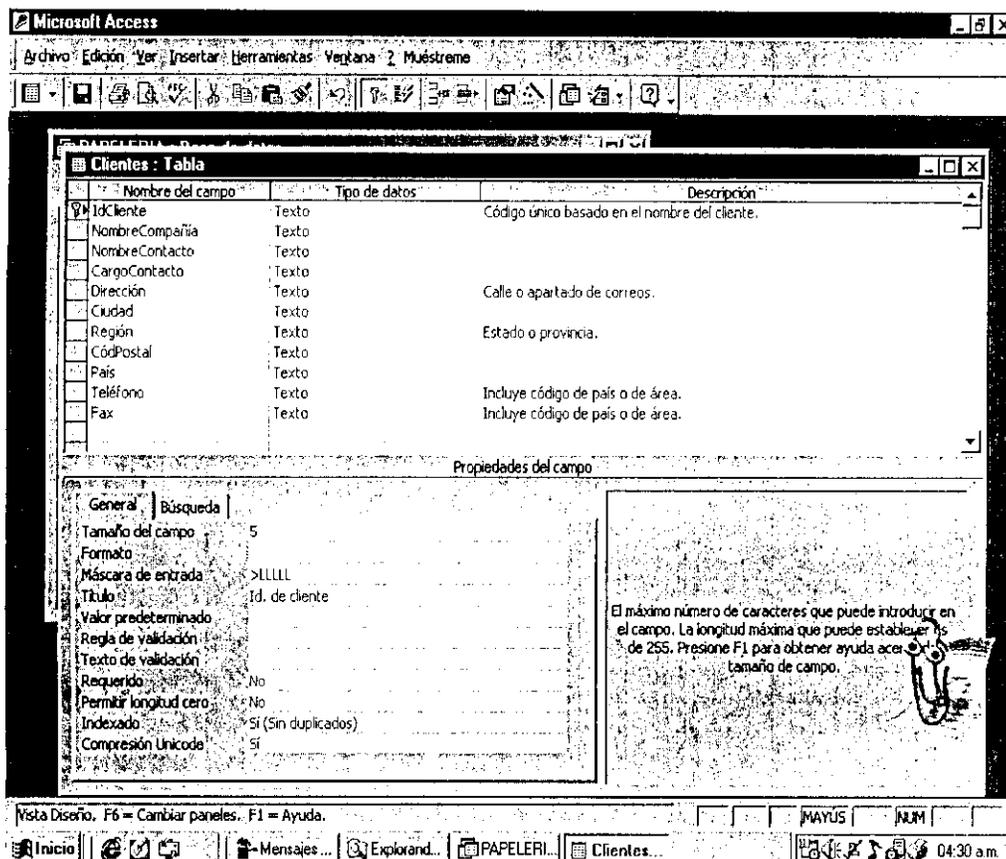
Un dominio D es un conjunto de valores atómicos. Por atómico queremos decir que cada valor del dominio es indivisible en lo tocante al modelo relacional. Un método común de especificación de los dominios consiste en especificar un tipo de datos a los cuales pertenecen los valores que constituyen el dominio.

Un dominio debe tener un nombre, un tipo de datos y un formato. También puede incluirse información adicional para interpretar los valores de un dominio. Por ejemplo, un dominio numérico como Pesos\_ de\_personas deberá especificar las unidades de medición: libras o kilogramos.

El tipo de una variable tiene una fuerte connotación de su representación. Es preferible definir un conjunto de valores permitidos: esto está más cerca del concepto de dominio como se utilizó en el modelo de base de datos. En algunos de los modernos lenguajes de programación se dispone de proposiciones para definir un dominio de datos como un anexo a las declaraciones de tipo.

Una proposición DEFINE describe en nuevo tipo de datos asociando un nombre de tipo con una especificación de dominio. El empleo, durante el procesamiento, de elementos dato fuera el dominio se detecta en forma semejante a los conflictos de tipo que ocurren cuando, por ejemplo, se mezclan cadenas de números y caracteres. Se dispone de definiciones de dominio en el lenguaje PASCAL.Ref:[6, P. 485]

Figura No.7



Un esquema de relación  $R$ , denotado por  $R(A_1, A_2, \dots, A_n)$ , se compone de un nombre de relación  $R$ , y una lista de atributos,  $A_1, A_2, \dots, A_n$ . Cada atributo  $A_i$  es el nombre de un papel desempeñado por algún dominio  $D$  en el esquema  $R$ . Se dice que  $D$  es el dominio de  $A_i$  y se denota con dominio ( $A_i$ ). Un esquema de relación sirve para describir una relación;  $R$  es el nombre de la relación. El grado de una relación es el número de atributos,  $n$ , de su esquema de relación.

Los atributos, que representan propiedades de la misma y que están caracterizadas por un nombre, y un conjunto de filas llamadas **tuplas**, que son las ocurrencias de la relación. El número de filas de una relación se denomina **cardinalidad**, mientras que el número de columnas es el grado. Existen también **dominios** de donde los atributos toman sus valores

El valor mismo de un atributo puede dividirse en un número de pares de nombre de atributo - valor para permitir la organización estructural del registro, como anteriormente se ve en el ejemplo.

Una relación se puede representar en forma de tabla, aunque tiene una serie de elementos característicos que la distinguen de la tabla:

- No puede haber filas duplicadas, es decir, todas las tuplas tienen que ser distintas.
- El orden de las filas es irrelevante.
- La tabla es plana, es decir, en el cruce de una fila y una columna sólo puede haber un valor (no se admiten atributos multivaluados).

Una relación siempre tiene un nombre, y en ella es posible distinguir una cabecera (esquema de relación o intensión) que define la estructura de la tabla; es decir, sus atributos con los dominios subyacentes, y un cuerpo, extensión, que está formado por un conjunto de tuplas que varían en el tiempo. Ref: [4, P.434.]

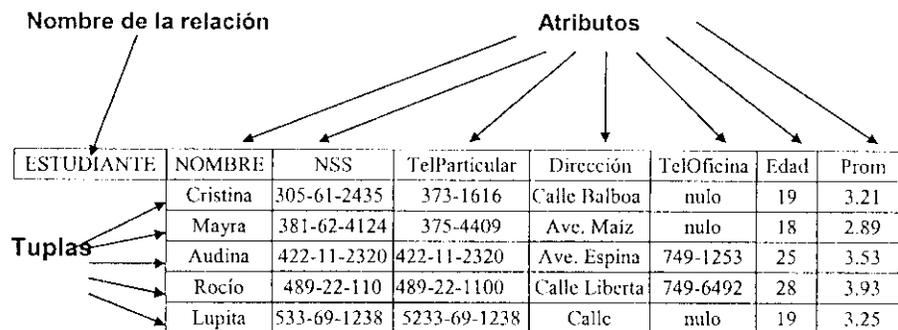
### 1.1.2.4 CARACTERÍSTICAS DE LAS RELACIONES

La primera definición de relación implica ciertas características que distinguen a una relación de un archivo o de una tabla.

**Orden de las tuplas en una relación.** Una relación se define como un conjunto de tuplas. Matemáticamente, los elementos de un conjunto no están ordenados; por tanto, las tuplas de una relación no tienen un orden específico. Los registros de un archivo se almacenan físicamente en el disco, de modo que siempre existe un orden entre ellos. Este ordenamiento indica el primero, segundo, i-ésimo y último registro del archivo. Cuando presentamos una relación en forma de tabla, las filas se muestran en cierto orden.

El ordenamiento de las tuplas no forma parte de la definición de una relación, porque la relación intenta representar los hechos en un nivel lógico o abstracto.

Figura No. 8



### 1.1.2.5 Definición de relaciones

Cuando se desea implementar una base de datos relacional para una aplicación compleja, los diseñadores suelen empezar por diseñar con mucho cuidado el esquema de la base de datos. Esto implica decidir cuáles atributos deben ir juntos en cada relación, elegir nombres apropiados para las relaciones y sus atributos, especificar los dominios y tipos de datos de los diversos atributos, identificar las claves candidatas y escoger una clave primaria para cada relación, y especificar todas las claves externas.

Todo Sistema de Gestión de Bases de Datos Relacional debe contar con un lenguaje de definición de datos (DDL) para definir los esquemas de relaciones. La mayoría de los DDL se basan en el lenguaje SQL.

#### 1.1.2.6 ASPECTOS ESTATICOS DEL MODELO RELACIONAL

**Relación:** El sistema de gestión de base de datos (SGBD) Oracle sigue el modelo entidad - relación, representando tanto el elemento entidad como el elemento relación de un sistema a través de la misma estructura: La **tabla**. Una tabla consta de filas y columnas. Ref: [4, P. 916].

La fila referencia una ocurrencia de la entidad o relación representada por la tabla. El orden interno de las tuplas en una tabla no es significativo; es la cláusula ORDER BY de la sentencia Select la que permite visualizar la información en un orden específico.

La columna queda identificada por su nombre, no por su posición. El orden interno de las columnas no es significativo. Se especifica la secuencia de columnas a visualizar en la lista de atributos de la sentencia Select. Se permiten hasta un total de 254 columnas por tabla. Ref: [4, P. 917].

La extensión de SQL Alter Table Add [Modify] permite añadir [aumentar el tamaño de ] columnas de una forma dinámica, sin descarga y recarga de información.

#### 1.1.2.7 DEFINICION

Los aspectos estáticos del modelo relacional se definen en una Metabase de forma asistida o bien directamente con sentencias S.Q.L:

- Los esquemas de relaciones estáticas se definen en la metabase. Se garantiza la unicidad de las claves primarias establecimiento un índice para el dominio y declarándolo único (la clave puede componerse de distintos campos). También pueden definirse claves secundarias. La clave principal (o cualquier clave secundaria) puede declararse multiocurrente.

- Los dominios pueden definirse como caracteres, enteros, reales, booleanos, fechas, o vectores de una columna. La longitud y demás características (ámbito de los dígitos aceptables como por ejemplo la obligatoriedad de introducir todos los dígitos, un determinado dígito en una posición determinada, mayúsculas, etc.) se definen en la metabase. Se dispone del valor nulo (desconocido) para cualquiera de estos tipos. Los datos se almacenan en longitud variable. Ref: [4, P. 929].

#### **RESTRICCIONES:**

- Integridad de dominio:

Se define directamente en la metabase, estas especificaciones son dinámicas; pueden variarse en cualquier momento (con la salvedad de que los datos existentes queden coherentes con la restricción introducida). La integridad de dominio incluye desautorización del valor nulo, comprobación del cumplimiento de una fórmula. En lo que a fechas se refiere, sólo se permite la introducción de los dígitos que corresponden a días existentes. Un dominio puede definirse como sensible a la distinción entre mayúsculas y minúsculas o no. Ref: [4, P. 930].

- Integridad referencial:

Se dispone de una función (CAN-FIND) directamente utilizable desde el metaesquema y que comprueba la existencia en una tabla determinada de una clave que cumpla la expresión que se especifique en el argumento de la misma. El tratamiento de la integridad referencial es flexible: no se trata de una simple búsqueda de la clave, sino del cumplimiento por el registro identificado por la clave de una expresión, además, la clave que se busca no tiene porque ser la principal, ni siquiera única (se puede indicar de forma específica el índice a utilizar). Y también dinámico (puede variarse en cualquier momento desde el metaesquema).

### **1.1.2.8 Modelado Relacional.**

Es un sistema de administración de Bases de Datos relacionales (RDBMS). Este permite crear y definir fácilmente la estructura de la Base de Datos. Se pueda insertar, cambiar, manipular, o borrar los datos y las estructuras de datos en la Base de Datos como sean requeridos los cambios.

También incluye un lenguaje de cuarta generación (4GL) y un conjunto de herramientas de desarrollo interactivas que permiten escribir aplicaciones para llevar a cabo trabajos.

La estructura física de la Base de Datos consiste de los elementos de una Base de Datos relacional que solamente es necesario leer acerca de: tablas, campos e índices. El Diccionario de Datos es usado para construir la estructura física de la Base de Datos. La descripción de la estructura de Base de Datos (los archivos que contiene, los campos dentro de los archivos, vista, etc) es llamada esquema de la Base de Datos o la definición de datos.

La estructura interna de una Base de Datos que hace posible almacenar y obtener datos de una Base de Datos es llamada el metaesquema.

### **1.1.2.9 Relación Uno a uno.**

Una relación uno a uno existe cuando cada registro en una tabla tiene solo un registro relacionado en una segunda tabla. Por ejemplo, una papelería puede decidir asignar una mercancía exactamente a un proveedor. Así, como lo muestro en la figura No.9. Ref: [17, P.2]

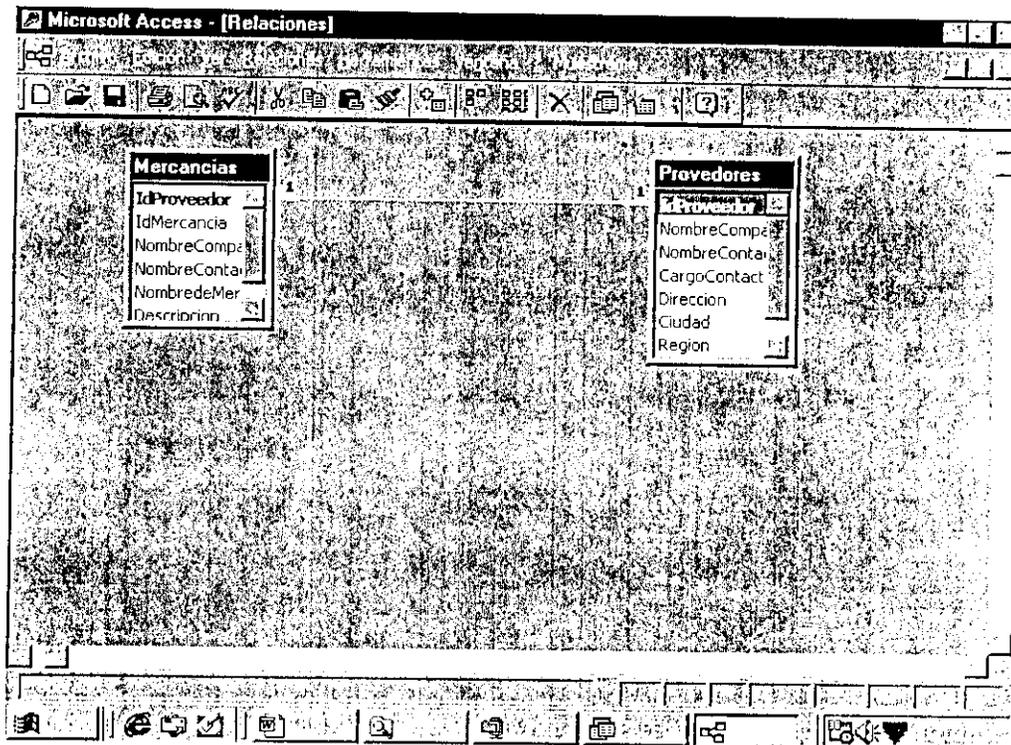


Figura No.9

Sin embargo, la papelería también puede decidir que para una mercancía haya cero o un proveedor; para que una mercancía no haya un solo proveedor o exista uno entonces la relación es descrita como una relación cero a uno.

#### 1.1.2.10 Relación uno a muchos.

Una relación uno a muchos existe cuando cada registro en una tabla tiene uno o muchos registros relacionados en una segunda tabla. La figura No. 10, muestra un ejemplo: un cliente puede colocar muchas órdenes. Un empleado puede tomar realizar muchas ventas, y una venta representativa puede tener, muchas cuentas de clientes.

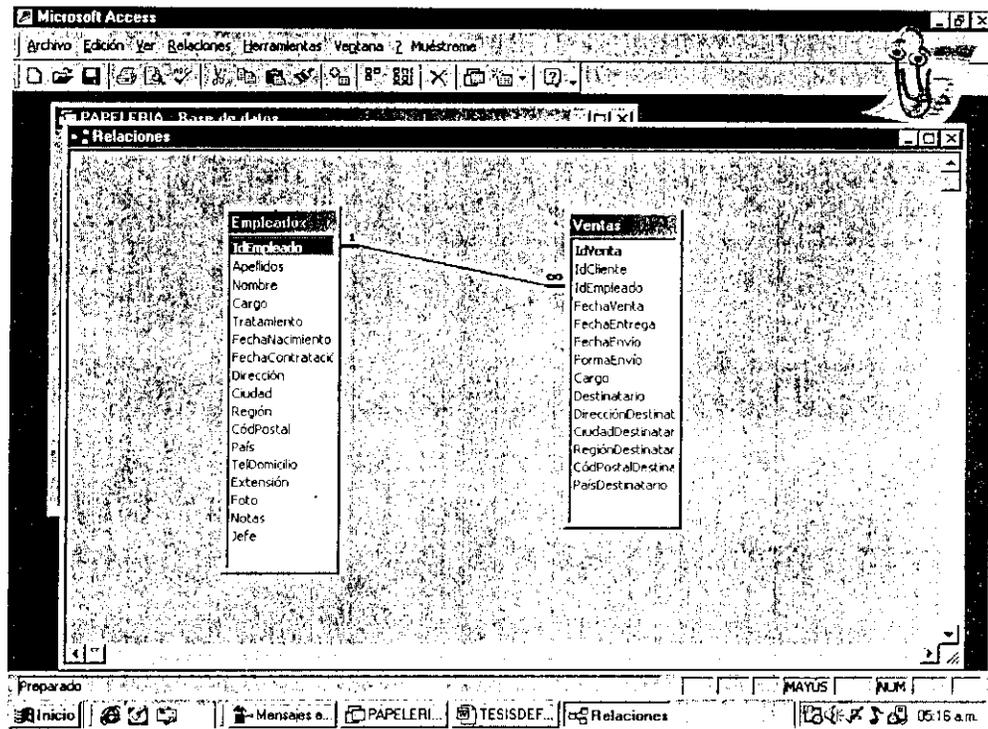


Figura No. 10

### 1.1.2.11 Relación muchos a muchos

Una relación muchos a muchos existe cuando un registro en una tabla tiene muchos registros relacionados en una segunda tabla. También esos registros relacionados tienen muchos registros en la primera tabla. Como se ve en la figura No. 11, una venta contiene muchas mercancías, y una mercancía puede aparecer en muchas ventas diferentes; un cliente puede trabajar en muchas ventas, y una venta puede tener muchas mercancías trabajando en la papelería. Ref:[17, P.4]

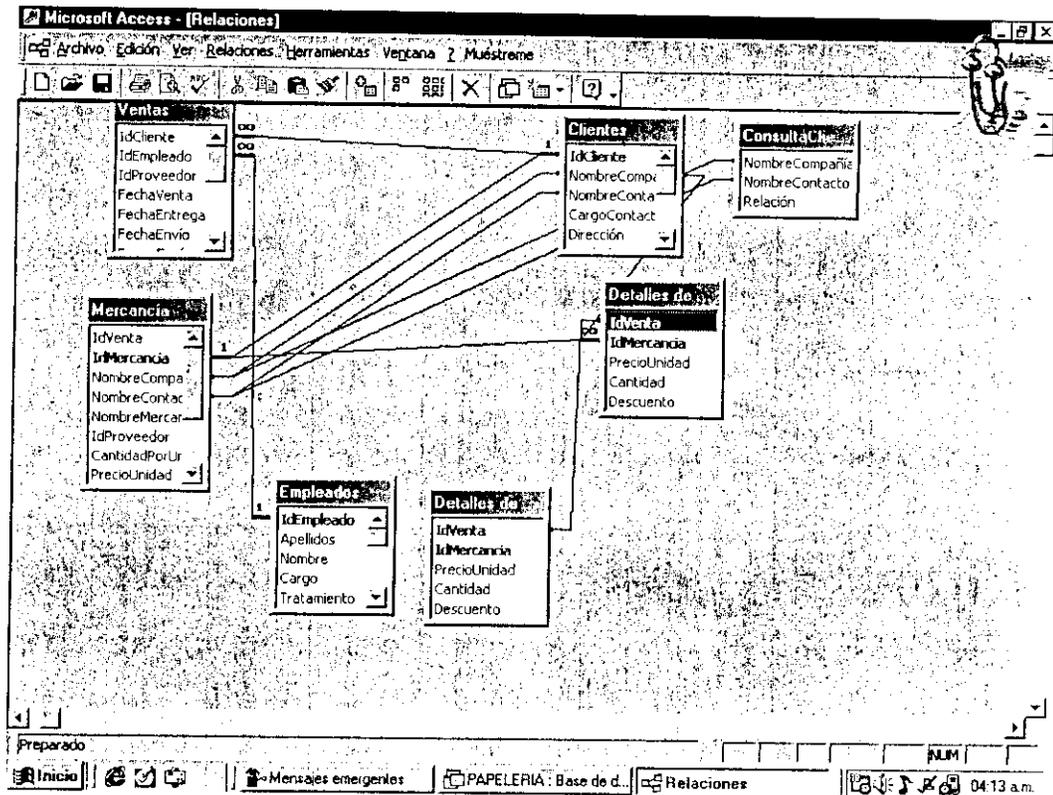


Figura No. 11

Adcesar a la información en tablas con una relación muchos a muchos es problemática y consume tiempo. Para un procesamiento eficiente, se pueden convertir la relación muchos a muchos de la tabla a dos relaciones uno a muchos conectando esas dos tablas con una tabla de referencias cruzada que contiene los campos relacionados. Ref:[17, P.6]

## 2. ORGANIZACION DE ARCHIVOS

**2.1.1 Archivos secuenciales.** En la organización secuencial de archivos, la secuencia física de los registros en el archivo coincide con la secuencia lógica del archivo. La actualización de los datos en un archivo secuencial se lleva a cabo reescribiendo el archivo original para producir un nuevo archivo maestro.

**2.1.2 Archivos relativos.** Los registros en un archivo de organización relativa se guardan continuamente en la unidad de almacenamiento interno. Un registro se puede acceder aleatoriamente, dando su dirección relativa (su número relativo desde el origen del archivo).

El método más común para asociar claves principales con sus direcciones relativas es la distribución. Un algoritmo de distribución deseable es el que puede distribuir claves principales de la manera más uniformemente posible en los espacios de direcciones disponibles. Ref.[4,P.7].

La ventaja principal del uso de una función de distribución es que los registros de datos se pueden localizar dentro de un sólo acceso a disco, esto siempre y cuando las colisiones de distribución estén bajo control. Las colisiones se pueden minimizar pero no eliminar. Se trataron distintos métodos para solucionar las colisiones de dispersión. Una solución sencilla es la del direccionamiento abierto, en la cual los registros con sobreflujo se guardan en los espacios libres más cercanos a la dirección inicial.

Las bases de datos pueden organizarse de muchas maneras. La cuestión ha sido estudiada en detalle por muchos organismos, por empresas que utilizan computadoras, por comisiones del gobierno, y por grupos para tareas confidenciales de los fabricantes de computadoras.

## **1. Versatilidad para la representación de relaciones**

Diferentes programadores requieren diferentes archivos lógicos. Estos archivos deben derivarse de la misma colección de datos. Existen diversas relaciones entre los rubros de datos almacenados. Algunas bases de datos comprenden un complejo entretrejo de relaciones. El método de organización debe tener capacidad para representar estas relaciones y acomodar los posibles cambios en el futuro. El sistema de administración de datos debe tener a su vez capacidad para derivar, de datos y relaciones y acomodar los posibles cambios en el futuro. El sistema de administración de datos debe tener a su vez capacidad para derivar, de datos y relaciones, los archivos lógicos que se requieran. Puede haber muy poco parecido entre el archivo lógico que se presenta a un programador de aplicaciones y el almacenamiento físico de los datos.

## **2. Desempeño**

Las bases de datos diseñadas para ser usadas por operadores de terminal deben asegurar un tiempo de respuesta adecuado para el diálogo entre el hombre y la terminal. Además, el sistema de bases de datos debe tener capacidad para manejar un adecuado caudal de transacciones.

En los sistemas en que el volumen de tráfico es reducido, el caudal de transacciones no tiene porqué imponer muchas restricciones al diseño de la base de datos. En cambio, en los sistemas de alto volumen de tráfico, por ejemplo los de reserva de plazas en las aerolíneas, el caudal de transacciones tiene una gran influencia sobre la organización del almacenamiento físico. Ref: [15, P.31]

En los sistemas diseñados solo para el procesamiento por lotes, el tiempo de respuesta no tiene una gran significación y el método de organización física se elegirá teniendo a la vista la mayor eficiencia con ese tipo de procesamiento.

El tiempo de respuesta adecuado en el caso de un sistema con terminales depende de la naturaleza del diálogo entre hombre y terminal. Para ciertas categorías de diálogo se requiere un tiempo de respuesta del orden de dos segundos (es decir, dos segundos en que aparece en ésta el primer carácter de la respuesta). En el caso de averiguaciones simples pueden tolerarse, en general, tiempos de respuesta mucho mayores.

### **3. Costo Mínimo**

Con el fin de mantener bajo el costo hay que elegir técnicas que minimizan las necesidades totales de almacenamiento. Apelando a estas técnicas, la representación física de los datos en el almacén puede ser muy distinta de la acción que usa el programador. La conversión entre ambas representaciones se hace por medio del software, o posiblemente del hardware o de la microprogramación. Debe llegarse siempre a una concordancia entre el costo de los algoritmos de conversión y la economía de almacén.

El costo por bit de almacenamiento está disminuyendo rápidamente gracias al progreso tecnológico, mientras que no ocurre lo mismo con el costo de la programación de aplicación, y las organizaciones lógicas de datos deben diseñarse con este objetivo. Ref: [15, P.32]

### **4. Redundancia mínima**

Antes de la aparición de las técnicas de base de datos, existía un altísimo nivel de redundancia en los sistemas de procesamiento de datos. La mayoría de las bibliotecas de cinta contiene muchos datos redundantes.

Aún con las técnicas de las bases de datos, a medida que crecientes volúmenes de información se combinan para formar bases de datos integradas, se crea una mayor posibilidad de redundancia. La redundancia es onerosa porque ocupa más espacio de almacén que lo necesario y requiere múltiples operaciones de actualización.

### **5. Capacidad de búsqueda**

El usuario de una base de datos suele plantear muchos interrogantes acerca de los datos almacenados. En la mayoría de las aplicaciones comerciales actuales, se han anticipado los tipos de averiguación y los datos se han organizado físicamente de modo de poder satisfacerlas con adecuada prontitud. Hay una creciente demanda para sistemas que sean capaces de atender consultas o producir informes que no han sido previstos detalladamente en la época del diseño. El usuario quiere presentar pedidos espontáneos de información en una terminal. Las averiguaciones no anticipadas (y algunas de las anticipadas) hacen necesario explorar algunas partes de la base de datos.

Si en la terminal se pretende una respuesta inmediata, la exploración tiene que ser rápida. La capacidad para explorar una base de datos rápidamente y con diferentes criterios de búsqueda depende mucho de la organización física de los datos.

Con algunas organizaciones los tiempos de búsqueda son excesivamente prolongados para poder considerar las respuestas como de tiempo real. Uno de los objetivos de la organización es, pues, el de lograr capacidad para la búsqueda rápida y flexible. Ref: [15, P.33.]

## **6. Integridad**

Cuando una base de datos incluye información utilizada por muchos usuarios, es importante que no puedan destruirse los datos almacenados ni las relaciones que existen entre los distintos ítems. Ocasionalmente se producirán fallos de hardware y diversos tipos de accidente. El almacenamiento de los datos y los procedimientos de actualización e inserción deben asegurar que el sistema pueda recuperarse de estas contingencias sin daño para los datos. Toda instalación debe garantizar la integridad de la información que almacena.

Además de proteger los datos contra posibles problemas sistémicos deben incluirse también procedimientos de chequeo que se aseguren que los valores de los datos se ajusten a ciertas reglas prescritas de antemano. Estas pruebas podrán hacerse verificando las relaciones que existen entre varios valores de datos. Ref: [15, P.34]

## **7. Reserva (privacidad) y seguridad**

Los datos albergados en una base de datos deben ser conservados con seguridad y reserva. La información almacenada es a menudo de gran valor para la empresa. Ref: [15, P.36.]

Deben estar protegidos contra su pérdida o robo. Cuando más vital es la información almacenada, tanto más importante es protegerla contra los fallos del hardware o software, contra las catástrofes y contra los criminales, vándalos, incompetentes y personas que pretendan darles uso ilegítimo.

## **8. La interface con el pasado**

Los organismos que han estado usando el procesamiento de datos durante algún tiempo han invertido mucho dinero en los programas, procedimientos y datos existentes. Cuando un organismo se decide a instalar un nuevo software de base de datos, es importante que éste pueda trabajar con los programas y procedimientos existentes y que los datos ya almacenados puedan ser convertidos a las nuevas formas.

