UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

FACULTAD DE CONTADURIA Y ADMINISTRACION

SEGURIDAD Y REGLAS DE INTEGRIDAD EN BASES DE DATOS RELACIONALES

SEMINARIO DE INVESTIGACION INFORMATICA

QUE PARA OBTENER EL TITULO DE

L I C E N C I A D O E N I N F O R M A T I C A

P R E S E N T A

ULISES LAZARINI CASTAÑEDA

ASESOR DEL SEMINARIO:
ACT. FRANCISCO DAVID MEJIA RODRIGUEZ

MEXICO, D. F. **1996**

1997

AGRADECIMIENTOS

Al todo Poderoso por darme un suspiro de vida.

A la Universidad Nacional Autónoma de México por darme los conocimientos necesarios para poder desenvolverme profesionalmente, y por ser para siempre, mi alma mater.

Al Actuario David Mejía Rodríguez por sus sabios consejos, confianza y disponibilidad para la elaboración de esta tesis.

A la Dirección de Computo para la Administración Académica por brindarme su apoyo incondicional y proporcionare los instrumentos necesarios para mi completo desarrollo y capacitación.

A todos ellos gracias.

SEGURIDAD Y REGLAS DE INTEGRIDAD EN BASES DE DATOS RELACIONALES

DEDICATORIAS

A la mujer que ha estado conmigo toda mi vida y a hecho un gran esfuerzo para hacer que esto sea posible, mi Mamá.

A ti padre por ser mi guía y apoyarme en todos los aspectos.

A mis amigos por estar conmigo siempre y en especial a tí Aída Covarruvias por ser mi ángel de la guarda, por brindarme siempre todo tu apoyo ilimitado, y a ti Hugo Reyes por ser un buen compañero.

A todos ustedes gracias por creer en mi.

SEGURIDAD Y REGLAS DE INTEGRIDAD EN BASES DE DATOS RELACIONALES

ÍNDICE **PÁGINA**

OBJETIVO

Esta tesis tiene como objetivo el ser un apoyo que sirva como guía para la toma de decisiones a cerca del nivel de seguridad que deberá tener una base de datos al momento de hacer su análisis, diseño e implementación, ya que esta dará un funcionamiento adecuado a la organización y no será blanco fácil de los constantes ataques a las que se ven envueltas.

INTRODUCCIÓN

El avance de la tecnología aunado al no tener una adecuada seguridad ha originado que cada vez sea mayor el número de universidades y empresas públicas y privadas que hayan sufrido algún tipo de ataque a sus bases de datos, y es por eso que hoy día es mayor el número de organismos que están implementando sistemas de seguridad para proteger uno de sus principales recursos con que cuentan, su *información*.

La información que se almacena en una base de datos debe protegerse contra el acceso no autorizado, la destrucción o alteración con fines indebidos y la introducción accidental de inconsistencias. Actualmente no es posible proteger de manera absoluta la base de datos contra abusos ilícitos, pero, ¿qué puede hacerse para dificultar el acceso sin autorización a la base de datos?.

Este documento propone la implantación de políticas, técnicas y herramientas de seguridad sobre bases de datos, al igual que expone las ventajas que traerá consigo la utilización de estas para frenar los ataques y la salvaguarda de los datos.

Adicionalmente en varios de los capítulos aquí expuestos, se muestra la implantación