Esta necesaria compatibilidad llega a menudo a tener el carácter de una severa restricción para pasar al nuevo sistema de base de datos. Ref:[15, P.37]

## **9. La interface con el futuro**

Más importante es la interface con el futuro. En el futuro ya presente, los datos y los medios de almacenamiento cambiarán de muchas maneras. Ninguna organización comercial es estática; el cambio es un modo de ser. Pero los cambios han resultado extremadamente costosos para los usuarios del procesamiento de datos. El enorme costo de cambios aparentemente triviales ha frenado seriamente el desarrollo de nuevas aplicaciones, convertir datos, y resolver los muchos problemas menores introducidos por los cambios.

Con el transcurso de los años, la cantidad de programas de una organización va creciendo hasta el punto de que en algún momento no podrá ya pensarse en volverlos a escribir.

Uno de los objetivos más importantes en el diseño de una base de datos será, pues, el de planearla de manera que se la pueda modificar sin necesidad de tener que alterar los programas de aplicación en uso. Ref:[15, P. 39.]

### **3. BASE DE DATOS**

#### **1.3. DEFINICION DE UN SISTEMA DE BASES DE DATOS.**

Un sistema de bases de datos es un sistema computarizado de información para el manejo de datos por medio de paquetes de software llamados sistemas de manejo de bases de datos (SGBD).Ref.[10,P.5].

Los tres componentes principales de un sistema de bases de datos (sistema de gestión de base de datos (SGBD)), son el hardware, el software sistema de gestión de base de datos (SGBD) y los datos que manejar. Ref.[10,P.5 ].

#### **A) CONCEPTO**

##### **1.1.1 Bases de datos**

Son muy numerosas las definiciones de base de datos, y si se analizan detenidamente se suele observar en casi todas estas coincidencias en ciertos elementos, aunque también se detectan la falta de otros fundamentales, o al menos muy importantes, que son característicos de las bases de datos y que marcan la diferencia entre este concepto y el de ficheros. Ref.: [4,P.43].

La aparición de la expresión *base de datos* se produce a comienzos de los años setenta. En 1963 tuvo lugar en Santa Mónica (USA) un simposio en cuyo título se encontraba la expresión *Data Base\**. En una de sus sesiones se propuso una definición de base de datos que, según las actas del simposio, no fue universalmente aceptada.

##### **1.1.2 Concepto**

"Colección de datos interrelacionados almacenados en conjunto sin redundancias perjudiciales o innecesarias; su finalidad es servir a una aplicación o más de la mejor manera posible; los datos se almacenan de modo que resulten independientes de los programas que los usan; se emplean métodos bien determinados para incluir nuevos datos y para modificar o extraer los datos almacenados" (Martín, 1975) Ref.: [ 4, P.44].

"Conjunto de datos de la empresa memorizado por un ordenador, que se utilizan por numerosas personas y cuya organización está regida por un modelo de datos " (Flory, 1982). Ref:[4,P.44]

Una base de datos es una colección de archivos interrelacionados creados con un sistema de gestión de base de datos (SGBD). El contenido de una base se obtiene combinando datos de todas las diferentes fuentes en una organización, de tal manera que los datos estén disponibles para todos los usuarios, y los datos redundantes puede eliminarse, o al menos minimizarse. Ref.[10, P.6].

## **B) DICCIONARIO DE DATOS**

### **1.1.1.1 Diccionario de datos**

Es preciso distinguir entre los conceptos de diccionario y de directorio de datos, que, aunque muy relacionados entre sí, cumplen finalidades distintas en el SI.

**Diccionario de datos:** Reúne la información sobre los datos almacenados en la base de datos.

**Directorio de datos:** Es el subsistema del sistema de gestión de base de datos (SGBD) encargado de describir dónde y cómo almacenan los datos de la base, el modo de acceso y otras características fichas de los datos, atendiendo de este modo las peticiones de los programas y procesos. Ref. [10, P.116 ].

La complejidad de los diccionarios de datos se ha ido haciendo cada vez mayor, ampliando el número de funciones que se le asignaron originalmente, así como el entorno operativo en el que utilizaban. En los años setenta aparecieron varios paquetes de soporte lógico de este tipo: DB/DC Datadictionary, Datamanager, Datadictionary System, ADR/Datadictionary, Léxico, etc.

## **C) SISTEMA DE GESTION DE BASE DE DATOS.**

### **1.1.1.1.1. Sistemas de gestión de base de datos (SGBD).**

Un sistema de gestión de base de datos (DBMS database management system) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. La colección de datos, normalmente denominada base de datos, contiene información acerca de una empresa determinada. El objetivo primordial de un sistema de gestión de base de datos es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos.

Los sistemas de bases de datos están diseñados para gestionar grandes bloques de información. La gestión de datos implica tanto la definición de estructuras para el almacenamiento de información como la provisión de mecanismos para la gestión de la información. Además los sistemas de bases de datos deben mantener la seguridad de la información almacenada, pese a caídas del sistema o intentos de accesos no autorizados. Si los datos van a ser compartidos por varios usuarios, el sistema debe evitar posibles resultados anómalos. Ref:[9, P.1]

#### **a) Tipos**

Las operaciones típicas que debe realizar un sistema manejador de bases de datos puede resumirse en aquellas que afecten a la totalidad de los datos o a todos los registros de un determinado tipo y las que tienen lugar sobre registros concretos.

Las funciones esenciales de un sistema manejador de bases de datos son las de descripción, manipulación y utilización.

##### **a.1 Función de descripción o definición**

Esta función debe permitir al administrador de la base especificar los elementos de datos que la integran, su estructura y las relaciones que existen entre ellos, las reglas de integridad semántica, los controles a efectuar antes de autorizar el acceso a la base, etc., así como las características de tipo físico y las vistas lógicas de los usuarios.

A nivel interno, se ha de indicar el espacio (volúmenes, cilindros y pistas) reservado para la base, la longitud de los campos o elementos de datos, su modo de representación (binario, decimal, alfanumérico, punto fijo o flotante, etc.). Además, se debe poder definir caminos de acceso, como punteros, índices, etc.

Para las estructuras externas y lógica global, la función de descripción ha de proporcionar los instrumentos para la definición de las entidades y su identificación, atributos de las mismas, interrelaciones entre ellas, autorizaciones de acceso, restricciones de integridad, etc.

### **a.2 Función de manipulación**

La función de manipulación permite a los usuarios de la base, informáticos, o no, buscar, añadir, suprimir o modificar los datos de la misma, siempre de acuerdo con las especificaciones y las normas de seguridad dictadas por el administrador.

La función de manipulación se llevará a cabo por medio de un lenguaje de manipulación de datos (LMD) que facilita los instrumentos necesarios para la realización de estas tareas.

### **a.3 Función de utilización**

Se trata de una función que muchos autores consideran integrada en las dos anteriores, por lo que no aparece explícitamente definida.

La función de utilización reúne todas las interfaces que necesitan los diferentes usuarios para comunicarse con la base y proporciona un conjunto de procedimientos para el administrador.

Las exigencias respecto a la forma de utilizar las base de datos son muy diferentes, según los tipos de procesos y según los usuarios, siendo preciso que la función de utilización responda a todas ellas.

En la mayoría de los sistemas de gestión de bases de datos existen funciones de servicio, como cambiar la capacidad de los ficheros, obtener estadísticas de utilización, cargar archivos, etc, y principalmente las relacionadas con la seguridad física (copias de seguridad, re arranque en caso de caída del sistema, etc) y de protección frente a accesos no autorizados. Todas ellas se consideran comprendidas en la función de utilización.

Estas funciones esenciales se subdividen en otras; así, la función de manipulación incluye la de recuperación y actualización, y esta última, a su vez, comprende el alta, la baja y la modificación.

## **b) Lenguaje de definición y manipulación de datos.**

Algunas veces, estos paquetes realizan tanto las funciones de diccionario como las de directorio, denominándose entonces diccionario/directorio de datos (DD/D).

En la figura 13, Se presentan dos posibles arquitecturas para los SI que posean SGBD, diccionario y directorio. En la primera, el diccionario y el directorio son dos elementos distintos sin ninguna interfaz directa.

El usuario puede alimentar el diccionario con datos sobre su sistema (sobre todo con fines de documentación), pero tiene, además, que indicar al SGBD los elementos a crear en la base a través de un LDD (lenguaje de definición de datos), almacenándose estas definiciones en el directorio. Este tipo de diccionarios que no tienen una relación directa con el SGBD se denomina diccionarios pasivos. Ref:[4, P.111]

En la segunda arquitectura, el diccionario y el directorio están integrados en un DD/D, que sirve tanto a los usuarios como al propio SGBD, por lo que se trata de un diccionario activo.

Tabla No.12 Jerarquía Lógica DL/1 para el esquema externo.

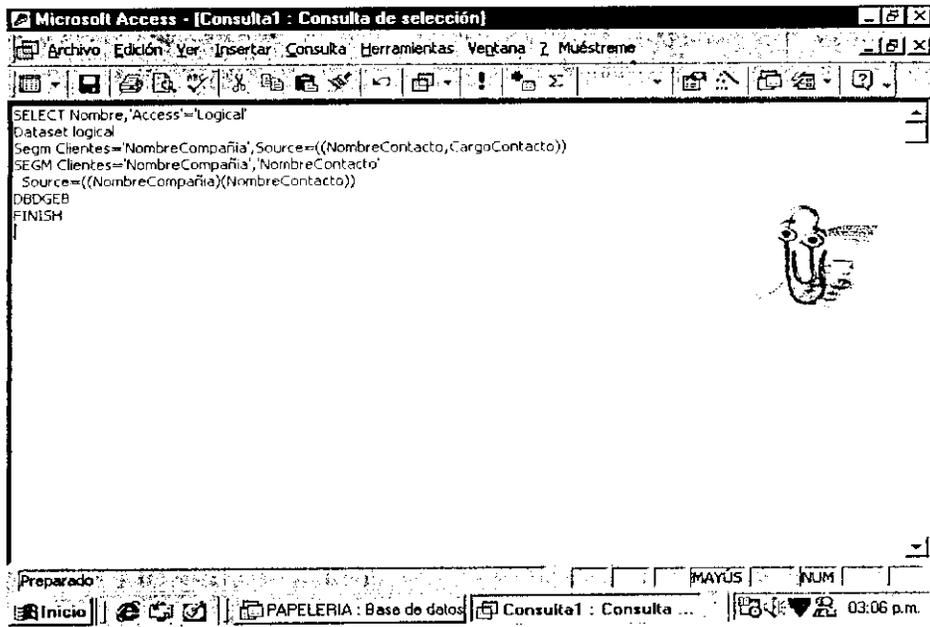
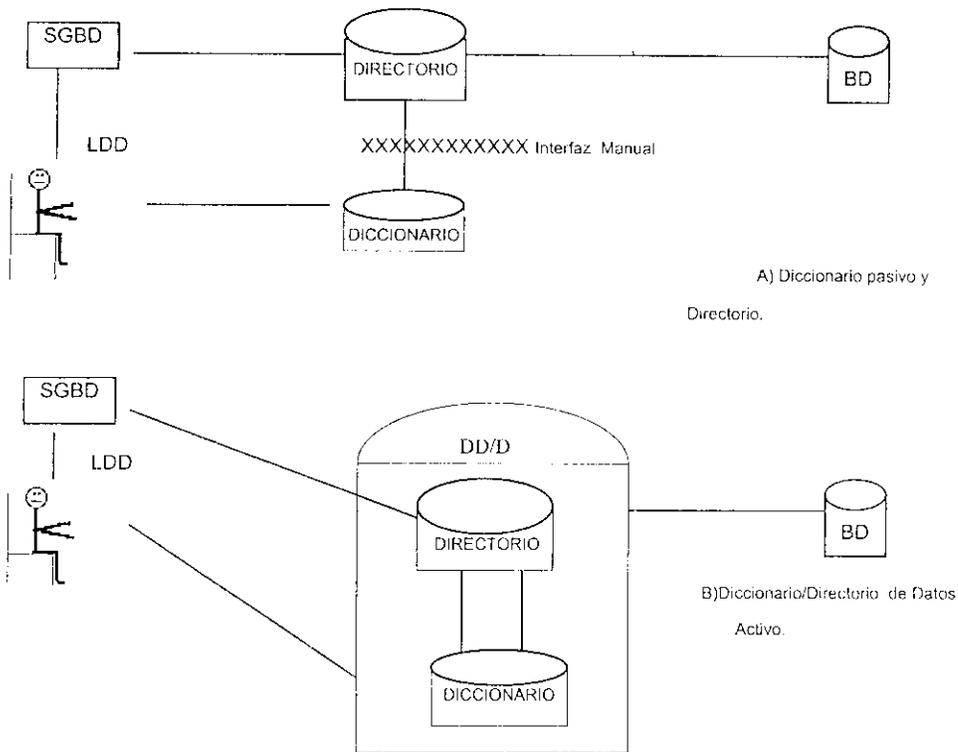


Figura No. 13. Posibles arquitectura de los SI apoyados en diccionarios y directorios.

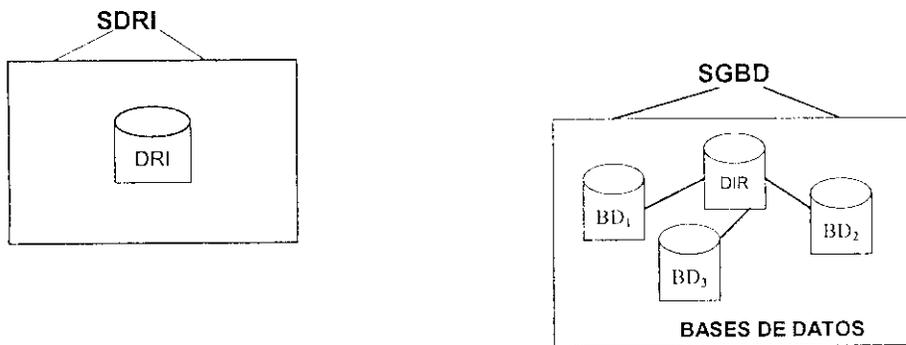


Por otro lado, la creciente complejidad de la concepción y diseño de los Sistemas de Información ha llevado a la construcción y comercialización de herramientas conocidas con el nombre genérico de CASE (Computer Aided Software / System Engineering), que contienen un diccionario llamado enciclopedia o repositorio, donde se almacenan los datos generados durante el ciclo de vida de un Sistema de Información (SI): Esquemas, grafos, matrices, información relativa a la gestión de proyectos, gestión de configuraciones, etc. Aunque este repositorio no suele ser activo, en algunos casos la herramienta CASE facilita instrumentos para cargar directamente en los catálogos propios de los SGBD más extendidos las descripciones de los datos obtenidas en la etapa de diseño.

Recientemente, ha aparecido un nuevo concepto, el de diccionario de recursos de información (DRI), que pretende ser el eslabón final en la evolución de los almacenes de datos. El DRI constituye el depósito integrado de todos los datos sobre la organización, automatizados o no, que son utilizados para efectuar las labores de planificación, control operación que permitan a la empresa cumplir sus objetivos. Los DRI engloban de algún modo las capacidades y funciones de todos los almacenes de datos anteriores.

En lo que se refiere al concepto de catálogo de un sistema de gestión de base de datos (SGBD), que surge con los SGBD relacionales, hay que destacar que se podría considerar como un diccionario/directorio de datos, pero de bajo contenido semántico. Un aspecto importante del mismo es que mientras en la mayoría de los casos el SDRI es un sistema de gestión propio del DRI e independiente del SGBD (véase Figura 13), en el caso de los catálogos el sistema de gestión es el propio SGBD, y el DRI, una base de datos más, por lo que se denomina a veces metabase.

Figura No.14. SDRI y SGBD independientes.



Para Codd (1990), uno de los requisitos para un SGBD pueda considerarse verdaderamente relacional es que "soporte un catálogo dinámico en línea en el que la descripción de la base de datos se represente como cualquier otro dato, permitiendo a los usuarios autorizados aplicar el mismo lenguaje relacional tanto a la descripción de la base de datos como a los datos regulares. Ref:[6, P.112]

## D) INTRODUCCION AL DISEÑO DE BASE DE DATOS

### a) Modelo de Red

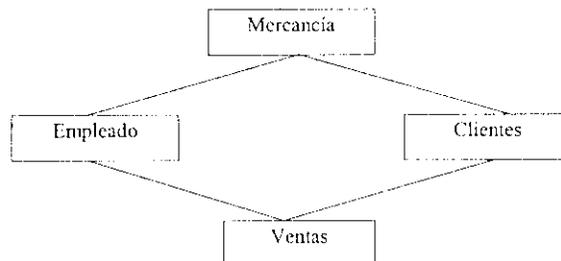
En el modelo jerárquico como en el de red el usuario percibe a los datos como un conjunto de registros con relaciones 1:M.

El modelo de red si permite que un registro tenga más de un padre.

En el modelo de Red las relaciones son llamadas conjuntos y cada conjunto esta compuesto de al menos dos tipos de registros; un registro propietario (padre en el modelo jerárquico) y uno o más registros miembros (hijos en el modelo jerárquico).

La característica diferente del modelo de red, con respecto al jerárquico, es que un miembro puede pertenecer a varios conjuntos, es decir, puede tener varios propietarios. Ref:[7,P.1 a 17]

Figura No.15. Estructura de Red.



### b) Modelo Jerárquico

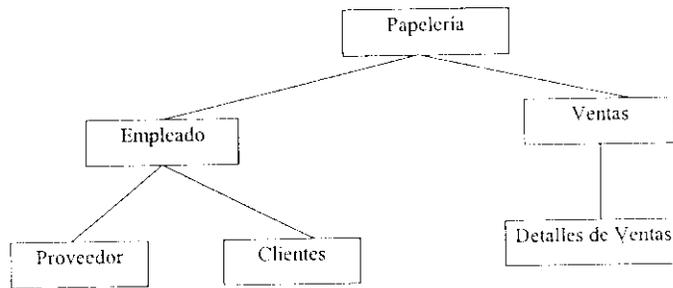
Tiene su origen en la estructura jerárquica de muchas organizaciones y en consecuencia en sus datos.

Las estructuras de datos jerárquicas son representadas por medio de árboles.

Los nodos de estos arboles representan un tipo especial de entidad (con todas sus características) y son llamados segmentos.

Los segmentos que conforman la estructura se relacionan entre si por medio de relaciones 1: M (padre - hijo), es decir, el nodo padre puede tener muchos hijos pero el nodo hijo un nodo padre.

Figura No.16. Estructura Jerárquica



### c) Modelo Relacional

El modelo relacional surgió como una solución a los problemas derivados de la dependencia estructural mostrada por los otros dos métodos. Ref:[7, P. 1 a 17]

La idea básica de este método fue lograr una separación entre la concepción lógica de los datos y la manera en que estos están almacenados físicamente.

Las bases de datos relacionales son un grupo de tablas donde los datos son almacenados.

Estas tablas, también llamadas relaciones, almacenan datos referentes a un tipo específico de entidad y están constituidas por filas y columnas.

Cada fila representa una instancia de una entidad y cada columna un atributo diferente de estas entidades.

El modelo relacional elimina los problemas del sistema de archivos al almacenar todos sus datos e incluso metadatos en un solo tipo de estructura.

Mejora las características de los modelos jerárquicos y de red al obtener tanto independencia de datos como estructural.

Se logra separar la estructura lógica de los datos de su almacenamiento físico. En general el modelo relacional consiste en un conjunto de tablas, donde cada una de ellas representa un tipo de entidad, y describe cada uno de los elementos de ésta.

## **E) ADMINISTRACION DE BASE DE DATOS**

El administrador de la base de datos (ABD) (también conocido por las siglas inglesas DBA, Data Base Administrator) es el responsable del diseño, control y administración de la base de datos. Se trata en realidad de una función (por lo que sería más correcto decir "administración de la base de datos") que puede ser desempeñada por una persona o por un grupo de personas dependiendo de la envergadura del proyecto. Ref:[4, P.110]

La función de administración ha sido reconocida por los diferentes grupos de estandarización. Por ejemplo, Codasyl creó en su seno un grupo de trabajo especial para tratar los temas relacionados con esta función, a la que llamó (Data Base Administration Working Group), que se ocupó, entre otros importantes temas, de especificar los lenguajes de descripción de los datos, de hacer un estudio sobre análisis y evaluación del SISTEMA DE GESTIÓN DE BASE DE DATOS (SGBD), etc.

También el grupo ANSI/X3/SPARC se refiere a esta función, proponiendo tres niveles distintos de administradores:

- Administrador de la empresa: Se encarga del diseño conceptual y lógico de la base de datos.
- Administrador de la base de datos: Es el responsable de las funciones de diseño físico, mantenimiento, seguridad, etc., y de la labor de ajuste o afinamiento (tuning) a efectos de eficiencia es más correcta la denominación de "Administración de la base de datos física").
- Administrador de aplicaciones: Tiene a su cargo la creación de las vistas o esquemas externos que necesitan los programadores de aplicaciones para escribir los programas de consulta y actualización de la base de datos.

La responsabilidad del administrador de base de datos son las siguientes:

- La estructura de la base de datos, en el sentido de determinar qué información va ser necesario almacenar en la misma, después de haber analizado los requisitos de los distintos usuarios (en algunos casos esta función se considera como de Administración de datos, es decir, del contenido, y se separa del resto de las funciones de administración más relacionadas con aspectos informáticos).
- La descripción conceptual y lógica de la base de datos. Una vez especificados los requisitos de la información, es preciso realizar el diseño conceptual de la base de datos, para después adecuar la estructura conceptual a un SGBD específico y, por tanto, también a un modelo convencional de datos concreto.
- La descripción física de la base de datos, para encontrar una estructura interna que soporte el esquema lógico y los objetivos de diseño con la máxima eficiencia de los recursos de máquina. Se trata de una labor que se extiende a lo largo de la vida de la base de datos, ya que este ajuste nunca puede darse por finalizado, y en tanto exista la base de datos el Administrador de la Base de Datos tendrá que variar parámetros, reorganizar los datos, modificar estructuras de almacenamiento, realizar nuevas distribuciones de los ficheros en los soportes, etc.
- Las especificaciones y vistas (o subesquemas) para los programas. Del estudio de los requisitos de los usuarios se obtendrá un conjunto de necesidades en cuanto a procesos a realizar sobre los datos. Con esta información será preciso definir las vistas externas y la normativa necesarias para los programas de aplicación. Aunque la función de programación de las aplicaciones no es en absoluto de la competencia del administrador, si lo es, sin embargo, dictar las normas y especificaciones que tienen que cumplir las aplicaciones en su acceso a la base de datos.

El administrador interviene en todas las etapas del ciclo de vida de una base de datos, teniendo en todas ellas un papel fundamental, que a veces no se reconoce en toda su amplitud. Ref:[4, P.111]

Para cumplir estas funciones, el Administrador de la Base de Datos tendrá que interactuar con todo el personal de la organización, y tiene las siguientes interfaces:

- Usuarios en general
- Directivos
- Analistas y programadores
- Operadores
- Suministradores
- Personal de formación

El administrador de base de datos ocupa un puesto más técnico y es el encargado del diseño lógico y físico de la base, él ajusta y la optimización de la misma. Ref: [4, P.112]

#### Ejemplo No. 17

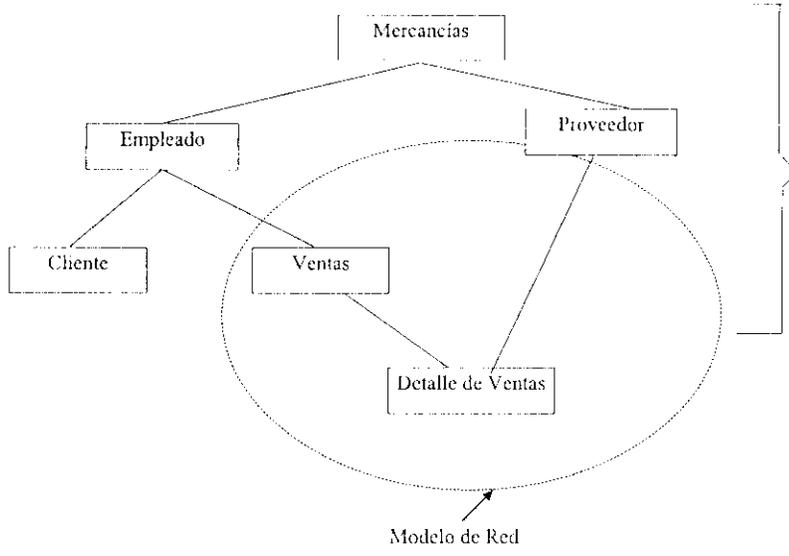
De acuerdo a lo antes, mencionado del modelo de red, jerárquico y relacional, la forma de diseñar una base de datos, en una papelería, la cual se divide en varias áreas y que estos a su vez tienen empleados, los cuales tienen que atender a clientes, y los cuales compran mercancías según sus necesidades, las cuales son registradas como ventas, y en ocasiones es necesario para la papelería contar más a detalle las ventas que a diario se realiza y las cuales son importantes.

Cuando se agotan las mercancías el dueño o empleado se ve en la necesidad de llamar al proveedor, el cual es encargado de surtir las mercancías.

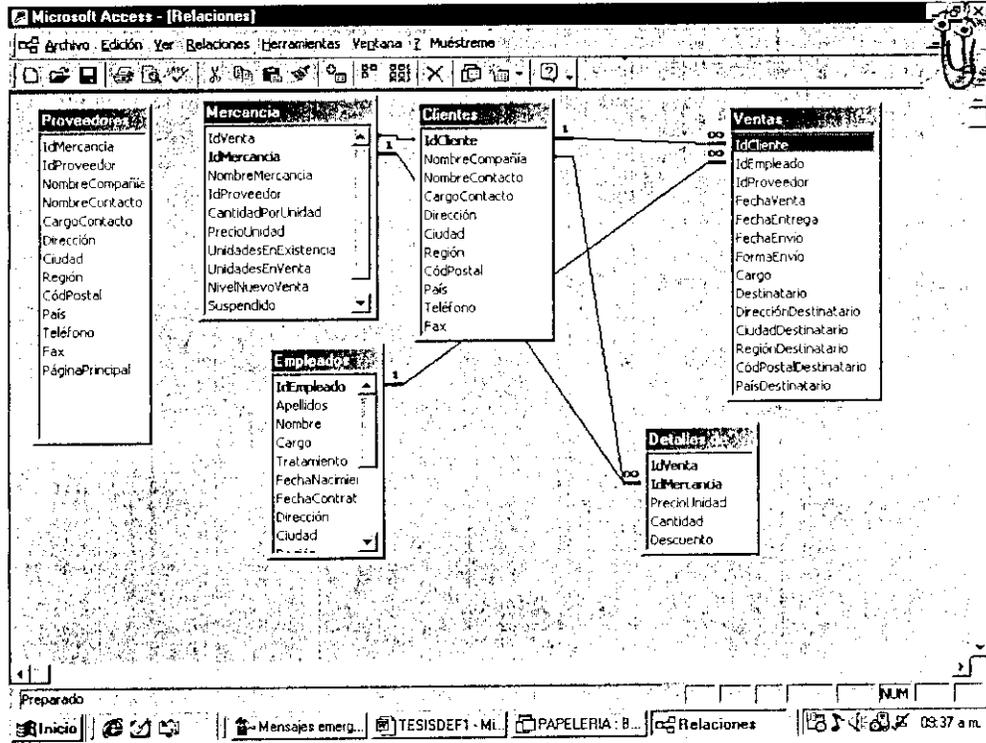
Pues, bien tomando en cuenta que contamos con las entidades siguientes:

- Cliente
- Empleado
- Proveedor
- Mercancías
- Ventas
- Detalle de ventas

Modelo Jerárquico



# Modelo Relacional



# **CAPITULO**

## **II**

## CAPITULO II

### 2. DISEÑO DE LAS BASES DE DATOS

#### 2.1 INTRODUCCIÓN

La historia de las bases de datos se extiende desde mediados de los años sesenta, y se ha caracterizado por su excepcional productividad y por su impresionante impacto económico.

A pesar de ello y de que los avances en el campo de las bases de datos han sido el fundamento de considerables progresos de la investigación básica y aplicada en diversas áreas, muchos autores [El informe Lagunita, Silberschatz (1990)], están de acuerdo en que el desarrollo y comprensión de los principios y de las técnicas de las bases de datos están todavía lejos de soportar las avanzadas aplicaciones de gestión de la información que se espera que revolucionen las economías industrializadas en los comienzos del próximo siglo. Ref. [4, P. 147].

Los científicos en computación desarrollaron teorías acerca de la mejor manera de rediseñar los datos que usados para ser conservados en archivos de papel, de manera que su manejo fuese más eficiente en la computadora. En muchos casos, simplemente usaron nuevos términos para referirse con mayor precisión a la misma clase de objetos de papel. Sin embargo, en otros casos desarrollaron nuevas maneras de manejar los datos, las cuales no hubieran sido posibles sin usar las computadoras.

El análisis de la evolución de los SGBD (Sistema de Gestión de Bases de Datos) nos va a permitir valorar los retos a los que se han de enfrentar los profesionales de hoy para conseguir una transición eficiente y no traumática de los sistemas actuales a los sistemas de la próxima generación, así como apreciar el enorme esfuerzo de investigación y desarrollo que han de realizar los científicos y suministradores para llevar a cabo los nuevos sistemas que las circunstancias imponen.

## 2.2. EVOLUCIÓN DE LA GESTIÓN DE DATOS

Cuando, en 1970, Codd propuso el modelo relacional, no podía pensar que lo que se consideraba más bien una elegante teoría matemática (un juego, según algunos) sin posibilidad de instrumentación eficiente en productos comerciales, iba a convertirse en los años ochenta, en un **señuelo** obligado en la promoción de todo SGBD.

Nadie puede discutir ahora que la aparición del modelo relacional ha marcado un hito en la historia de las bases de datos. La evolución de los sistemas de información ha tenido una considerable repercusión en la gestión de los datos, al exigirse cada vez mayores prestaciones de la información almacenada en el sistema. En realidad, el mismo término de gestión de datos no aparece hasta épocas relativamente recientes, cuando los diseñadores de sistemas de información comprendieron la importancia fundamental que una adecuada gestión de los datos tiene para un desarrollo coherente de los sistemas.

Poco a poco, el centro de gravedad de la informática, que estaba situado en el proceso, se desplazó hacia la estructuración de los datos, siendo actualmente los aspectos relacionados con este tema son un eje fundamental alrededor del cual gira una gran parte del conjunto de problemas con los que se enfrenta todo diseñador de un sistema de información. Ref. [Codd, 1970, Ídem,P.148].

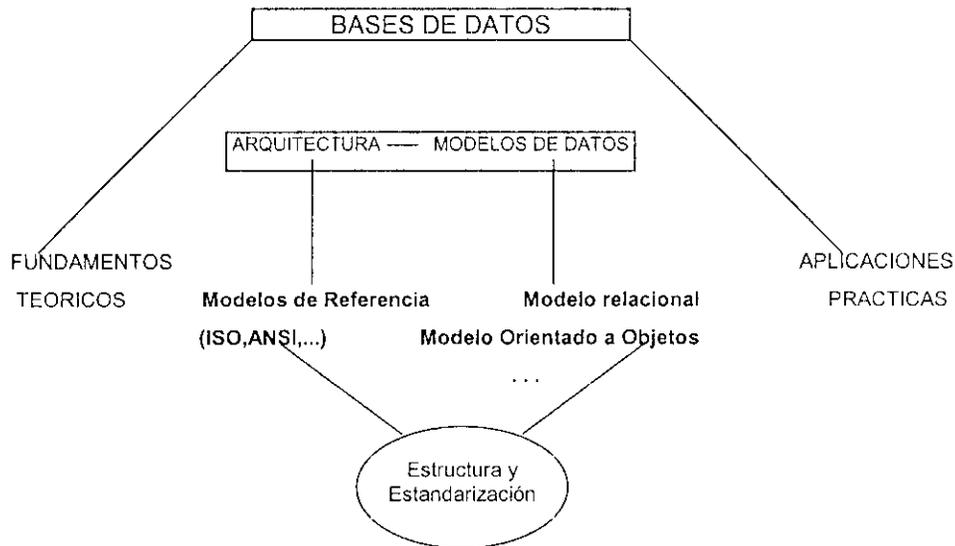
Se cambia, por tanto, de sistemas orientados hacia el proceso a sistemas orientados hacia los datos, donde estos adquieren el protagonismo, pasando desde el plano más bien oscuro y difuso en que estaban situados a ocupar un lugar privilegiado en el interés de todo informático.

En los últimos años venimos asistiendo a un avance espectacular en la tecnología de las bases de datos. Temas que hasta hace poco tiempo parecían exclusivos de laboratorios y centros de investigación, comienzan a aparecer en las últimas versiones de algunos SGBD comerciales, bases de datos distribuidas, orientación a objetos, bases de datos multimedia, arquitecturas cliente/ servidor, etc. Todo ello en el marco de la conectividad y de los sistemas abiertos.

Hito: Avance, Evolución.

La evolución de las bases de datos se ha basado en dos pilares: Por un lado, se han ido desarrollando aplicaciones prácticas en las que se busca una mayor eficiencia, facilidad de uso, seguridad, en definitiva, una mejor utilización de la tecnología a fin de responder a las exigencias de los usuarios. Por otro lado, se ha profundizado en los fundamentos teóricos de las bases de datos, que son imprescindibles para el desarrollo de sistemas cada vez más robustos, inteligentes y eficientes.

Todos estos avances, tanto en el campo teórico como en el práctico, se apoyan en dos elementos clave de los SGBD, la arquitectura y los modelos de datos, elementos fuertemente interrelacionados entre sí (Ver figura No. 18).



Cood no sólo propuso una nueva definición del modelo relacional, si no que también proponía un conjunto de doce reglas "para determinar cuán relacional es un producto de SGBD(Sistema de Manejo de Base de Datos)" con la implicación de que:

- a) Cumpliendo con esas reglas es factible en general en el aspecto técnico.
- b) El usuario obtendría claros beneficios prácticos si el sistema se ajustará en efecto a esas regla.

Así pues, se dice que un SGBD era totalmente relacional (en la acepción de mediados de la década de los ochenta, lo cual podemos ahora enmendar como de finales de dicha década), sólo se cumple con las doce reglas y poseía las "nueve características estructurales, tres de integridad y dieciocho manipulativas" del modelo relacional.

Las doce reglas propuestas por Codd eran las siguientes:

1. La regla de información
2. La regla de acceso garantizado
3. El manejo sistemático de los valores nulos
4. Un catálogo activo en línea basado en el modelo relacional
5. La regla del sublenguaje de datos completo
6. La regla de actualización de vistas
7. Inserción, modificación y eliminación de alto nivel
8. Independencia física de los datos
9. Independencia lógica de los datos
10. Independencia de la integridad
11. Independencia de la distribución
12. La regla de no subversión

Todas estas reglas se derivan de una sola regla fundamental, la "regla cero", la cual puede expresarse en la siguiente forma:

Para que un sistema se califique como sistema Relacional de Administración, de Bases de Datos, debe utilizar sus recursos Relacionales (exclusivamente) para Administrar la Base de Datos.

Dicho de otro modo, a finales de la década de los ochenta no bastaba con ser relacional según la definición de principios de la década; no es suficiente tomar un sistema de naturaleza no relacional y añadir una interfaz que maneje (sólo) restricciones, proyecciones y reuniones, para entonces aseverar que el sistema ya es relacional. Por el contrario, todo debe hacerse de manera relacional: debe existir un mecanismo relacional de vistas, un catálogo relacional, operaciones relacionales de actualización (además de recuperación), etc. Ref[ Idem,P.384].

Enseguida explicamos en forma somera las doce reglas subsidiarias.

**1. La regla de información.**

La regla de información requiere tan sólo que toda la información se presente en la base de datos de una manera y sólo una, a saber, mediante valores en posiciones de columna dentro de filas de tablas.

**2. La regla de acceso garantizado**

Esta regla no es en esencia más que otra forma de expresar el requerimiento fundamental de las claves primarias. Dice que debe ser posible obtener la dirección de cada valor escalar individual en la base de datos especificando el nombre de la tabla que lo contiene, el nombre de la columna que lo contiene, y el valor de la clave primaria de la fila que lo contiene.

**3. El manejo sistemático de los valores nulos.**

El SGBD debe manejar una representación de " la información faltante y la información no aplicable" que sea sistemática, distinta de todos los valores normales (por ejemplo, "distinta de cero o cualquier otro número", en el caso de valores numéricos), e independiente del tipo de datos. Se sobreentiende que el SGBD debe manipular representaciones de manera sistemática.

**4. Un catálogo activo en línea basado en el modelo relacional**

El sistema debe manejar un catálogo relacional en línea, accesible para los usuarios autorizados a través de su lenguaje de consulta normal.

**5. La regla del sublenguaje de datos completo**

El sistema debe manejar por lo menos un lenguaje relacional que a) tenga una sintaxis lineal, b) se pueda usar tanto en forma interactiva como dentro de programas de aplicación, y c) maneje operaciones de definición de datos (incluyendo definición de vistas), de manipulación de datos (actualización y recuperación), restricciones de seguridad e integridad, y operaciones de administración de transacciones (iniciar, comprometer y retroceder).

**6. La regla de actualización de vistas**

El sistema debe poder actualizar todas las vistas que son actualizables en teoría.

**7. Inserción, modificación y eliminación de alto nivel**

El sistema debe manejar operadores de inserción, modificación y eliminación de "todo un conjunto a la vez".

**8. Independencia física de los datos**

No requiere explicación.

**9. Independencia lógica de los datos**

No requiere explicación.

**10. Independencia de la integridad**

Las restricciones de integridad deben especificarse por separado respecto a los programas de aplicación y almacenarse en el catálogo. Debe ser posible alterar esas restricciones cuando sea apropiado sin afectar innecesariamente las aplicaciones ya existentes.

**11. Independencia de la integridad**

Las aplicaciones ya existentes deben seguir funcionando con éxito a) cuando se introduzca por primera vez una versión distribuida del SGBD; b) cuando los datos distribuidos ya existentes se redistribuyan en el sistema.

**12. La regla de no subversión**

Si el sistema ofrece una interfaz de bajo nivel (registro por registro), esa interfaz no podrá ser utilizada para subvertir el sistema pasando por alto (por ejemplo) una restricción relacional de seguridad o de integridad.

Sin duda es posible criticar estas reglas en varios aspectos técnico; no obstante, es innegable que han tenido una influencia importante en el mercado. Desde la publicación del artículo en el cual aparecieron por primera vez, Codd ha estado revisando, ampliando, aclarando y puliendo tanto las reglas en sí como las ideas técnicas en las cuales se fundamentan, desarrollando la "versión 2" del modelo relacional, RM/V2. Codd llama ahora al modelo original la "versión 1". En tanto que la versión 1 comprendía tres partes - estructura, integridad y manipulación la versión 2 tiene 18 (en las cuales se incluyen todavía las tres originales, desde luego). Las nuevas se ocupan de la asignación de nombres, los tipos de datos, las funciones, los calificadores e indicadores, el catálogo, la autorización, etc. Así pues, mientras que la versión 1 era un plano abstracto de un aspecto particular de los sistemas de bases de datos en esencia el aspecto del lenguaje para el usuario, la versión 2 es un plano abstracto de todo el sistema.

### **2.2.1 AVANCES EN LAS BASES DE DATOS.**

Respecto a la arquitectura, las propuestas de ANSI( Instituto Standard Americano Nacional) e ISO (Sistema de Información Operativa) en sus modelos de referencia han influido y están influyendo positivamente, no sólo en las investigaciones teóricas, sino también en las aplicaciones prácticas. Ref[ ídem, P.149,150].

En muchas de las normas de ISO (algunas ya aprobadas y otras en fase de elaboración) subyacen dos principios de los modelos de referencia, la arquitectura a tres niveles y la descripción recursiva de los datos. La arquitectura a tres niveles. Es aquella que existen en el mercado varios paquetes de SGBD con diferentes arquitecturas, además de las nuevas posibilidades que aparecen año con año. De acuerdo con ANSI/X3/SPARC (Standard Planning and Requirements Commite of the American National Standards Institute on Computers and Information Processing:Comite de Planeación y Requerimientos del Instituto Nacional Estadounidense de Estándares en Computación y Procesamiento de la Información), la arquitectura de una base de datos tiene tres niveles: interno, conceptual y externo, los cuales cada uno corresponde a un punto de vista distinto de la base de datos. Ref.[14, P.136,137].

La descripción recursiva de los datos. Durante estos últimos años se han realizado muchas investigaciones sobre técnicas para llevar a la práctica esa recursión; en efecto, prácticamente todas las conferencias sobre bases de datos desde 1986 han incluido uno o más artículos como ["Alpha: An Extension of Relational Algebra to Express a Class of Recursive Queries"IEEE Transactions on Software Engineering 14,num.7 (Julio de 1988)] sobre el tema.Como el manejo de consultas recursivas es un problema que por lo regular no se presenta en los SGBD clásicos.

En lo que se refiere a los modelos de datos, ha sido el modelo relacional el que ha marcado las líneas de investigación en los últimos veinte años y el que, como acabo de señalar, se ha asentado, por fin, en los productos comerciales actuales. Aunque están surgiendo otros modelos, como los orientados a objeto, en la opinión de Adoración Miguel/Mario Piattini los sistemas relacionales sobrevivirán todavía largos años y su impacto en las bases de datos será perdurable, ya que el esfuerzo de desarrollo invertido en estos sistemas y en las aplicaciones que se apoyan en los mismos obligará a mantenerlos, si bien adaptándolos paulatinamente a los avances tecnológicos lo cuál no implica que no se extiendan también los modelos orientados al objeto, así como que aparezcan otros modelos, pero compartiendo el mercado durante años con los sistemas relacionales.

### **2.1.1 Modelo Conceptual**

El administrador de la base de datos define el modelo conceptual (database administrator, DBA) por medio de un esquema conceptual. Este modelo representa la visión organizacional de la base que se obtiene al integrar los requerimientos de todos los usuarios en una empresa. Un esquema conceptual consta de las siguientes definiciones:

i) Definición de los datos: En el esquema se describen el tipo de datos y la longitud de campo de todos los elementos direccionables en la base. Los elementos por definir incluyen artículos elementales (atributos), totales de datos (artículos de grupo), y registros conceptuales (entidades).

ii) Relaciones entre datos: En el esquema se definen relaciones entre datos para enlazar tipos de registros relacionados para el procesamiento de archivos múltiples.

La arquitectura a tres niveles del grupo ANSI, con su esquema conceptual, ha marcado una clara línea de investigación en el campo de las bases de datos. Aun cuando en trabajos y propuestas de normalización anteriores ya se había indicado la conveniencia de separar los tres niveles de estructuras, ninguno de estos estudios había tenido un impacto semejante al del esquema conceptual de ANSI.

Una de las primeras tareas del grupo de estudio consistió en buscar una terminología común e intentar desarrollar un vocabulario consistente y comprensible. Otro trabajo que se abordó desde las primeras etapas fue el análisis de los componentes básicos de un sistema de información.

La arquitectura ANSI/SPARC está parcialmente basada en el concepto de máquinas anidadas (lo que se llama a veces tipo cebolla). El flujo de datos pasa a través de las distintas capas, que están separadas por interfaces y cuyas funciones se describen con cierto detalle en el documento.

Las múltiples interfaces, cuyo número se ha considerado excesivo, tienden a aislar los diversos componentes del sistema con vistas a conseguir el objetivo de independencia.

### **a) Análisis de datos**

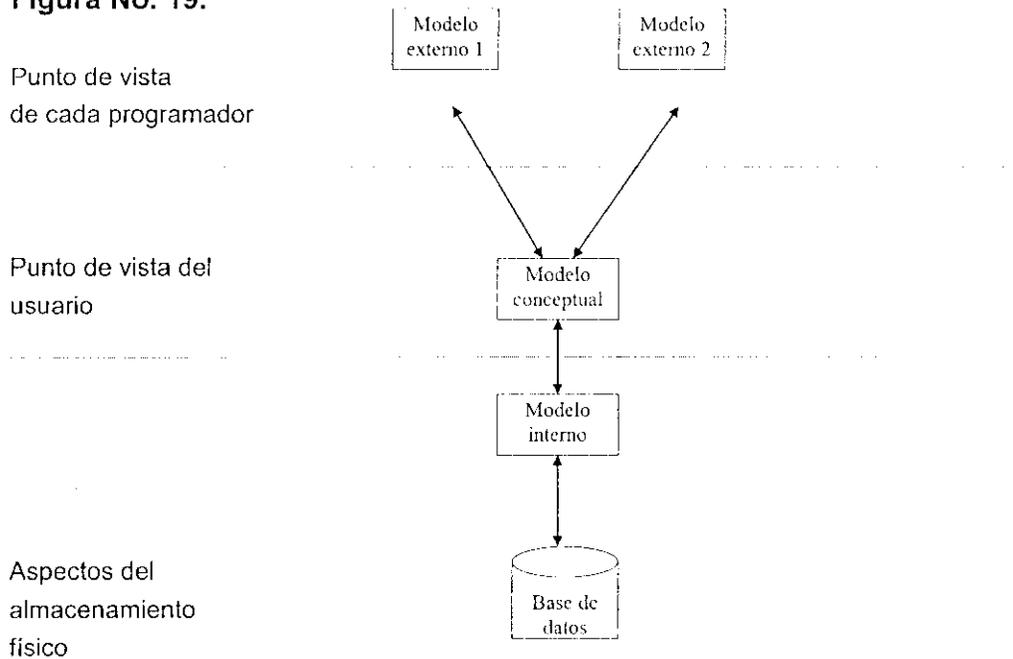
El proceso de diseño de la base de datos comienza con los requerimientos conceptuales de varios usuarios. También se pueden especificar los requerimientos conceptuales de algunas aplicaciones que no se iniciarán en un futuro cercano. Estos requerimientos de los usuarios individuales están integrados en un solo enfoque "comunitario" llamado modelo conceptual, el cual representa las entidades y sus relaciones. Nos da la capacidad de visualizar todas las entidades de datos y sus interrelaciones, sin tener que ocuparnos de su almacenamiento físico. Este modelo es inherente en toda empresa y en la comunidad.

Es posible que las relaciones entre entidades, tal como quedaron reflejadas en el modelo conceptual, no sean practicables con el paquete DBMS seleccionado. En tales casos se deberán hacer modificaciones al modelo conceptual que reflejen estas limitaciones.

Un esquema conceptual se formula sin importar el almacenamiento físico de los registros correspondientes.

En el nivel conceptual la base de datos aparece sólo como una colección de registros lógicos, sin descriptores de almacenamiento. En realidad, los archivos conceptuales no existen físicamente. La transformación de registros conceptuales en registros físicos para su almacenamiento se lleva a cabo por el sistema y es enteramente transparente para el usuario.

**Figura No. 19.**



Tres puntos de vista del almacenamiento físico, la visión del usuario y la de cada programador, están representados por un esquema interno, un esquema conceptual y un esquema externo, respectivamente.

Esquema interno: detalla cada registro almacenado y sus señaladores asociados para la recuperación sobre claves secundarias.

Esquema conceptual: Especifica todos los registros conceptuales (entidades) y su relación con la base de datos. Un registro conceptual no contiene encabezado con detalles de almacenamiento.

Esquema externo: Especifica los registros lógicos requeridos por el programa del usuario.

## b) Representación gráfica

Es bien sabido que un dibujo puede comunicar mejor que las palabras. Para describir nuestro proceso de desarrollo de un modelo conceptual, usaremos una notación para los dos elementos básicos de cualquier modelo de base de datos: la entidad y la normalización.

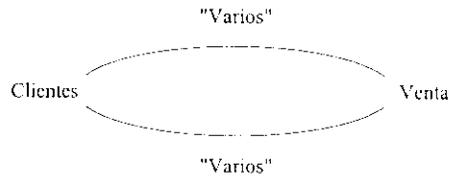
Considere el siguiente ejemplo:

En una papelería, un cliente puede comprar muchas mercancías y una sola mercancía y un cliente puede comprar muchas mercancías. Por cada mercancía comprada, el cliente recibirá una factura o nota de remisión.

Por otra parte, la información de la factura o nota, no es suficiente. Por lo cual recurrimos a hacer detalle de las ventas.

Figura No.20. Representación gráfica.

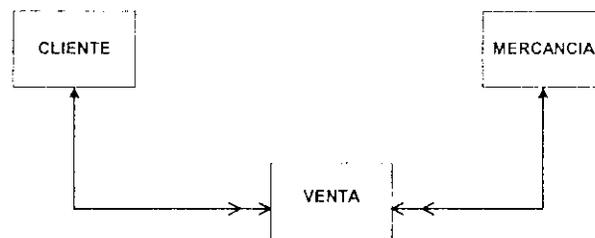
	Mercancías	LAPIZ	CUADERNO	LIBRETA	CLIPS	BORRADOR
Cristina	*	*	*	*		
Mayra			*		*	
Audina			*	*		
Rocio	*					*
Lupita	*				*	



La figura No.21. representa el modelo conceptual básico de la figura y se puede interpretar de la siguiente manera:

- Cualquier CLIENTE puede comprar cualquier cantidad de mercancías.
- Cualquier MERCANCIA lo puede comprar cualquier cliente.
- Una VENTA específica puede, y debe, estar relacionado con uno y sólo un cliente y con uno y sólo un cliente.
- La flecha de doble cabeza que une CLIENTE con VENTA representa el hecho de que un cliente pueda tener ninguno, una o varias compras, dependiendo del número de mercancías tomadas por el cliente y así realizar una venta.

Figura No.22. Relaciones entre CLIENTE, MERCANCIA y VENTA. Un cliente dado puede tomar varias ventas. Una venta determinada lo puede hacer varios clientes. Una venta específica dice más sobre el cliente y la venta.



La flecha de doble cabeza se usa para denotar este tipo de relación la que se describe como "varios".

Debemos determinar la manera de extraer las entidades, las relaciones entre los campos de datos que representan a las entidades y las relaciones entre las mismas entidades.  
Ref:[18,P.159]

## **2.1.2 Modelo Lógico**

### **a) Mapeo del diseño**

La versión del modelo conceptual que puede adaptarse al sistema de manejo de bases de datos se le conoce con el nombre de modelo lógico. A los usuarios se les entregan subconjuntos de este modelo lógico. Estos subconjuntos, llamados modelos externos (también conocidos como subesquemas), son los puntos de vista de los usuarios.

Los modelos externos son los puntos de vista que los usuarios "obtienen" basados en el modelo lógico. Los requerimientos conceptuales son los enfoques que los usuarios deseaban inicialmente y constituyen la base sobre la que se desarrolló el modelo conceptual.

#### **2.1.2.1 Cambios en aspectos lógicos**

1. En los atributos. Tanto a nivel externo como a nivel lógico global hay cambios en el nombre, tamaño, tipo, modo de cálculo (datos derivados), contraseñas, etc. Además, a nivel externo pueden producirse cambios en la presentación de los datos.

2. En las entidades (o en los agregados de datos en los sistemas que los admiten).

A nivel externo y lógico global, los siguientes: cambio en los nombres, introducción de nuevos atributos, borrado de algún atributo, alteración del orden de los atributos, división de un tipo de entidad en dos (mediante el proceso de normalización, por ejemplo), unión de dos tipos de entidades en una, contraseñas de las entidades, etc.

3. En las estructuras (interrelaciones). Puede cambiar el nombre de una interrelación o puede introducirse (o desaparecer) en una interrelación un nuevo tipo de entidad; cambios en las cardinalidades de una interrelación, aparecer o desaparecer interrelaciones, incluso una estructura jerárquica puede convertirse en red; cambios en las contraseñas de las interrelaciones, etc.

## 2.1.3 Modelo Físico

### a) Interfases entre usuario y Base de Datos

El modelo lógico se transfiere a un almacenamiento físico, como un diseño, una cinta o un tambor.

El modelo físico, que toma en cuenta la distribución de los datos, los métodos de acceso y las técnicas de clasificación, se llama modelo externo.

Los modelos externos no deben ser afectados por cambios en el almacenamiento físico o en el método de acceso a la base de datos. Esta es la primera etapa de la independencia de los datos.

Por otro lado, el modelo conceptual se diseña de tal manera que refleje las necesidades de expansión futuras, los cambios en el modelo conceptual no deben afectar tampoco a los modelos externos.

Esta es la segunda etapa de la independencia de los datos. Algo que debemos recordar aquí, es que el modelo lógico está forzado a ajustarse al DBMS; por lo tanto, cambiará si el sistema de manejo de bases de datos lo hace.

Existen dos etapas de la independencia de los datos:

1. Independencia lógica.
2. Independencia física.

- Modelo conceptual. Los requerimientos conceptuales de los usuarios individuales están integrados en un solo enfoque "comunitario" llamado modelo "conceptual".
- Modelo lógico. La versión del modelo conceptual que se adapta al sistema de manejo de la base de datos se llama modelo "lógico".

### 2.1.3.1 Cambios en aspectos físicos

1. En el nivel interno. Podemos encontrar cambios en el tamaño de los bloques, longitud de los registros almacenados, pasar de registros almacenados de longitud fija a longitud variable o cambiar la posición relativa de los registros almacenados.

Pueden existir cambios también en la organización física (como ISAM o VSAM), métodos de direccionamiento (como el algoritmo de aleatorización), tratamiento de desbordamientos, inserciones y eliminaciones, ubicación de los conjuntos de datos en diferentes volúmenes, etc.

Asimismo, pueden introducir o borrarse índices, reorganizarse algún volumen, cambiar la dirección relativa o absoluta del almacenamiento, cambiar o introducir técnicas de compactación, pasar a organizaciones de tipo autoadaptativo, replicar conjuntos de datos por razones de disponibilidad o seguridad, cambiar o introducir técnicas criptográficas, etc.

2. En el nivel físico (equipo). Por último, en este nivel puede prever cambios en el tipo de soporte, tamaño de los volúmenes, longitud de las pistas o tamaño de las celdas, número de pistas/cilindros, número de cilindros/paquete, codificación física, dirección de dispositivos, tipo de dispositivo (como discos removibles o fijos), sistema operativo, unidad central, distribución geográfica de los datos físicos, etc.

#### **b) Modelo interno**

Modelo interno. El modelo físico que toma en cuenta la distribución de los datos, los métodos de acceso y las técnicas de indexación se llama modelo "interno".

El modelo interno es la representación del nivel inferior de una base de datos. Mapea a la base lógica hacia el almacenamiento físico y establece trayectorias de datos (por ejemplo, mediante señaladores o índices) para el acceso aleatorio a las bases de datos.

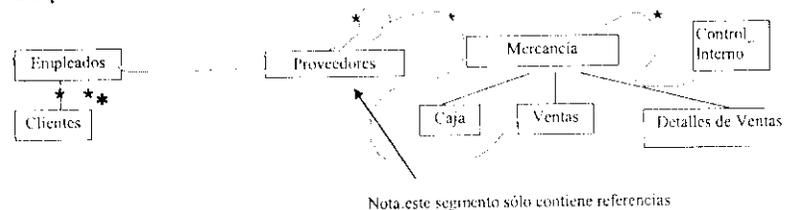
Un registro interno almacenado difiere de un registro lógico en lo que a la percepción del programador se refiere.

Un registro en una unidad física puede consistir en uno o más registros lógicos, además de los elementos descriptivos o identificadores del sistema, como son longitud del registro, tipo de registro, banderas y apuntadores. (Vale la pena recordar que el registro físico consta no de palabras o ideas sino de ceros y unos codificados electrónicamente). Por otro lado, un registro lógico es la visión del programador de un registro, el cual sólo consiste en datos, sin ningún extra del sistema.

Un modelo interno es descrito por el Sistema de Manejo de Base de Datos como un **esquema interno**. El esquema contiene especificaciones detalladas del almacenamiento de todos los registros almacenados en la base de datos así como los descriptores del sistema: señalizadores, palabras de control y trayectorias de datos necesarias para recuperación sobre claves secundarias. La base de datos física contiene tres tipos distintos de registro y sus índices asociados. Ref:[6,P.597]

Figura No. 23

### Esquema Interno



Empleado\_1    Proveedor\_2    Mercancia\_1    Ventas-2

### c) Modelo externo

El nivel externo representa la percepción individual de cada programador de la base de datos porque los programadores pueden estar trabajando en distintos subconjuntos de una base integrada.

Así como en el modelo conceptual, los programadores pueden imaginar que los archivos externos usados por sus programas en la base de datos, existen de la manera que ellos los perciben. De hecho, los archivos externos tampoco existen físicamente. Cuando un programa convoca un comando E/S, el sistema de manejo de base de datos extrae los datos requeridos por los registros lógicos externos de uno o más registros físicos de la base de datos.

Como que los programas individuales adcesan sólo subconjuntos de datos de una base ya integrada, los datos requeridos por el programa se especifican por medio de un esquema externo, habrá uno por cada programa.

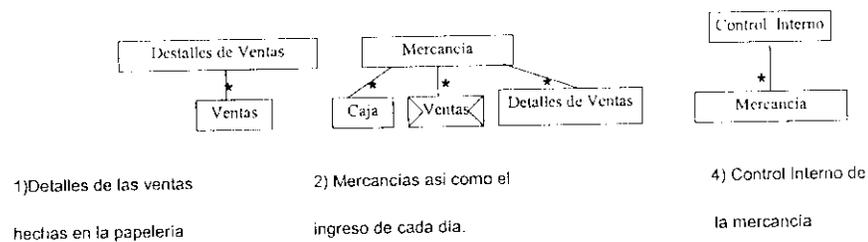
Sin embargo, algún usuario podrá tener derecho de acceso a más de un esquema externo y un esquema externo podrá ser compartido por diversos usuarios. Al construir un esquema externo para uso del programa, se deben tener en cuenta los siguientes puntos para la elección de datos por incluir:

- i) en el esquema pueden omitirse uno o más tipos de registro.
- ii) en el registro conceptual elegido pueden omitirse uno o más campos, y puede disponerse el orden relativo de los campos en un tipo de registro.
- iii) en el esquema conceptual pueden omitirse una o más relaciones entre los datos.

Un esquema externo también se llama subesquema. A nivel conceptual, el esquema representa la visión de la comunidad de usuarios de la base de datos entera. Al mismo tiempo, en el nivel externo, los datos requeridos externos). Ambos, esquema y subesquemas, están definidos con el lenguaje de definición de datos (Data Definition Language, DDL) proporcionado por el sistema de manejo de base de datos. El esquema "objeto" y los subesquemas que serán referenciados por el sistema de manejo de base de datos se crean al compilar los programas fuente DDL correspondientes. Automáticamente, el sistema de manejo de base de datos genera un esquema interno que consiste en el mapeo de la estructura lógica en almacenamiento externo; no requiere que el usuario especifique información detallada de almacenamiento. Ref:[6,P.597]

Figura No. 24

Esquema Externo que se puede desarrollar a partir del esquema interno



### **c.1) Dos pasos en el diseño de la base de datos (modelo físico)**

El modelo físico es una estructura de la base de datos que se ha de almacenar en dispositivos físicos. Puesto que un gran porcentaje de bases de datos se utilizan en un medio de línea, se tiene que estar muy relacionado con el funcionamiento del modelo físico antes de implantar la base de datos. La predicción del funcionamiento es un análisis multivariable, en el cual el principal variable es el modelo físico.

Se debería efectuar un análisis cuantitativo del modelo físico con frecuencias promedio de ocurrencias de grupos de datos, con las estimaciones esperadas de espacio auxiliar y con estimaciones esperadas de tiempo para la recuperación y el mantenimiento de los datos.

Las aplicaciones que emplean la base de datos se pueden categorizar en grupos, basándose en su importancia para la empresa, así como en el patrón de uso.

La estimación del funcionamiento en la fase de diseño debe tener un amplio apoyo de la administración, debido a que una vez que la base de datos se ha instalado, es extremadamente difícil, si no imposible, rediseñarla.

Un diseñador de bases de datos generalmente trata de optimizar el modelo físico, en cuanto a consideraciones de espacio y tiempo.

Se realizan concesiones para obtener equilibrio entre espacio y tiempo en la mayoría de los Sistemas de Manejo de Bases de Datos, es decir, se pueden eliminar varias entradas/salidas se puede salvar espacio, pero esto costaría más tiempo. Sin embargo, uno se debe dejar llevar por los requerimientos comerciales. Es posible que, desde el punto de vista comercial, sea necesario tener múltiples puntos de entrada a una base de datos o acceder un tipo de acceso, puede ser necesario invertir un archivo en esas claves, haciendo, por lo tanto, algunos gastos de espacio y/o tiempo. Si es así, ése es el precio que se tiene que pagar para satisfacer este requerimiento comercial particular.

## c.2 Diseño de un Modelo Físico de una Base de Datos

Varias de las directrices proporcionadas son "universales", es decir, son bastantes independientes del Sistema de Manejo de Base de Datos utilizado. Para ser específicos, sin embargo, consideremos los siguientes tres sistemas de manejo de bases de datos:

1. Sistema de Manejo de Información (IMS), comercializado por International Business Machines Corporation; este sistema mantiene un modelo de datos jerárquico con algunas capacidades reticulares y de archivo invertido.
2. Sistema de Manejo de Bases de Datos 10/-20 (DBMS -10/20), comercializado por Digital Equipment Corporation para el DECSYSTEM -10/-20, que emplea un modelo de datos reticular con especificaciones de CODASYL.
3. Sistema de Base de Datos Adaptable (ADABAS), comercializado (en Estados Unidos y Canadá) por Software AG of North America, que es un sistema de manejo de base de datos que emplea inversión de archivos.

### d) Sistema de manejo de base de datos (red, jerárquico, relacional).

#### Modelo relacional

El modelo relacional representa los datos y las relaciones entre los datos mediante una colección de tablas, cada una de las cuales tiene un número de columnas con nombres únicos. La figura es un ejemplo de base de datos relacional que muestra clientes y las ventas que ellos tienen. Por ejemplo, que el cliente Cristina vive en Sidehill, en Brooklyn, y tiene dos ventas, una a detalle con factura, con un saldo de 10,000 dólares, y la venta por 801, con un saldo de 10533 dólares. Ref:[7].

### **Ventajas**

1. Permite que el diseñador se olvide de las características físicas del almacenamiento de los datos y que se concentre únicamente en la estructura lógica de la base de datos.
2. Logra la independencia de datos y la estructural.
3. Tiene gran capacidad para llevar a cabo búsquedas y peticiones.

### **Desventajas**

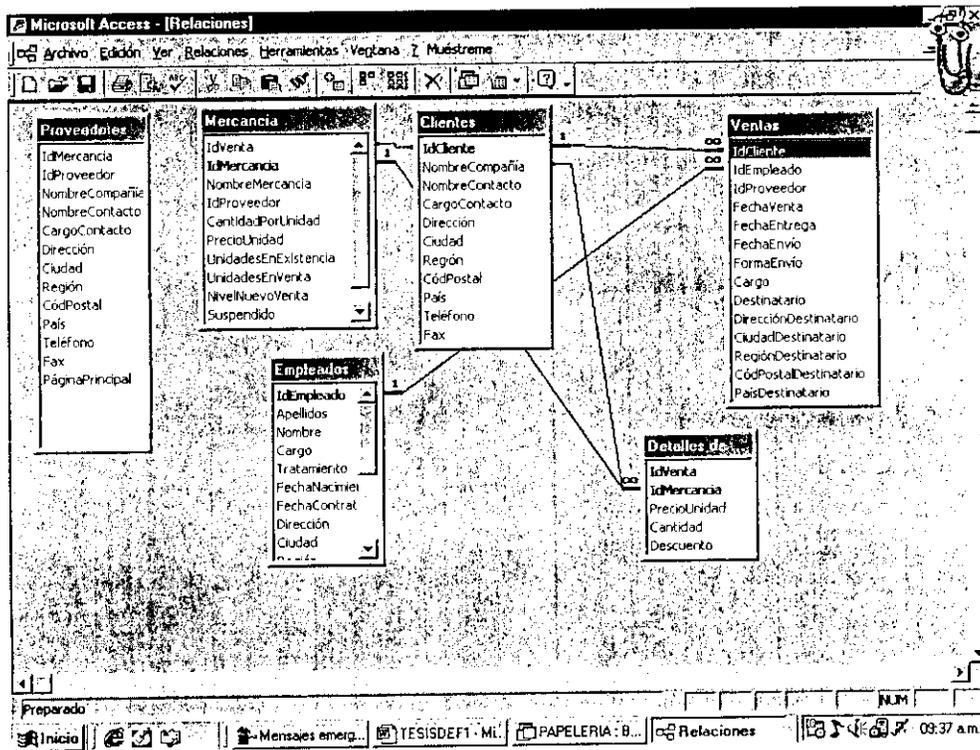
1. Relativa lentitud.
2. Mayor demanda de recursos del sistema.

### **Modelo de red**

Los datos en el modelo de red se representan mediante colecciones de registros (en el sentido de la palabra en Pascal o PL/I) y las relaciones entre los datos se representan mediante enlaces, las cuales pueden verse como punteros. Los registros en la base de datos se organizan como colecciones de grafos arbitrarios.

Tabla No. 25

Ejemplo de una base de datos relacional.



### Ventajas

1. Las relaciones M:N son más fáciles de implementar en este modelo que en el jerárquico.
2. Este modelo asegura la integridad de la base de datos, ya que para definir un miembro se tiene que definir primero a su propietario.
3. Logra la independencia de dato.

## **Desventajas**

1. Es difícil de diseñar y usar apropiadamente. Sobre todo porque se requiere de conocimiento de la estructura que utiliza.
2. Es difícil de realizar cambios en la base de datos, sobre todo los referentes a la definición de la estructura. Lo cual implica que no logra la independencia estructural.
3. En este modelo al igual que en el jerárquico solo se puede acceder un registro a la vez.

## **Modelo Jerárquico**

El modelo jerárquico es similar al modelo de red en el sentido de que los datos y las relaciones entre los datos se representan mediante registros y enlaces, respectivamente. Se diferencia del modelo de red en que los registros están organizados como colecciones de árboles en vez de grafos arbitrarios.

### **Algunas ventajas de este modelo son:**

1. La compartición de datos y su seguridad son buenas debido a que toda la información esta contenida en una estructura común. Ref:[7]
2. Mantiene la independencia de los datos.
3. Debido a que cada segmento hijo por necesidad esta ligado a un padre, este modelo incrementa la integridad de la base de datos.
4. Es aplicable a muchas instituciones, ya que la mayoría de su información es de carácter jerárquico.

## Desventajas

1. A pesar de que el modelo logra la independencia de datos, no consigue la independencia estructural.

2. Muchas relaciones comunes no pueden modelarse como relación 1:M y por lo tanto es muy difícil llevarlas satisfactoriamente al modelo jerárquico, tal es el caso de las relaciones M:N.

3. Relaciones basadas en un hijo y múltiples padres son muy complicadas de representar por este modelo.

4. El sistema no permite hacer consultas, por lo tanto es muy poco flexible desde la perspectiva del usuario final.

Figura No.26

Ejemplo de una base de datos jerárquica (Relaciones en un bosque)

Manipulación de árboles

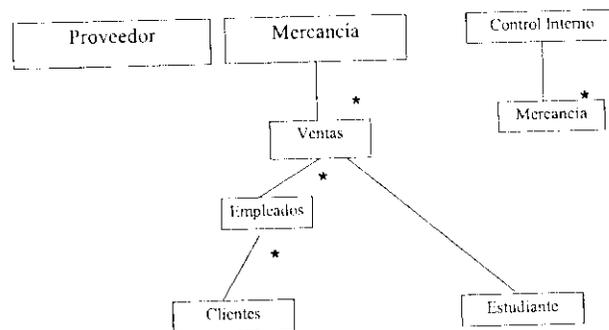


Figura No.27

Ejemplo de una base de datos de red.

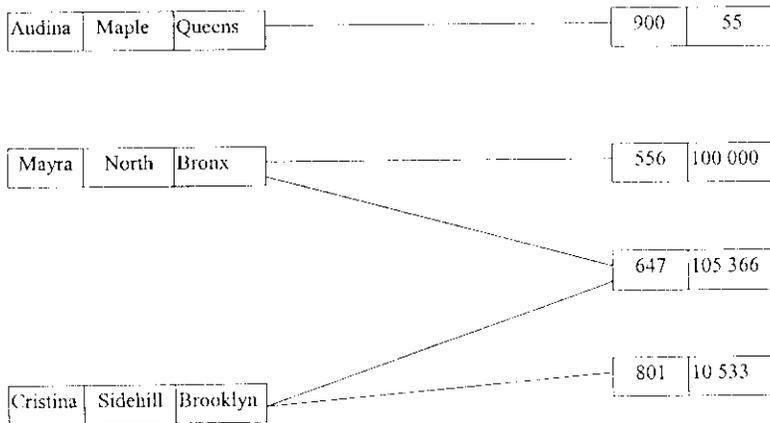
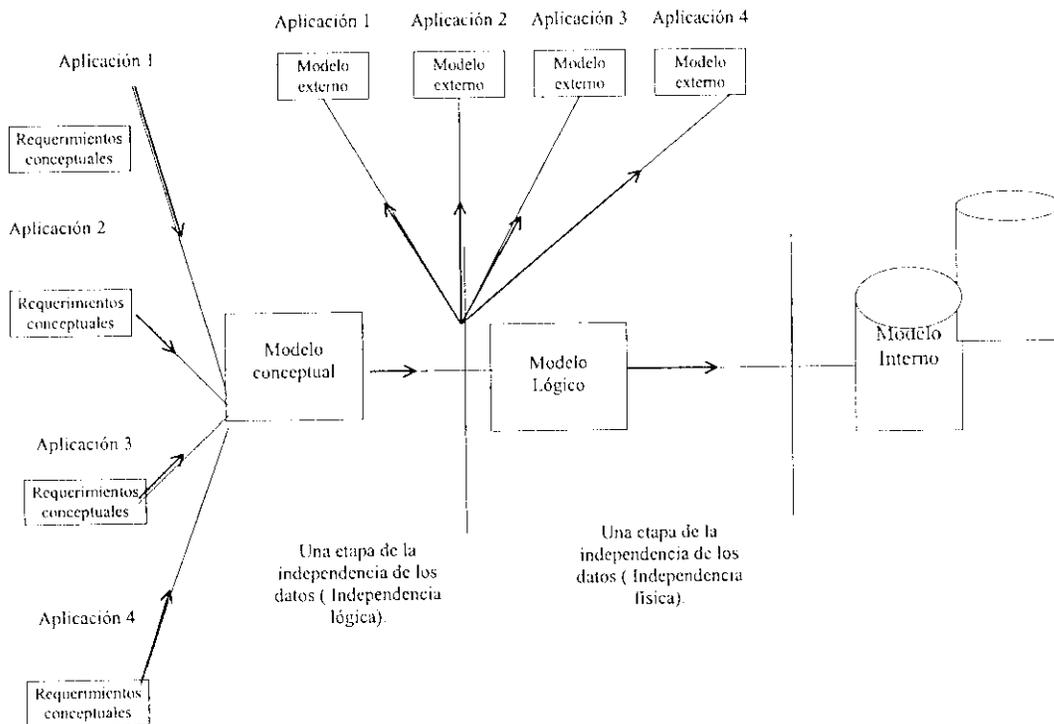


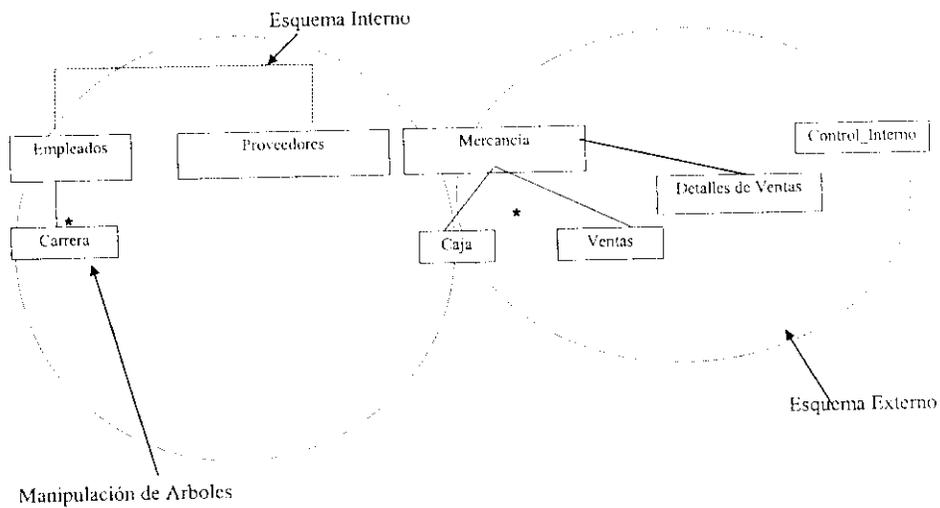
Figura No. 28. Etapas de Independencia de los datos



## Ejemplo No.29

Según el esquema interno y el esquema externo, una estructura de datos debe estar relacionada, ya sea mediante un puntero o enlaces que muestren la relación entre una entidad y otra, por lo que en este ejemplo, muestro como una Papelería que cuenta con las siguientes entidades como son:

- Empleado
- Clientes
- Mercancia
- Ventas
- Detalles de Ventas
- Proveedor
- Control Interno



# CAPITULO

## III

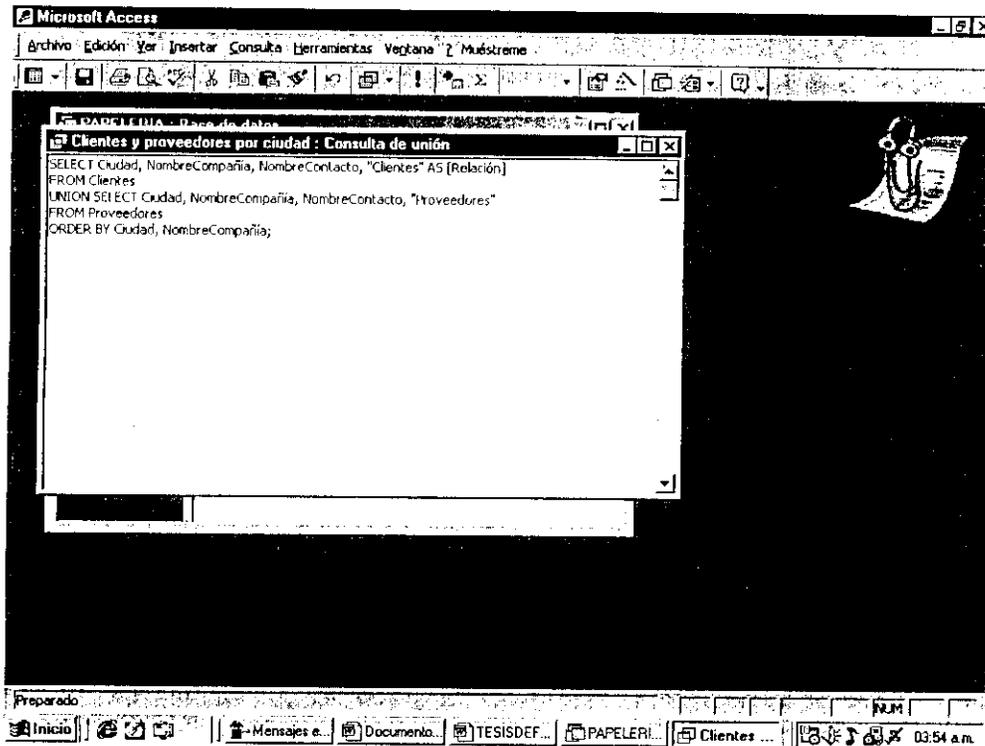
## CAPITULO III.

### 3. ASPECTOS DE DESEMPEÑO.

#### 3.1 Consultas en SQL

SQL tiene una instrucción básica para obtener información de una base de datos: la **instrucción SELECT** (seleccionar). Esta instrucción no tiene que ver con la operación SELECCIONAR del álgebra relacional.

Tabla No.30



Ref:[6,P.459]

### 3.1.1 Definición de datos en SQL

SQL es un lenguaje de modo dual. Es a la vez un lenguaje de base de datos interactivo utilizado para consultas y actualizaciones, y un lenguaje de base de datos programado utilizado por programas de aplicación para acceso a la base de datos. En su mayor parte, el lenguaje SQL es idéntico en ambos modos. La naturaleza de modo dual de SQL tiene varias ventajas: Ref: [6, P. 395]

- Es relativamente fácil para los programadores aprender a escribir programas que acceden a la base de datos.
- Las capacidades disponibles a través del lenguaje de consulta interactivo están también automáticamente disponibles para los programas de aplicación.
- Las sentencias SQL a utilizar en un programa pueden ser probada por primera vez utilizando SQL interactivo y luego codificadas en el programa.
- Los programas pueden trabajar con tablas de datos y resultados de consulta en vez de navegar a través de la base de datos.

SQL emplea los términos tabla (table), fila (row) y columna (column) en vez de relación, tupla y atributo respectivamente. Las órdenes de SQL para definir los datos son CREATE (crear), ALTER (alterar) y DROP (desechar).

### 3.1.2 SQL: Un lenguaje de bases de datos relacionales.

#### 3.1.2.1 Conceptos de esquema y catálogo en SQL2

El concepto de esquema SQL se incorporó en SQL2 con el fin de agrupar tablas y otros elementos pertenecientes a la misma aplicación de bases de datos. Un esquema SQL se identifica con un nombre de esquema, y consta de un identificador de autorización que indica el usuario o la cuenta que es propietario (a) del esquema, además de los descriptores de cada elemento del esquema. Dichos elementos comprenden tablas, vistas, dominios y otros (como las concesiones de autorización y las aserciones), que describen el esquema. Los esquemas se crean mediante la instrucción CREATE SHEMA, que puede contener todas las definiciones de los elementos del esquema.

### 3.1.3 SQL incorporado

En este enfoque, las sentencias SQL están incorporadas directamente al código fuente del programa, entremezcladas con las otras sentencias del lenguaje de programación. Se utilizan sentencias especiales de SQL incorporado para recuperar datos en el programa. Un precompilador de SQL especial acepta el código fuente combinado y, junto con otras herramientas de programación, lo convierte en un programa ejecutable. Ref:[3, P.396]

#### 3.1.3.1 Conceptos de SQL incorporado

SQL incorporado utiliza las siguientes técnicas para incorporar las sentencias SQL:

- Las sentencias SQL se entremezclan con las sentencias del lenguaje principal en el programa fuente. Este "programa fuente con SQL incorporado" se remite a un precompilador de SQL, que procesa las sentencias SQL.
- Las variables del lenguaje de programación anfitrión pueden ser referenciadas en las sentencias SQL incorporado, permitiendo que valores calculados por el programa sean utilizados por las sentencias SQL.
- Las variables del lenguaje de programación son también utilizadas por las sentencias de SQL incorporado para recibir los resultados de consultas SQL, permitiendo al programa utilizar y procesar los valores recuperados.
- Variables de programa especiales son utilizados para asignar valores NULL a columnas de la base de datos y para soportar la recuperación de valores NULL de la base de datos.
- Varias nuevas sentencias SQL que son propias del SQL incorporado se añaden al lenguaje SQL interactivo, para permitirle el procesamiento fila a fila de resultados de consulta. Ref:[3, P.399]

SQL también puede usarse junto con un lenguaje de programación de aplicación general como C, ADA, PASCAL, COBOL o PL/1. El lenguaje de programación se denomina **lenguaje anfitrión**. Cualquier instrucción de SQL de definición de datos, consulta, actualización o definición de vistas se puede incorporar en un programa escrito en el lenguaje anfitrión.

La instrucción de SQL incorporado se distingue de las instrucciones del lenguaje anfitrión anteponiéndole un carácter o una orden especial como prefijo para que el preprocesador pueda separar las instrucciones de SQL incorporadas y el código del lenguaje anfitrión. En SQL2, las palabras reservadas EXEC SQL o la secuencia & SQL (preceden a todo enunciado de SQL incorporado, y las instrucciones pueden terminar con un END-EXEC, correspondiente, un ")" o un ";". En algunas de las primeras implementaciones, las instrucciones de SQL se pasaban como parámetros en llamadas a procedimientos.

La característica fundamental de los sistemas de base de datos para computadores es su sencillez. La capacidad limitada de los computadores personales restringe tanto el tamaño de la base de datos como el grado de complejidad del sistema. Aunque casi todos los sistemas de base de datos se diseñan pensando en la facilidad de uso, la importancia de este factor en el mercado de los computadores personales es extraordinaria, ya que los usuarios no pueden contar con la ayuda de un administrador de base de datos experimentado. Cada uno de los usuarios de un sistema de base de datos de computadores sirve como administrador de base de datos. Ref:[3,P.400]

El lenguaje estructurado de consultas, o SQL, es el principal medio de comunicación con la base de datos, indicándole que información se desea seleccionar (select), insertar (insert), actualizar (update) o borrar (delete).

### 3.1.4 SQL

Al crear consultas, el mejor sitio para comenzar es la vista Diseño de consultas :

1. Abra la base de datos creada.
2. Haga clic en la ficha Consultas.
3. Haga clic en el botón Nuevo (New) ubicado en el extremo derecho de la ventana Base de datos. Se despliega el cuadro de dialogo Nueva consulta. Ver en la figura No.31.
4. Haga clic en Aceptar(OK) para continuar. Antes de entrar en la vista Diseño de consultas, aparece el cuadro de dialogo Mostrar tabla (Show Table), como se muestra. Ref:[1,P173]
5. Seleccione la tabla creada y haga clic en Agregar (Add) Access 97 agrega entonces la tabla a la vista Diseño de consultas. Haga clic en el botón Cerrar (Close). La vista Diseño de consultas despliega la tabla No.32 y No.33, en el extremo superior de la pantalla. Como se muestra. Ref:[1,P174]

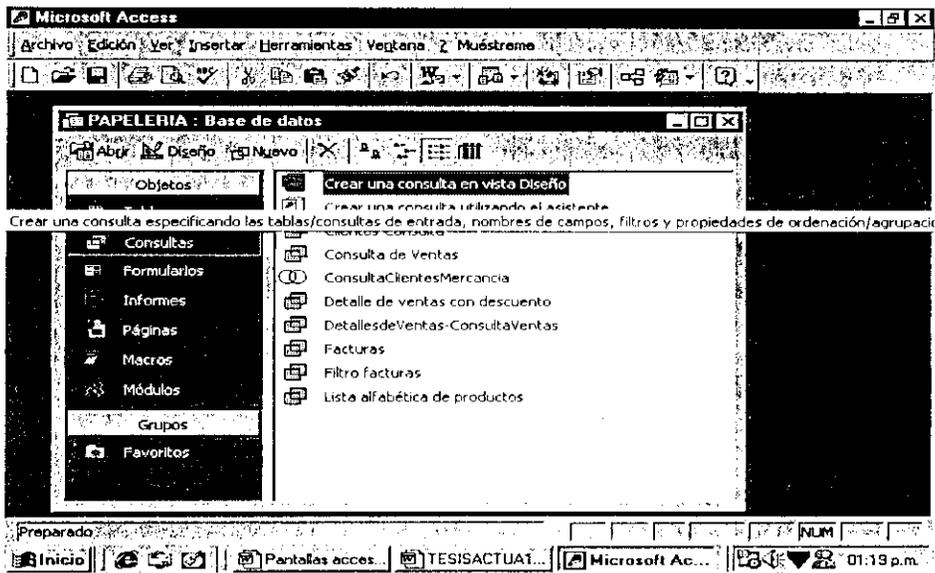


Tabla No.31

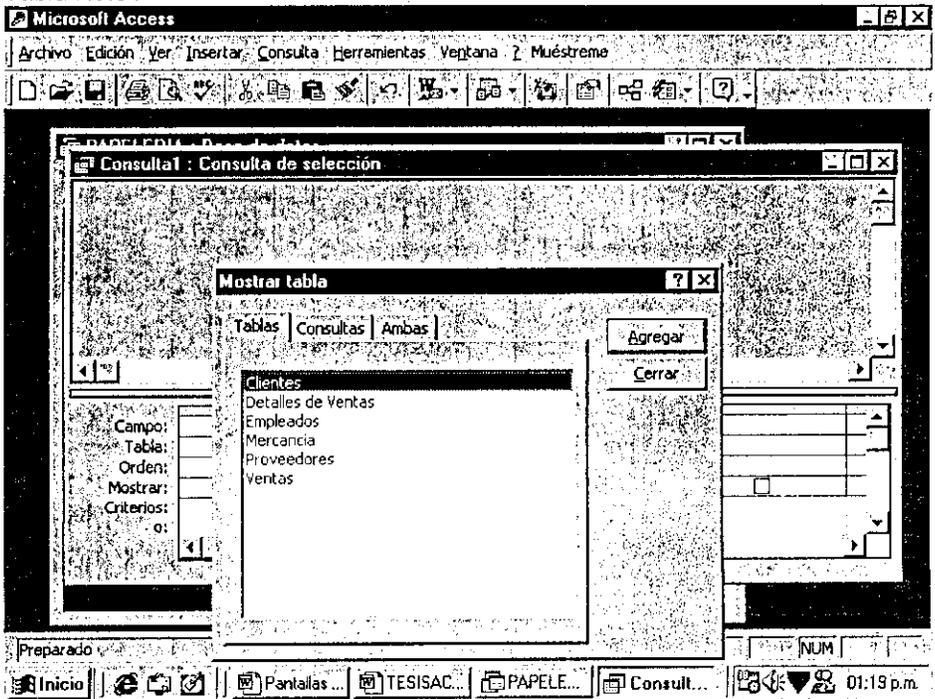


Tabla No.32

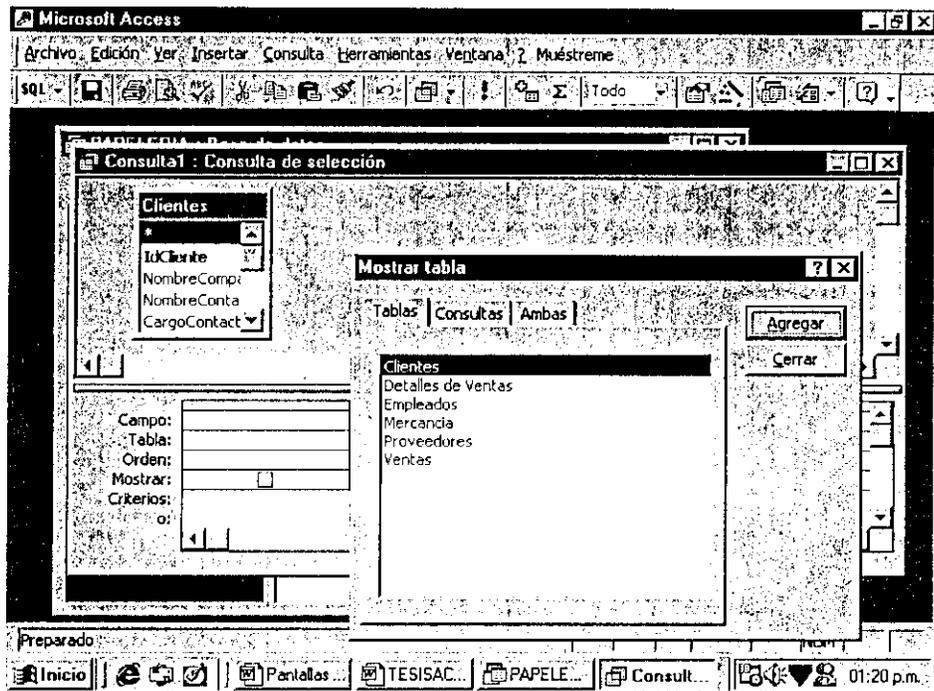


Tabla No. 33

## Estilo

En primer lugar, se van a ver algunos comentarios sobre el estilo. SQL no tiene en cuenta si los mandatos se escriben en mayúscula o minúscula. El mandato.

SeLeCt feaTURE, section, NOmbre FROM papEleRia;

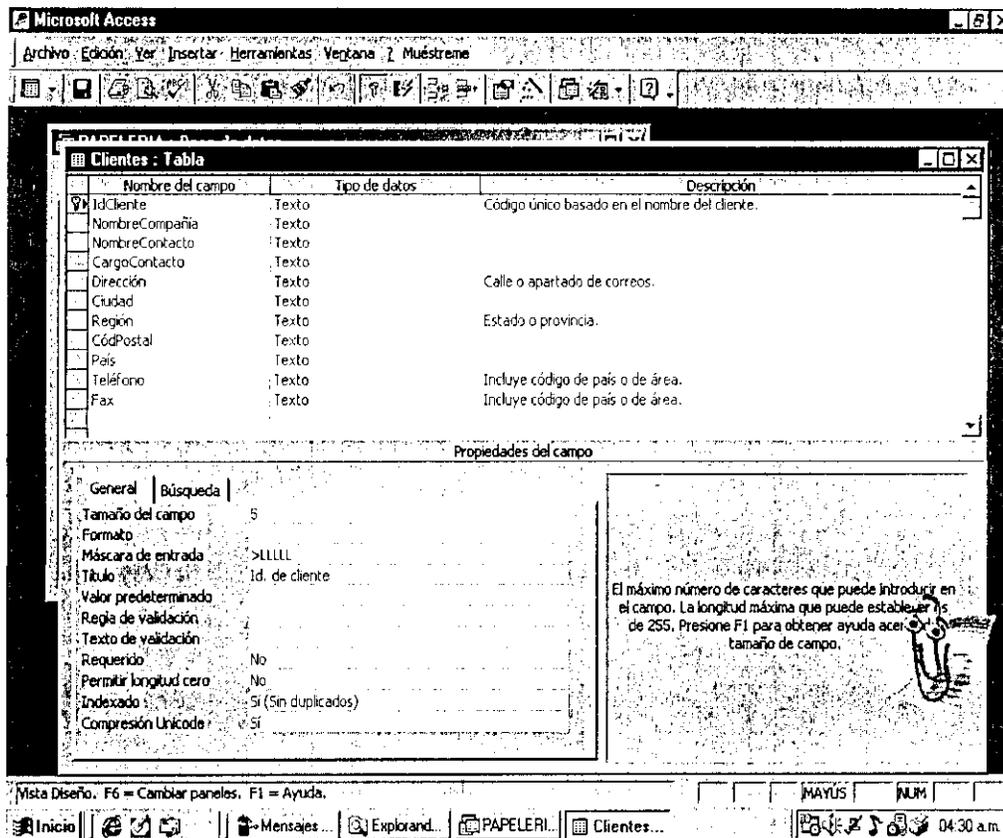
producirá exactamente los mismos resultados que:

select Feature, Section, Nombre from PAPELERIA;

## Utilizar SQL para seleccionar datos de las tablas

El ejemplo, muestra una tabla de características de una papelería. SQL mostrara lo siguiente después de escribir: Ref:[16,P.48]

Select Feature, Section, Nombre from PAPELERIA;



Ejemplo. Tabla de las secciones de una papelería.

¿Se puede observar las diferencias entre esta tabla y la de la papelería del ejemplo? Hay la misma información en la tabla de la base de datos que en la tabla pero están formateadas de maneras diferentes. Aunque la información de las filas es exactamente la misma, las cabeceras de las columnas son ligeramente diferentes. En efecto son diferentes las del ejemplo y las obtenidas con la sentencia **select**. Ref:[16,P.49]

select Feature, Section, Nombre from PAPELERIA;

Las columnas vendrán con todas las letras en mayúsculas.

Lo que se está viendo es la presentación de información que da el SQL por defecto.

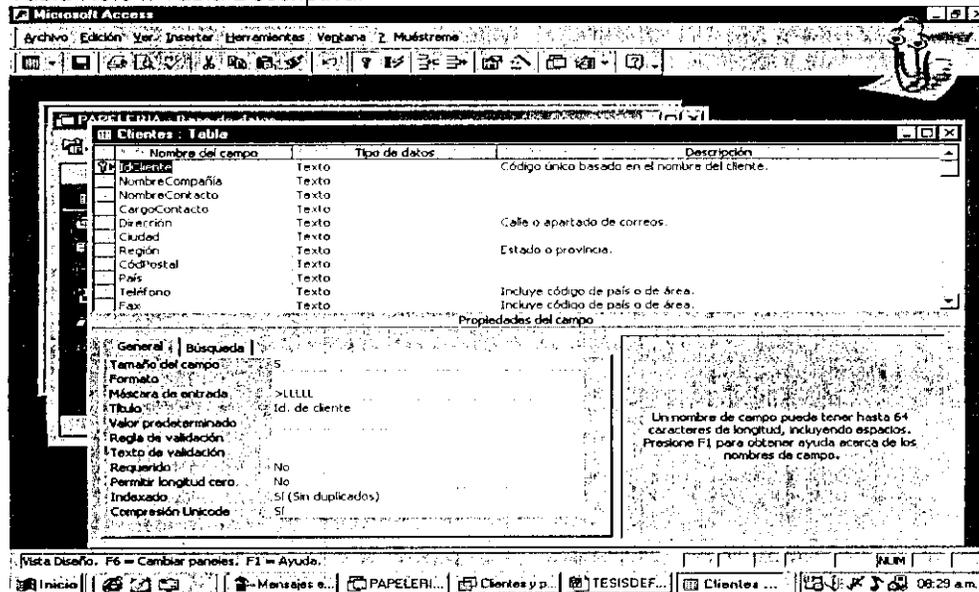
Esta se puede cambiar, y se hará, pero hasta que se explique las órdenes correspondientes, ésta es la forma en que trabaja SQL:

- Cambia todas las cabeceras de columna a mayúsculas.
- Hace las columnas tan anchas como el tamaño en que están definidas en la base de datos.
- Comprime cualquier espacio si la cabecera de columna es una función.

El primer punto es obvio. Los nombres de columna que se usan fueron cambiados a mayúsculas. El segundo punto ya no es tan obvio. ¿Cómo son las columnas definidas en la base de dato? Simplemente habrá que decirle a SQL que describa la tabla:

Describe PAPELERIA;

Tabla No.34. Tabla Descriptiva.



Id. de cliente	Nombre de compañía	Nombre del contacto	Cargo del con
AMORS	SALON CRISTY	MAYRA	Gerente
AROUT	Around the Horn	Thomas Hardy	Representante de ven
BERGS	Berglunds snabbköp	Christina Berglund	Administrador de pec
BLAUS	Blauer See Delikatessen	Hanna Moos	Representante de ven
BLONP	Blondel père et fils	Frédérique Citeaux	Gerente de marketin
BOLID	Bólido Comidas preparadas	Martin Sommer	Propietario
BONAP	Bon app'	Laurence Lebihan	Propietario
BOTTM	Bottom-Dollar Markets	Elizabeth Lincoln	Gerente de contabili
BSBEV	B's Beverages	Victoria Ashworth	Representante de ven
CACTU	Cactus Comidas para llevar	Patricio Simpson	Agente de ventas
CENTC	Centro comercial Muctezuma	Francisco Chang	Gerente de marketin
CHOPS	Chop-suey Chinese	Yang Wang	Propietario
COMMI	Comércio Mineiro	Pedro Afonso	Asistente de ventas
CONSH	Consolidated Holdings	Elizabeth Brown	Representante de ven
DRACD	Drachenblut Delikatessen	Sven Ottlieb	Administrador de pec
DUMON	Du monde entier	Janine Labrune	Propietario
EASTC	Eastern Connection	Ann Devon	Agente de ventas
ERNSH	Ernst Handel	Roland Mendel	Gerente de ventas

Tabla No.35. Hoja de datos de la tabla clientes.

Esto que se muestra es una tabla descriptiva que contiene las columnas y sus definiciones para la tabla PAPELERIA; el mandato describe funciona con cualquier tabla.

La primera columna indica los Id de los clientes de la tabla que se está describiendo.

La segunda columna, se describe el nombre de la compañía de la tabla descripta.

Por supuesto, en una tabla como PAPELERIA, probablemente hubiera sido valioso tener la misma regla en las tres columnas. No sólo es bueno conocer el título de un artículo (nombre), sino también en qué sección se encuentra y en qué página. Pero esto no se ha hecho aquí para que el ejemplo sea más ilustrativo. Sólo la columna Nombre se creó con la regla NULL.

La razón por la que Section y Nombre no tiene nada en la columna Null?, Es porque a ellas les está permitido estar o vacía o llenas en cualquier fila de la tabla PAPELERIA.

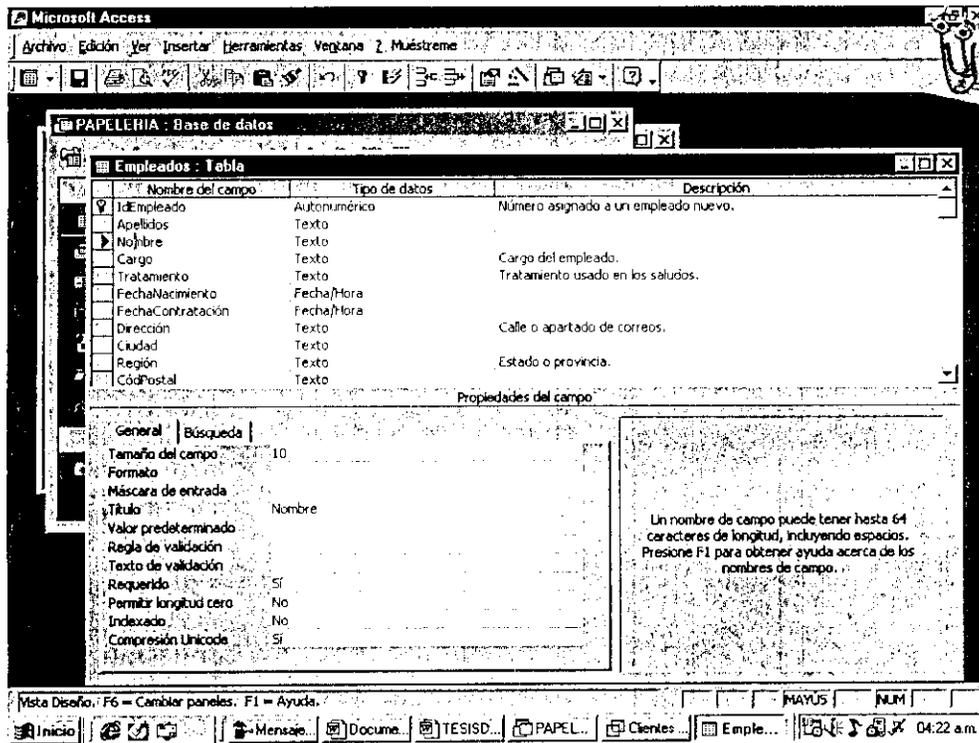
La tercera columna, Type, indica la naturaleza básica individual de las columnas. Feature es una columna de caracteres (CHARacter) que puede tener 15 caracteres (letras, números, símbolos o espacios).

"Section" es una columna de caracteres, que sólo puede tener un carácter. Resultados de SQL mostraron los resultados de la consulta: Ref:{16,P.50}

```
select Feature, Section, Nombre from PAPELERIA;
```

conocía de la base de datos que la "Section" tenía un máximo de un carácter. Suponía que no se quería consumir un espacio mayor a éste, por eso mostró una columna con una anchura justa de un carácter y usó como nombre de la columna el único carácter que podía: "S".

Tabla No.36



Se habrá notado que la columna Nombre en la tabla EMPLEADO, muestra una anchura de diez espacios, aunque las páginas no usen más de dos números.

Esto se debe a que los números no se suelen definir con una anchura específica, por eso SQL asume esta anchura por defecto al arrancarse.

Se puede comprobar que las columnas compuestas de números están justificadas a la derecha, o sea, se asientan sobre el lado derecho de la columna, mientras que las columnas compuestas de caracteres se justifican a la izquierda. Esta es la alineación estándar de las columnas en SQL. Como otras características de las columnas, ésta también se podrá cambiar en el futuro.

Finalmente, SQL indica cuántas filas (se puede usar como sinónimo registros) ha encontrado en la tabla PAPELERIA.

**Set** es un mandato de SQL. Esto quiere decir que es una instrucción para SQLPLUS que le indica cómo debe actuar. Hay muchas características de SQLPLUS, como feedback, que se pueden fijar.

**Set** tiene un complemento llamado **show** que permitirá ver qué instrucciones se han dado a SQL. Por ejemplo, se puede mirar el valor dado a **feedback** marcando:

```
show feedback
```

SQL responderá:

```
feedback ON for 25 or more records
```

La anchura que se usa para mostrar los números también se cambia mediante el comando **set**. Se puede comprobar escribiendo:

```
show numwidth
```

SQL responderá:

```
numwidth 10
```

Puesto que 10 es demasiado ancho para mostrar números de páginas, que nunca serán mayores de dos dígitos, se puede cambiar escribiendo :

```
set numwith 5
```

Sin embargo, esto significa que todas las columnas de números tendrán una anchura de cinco dígitos. Si se prevén números de más de cinco dígitos se debe usar una anchura superior.

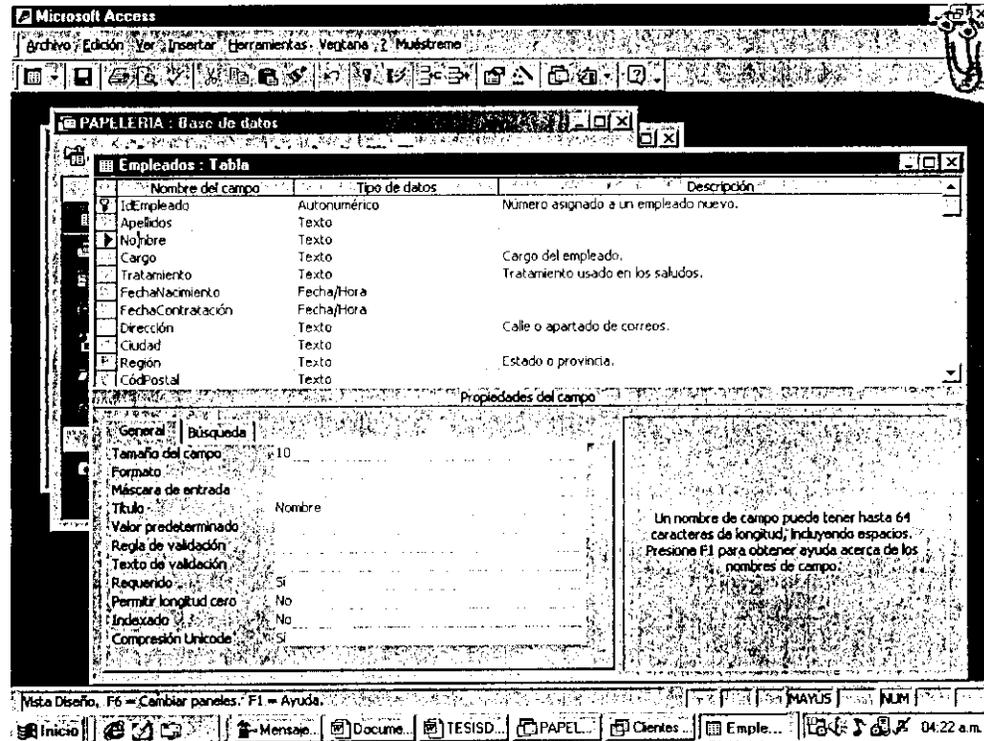
### **Select, from, where y order by**

Existen cuatro palabras claves en SQL para obtener información de las tablas de la base de datos: **select**, **from**, **where** y **order by**. Las dos primeras se usan en todas las consultas de la base de datos que se hagan. Ref:[16,P.51]

La palabra clave **select** le dice a la base de datos las columnas que se desean, y **from** le indica a la base de datos los nombres de la tabla o tablas cuyas columnas se desean.

Notar que la consulta en SQL es bastante parecida a la que se realiza en inglés. Una consulta en SQL acaba normalmente con un punto y coma (que algunas veces se llama "terminador de SQL"). La palabra clave **where** le indica a la base de datos las restricciones que se quieren poner a la información seleccionada. Por ejemplo: Ref:[16,P.52]

Tabla No.37



Para indicar a la base de datos que la información se encuentra en un orden se utiliza **order by**. Considero los siguientes ejemplos importantes:

Tabla No.38

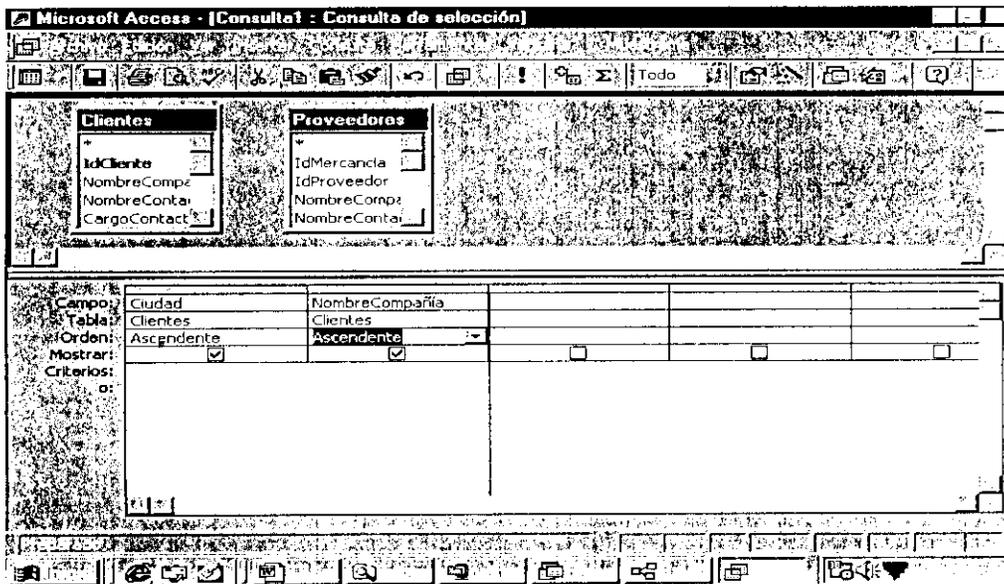
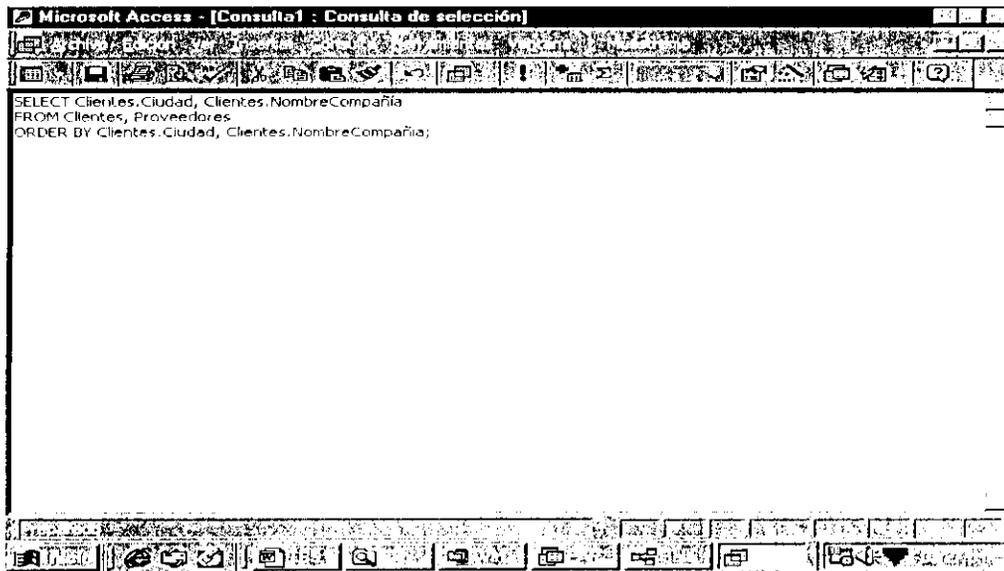


Figura No.39

Tabla No.40

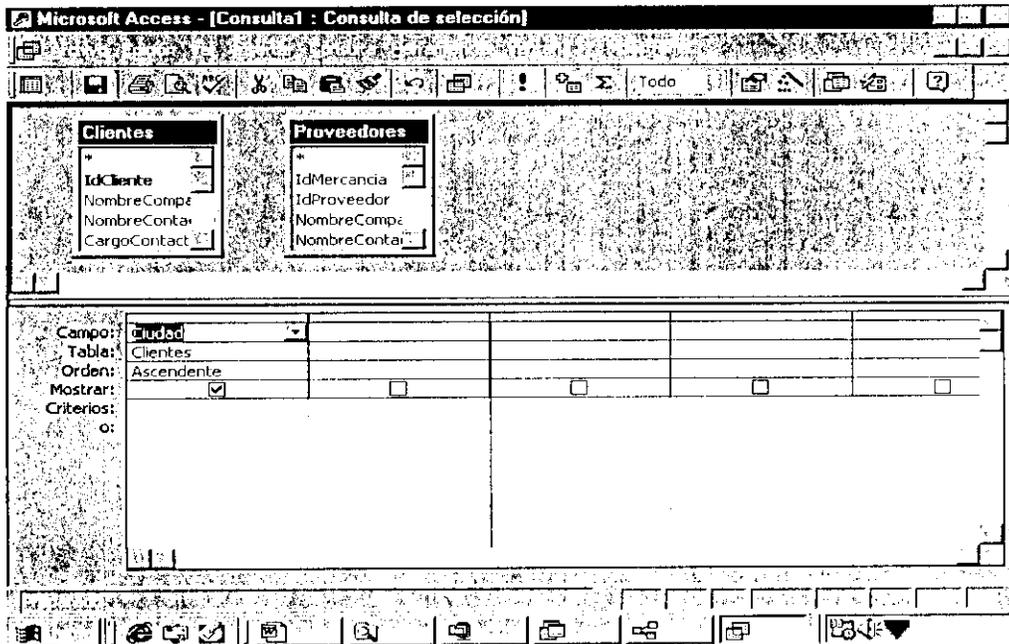
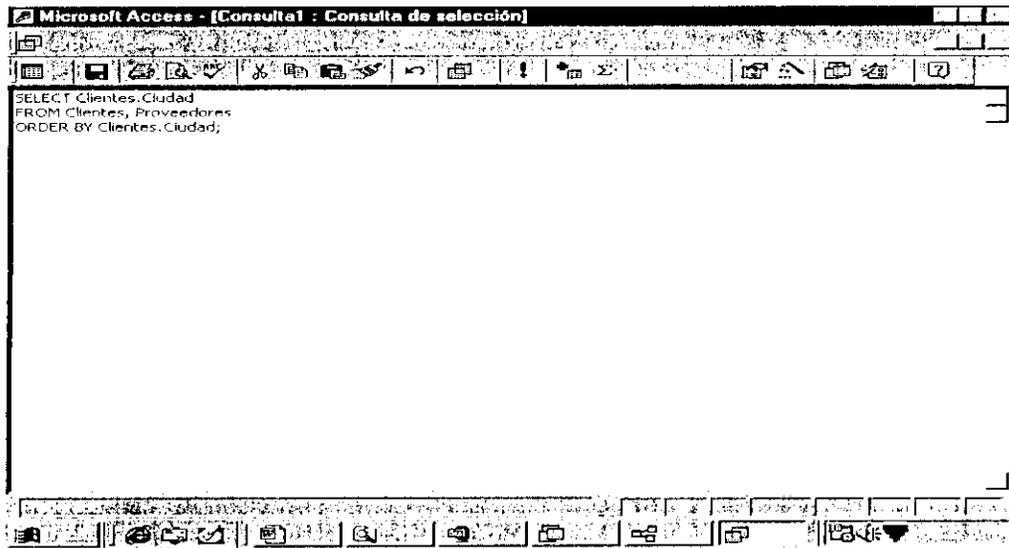


Figura No.41

### 3.2 IMPLANTACION DE LAS BASES DE DATOS.

A continuación compararemos las características comunes de los sistemas de bases de datos para microcomputadores con las de los sistemas más grandes. Ref: [9, P.589] .

- **Modelo de datos.** Dado que los sistemas de base de datos para microcomputadores son relativamente nuevos, casi todos están basados en el modelo relacional. Algunos de esos sistemas deben considerarse más bien tabulares, ya que, aunque utilizan tablas, son demasiado primitivos para llamarse relacionales.
- **Lenguaje de consultas.** Es posible que los lenguajes de más alto nivel que se analizaron sean demasiado complejos para el usuario casual de un sistema de base de datos para microcomputadores. Muchos de los lenguajes se basan en una interfaz de forma, con la cual el usuario puede interactuar con el sistema relleno una forma.
- **Realización física.** Para los realizadores de un sistema, uno de los factores importantes es el espacio que ocupa el código objeto de un sistema de base de datos para microcomputadores. Si se reduce el espacio requerido, es posible utilizar el sistema de esquinas que cuenten con menos memoria principal. Esto puede tener una influencia determinante sobre el mercado potencial. Por esta razón son pocos los sistemas que utilizan técnicas de gestión de memoria y de indexación complejas. Por lo general se elige un solo tipo de índice, y la optimización de consultas, cuando se lleva a cabo, es pequeña.

### 3.3. OPERACIONES DE MANTENIMIENTO

#### 3.3.1 OPERACIONES DE ENTRADA/SALIDA

Las operaciones de Entrada/Salida son la manera en la que la computadora se comunica con el mundo externo. Ya que el CPU sólo puede tener acceso a programas o datos en la memoria principal, deben idearse métodos para poder introducir datos o programas que estén almacenados externamente, a la memoria principal y para poder obtener resultados de la ejecución de programas, sobre los dispositivos de almacenamiento externos. Ref. [9,P.9].

Un programador que desee mandar el comando Read para recobrar un registro de un archivo, debe especificar el nombre simbólico del mismo, y tal vez el valor de la clave del registro, en caso de acceso aleatorio. En la práctica no todos los comandos Read/Write convocados desde un programa, proporcionan una transferencia física de datos. Debido a la lentitud en la velocidad de las operaciones de E/S, es necesario, minimizar el número de estos pasos. Una de las técnicas usadas para minimizar el número de estos pasos. Una de las técnicas usadas para minimizar las operaciones de E/S consiste en guardarlas en memorias intermedias.

Un **buffer** o depósito de transferencia es un lugar para guardar algún bloque de datos que va a ser transferido entre un almacenamiento secundario y la memoria del programa. El tamaño del buffer puede ser definido por el usuario como un número específico de registros físico.

El buffer asociado con el archivo es asignado por un **método de acceso** del sistema operativo cuando se abre el archivo y es liberado cuando el archivo se cierra. Al recibir cada petición de Read, el método de acceso transfiere otro bloque de datos del almacenamiento secundario hacia el buffer. La operación Write causa que los datos fluyan en la dirección opuesta a la de Read.

El uso del buffer es algo semejante a comprar productos en el supermercado y ponerlos en el refrigerador, de donde se tomarán uno a la vez, sin necesidad de regresar a la tienda cada vez que se necesite algún artículo.

## **A) Respaldo**

Un sistema informático, como cualquier otro dispositivo mecánico o eléctrico, está sujeto a fallos. Las causas de los fallos incluyen rotura de disco, problemas del suministro de energía y errores software. En cada uno de estos casos, se pierde la información referente a la base de datos. Es responsabilidad del gestor de la base de datos detectar tales fallos y restaurar la base de datos al estado que existía antes de ocurrir el fallo. Esto se lleva a cabo normalmente a través de la iniciación de varios procedimientos de copias de seguridad o respaldo.

### **3.3.2 TRANSFERENCIA DE DATOS ENTRE LA MEMORIA PRINCIPAL Y LA UNIDAD DE ALMACENAMIENTO SECUNDARIO.**

Una operación de entrada causará que los datos requeridos sean llevados desde un almacenamiento secundario a través de una unidad de control y un procesador de E/S hasta la memoria principal, de donde una operación de salida causará su movimiento a través de pasos similares en orden inverso, para ser guardados en la base de datos. Ref[9,P.10 ]

Cuando se emite un comando Read o Write desde un programa, el CPU cambia su control hacia un método de acceso apropiado del sistema operativo. El método de acceso establecerá una trayectoria para los datos y determinará la dirección del registro físico por acceder. Bajo el control del sistema operativo, el CPU activa un procesador ocioso de E/S, para que realice la transferencia de datos. Mientras se transfieren los datos el CPU continúa su trabajo con otros programas. Así, la operación de E/S se traslada con la ejecución del CPU.

Un procesador de E/S inicia la transferencia de datos entre el dispositivo de almacenamiento externo y la memoria principal, mandando señales a la unidad de control (o controlador), el cual posiciona la cabeza de lectura/escritura (Read/Write) sobre el área de datos objetivo en el dispositivo de almacenamiento para la transmisión de los datos. Ref[9,P.10]

### **3.3.3 PROCESADOR DE ENTRADA/SALIDA (E/S) Y ADITAMENTO DE UNIDAD DE CONTROL**

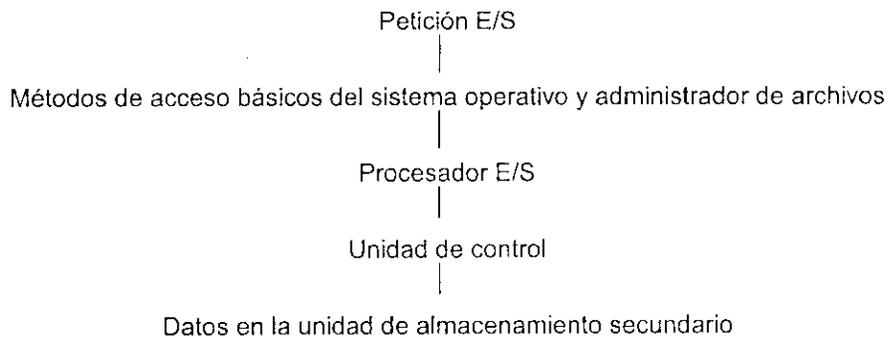
Los procesadores de E/S están conectados a la memoria principal para controlar la transmisión de datos entre el dispositivo de almacenamiento y la memoria principal, para que las lentas operaciones de E/S puedan llevarse a cabo simultáneamente con la ejecución del CPU. El procesador de E/S es una computadora en sí misma; ejecuta sus propios programas. Un procesador de E/S, generalmente está conectado a varios dispositivos de almacenamiento secundarios. La mayoría de las computadoras medianas o grandes tienen varios procesadores de E/S, el cual es activado por el sistema operativo al enviar una petición Read/Write desde un programa cuando el buffer está vacío. El programa canal de un sistema IBM reside en la memoria principal. Ref.[9,P.11].

Una unidad de control es un procesador con una función especial que sirve como interfase entre un procesador de E/S y una unidad de almacenamiento. Interpreta las señales que envía el procesador de E/S a la unidad de almacenamiento y luego controla la operación.

Es posible agregar al sistema una variedad de dispositivos de almacenamiento, siempre y cuando esté disponible un controlador adecuado para interpretar las señales enviadas a o desde la computadora huésped. La unidad de control puede integrarse como parte del dispositivo E/S o separadamente. Ref.[9,P.12].

Figura No.42

La trayectoria de los datos en una operación E/S es como sigue:



- **Recuperación.** Muchos sistemas no cuentan con subsistemas de recuperación. El usuario tiene la responsabilidad de sacar copias de seguridad de sus datos de forma regular.
- **Concurrencia.** No se requiere control de concurrencia para computadores personales de un solo usuario.

La distinción entre los sistemas de base de datos para microcomputadores y los sistemas más grandes se hace menos clara. Los primeros sistemas de bases de datos para computadores no eran mucho más que interfaces para lograr acceso a un solo archivo de registros de longitud fija.

Al crecer la capacidad de los computadores personales, ha crecido también la complejidad de los sistemas de base de datos para microcomputadores. De hecho, han empezado a aparecer versiones de alguno de los sistemas de base de datos de gran escala en los computadores personales de mayor tamaño.

## **B) Recuperación de base de datos**

En una base de datos pueden existir errores debido a varios motivos, tales como la actualización incompleta de datos redundantes, validación inadecuada de los datos de entrada, mal funcionamiento físico o errores de lógica en la programación. Desgraciadamente los errores de lógica en la programación. Desgraciadamente los errores no siempre se pueden evitar, así que los DBMS proporcionan mecanismos para recuperar los datos correctos después de que ocurren los errores. La restauración de una base de datos a su estado normal es problema real e importante que no se puede pasar por alto.

Generalmente, quien opera la base (DBA), es responsable de implantar procedimientos de detección de errores.

Utilerías de detección son proporcionadas como parte del sistema de manejo de base de datos o se pueden desarrollar particularmente. Sin importar el procedimiento utilizado, éstos se corren a intervalos regulares para mantener la integridad. Una vez detectados, es esencial que el manejador los corrija inmediatamente.

Para que el manejador obtenga la información necesaria para efectuar la recuperación de la base, se deben hacer regularmente copias de respaldo (back-up copies). Por lo común se usan cintas magnéticas para guardar las copias, debido a que son compactas y de bajo costo. La última copia de respaldo anterior a un error, se usa para restaurar los datos afectados hasta ese día.

En un archivo de varias claves es usual que una de éstas sirva para la identificación unívoca de los registros. No es normal que una clave secundaria identifique también de forma exclusiva un registro. Hay a menudo muchos registros con ese mismo valor de clave. NUMERO-DE-PIEZA podría ser una clave primaria puesto que identifica unívocamente un registro de pieza al no existir dos piezas que tengan el mismo número. En cambio TIPO-DE-PIEZA podría ser una clave secundaria. Pueden existir muchas piezas del mismo tipo, pero TIPO-DE-PIEZA es a menudo una clave importante para muchas averiguaciones.

Una manera de hallar un registro a partir de una clave secundaria consiste en construir un índice para esa clave. Los índices basados en un atributo que no es la clave primaria se llaman índices secundarios. En algunos sistemas de información resulta inconveniente indizar muchos atributos diferentes, por lo que los índices secundarios suelen resultar numerosos.

### **Ejemplo de una base de datos.**

Es un ejemplo típico de sistema de información con índices secundarios el sistema que sirve para obtener información acerca de las ventas de una compañía. El usuario plantea al sistema interrogantes del tipo de los siguientes:

El nombre y la dirección postal del cliente forman, en conjunto una clave prima para el direccionamiento de los registros de clientes. El número de artículo constituye una clave prima para direccionar los registros de artículos.

Pero se necesitan claves secundarias para vincular estos dos archivos y para indizar ítems tales como zona, ventas netas, número de artículos vendidos, número de ítems pendientes y categoría industrial.

En los sistemas de procesamiento por lotes no se requieren por lo general índices secundarios. Durante las exploraciones periódicas de los archivos es fácil producir una variedad de listas. Pero en los sistemas de información en línea no es posible acumular grandes cantidades de averiguaciones y clasificarlas antes de procesarlas, como se haría en el procesamiento por lotes. Deben por lo tanto preverse los medios para ir directamente de la averiguación planteada a los registros que permiten evacuarla.

Las opciones que se ofrecen al diseñador de un archivo de claves múltiples permiten llegar a una adecuada concertación entre el aumento de la cantidad de claves secundarias utilizadas, el aumento de la utilización de la memoria principal, y el aumento del tiempo requerido para localizar los registros necesarios.

## C) Reorganización y estructuración

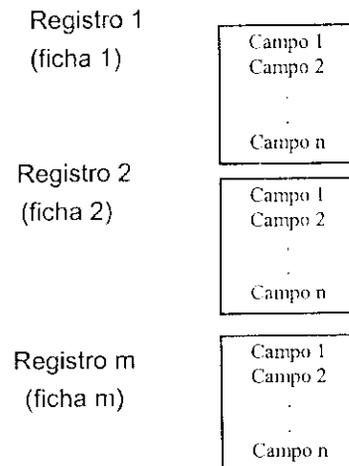
La mayoría de los sistemas manejadores de bases de datos proporcionan un elemento de rastreo para registrar lo sucedido en cada transacción actualizada por la base de datos. El registro, es un prerequisite para la recuperación de la base de datos, restaura un archivo a su estado anterior cuando ocurre alguna falla en una transacción. El manejador del registro (log manager), es una componente del sistema de manejo de base de datos que efectúa el registro escribiendo cualquiera de los dos tipos de registros siguientes en un archivo de búsqueda:

1) Ante-imagen: Se refiere a los bloques antiguos de datos originales en la base que se guardaron antes de la actualización de una transacción. Si ocurre un error, esta copia se puede usar para cancelar el efecto de la transacción.

2) Post-imagen: Es el nombre que recibe un bloque procesado por una transacción listo para ser reescrito en la base.

Las bases de datos se organizan en tablas. Pero cualquier sistema de bases de datos se construye mediante registros que podrían compararse a las fichas de un fichero, que sería la base de datos o una parte integrante de ésta. A su vez, cada registro de la base está compuesto por campos, que en equivalencia son los datos distribuidos en las fichas del fichero y que se repiten para cada ficha (los campos, no su contenido).

Ejemplo:



Como vemos en el esquema, la estructura de los campos se repite para cada registro. Así, si una estructura de base de datos contiene los campos Nombre, Apellidos y Teléfono, cada registro tendrá un Nombre, unos Apellidos y un Teléfono.

La forma de representar estos datos mediante bases de datos relacionales es la tabla, que contiene en sus filas y columnas los datos que la llenan. Por ejemplo, he aquí una sencilla tabla para el almacenamiento de números de teléfono:

Tabla No. 43

	Campo 1	Campo 2	Campo 3
	Nombre	Apellidos	Teléfono
Registro 1	Cristina	Tomía	123-45-67
Registro 2	Mayra	Nazas	765-43-21
Registro 3	Audina	Derrana	135-79-11
Registro 4	Rocio	Doadistancia	119-75-31

Ejemplo:

En esta estructura se repite para cada registro, aunque, naturalmente, los datos son distintos para cada uno. Los campos, por su parte, forman la primera fila de la tabla (Nombre, Apellidos y Teléfono).

Una base de datos tiene la ventaja de poder relacionar los datos entre sí, ya que una base de datos no tiene por qué estar compuesta exclusivamente por una tabla, sino que, por el contrario, puede estar constituida por varias conectables entre sí.

En los métodos de direccionamiento por clave única, es ésta la que determina la posición física de los registros en los archivos. Los registros, en los archivos secuenciales indezados, se ordenan según la secuencia ascendente de la clave. En los dispositivos de acceso directo, la posición de los registros es la que se determina por medio de un algoritmo de conversión de la clave. En el caso de los archivos de claves múltiples, sólo una de las claves debe determinar la posición física del registro. Si los registros, que se muestran en el ejemplo, están ordenados en secuencia ascendente según la clave 1, las otras claves no pueden estar ordenadas del mismo modo ni según el orden determinado por un algoritmo.

En la mayoría de los archivos de claves múltiples de hoy en día, es la clave prima la que determina la ubicación física de los registros, tal como ocurre en el caso de casi todos los archivos de clave única. El método de direccionamiento de clave secundaria debe basarse por lo tanto en una técnica que no dependa de la posición física de los registros. A veces existen buenas razones para que esta independencia se extienda también a la clave prima. Si la técnica para la localización de los registros es independiente de la posición física de éstos, los registros podrán dejarse donde se escribieron por primera vez, sin necesidad de ser redistribuidos periódicamente para hacer lugar a los recién venidos. Muchas de las dificultades que se presenta en el mantenimiento de una base de datos tienen su origen en las reorganizaciones periódicas a que se someten los registros. Además, los archivos se agrandan a medida que aumenta el número de aplicaciones y se abarata el almacenamiento. La reorganización periódica de los grandes archivos es una operación que insumo tiempo y dinero.

## **D) Monitoreo y ajustes del desempeño**

Se mejora en performance, productividad y seguridad, también incrementa el cumplimiento con los estándares de la industria. Ref:[17]

### **Operaciones de SQL paralelas**

Esta opción mejora significativamente el performance para las operaciones que procesan una gran cantidad de datos.

Con la opción de query paralelo, puede desglosar la ejecución de consultas, la carga de datos y la creación de índices, para ser ejecutadas en varios CPU'S. Así, se explota el poder de procesamiento paralelo de las configuraciones de multiprocesadores, escalando el performance a medida que se añaden procesadores. La opción de query paralelo incremento el performance en Multiprocesamiento Simétrico (Symmetric multi-processing, SMP), sistemas en cluster (Cluster System ej. dos procesadores accedando al mismo disco), y masivamente paralelos.

## **Ejecución de Query Paralelo**

La ejecución de query paralelo incrementa el performance de las aplicaciones que envuelven consultas de SQL en sus bases de datos grandes.

Estas operaciones que son: table scans, joins, ordenamientos, operaciones de conjuntos (Unión y Unión All), y agrupaciones (Group By, Distinct) mejoran el tiempo de respuesta.

Excepto en la mejora de performance, las operaciones de query paralelo son transparentes al usuario.

Las opciones de Configuración y hints de SQL permiten controlar la descomposición de queries y ejecución paralela.

## **Desarrollo de Aplicaciones**

Incluye varias facilidades que mejoran el ambiente de desarrollo de aplicaciones para los clientes con Procedural Option. Los usuarios pueden extender el lenguaje SQL con funciones almacenadas de PL/SQL específicas de las aplicaciones desarrolladas por el mismo usuario.

La capacidad de referenciar funciones definidas por el usuario en SQL permite a los desarrolladores de la aplicación hacer encapsulación lógica de la aplicación en el servidor de la base de datos, sin exponer el sistema a problemas de integridad causados por user exits.

El performance de la aplicación a menudo puede ser mejorado enormemente, en especial en ambientes cliente/servidor, ya que la lógica de la aplicación se ejecuta como parte del procesamiento de la instrucción de SQL, reduciendo la necesidad de procesar datos en el programa de la aplicación.

## **Mejoras Administrativas**

Un administrador de la base de datos puede definir un tablespace en modo read-only. Los sistemas administradores de bases de datos cuyas bases de datos contienen grandes cantidades de datos estáticos pueden ahorrar tiempo porque los tablespaces de solo-lectura no requieren backup ni recovery. Para aplicaciones de multimedia, los tablespaces solo-lectura pueden ser almacenados a un bajo costo por bit del dispositivo de almacenamiento como CD-ROM o drivers WORM.

## **Mejoras en Seguridad**

Este release soporta las sesiones del administrador de bases de datos y OPER autorizadas por el Sistema Operativo sobre una conexión no segura, aún cuando la base de datos no esté disponible. Se provee la protección de passwords para las conexiones, haciendo más fácil la labor del administrador de bases de datos de auditar.

## **E) SEGURIDAD Y PRIVACIDAD**

La seguridad de los datos está fuertemente relacionada con la integridad de los mismos. La seguridad se refiere a la protección de la base contra accesos o modificaciones no autorizados. Como en las bases de datos los datos se comparten ampliamente, la información confidencial en la base es vulnerable a las intromisiones. Un acceso ilegal puede tener como resultado la destrucción accidental o maliciosa de la base. Sin control de seguridad, los usuarios no tendrán la privacidad requerida en sus datos confidenciales y el sistema no podrá mantener la integridad de los mismos.

Por lo cual, se proporciona un mecanismo en el sistema administrador de bases de datos para reforzar la seguridad de los datos. Algunos mecanismos para el control de seguridad son el subesquema, los seguros de control de acceso y al poner los datos en claves secretas.

La seguridad se aplica generalmente a tablas y vistas, pero otros objetos tales como formularios, programas de aplicación y bases de datos enteras también pueden ser protegidos. La mayoría de los usuarios tendrán permiso para utilizar ciertos objetos de la base de datos, pero tendrán prohibido el uso de otros. Ref:[3,P.532]

## Componentes del Problema de Protección

Tres tipos de elementos se combinan para formar el sistema que se utiliza para analizar los métodos de protección:

1. Los usuarios con acceso a la base de datos, a los que por brevedad denominaremos accesoros.
2. El tipo de acceso deseado.
3. Los elementos a los que se realizará el acceso.

Cada uno de estos elementos debe estar adecuadamente identificado a fin de lograr el control del acceso a los datos. También es necesario considerar el entorno o frontera del área dentro de la cual es válido el sistema de protección. Ref:[Ídem, P.685]

Un ejemplo de privacidad individual. Es el estimar el costo operativo para el próximo mes de una Empresa, se escribe un programa que utiliza los datos de personal para extrapolar costos a partir de salarios, horas normales de trabajo y días de vacaciones. Cristina Ordoñez, quien prepara el informe, no debería tener acceso a los antecedentes psiquiátricos del personal, que se conservan en línea junto con los datos de personal. Ref: [6, P. 684]

Ejemplo No. 44:

Partiendo de la tabla No.44, y utilizando todos los conceptos antes vistos de SQL, como son: Select, Where, Order by, From, Describe, como a continuación muestro:

Describe Agenda;

Ejemplo.

Name	Null?	Type
NOMBRE	NOT NULL	CHAR(10)
APELLIDO		CHAR(15)
TELEFONO		NUMBER

Lo cuál nos da la siguiente tabla, con datos ya capturados.

Campo 1          Campo 2          Campo 3

	Nombre	Apellidos	Teléfono
Registro 1	Cristina	Villa	123-45-67
Registro 2	Mayra	Villa	765-43-21
Registro 3	Audina	Villa	135-79-11
Registro 4	Rocio	Villa	119-75-31

En dicha tabla se pueden hacer varias operaciones ya sea ubicarse en cierta parte de la misma, así como ordenar por Nombre, Apellidos, Teléfono, utilizando la instrucción Order by de la siguiente forma:

```
select Nombre, Apellidos, Telefono from Agenda
Where Apellidos = "Villa"
Order by Nombre, Apellido
```

Y queda la tabla ordenada de la siguiente forma como a continuación se muestra:

Nombre	Apellidos	Teléfono
Audina	Villa	135-79-11
Cristina	Villa	123-45-67
Mayra	Villa	765-45-21
Rocio	Villa	119-75-31

O también, se puede ordenar en forma descendente, utilizando las mismas instrucciones básicas de SQL, anteriormente mencionadas, solo agregando la de desc, de la siguiente forma:

```
select Nombre, Apellido, Teléfono from Agenda
Where Apellidos= "Villa"
Order by Teléfono, desc Nombre
```

Y queda la tabla, así.

Nombre	Apellidos	Teléfono
Mayra	Villa	765-45-21
Audina	Villa	135-79-11
Cristina	Villa	123-45-67
Rocio	Villa	119-75-31

# **CAPITULO**

## **IV**

## CAPITULO IV.

### 4. TOPICOS AVANZADOS DE BASES DE DATOS.

#### 4.1 BASES DE DATOS DISTRIBUIDAS

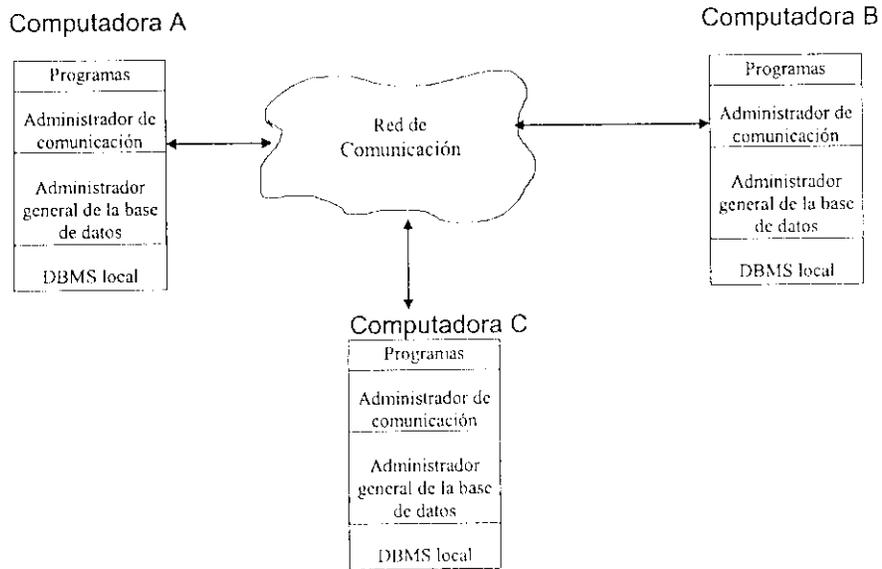
Cuando distintos usuarios en distintos lugares geográficos van a compartir una gran base de datos, las compañías pueden optar por dividir la base de datos en bases más pequeñas llamadas fragmentos, las cuales se pueden almacenar en ubicaciones distintas. Una base de datos distribuida consta de fragmentos almacenados en sitios distintos. Cada sitio cuenta con una computadora y un DBMS para administrar su propia base local. Los fragmentos de una base distribuida se conectan entre sí por medio de una red de comunicación.

Cuando surge una petición de consulta de algún sitio en particular, el administrador general de la base analiza la petición para determinar qué fragmentos de la base se requieren. Un programa en un sitio puede invocar programas en otros sitios remotos para acceder sus fragmentos locales con el DBMS local.

Aunque el concepto de base distribuida es joven aún, se han desarrollado diversas implantaciones donde las principales son:

- 1) SDD-1(System for Distributed Database) de la Computer Corporation of América: Sistemas relacionales (Data Computer) en distintos lugares conectados por redes de comunicación.
- 2) IMS/MSC (Information Management System/Múltiple Systems Coupling), de IBM: El MSC permite la intercomunicación de dos o más sistemas de bases de datos IMS.
- 3) CICS/ISC (Customer Information Control System/Inter-Systems Communication), de IBM: ISC conecta dos o más sistemas CICS.

Figura No.45. Sistema Distribuido de Bases de Datos.



#### 4.1.2 CONSTRUCCION DE UNA BASE DE DATOS DISTRIBUIDA

Existen tres tipos de distribución de una base de datos, que son:

1) Base distribuida particionada

Con este tipo de distribución, una base global se parte en tres (A,B y C), cada una de las cuales se almacena en un sitio distinto.

2) Base de datos multiplicada y distribuida

Toda la base global se duplican cada sitio. El objeto de estas réplicas es reducir los costos de comunicación y aumentar el desempeño del sistema eliminando la necesidad de la transmisión de datos entre ubicaciones distintas.

Este tipo de distribución es rara vez implantado debido al costo global de las réplicas de la base (redundancia extrema) y a problemas en la actualización de los datos redundantes.

Distribución de la base de datos combinando réplicas y particiones. Cada nodo contiene un subconjunto único de la base así como una selección de fragmentos réplica de otros sitios.

En la práctica, esta distribución es la que se emplea con mayor frecuencia. Un archivo se parte basándose en su patrón de uso, es decir, los datos se guardarán en los lugares donde tengan mayor probabilidad de ser accedados. Sin embargo, cuando se requiere cierto fragmento en más de un sitio, el fragmento puede estar almacenado redundantemente en cada uno de esos sitios.

Ejemplo No. 46

NO. ESTUDIANTE	LOCATION	BALANCE
C1	CRISTINA	2005.75
C2	CRISTINA	575.00
C3	CRISTINA	1200.75
C4	MAYRA	7982.00
C5	MAYRA	45.00

a) Un archivo global de datos

NO. ESTUDIANTE	LOCATION	BALANCE
C1	CRISTINA	2005.75
C2	CRISTINA	575.00
C3	CRISTINA	1200.75

b) Fragmentación horizontal del archivo NUMERO a): escogiendo los renglones que contienen CRISTINA en LOCATION.

NO. ESTUDIANTE	BALANCE
C1	2005.75
C2	575.00
C3	1200.75

c) Fragmentación horizontal y vertical del archivo NUMERO en a): escogiendo renglones que contienen CRISTINA en LOCATION y columnas NO-ESTUDIANTE y BALANCE.

Existen diferentes tipos de particiones que son las siguientes:

1)Participación horizontal: Un fragmento se genera escogiendo registros que corresponden a clientes en CRISTINA.

2)Partición vertical: El fragmento se crea escogiendo de la base global las columnas NO.ESTUDIANTE y BALANCE.

3)Partición horizontal y vertical (arborescente):La relación se genera escogiendo renglones que corresponden a los clientes en CRISTINA. Los registros escogidos son a su vez divididos para escoger sólo las columnas NO.ESTUDIANTE y BALANCE.

Los usuarios en un sistema distribuido no necesitan saber que el almacenamiento de la base está distribuido en distintos sitios, o que algunos fragmentos están repetidos en más de un sitio. Como sucede en un sistema integrado, los puntos de vista de los programadores individuales están representados por medio de esquemas externos y los usuarios no tienen necesidad de saber los detalles de su almacenamiento físico. Uno de los objetivos del sistema distribuido es proteger a los programas de cambios cuando un fragmento se mueve de un sitio a otro. Estos conceptos se llaman transparencia de la ubicación y transparencia de la réplica.

### **4.1.3 ORIENTACION EN LA INVESTIGACION SOBRE BASES DE DATOS DISTRIBUIDAS**

Un sistema de bases de datos distribuidas ofrece las siguientes ventajas:

1) Costo reducido en la comunicación de datos. Una base distribuida reduce el tráfico en las intercomunicaciones ya que la mayoría de las preguntas o consultas se pueden contestar sin necesidad de acceder los archivos de lugares distantes.

2) Autonomía local. En un sistema distribuido, los datos obtenidos en algún lugar en particular son generalmente procesados sólo por usuarios locales.

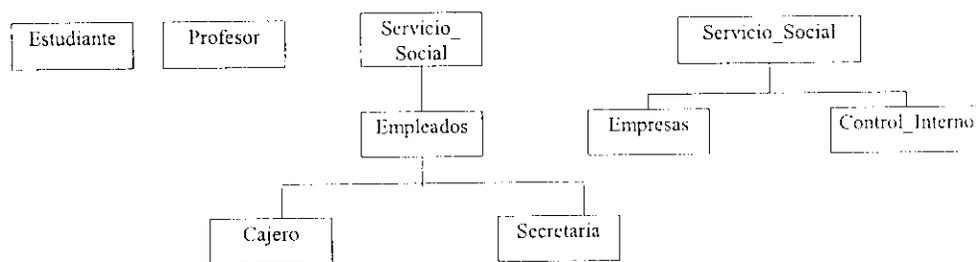
#### 4.1.4. Relaciones Jerárquicas

Las relaciones jerárquicas son la forma más natural de la que dispone el modelo orientado a objetos para relacionar entidades (Clases).

Este tipo de relaciones se deriva de la capacidad de herencia que tiene el modelo OO, por medio del cual un objeto (clase) puede heredar sus características y métodos a otra, formando de esta manera relaciones del tipo "es un".

Utilizar relaciones jerárquicas conlleva a manejar 2 tipos de clases, las superclases y subclases, y a relacionarlas por medio de relaciones del tipo "es un".

Ejemplo No.47



#### 4.1.5 Relaciones interclases

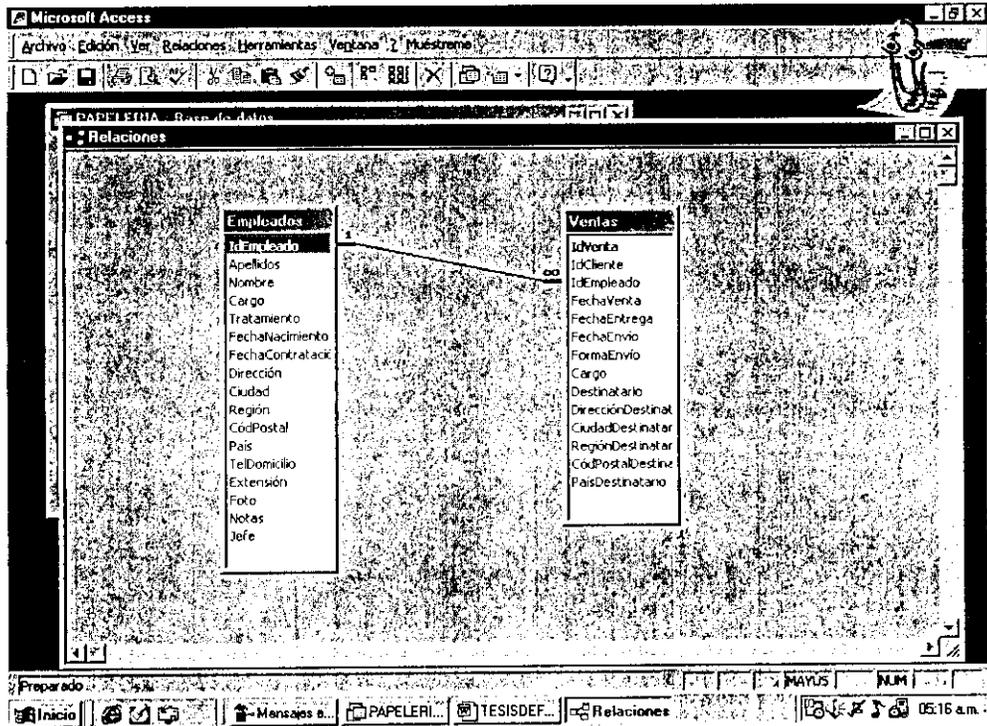
Las relaciones interclases se forman cuando un atributo de una clase es descrito con referencia a otro tipo de clase (tipo abstracto de datos).

Por ejemplo, existe una relación 1:M entre las clases instalación y empleados, ya que cada empleado puede trabajar en una sola instalación puede tener a muchos empleados trabajando en ella.

Cuando se tiene una relación 1:M, la parte de muchos se construye como una lista de objetos de la otra clase.

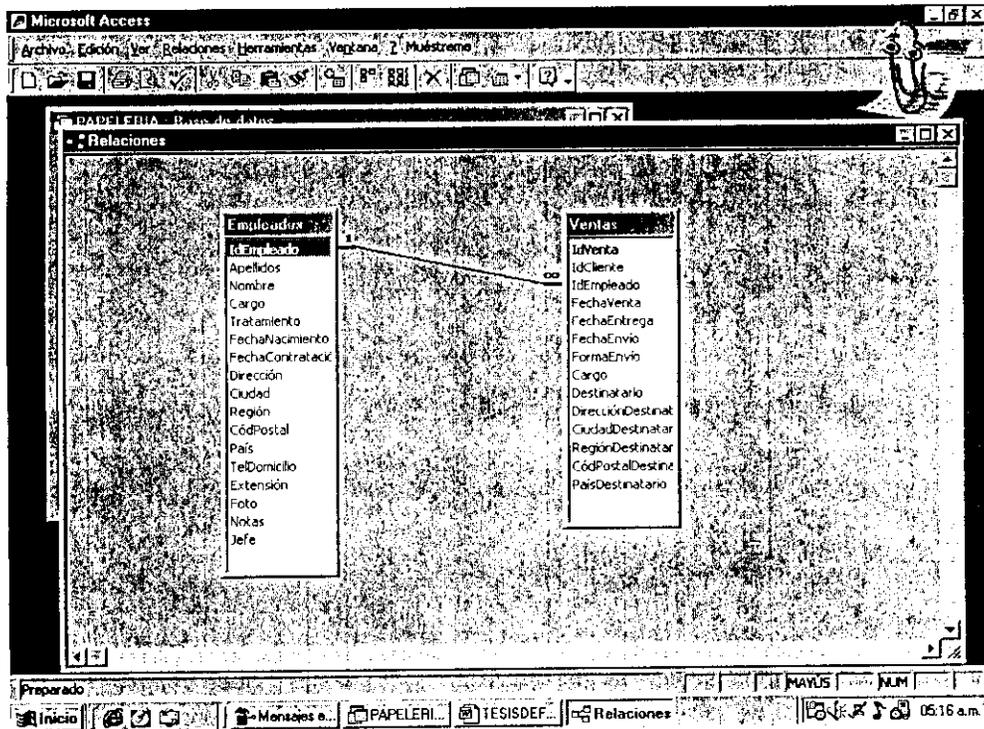
Por ejemplo, cada objeto de instalación tendrá como atributo trabajadores una lista de objetos de la clase empleado.

Ejemplo No.48. Representación de relaciones 1:M.



Las relaciones del tipo M:N se representan usando la misma metodología que las relaciones 1:M, es decir, conectando objetos (clases) a través de sus atributos.

#### Ejemplo No.49: Representación de relaciones M:N.



#### 4.2 BASES DE DATOS ORIENTADAS A OBJETOS

Entre los años de 1980 y 1990, muchos de los expertos programadores y diseñadores de bases de datos se enfrentaron a problemas más complejos.

Estos implicaban requerimientos de datos mayores y ya no podían ser manejados fácilmente por los lenguajes tradicionales (estructurados) ni por las bases de datos relacionales. Ref :[14, P.672]

Por ejemplo, ahora tanto los programas como las bases de datos tenían que incluir gráficas, vídeo, sonidos así como otro tipo de datos compuestos.

Como una respuesta a todo esto surgió el enfoque de orientación a objetos.

Definiremos un objeto como un cuadro, ventana, campo, línea, menú, popup o texto que se muestra en la pantalla. Algunos de los objetos son muy simples, como un campo de un registro; otros, como una ventana, son mucho más complejos.

Algunos objetos contienen atributos, que determinarán cómo visualizar el objeto o cómo trabajar con él. El programador podrá controlar algunos de los atributos, como el color, el tipo o el tamaño de un objeto. Otros atributos, denominados asociados, no podrá controlarlos, como son los atributos de control asociados a los objetos. Por ejemplo, cuando ejecutamos la orden de revisión de una ventana podemos controlar cómo presentaría en pantalla y su contenido, pero no podemos controlar atributos de control asociados automáticamente a las ventanas, como cerrar cuadro, zoom, agrandar cuadro o desplazamiento.

Utilizando objetos, como la creación de bloques, podremos generar pantallas e informes, y aplicar posteriormente otros comandos hasta completar el sistema o la aplicación.

La búsqueda de mejores formas de administrar los datos, a provocado que los nuevos modelos incorporen mas características semánticas y soporten el manejo de datos más complejos.

Un ejemplo de esto es el modelo orientado a objetos que presenta las siguientes propiedades básicas:

1. Permite la representación de objetos complejos.
2. Es extensible, es decir, debe ser capaz de definir nuevos tipos de datos y las operaciones permitidas sobre estos.
3. Permite la encapsulación, lo que significa que la representación interna de los datos y métodos debe permanecer oculta para entidades externas.
4. Usa la herencia. Esto permite que un objeto herede las características y métodos de otros objetos.
5. Maneja el concepto de OID (identificador único de objeto).

Figura No.50

Modelo OO vs. Modelo E-R

Modelo OO	Modelo E-R
Tipo Clase	Definición de entidad
Objeto	Entidad
Clase	Conjunto de entidades
Variable (propiedades)	Atributos
---	Llave primaria
OID	---
Métodos	---
Jerarquía de clases	Diagrama E-R
Relaciones interclases	

#### 4.2.1 Representación de objetos

Los objetos son las instancias formadas de una clase, por lo tanto, es en la definición de la clase donde se especifican todas las características que van a tener los objetos que se derivan de ésta. Ref :[15, P.65]

La estructura de una clase es similar a la de un conjunto de entidades en el modelo E-R y corresponde a lo representado por medio de una tabla en el modelo relacional.

En el modelo orientado a objetos las clases (objetos) no solo son un grupo de características sino que incluyen un grupo de métodos aplicables a los objetos de esa clase.

## 4.2.2 Relaciones entre objetos

Uno de los aspectos más importante de todo modelo de dato es su capacidad de relacionar información.

El modelo E-R hace uso de relaciones o en su defecto de entidades relación para relacionar dos entidades diferentes pero pertenecientes a la base de datos.

En el modelo relacional, las relaciones entre tablas (implementación de conjunto de entidades) se realizan por medio de campos comunes.

El modelo orientado a objetos hace uso de dos mecanismos para conectar información (representada por medio de clases), a saber: Las relaciones jerárquicas y las relaciones interclases.

## A) CONCEPTOS DE BASES DE DATOS ORIENTADA A OBJETOS

### 4.3.1 DEFINICION DE UN OBJETO

Hacia cualquier lugar que miremos lo que siempre vamos a ver van a ser muchos y diferentes objetos, ya sea personas, animales, plantas, muebles, edificios, computadoras y demás.

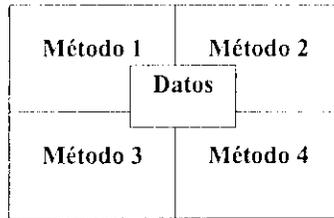
Todos estos objetos son en esencia diferentes, partiendo de que unos son animados y otros no, pero todos ellos tienen algo en común:

- Todos tienen atributos, como por ejemplo, color, tamaño, forma, etc.
- Todos exhiben varios comportamientos, por ejemplo, una persona camina, come, duerme; un animal ladra, corre; un carro acelera, frena, gira, etc.

En la tecnología de bases de datos, un objeto es una representación abstracta de una entidad del mundo real que tiene una identidad única, que tiene ciertas propiedades intrínsecas y tiene la habilidad de interactuar con otros objetos y con él mismo.

En otras palabras, los objetos están conformados por un grupo de datos (atributos) y un conjunto de métodos (acciones que puede realizar o se pueden realizar con el objeto).

Figura No.51 Estructura de un Objeto



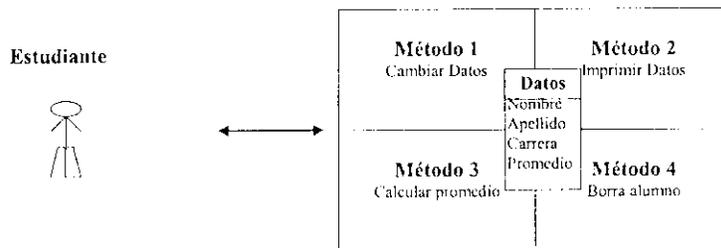
Se puede indentificar tres partes fundamentales: a identidad única, las propiedades y el comportamiento. Ref :[7]

La identidad única, representada por un identificador asignado por el sistema (OID) sirve para poder distinguir un objeto de otro, a pesar de que estos comportan características.

Las propiedades o atributos de los objetos del mundo real son simulados por medio de un conjunto de variables (datos) en los sistemas orientados a objetos.

El comportamiento es simulado mediante un grupo de métodos, es decir, cada acción relacionada con el objeto se implementa por medio de un método. En este sentido los métodos representan las acciones del mundo real

Figura No.52 Representación de entidades reales con objetos



### **4.3.2 DEFINICION DE CLASES**

Los sistemas Orientados a Objetos se clasifican a los objetos de acuerdo a sus similitudes y diferencias, agrupando a los objetos que comparten las mismas características dentro de clases.

Una clase es por consiguiente un conjunto de objetos similares que comparten su estructura (datos) y su comportamiento (métodos).

Las clases contienen la descripción de la estructura de los datos y los detalles de implementación de los métodos que formaran parte de los objetos pertenecientes a dicha clase.

A cada uno de estos objetos se le conoce con el nombre de instancia de la clase.

El concepto de clase es muy importante porque con fundamento en ellas, la tecnología Orientada a Objetos introduce conceptos como: Encapsulación, Polimorfismo y Herencia, características que son la base del comportamiento de los sistemas Orientados a Objetos, entre los cuales se encuentran las bases de datos. Ref:[15, P.677]

### **B) PARADIGMA DE BASES DE DATOS ORIENTADAS A OBJETOS**

La programación orientada a objetos (POO) \* se suele conocer como un nuevo paradigma de programación. Otros paradigmas conocidos son: el paradigma de la programación imperativa (con lenguajes tales como Pascal o C), el paradigma de la programación lógica (PROLOG) y el paradigma de la programación funcional (Lisp). El significado de paradigma (paradigma en latín; paradigma en griego) en su origen significaba un ejemplo ilustrativo, en particular enunciado modelo que mostraba todas las inflexiones de una palabra. En el libro The Structure of Scientific Revolutions, el historiador Thomas Kuhn describía un paradigma como un conjunto de teorías, estándares y métodos que juntos representan un medio de organización del conocimiento: es decir, un medio de visualizar el mundo.

En este sentido, la programación orientada a objetos es un nuevo paradigma. La orientación a objetos fuerza a reconsiderar nuestro pensamiento sobre la computación, sobre lo que significa realizar computación y sobre cómo se estructura la información dentro del computador.

Jenkins y Glasgow observan que "la mayoría de los programadores trabajan en un lenguaje y utilizan sólo un estilo de programación. Ellos programan en un paradigma forzado por el lenguaje y utilizan. Con frecuencia, no se enfrentan a métodos alternativos de resolución de un problema, y por consiguiente tienen dificultad en ver la ventaja de elegir un estilo más apropiado al problema a manejar". Bobrow y Stefik definen un estilo de programación como "un medio de organización de programas sobre la base de algún modelo conceptual de programación y un lenguaje apropiado para hacer programas en un estilo claro". Ref:[9, P.15]

#### **4.4.1 ENCAPSULACION**

La encapsulación es un mecanismo que agrupa el código y los datos que maneja un objeto y los mantiene protegidos frente a cualquier interferencia y mal uso.

Este concepto tiene su origen y principal uso en los lenguajes orientados a objetos, en donde se construyen los objetos de tal forma que sean como una "caja negra" que encierra a los datos y que solo permite el acceso a estos a través de varios métodos definidos por el propio objeto. Ref :[7]

El encapsulamiento es un concepto muy importante en la construcción de bases de datos, ya que permite que un tipo de objeto (representación de una entidad del mundo real) este desligado completamente de otros en lo que se refiere a su comportamiento intrínseco, cosa que sucede en el mundo real. Ref :[7]

#### **4.4.2 POLIMORFISMO**

Polimorfismo (del griego, cuyo significado es "muchas formas") es la cualidad que permite que un nombre se utilice para dos o más propósitos relacionados, pero técnicamente diferentes.

El propósito del polimorfismo en el ambiente Orientado a Objetos es lograr que diferentes objetos respondan al mismo mensaje en diferentes formas, dependiendo de sus características específicas.

En forma general, el polimorfismo en los sistemas Orientados a Objetos significa:

- Que se pueda usar el mismo nombre para un método definido en diferentes clases de la jerarquía de clases.
- El usuario puede mandar el mismo mensaje a diferentes objetos que pertenecen a diferentes clases y siempre generar la respuesta correcta.

La idea del polimorfismo se puede resumir en una idea "una interfaz, múltiple métodos".

#### 4.4.3 HERENCIA

La herencia es el proceso mediante el cual un objeto puede adquirir las propiedades de otro.

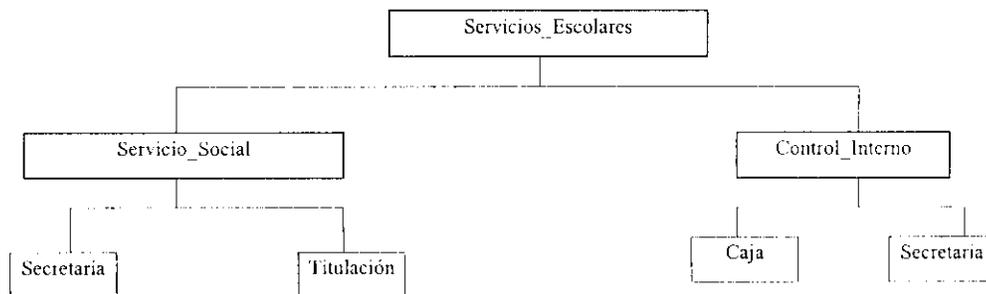
Este concepto permite que un objeto herede un conjunto general de propiedades (de otros objetos) a las cuales añada otras características que son específicamente suyas.

La herencia es muy importante porque admite que los objetos soporten el concepto de clasificación jerárquica.

Hay dos tipos de herencia, la herencia sencilla y la múltiple:

- En la sencilla las subclasses únicamente heredan características de una superclase.
- En la herencia múltiple puede haber una subclase que herede características de dos o más superclases. Ref :[14, P.590]

Ejemplo No. 53 Jerarquía de clases; un ejemplo de herencia.



## C) LENGUAJES PARA BASES DE DATOS ORIENTADAS A OBJETOS

### 4.5.1 PROGRESS

En la versión 6 de Progress soporta bases de datos distribuidas Progress puede conectarse y desconectarse de forma dinámica con 240 bases de datos simultáneamente.

Progress utiliza una arquitectura distribuida, que soporta acceso completo de lectura y escritura a todas las bases de datos a través de todas las bases de datos. Para ello, el cliente verifica que todas las bases de datos están activas y preparadas para realizar una transacción. Una vez que todas las bases de datos devuelven la confirmación del cliente, conforme están preparadas, la transacción es ejecutada. Si cualquier base de datos no confirma la actualización, ésta es cancelada, volviendo todas las bases de datos a su estado original. Ref: [4, P. 934].

#### 4.5.1.1 Control de concurrencia

El control automático de concurrencia con total soporte transaccional asegura que ningún usuario tenga una visión inconsistente ni pierda ninguna actualización de la base de datos.

Esto se consigue sin utilizar complejos protocolos de bloque de registros. En cualquier caso, el usuario puede fácilmente controlar el ámbito de la transacción y los tipos de bloqueo a cualquier nivel.

Los menús de FoxPro forman parte de la interfaz orientada a objetos.

El control de objetos de FoxPro se realiza por medio de las cajas de diálogo que muestra en las ejecuciones de los bloques de menús. Al ejecutar un menú, los tipos de control de objetos que aparecen en el mismo pueden ser:

#### \* Las cajas de comprobación (Check boxes):

Representan una selección múltiple de opciones. Las opciones van precedidas de corchetes ([ ]). Al seleccionar una caja aparece la letra X al lado, entre corchetes [X], lo que muestra el estado de la opción ON u OFF.

### **\*Controles Popup:**

Objetos que abren ventanas popup con más opciones. Una de las funciones típicas es permitir al usuario el cambio de directorio activo. Otros controles popup se aplican a la visualización de los nombres de ficheros, comandos y funciones.

### **\*Botones activos (Pushbuttons):**

Aportan el control de opciones por medio de una variada selección de botones en las ventanas de diálogo.

Cada opción está encerrada entre ángulos simples, salvo la opción activa que está encerrada entre ángulos dobles.

Una caja de diálogo típica de este tipo es:

<<Cancel>> <Ok> <Ignore>

### **\*Botones de radio (Radio Buttons):**

Permiten al usuario seleccionar una opción entre un conjunto de ellas. Cada opción está precedida por paréntesis vacíos ().

Al seleccionar una opción, aparecerá un punto entre los paréntesis (.). Al activar una opción, las demás se desactivan automáticamente.

### **\*Lista de opciones deslizantes (Scrolling List):**

Representa la lista que muestra en la ventana sólo una parte de las opciones que comprende. Cuando nos deslizamos por la ventana hasta la última opción de la misma, si seguimos deslizándonos hacia abajo de la ventana irán apareciendo las demás opciones hasta completar la lista.

Este tipo de listas se reconocen fácilmente porque en el marco de la ventana aparecen dos flechas, hacia arriba y hacia abajo, que indican que hay más opciones de las que se muestran en la ventana.

### **\*Cajas de texto (Text Boxes):**

Corresponden a las cajas de diálogo donde solamente el usuario puede escribir datos. Un ejemplo de estas cajas es cuando el sistema nos pide que escribamos el nombre de un archivo nuevo. Sólo el usuario puede suministrar esta información.

Los sistemas Orientados a Objetos surgen en los lenguajes de programación Orientados a Objetos ya lo que pretenden ambos es que el usuario no tenga problemas y que todos los términos sean más cercanos al mundo real.

En los años ochentas algunos investigadores como Brad. J. Cox, Jacob Stein, vierón a los Sistemas Orientados a Objetos como el futuro de las bases de datos. Los Sistemas Orientados a Objetos hacen que los sistemas se vuelvan inteligentes.

Las Bases de Datos deben ser controladas y administradas por una persona la cual es conocida con el nombre de Administrador de la Base de Datos (ADB), su función es reconocida por Codasyl, es importante mencionar que existen diferentes tipos de administradores en los cuales sus nombres varían según su función entre los cuales son:

- Administrador de la base de datos
- Administrador de aplicaciones
- Administrador de la empresa, etc.

La Base de Datos cuenta con recursos como son los Diccionarios de Recursos de Información, los cuales surgen por el aumento de la información, en los cuales se conocen dos tipos de almacenes de datos que son Diccionario de datos y Directorio de datos, los primeros sólo guarda información de datos y los otros es el subsistema de el Sistema de Gestión de Bases de Datos y encargado de decir donde y como se almacenan los datos.

## 4.5.2 INGRES

A mediados y fines de la década de 1970, se estaba desarrollando el prototipo de “**System R**” dentro de IBM, se había emprendido otro proyecto relacional importante en la Universidad de California en Berkeley. El resultado de esos trabajos fue el prototipo INGRES (en un principio, INGRES fue un acrónimo, y significaba "sistema interactivo de gráficos y obtención de datos", "Interactive Graphics and Retrieval System"). El prototipo de INGRES (al cual hoy día se le conoce como "INGRES académico" para distinguirlo de la versión comercial tuvo amplia difusión en ambientes universitarios a finales de la década de 1970 y a principios de 1980, tanto en Estados Unidos como en otros lugares. Además, desde luego, hasta la fecha continúa en Berkeley un programa activo de investigación y desarrollo con base en INGRES, así como continúan dentro de IBM actividades similares con base en System R. Ref: [10, P.217].

A principios de la década de 1980 se formó una compañía llamada “ **Relational Technology Inc.** “(La cual cambió su nombre hace poco a Ingres Corporation) con el propósito de desarrollar y promover una versión comercial de INGRES.

El INGRES académico se ejecutaba en máquinas DEC PDP con el sistema operativo UNIX; el INGRES comercial se ejecutaba en diversas máquinas con varios sistemas operativos distintos, incluyendo, por ejemplo, máquinas DEC VAX con UNIX o VMS, máquinas IBM System/370 y similares con VM, computadores personales IBM y similares con PC/DOS, y varias máquinas con base en MC68000 y otros procesadores, con UNIX.

Su versión original, INGRES no fue un sistema SQL; el prototipo original universitario manejaba un lenguaje llamado QUEL ("Query Language", lenguaje de consulta). INGRES comercial maneja en la actualidad QUEL y SQL como alternativas iguales (en principios), aunque a juzgar por la publicidad reciente - la orientación parece estar desplazando a últimas fechas hacia SQL. (Situación desafortunada, en opinión de este autor, pues QUEL es en muchos aspectos técnicamente superior a SQL. QUEL puede considerarse como realización bastante pura el cálculo relacional, y, como tal, es bastante menos peculiar que SQL). Pero QUEL todavía constituye la base de muchas investigaciones modernas sobre base de datos, y en la literatura de investigación aparecen con frecuencia ejemplos expresados en QUEL.

Al igual que SQL, QUEL puede utilizarse como un lenguaje de consulta interactiva (a través del monitor de terminal de INGRES, INGRES Terminal Monitor) y también como un lenguaje de programación de bases de datos embebido en diversos lenguaje anfitriones (a través de QUEL embebido, Embedded QUEL o EQUEL).

Nota: El producto INGRES comercial ofrece también un conjunto de interfaces integradas basadas en formas para tareas tales como consultas especiales, redacción de informes, gráficas empresariales, etc. Ref: [Idem, P.218]

Existen muchos paralelismos entre INGRES y DB2; por ejemplo, INGRES admite también múltiples usuarios, incluye un componente optimizador de consultas, permite la definición dinámica de los datos, ofrece un mecanismo de vistas relacionales, etc.

En INGRES no existe la idea de un componente ligador separado, como en DB2. Las dos operaciones:

a) Produciendo un "plan de aplicación" optimizado (llamado "plan de consulta" en INGRES) para una consulta dada.

b) Ejecutando ese plan.

La ventaja es de no separar INGRES lo cual no precisa del complicado mecanismo de compilación/recopilación de DB2; el optimizador siempre produce código de acuerdo con él esto actúa en la base de datos, y las operaciones como desechar un índice u otro objeto no requieren actividad adicional del sistema para supervisar la utilización del objeto desechado. Por añadidura, el optimizador de INGRES quizá produzca mejor código que el optimizador de DB2, precisamente porque trabaja en términos de información más actualizada. La desventaja, desde luego, radica en la necesidad de optimizar los programas de ejecución repetitiva cada vez que se ejecutan.

Digresión: INGRES comercial, incluye dos características que se abocan a este problema. En primer lugar, si una solicitud determinada (proposición QUEL o SQL) se va a aplicar varias veces dentro de una misma ejecución del programa, sólo será precisa optimizarla primera vez que se aplique, no en cada ejecución. En segundo lugar, el recurso de procedimiento almacenado hace posible que un programa de aplicación invoque un procedimiento (o sea, subrutina) ya compilado desde antes, en forma bastante similar a DB2. Ref: [10, P.218].

## B) APLICACIONES

**Progress** es un producto de P.S.C.(Bedford, Massachusetts).La primera versión de Progress se difundió a partir de 1984 en Estados Unidos.

A la fecha de enero de 1991 se habían vendido 67.000 licencias de Progress en el mundo, 26.000 de las cuales eran Unix. El número de usuario de Progress se encuentra en torno a los 500.000.

Política de actualización. Hasta la fecha, PSC ha lanzado al mercado una nueva versión del producto con una periodicidad anual (a título de ejemplo: se espera para septiembre de 1991 la versión siete, que entre otras cosas podrá trabajar en entorno gráfico G.U.I.).Ref: [4, P. 926] .

Progress constituye una herramienta de desarrollo de aplicaciones transaccionales de alto rendimiento, integrada por un potente lenguaje de 4ª. generación, un gestor de bases de datos relacionales (multivolumen y distribuidas), un diccionario de datos (metaesquema), herramientas de productividad (Fast-Track,Results,Report Writer), y módulos para portar aplicaciones entre diferentes sistemas operativos y redes (Developer's Toolkit). Progress se encuentra disponible para la mayoría de entornos hardware y redes locales. Utiliza arquitectura cliente/servidor y dispone de versión run-time para todos los entornos así como interface de desarrollo con Oracle y RDB o RMS de Dec y en breve plazo con otras bases de datos del mercado.

### 4.6.1 Indices

En progress se usan índices para definir llaves. Un índice en una Base de Datos opera como un índice tab en un folder de archivo. Estos señalan un campo indentificador, tal como el nombre de un cliente, haciendo más fácil y rápida la búsqueda de la información que se requiere.

Progress permite usar un campo simple para definir un índice simple, o una combinación de campos para definir un índice compuesto.

Para decidir que campos usar, debemos determinar como serán accedidos los datos en la tabla. Si el usuario frecuentemente busca clientes por apellido, entonces el apellido es el candidato para un índice. Es típico basar índices en campos que contienen información única.

Un índice tiene las siguientes ventajas:

- **Búsqueda y acceso rápido de registros.** Es más eficiente localizar un registro buscando en una tabla ordenada por índices que buscar en una tabla desordenada.
- **Los registros son ordenados automáticamente para soportar las preferencias particulares de acceso de datos.** No importan los cambios en la tabla, cuando se busca o imprime información de la tabla, los registros aparecen ordenados por índice sin importar el orden de almacenamiento físico en disco.
- **Cada registro es único cuando se define un índice como único.** Esto garantiza que los registros duplicados no existan.
- **Una combinación de campos puede estar juntos al índice permitiendo ordenar una tabla de diferentes modos a la vez** (por ejemplo, ordenar la tabla Proyectos como una combinación de los campos empleado y fecha).
- Acceso eficiente a datos en múltiples tablas relacionales.

#### 4.6.2 PRODUCTOS DE QUE SE COMPONE

##### **Sistema de desarrollo de aplicaciones Progress**

Es el principal módulo de la familia. Combina toda la potencia y la flexibilidad del Lenguaje de Cuarta Generación (4GL) y el Sistema de Gestión de Bases de Datos Relacionales (RSGBD) con los beneficios que se obtienen con la alta productividad de Progress Fast Tack.

##### **Progress 4GL y RSGBD**

Proporciona todas las herramientas necesarias para desarrollar por completo complejas aplicaciones.

### **Progress Fast Track**

Generador de aplicaciones asistido mediante menús, que acelera la construcción de prototipos y facilita el desarrollo de aplicaciones Progress.

### **Progress Query/Report**

Versión Run-Time, que adicionalmente proporciona a los usuarios la capacidad de generación de informes propia de Progress Fast Track en combinación con la capacidad de acceso al Diccionario de Datos propia de Progress 4FL/RSGBD.

Ref: [4, P. 935].

### **Progress Run-Time**

Producto que posibilita el funcionamiento de aplicaciones Progress en sistema donde no esté instalado éste.

### **Progress Developer's Toolkit**

Posibilita el traspaso de aplicaciones entre diferentes entornos.

### **Progress Test Drive**

Versión totalmente funcional de Progress acompañada de una aplicación de muestra, que incluye todas las capacidades y la mayoría de la documentación. La única limitación es que el usuario solo puede acceder un número determinado de veces a la misma base de datos.

### **Host Languaje Interface**

Mediante este módulo se pueden intercalar sentencias SQL en aplicaciones 3GL ya existentes, y acceder a bases de datos distribuidas Progress, directamente a través de una interface SQL. Ref: [4, P. 935].

### **Progress Databases Gateway**

Permite el acceso a múltiples Bases de Datos simultáneamente.

## **Progress Results**

Módulo de usuario final que permite el acceso y generación de informes, etiquetas,.. de la base datos. Ref: [4, P. 936].

### **4.6.3 OTRAS CARACTERISTICAS**

#### **Memory Saver**

Cuando se trabaja bajo MS-DOS, Progress tiene la característica "Memory Saver", que permite que en cualquier PC con un mínimo de 640 K de memoria funciones Progress por encima de la frontera de un megabyte, liberando de esta forma parte de las 640 K para otros softwares de utilidades y programas.

#### **Host Language Calls**

Independientemente que con el 4GL de Progress se pueda desarrollar enteramente la mayoría de las aplicaciones, los programadores tienen la libertad de acceder al sistema operativo, o utilizar rutinas escritas en C para realizar cálculos especializados, que interactúa con hardware específico. Ref: [4, P. 932].

Un programa Progress puede enviar un output o recibir un input, a o desde terminales, ficheros de un sistema operativo, impresoras, discos ópticos, o cualquier grupo de éstos concurrentemente. De igual forma, las aplicaciones desarrolladas con Progress pueden intercambiar información con otras aplicaciones.

Progress proporciona ANSI-estándar SQL, permitiendo al programador integrar sentencias SQL con sentencias Progress 4GL en el mismo programa. El editor de Progress permite trabajar con ambos lenguajes y generar un único programa ejecutable. Los programas pueden acceder a ficheros SQL y de igual forma mediante sentencias SQL se puede acceder a ficheros Progress.

#### **4.6.4 Arquitectura Cliente/Servidor**

La arquitectura cliente/servidor de Progress proporciona unas interfaces transparentes entre las actividades del front end client y las actividades del back end database.

Progress separa los procesos que realiza el cliente de las funciones del servidor de la base de datos. El cliente se comunica con el usuario, manejando todas las funciones lógicas de las aplicaciones y los procesos de interfase con el usuario. El servidor gestiona el acceso y actualización de la base de datos. Ref: [4, P. 933].

#### **4.6.5 Sistema de gestión de bases de datos multi-usuario**

Progress tiene una arquitectura multi-hembra-multi-usuario que le permite eliminar tiempos de espera y procesos de cuellos de botella normalmente asociados con redes.

#### **4.6.6 Recuperación automática en caso de caída del sistema**

El sistema de gestión de Progress mantiene un fichero "before image" donde se guardan copias sin modificar de todos los registros actualizados.

En el caso de un conflicto de transacciones, o caída del sistema, este fichero se utiliza para deshacer todas las transacciones incompletas, asegurando así la integridad de la base de datos. Así, las capacidades de recuperación de bases de datos, hasta hace poco limitada a los mainframes, se incorporan a los micros y miniordenadores.

# **CAPITULO**

## **V**

## CAPITULO V

### 5.1 CONCLUSION Y RECOMENDACIONES

Con el desarrollo del presente trabajo, compruebo la validez de la hipótesis planteada, dado que cubre con las características precisas del plan de estudios de la Licenciatura en Informática, permitiendo al profesor planear el curso con mejores resultados; y tener una referencia directa a los alumnos y facilitar el momento educativo.

Antes de haber elaborado esta tesis siempre había existido el problema de como enseñar las bases de datos, ya que la mayoría del profesorado que es contratado por la Universidad Americana de Acapulco, no cuenta con el tiempo suficiente en documentarse, debido a que tienen otro trabajo en empresas, por lo cual mi trabajo viene a servir como apoyo fundamental del aprendizaje de las bases de datos para el alumnado de recién ingreso a esta institución.

Otro de los problemas que se presenta en el aprendizaje es, que el profesor contaba con otros conceptos que en ocasiones eran muy complejos su comprensión y lo cuál dificultaba más que el alumno aprendiera, por ello en mi trabajo se encuentran conceptos lo suficientemente entendibles.

En él demuestro que tan importante es una base de datos para el manejo de datos, ya que actualmente según mis investigaciones realizadas, cada vez hay más empresas que manejar alguna Base de Datos, dígase Access, Oracle, Progress, Ingress, Clipper o Manejador de Bases de Datos como es Dbase y sobre todo en las instituciones de Gobierno.

En muchas ocasiones cuando un alumno es nuevo en el conocimiento computacional, es decir que no conoce términos básicos se le complica aún más el saber que es una tabla y cual es su función y objetivo.

Pues bien, una base de datos es tan poderosa según el programador lo quiera, es decir, que los datos que son manejados en cualquiera de las bases de datos antes mencionadas, pueden ser almacenadas para un sólo usuario lo cuál es una gran ventaja ya que se cuenta con la información más importante de la empresa en cualquier momento.

Los sistemas administradores de bases de datos orientadas a objetos, (OODBMS), surgieron por la necesidad de satisfacer nuevos requerimientos en el manejo de la información. Ya que permiten administrar datos de naturaleza más compleja como: grandes cantidades de texto, imágenes y sonidos.

Estos sistemas combinan características de los conceptos de la orientación a objetos, como por ejemplo, su modelo, el encapsulamiento, la herencia de propiedades y el polimorfismo, y las principales propiedades de los sistemas administradores de bases de datos tradicionales, el manejo de la concurrencia, el acceso, la seguridad, integridad y respaldo de los datos.

Algunas de sus principales ventajas de los sistemas administradores de bases de datos orientadas a objetos son que incluyen mayor información semántica y la mejor representación de entidades reales y que facilitan la reutilización del código.

Y entre sus principales desventajas es que no está completamente finalizado, y que por consiguiente carecen de estándares, en ocasiones este modelo es considerado como retroceso al antiguo sistema de vectores utilizado por los modelos jerárquicos y de red y que el tiempo requerido para su aprendizaje es mayor que el ocupado para aprender el modelo relacional.

No es tan solo, el hecho de contar con una base de datos o hacer una base de datos y elaborar un sistema de contabilidad, nominas, presupuestos, inventarios, en una empresa, si no el saber como funcionan las bases de datos en general.

Pues bien, las bases de datos van evolucionando al ritmo de la tecnología, ya que se puede utilizar bases de datos distribuidas incluso en redes locales, o en cualquier otro tipo de red.

## BIBLIOGRAFIA:

[1] ACCESS 97,PASO A PASO LA FORMA RAPIDA Y FACIL DE APRENDER,  
F. Scott Barker y Diana Barker,PHH. A. Simon, Shuster,Company Prentice Gue  
Hall.

[2] ANALISIS Y DISEÑO DE SISTEMAS, KENNETH E. KENDALL, JULIE E.  
KENDALL,P.H.H. PRENTICE HALL.

[3] APLIQUE SQL,JAMES R. GROFF,PAUL N. WEINBERG,MC GRAW HILL.

[4] CONCEPCION Y DISEÑO DE BASES DE DATOS DEL MODELO E/R AL  
MODELO RELACIONAL,Adoración Miguel/Mario Piattini, Addison-Wesley  
Iberoamericana.

[5] DICCIONARIO DE COMPUTACION, FREEDMAN,MCGRAW HILL.

[6] DISEÑO DE BASES DE DATOS, GIO WIEDERHOLD,MC GRAW HILL.

[7] DIPLOMADO DE BASE DE DATOS

[8] dBASE IV MASTER INFORMATICA V.1.1.,JULIO GONI.

[9] FUNDAMENTOS DE BASES DE DATOS,Henry F. Korth, Abraham  
Silberschatz y S. Sudarshan, Mc Graw Hill.

[10] INTRODUCCION A LOS SISTEMAS DE BASES DE DATOS, C.J. Date,  
Quinta Edición, Addison-Wesley Iberoamericana.

[11] INTRODUCCION A LA INFORMATICA, JOSE LUIS MORA ENZO  
MOLINO  
TRILLAS.

[12] NOTAS DE INFORMATICA II, Semestre 1, 1995, Ing. Gonzalo Trinidad Garrido.

[13] SISTEMAS DE BASES DE DATOS CONCEPTOS FUNDAMENTALES  
Elmasri/ Navathe, 2 do. EDICION.

[14] SISTEMAS DE BASES DE DATOS, ADMINISTRACION Y USO, ALICE Y  
H. TSAI, PRENTICE HALL.

[15] ORGANIZACION DE LAS BASES DE DATOS, JAMES MARTIN,  
PRENTICE HALL.

[16] ORACLE MANUAL DE REFERENCIA, GEORGE KOCK, VERSION 5 Y 6,  
MC GRAW HILL.

[17] ORACLE MANUAL, RELACIONES DE TABLA Y NORMALIZACION.

[18] TECNICAS DE BASES DE DATOS, SHAKUNTALA ATRE, TRILLAS.

[19] PROGRAMACION ORIENTADA A OBJETOS, LUIS JOYANES AGUILAR,  
P.15.

[20] PC MAGAZINE EN ESPAÑOL, GUIA INDEPENDIENTE PARA EL  
USUARIO DE COMPUTADORAS, VOLUMEN 5 - NUMERO 8.

[21] [HTTP:// WWW.AMAZON.COM](http://www.amazon.com).

# **ANEXOS:**



## BASES DE DATOS



CLAVE:  
PLAN: 93

LICENCIATURA (SEMESTRE): INFORMATICA (6º)  
DEPTO. ACADEMICO: INFORMATICA AVANZADA  
AREA: APLICACIONES  
REQUISITOS: NINGUNA ASIGNATURA

CREDITOS: 8  
HORAS POR CLASE: 2  
CLASES POR SEMANA: 2  
HORAS POR SEMESTRE: 68

OBJETIVO GENERAL: AL FINALIZAR EL CURSO, EL ALUMNO DISEÑARA E IMPLANTARA BASES DE DATOS EN LOS SISTEMAS DE INFORMACION ORGANIZACIONAL.

HORAS	TEMATICA	OBJETIVOS EDUCACIONALES	SUGERENCIAS DIDACTICAS	REFERENCIAS BIBLIOGRAFICAS																																	
15	<p>I. PLATAFORMA TEORICO CONCEPTUAL</p> <p>1. Datos. A) Fuentes de Datos. B) Modelado de Datos. Relaciones, entidades y atributos.</p> <p>2. Organización de archivos.</p> <p>3. Base de Datos. A) Concepto. B) Diccionario de Datos. C) Sistemas manejadores de base de datos (DBMS). a) Tipos. b) Lenguaje de definición y manipulación de datos.</p> <p>D) Introducción al diseño de Base de Datos. a) Modelo de Red. b) Modelo Jerárquico c) Modelo Relacional. E) Administración de Base de Datos.</p>	<p>1. PARTICULAR DE LA UNIDAD.</p> <p>Al finalizar la unidad el alumno organizará los datos en estructuras de red, jerárquicas o relacionales.</p> <p>2. ESPECIFICOS.</p> <p>El alumno será capaz de: Describir las relaciones entre los datos. Seleccionar la estructura de datos adecuadas a sus necesidades de información. Integrar los elementos necesarios para el manejo de bases de datos. Diferenciar las características de distintos manejadores de bases de datos.</p>	<p>Debate dirigido de las lecturas referidas en esta unidad.</p> <p>Conferencias.</p> <p>Prácticas en laboratorios de cómputo.</p> <p>Seminarios.</p> <p>Solución de casos prácticos propuestos por el profesor.</p> <p>Trabajos de revisión bibliográfica de libros y publicaciones no referidos en esta unidad.</p> <p>Lecciones, tutoriales y simuladores.</p>	<table border="1"> <thead> <tr> <th>No.</th> <th>Cap. o Unid.</th> <th>Págs.</th> </tr> </thead> <tbody> <tr> <td>2</td> <td>1</td> <td>2 a 26</td> </tr> <tr> <td>4</td> <td>1 a 5</td> <td>2 a 108</td> </tr> <tr> <td>4</td> <td>11</td> <td>301 a 320</td> </tr> <tr> <td>5</td> <td>1</td> <td>3 a 53</td> </tr> <tr> <td>5</td> <td>11</td> <td>299 a 336</td> </tr> <tr> <td>6</td> <td>1,2</td> <td>1 a 44</td> </tr> <tr> <td>3</td> <td>1 y 19</td> <td>11 a 20,</td> </tr> <tr> <td></td> <td></td> <td>479 a 511</td> </tr> <tr> <td>1</td> <td>1 a 5</td> <td>2 a 241</td> </tr> <tr> <td>12</td> <td>1</td> <td>1 a 14</td> </tr> </tbody> </table>	No.	Cap. o Unid.	Págs.	2	1	2 a 26	4	1 a 5	2 a 108	4	11	301 a 320	5	1	3 a 53	5	11	299 a 336	6	1,2	1 a 44	3	1 y 19	11 a 20,			479 a 511	1	1 a 5	2 a 241	12	1	1 a 14
No.	Cap. o Unid.	Págs.																																			
2	1	2 a 26																																			
4	1 a 5	2 a 108																																			
4	11	301 a 320																																			
5	1	3 a 53																																			
5	11	299 a 336																																			
6	1,2	1 a 44																																			
3	1 y 19	11 a 20,																																			
		479 a 511																																			
1	1 a 5	2 a 241																																			
12	1	1 a 14																																			

HORAS	TEMATICA	OBJETIVOS EDUCACIONALES	SUGERENCIAS DIDACTICAS	REFERENCIAS BIBLIOGRAFICAS
				Nº. Cap. o Unid. Págs.
16	<p>II. DISEÑO DE BASES DE DATOS</p> <p>1. Modelo Conceptual. A) Análisis de datos. B) Representación gráfica.</p> <p>2. Modelo Lógico. A) Mapeo del diseño. (relacional, jerárquico, red).</p> <p>3. Modelo Físico. A) Interfases entre usuario y Base de Datos. B) Modelo Interno (métodos de acceso). C) Modelo Externo (vistas de usuario). D) DBMS (red, jerárquico, relacional).</p>	<p>1. PARTICULAR DE LA UNIDAD.</p> <p>Al finalizar la unidad el alumno aplicará las diferentes fases de modelado para el diseño de bases de datos.</p> <p>2. ESPECIFICOS.</p> <p>El alumno será capaz de: Determinar las vistas de usuario necesarias para satisfacer las necesidades de información del mismo. Expresar el análisis de un sistema del mundo real en términos de estructuras de datos. Identificar las entradas y salidas de las etapas del diseño de bases de datos. Seleccionar de los recursos de cómputo necesarios para la implantación adecuada de un modelo físico.</p>	<p>Películas educativas.</p> <p>Debate dirigido de las lecturas referidas en esta unidad.</p> <p>Conferencias.</p> <p>Seminarios.</p> <p>Solución de casos prácticos propuestos por el profesor.</p> <p>Trabajos de revisión bibliográfica de libros y publicaciones no referidos en esta unidad.</p> <p>L e c c i o n a r i o s , tutoriales y simuladores.</p> <p>Películas educativas.</p> <p>Práctica de campo: diseño de una base de datos para un caso real.</p>	<p>2 2 a 9 27 a 302 4 6 a 10 109 a 300 5 7 a 10 175 a 298 6 3 a 7 45 a 223 3 3 a 6 21 a 138 1 4 a 11 136 a 515 12 3 a 7 3 a 200 7 7 a 10 409 a 640</p>
16	<p>III. ASPECTOS DE DESEMPEÑO</p> <p>1. Consultas SQL. 2. Implantación de la Base de Datos. 3. Operaciones de mantenimiento. A) Respaldo.</p>	<p>1. PARTICULAR DE LA UNIDAD.</p> <p>Al finalizar la unidad el alumno aplicará técnicas para garantizar el desempeño de la base de datos dentro de un sistema.</p>	<p>Debate dirigido de las lecturas referidas en esta unidad.</p> <p>Conferencias.</p>	<p>2 10 303 a 334 4 12,13 321 a 336 5 12 a 17 337 a 499 6 7 a 11 224 a 399 6 13,14 435 a 466 3 7 a 16 139 a 431</p>

HORAS	TEMATICA	OBJETIVOS EDUCACIONALES	SUGERENCIAS DIDACTICAS	REFERENCIAS BIBLIOGRAFICAS
	<p>B) Recuperación.  C) Reorganización y restructuración.  D) Monitoreo y ajustes del desempeño.  E) Seguridad y Privacidad.</p>	<p>2. ESPECIFICOS.  El alumno será capaz de:  Aplicar una herramienta de cuarta generación para la implantación y búsquedas en una base de datos.  Diseñar estrategias para el mantenimiento eficaz de las bases de datos.  Analizar el desempeño de las bases de datos en un sistema computacional.</p>	<p>Prácticas en laboratorios de cómputo utilizando una herramienta 4GL-SQL.  Seminarios.  Trabajos de revisión bibliográfica de libros y publicaciones no referidos en esta unidad.  Leccionarios, tutoriales y simuladores.</p>	<p>No. Cap. o Unid. Págs.  1 12 520 a 558  7 11 a 15 641 a 804</p>
15	<p>IV. TOPICOS AVANZADOS DE BASES DE DATOS</p> <p>1. Bases de Datos Distribuidas.  2. Base de Datos orientadas a objetos (OODB).  A) Conceptos de OODB.  B) Paradigma de OODB.  C) Lenguajes para OODB.  D) Aplicaciones.</p>	<p>1. PARTICULAR DE LA UNIDAD.  Afinalizar la unidad el alumno integrará enfoques novedosos al diseño de bases de datos.  2. ESPECIFICOS.  El alumno será capaz de:  Planear el almacenamiento de información en distintos dispositivos físicos y lógicos.  Construir una base de datos bajo el paradigma de bases de datos orientadas a objetos.</p>	<p>Debate dirigido de las lecturas referidas en esta unidad.  Conferencias.  Prácticas en laboratorios de cómputo.  Seminarios.  Solución de casos prácticos propuestos por el profesor.</p>	<p>13 1 a 21 1 a 406  4 14,15 357 a 384  5 4 a 6 77 a 174  5 18 491 a 512  6 12 391 a 434</p>

HORAS	TEMATICA	OBJETIVOS EDUCACIONALES	SUGERENCIAS DIDACTICAS	REFERENCIAS BIBLIOGRAFICAS Nº.      Cap. o Unid.      Págs.
<p style="text-align: center;">6 <hr/><u>68</u></p>	<p style="text-align: center;">EVALUACION DEL APRENDIZAJE</p>		<p>Trabajos de revisión bibliográfica de libros y publicaciones no referidos en esta unidad.</p> <p>Leccionarios, tutoriales y simuladores.</p> <p>Películas educativas.</p>	

## BIBLIOGRAFIA:

### BASICA.

1. ALJCE & H. TSAI, Sistemas de bases de datos: administración y uso, México: Prentice-Hall Hispanoamericana, 1990, 590 pp.
2. ATRÉ, S., Data Base, Structured techniques for design, performance & management, EEUU: John Wiley & Sons, 1980, 435 pp.
3. ESEN Ozkarahan, Data base management. Concept, design & practice, EEUU: Prentice-Hall, 1990, 557 pp.
4. GUILLENSON, Mark L., Introducción a las bases de datos, México: McGraw-Hill, 1988, 385 pp.
5. HANSEN, G.W. & J. V. HANSEN, Data base management and design, EEUU: Prentice-Hall, 1992, 548 pp.
6. KORTH, H.F. & A. SILBERSCHATZ, Fundamentos de bases de datos, México: McGraw-Hill, 1987, 525 pp.
7. WIEDERHOLD, G., Diseño de bases de datos, (2ª ed.), México: McGraw-Hill, 1985, 903 pp.

### COMPLEMENTARIA.

8. DATAPRO: Management of Applications Software, Vol. I y II, EEUU: McGraw-Hill, 1991, 532 pp., 512 pp.
9. DATAPRO: Management of Applications Software, Vol. I y II, EEUU: McGraw-Hill, 1991, 532 pp., 512 pp.
10. DATAPRO: Computer Systems Series SOFTWARE, EEUU: McGraw-Hill, 1991, 630 pp.
11. DATAPRO: Computer Systems Series SOFTWARE, EEUU: McGraw-Hill, 1991, 630 pp.
12. HAWRYSZKIEWYCZ, J. T., Database analysis and design, EEUU: MacMillan Publishing Company, 1984, 573 pp.
13. RAJIV Gupta & E. HORROWITZ, Object oriented databases with applications to Case, Networks & Visicad, EEUU: Prentice-Hall, 1991, 437 pp.

### PERFIL PROFESIOGRAFICO DEL DOCENTE:

Los requisitos que debe reunir el docente para impartir la asignatura de BASES DE DATOS son los siguientes:

#### ACADEMICOS.

Tener como mínimo la Licenciatura en Informática, Computación, o experiencia equivalente.

Tener conocimientos a nivel de comprensión del idioma inglés.

Tener conocimientos en el manejo de sistemas de cómputo.

#### PROFESIONALES.

Poseer al menos tres años de experiencia en el análisis, diseño e implantación de bases de datos.

#### DOCENTES.

Acreditar dos cursos de Didáctica de 30 horas en la Coordinación de Calidad Académica de la Facultad de Contaduría y Administración.

Tener experiencia en la impartición de las asignaturas del área de informática.

**SUGERENCIA DE EVALUACION:**

**PORCENTAJE DE LA  
EVALUACION FINAL**

Examen "A", puede ser equivalente a lo siguiente:

Practicar tres (3) exámenes durante el semestre.

50%

Los exámenes parciales comprenderán:

1er. parcial: Tema I y II

2do. parcial: Tema III

3er. parcial: Tema IV

Trabajos y tareas

40%

Participación en clase

5%

Asistencia a clase

5%

**TOTAL**

100%

Examen "B" lo presentará el alumno en el caso de no haber acreditado el examen "A"

100%

# GLOSARIO

## B

**Base De Datos:** Es el conjunto de tablas relacionadas entre sí con un mismo objetivo.

## C

**Campo:** Es el lugar en el que se guarda un dato con significado y con un mismo objetivo.

**Client/Server (Cliente/Servidor):**

Arquitectura donde el cliente es la máquina solicitante (computador personal o estación de trabajo) y el servidor es la máquina proveedora. El cliente suministra la interfaz del usuario y realiza una o la mayor parte del procesamiento de aplicación. El servidor mantiene las bases de datos y procesa las solicitudes del cliente para extraer o actualizar los datos de la base correspondiente. El servidor además controla la integridad y seguridad de la aplicación. Existen una diferencia con centralized processing, donde las terminales no inteligentes (no procesamiento) se conectan a un mini o a un mainframe.

## D

**Data Administrator (administrador de base de datos):**

Persona responsable del diseño físico y de la administración de la base de datos además de la evaluación, selección e implementación del DBMS. En organizaciones pequeñas, el administrador de bases de datos y el administrador de datos son una sola persona; sin embargo, cuando las dos responsabilidades son administradas en forma separada, la función del administrador de bases de datos es más técnica.

**Data Administration (Administración de datos):**

Análisis, clasificación y mantenimiento de los datos y las relaciones de éstos de una organización. Incluye el desarrollo de modelos y diccionarios de datos, que combinados con el volumen de transacciones, representan las materias primas para el diseño de bases de datos.

**Data bank (banco de datos):**

Cualquier depósito electrónico de datos.

**Data manager (administrador de bases de datos):**

⇒ Con computadores personales, software que permite a un usuario manejar múltiples archivos de datos (lo mismo que DBMS).

⇒ Software que provee la capacidad de gestión de bases de datos para lenguajes de programación tradicionales, como COBOL, BASIC y C, pero sin las capacidades interactivas.

⇒ La parte del DBMS que almacena y recupera los datos.

**Data element: (elemento de datos)**

Estructura fundamental de datos en un sistema de procesamiento de datos. Cualquier unidad de datos definida para procesamiento es un elemento de datos; por ejemplo: NUMERO DE CUENTA, NOMBRE, DIRECCION y CIUDAD. Un elemento de datos se define por su tamaño (en caracteres) y su tipo (alfanumérico, sólo numérico, verdadero/falso, fecha, etc). Un conjunto específico de valores o rango de valores también puede formar parte de la definición.

Técnicamente, un elemento de datos es una definición lógica de datos, mientras que un campo es la unidad física de almacenamiento en un registro. Por ejemplo, el elemento de datos NUMERO DE CUENTA, que sólo existe una vez, se almacena en el campo NUMERO DE CUENTA en el registro del cliente, como también en el campo NUMERO DE CUENTA en los registros de pedidos.

Data element (elemento de dato), data item (ítem de dato), field (campo) y variable describen la misma unidad de dato y se utilizan en forma distinta.

**DATA DEFINITION (Definición de datos):**

En un programa de lenguaje fuente, las definiciones de estructuras de datos (variables, arreglos, campos, registros, etc).

**Data management:(administración de datos)**

Se refiere a varios niveles de manejo de datos, a partir de métodos de acceso hasta administradores de archivos y DBMS para manejar los datos como un recurso organizacional.

**Data Independence:(Independencia de los datos)**

Técnica de DBMS que separa los datos desde el procesamiento y permite cambiar estructuralmente la base de datos sin afectar los sistemas existentes.

# I

**Información:** Es el conjunto de datos relacionados entre sí, que tienen un significado y una utilidad.

# O

**OSI (Open System Interconnection):**

Interconexión de sistemas abiertos

Estándar ISO(International Standards Organization) para comunicaciones a nivel mundial que define una estructura con el fin de implementar protocolos en siete estratos o capas. Es similar a las capas SNA (Systems Network Architecture (arquitectura de redes de sistemas) de IBM.

**Object-Oriented Technology:** ( tecnología orientada a objetos)

Variedad de disciplinas que respaldan la programación orientada a objetos (OOP), que incluyen análisis orientado a objetos (OOA) y diseño orientado a objetos (OOD).

**Object:(Objeto)**

En programación orientada a objetos, módulo autónomo de datos y su procesamiento asociado.

En un documento compuesto, bloque independiente de datos, texto o gráficas que se crearon mediante una aplicación separada.

**Object - Oriented Programming:(Programación Orientada a Objetos)**

Programación orientada a objetos Abreviada "OOP", tecnología de programación que es más flexible que la estándar. Es una forma evolutiva de programación modular con reglas formales que permite con mayor facilidad que segmentos de software sean reutilizados e intercambiados entre diversos programas.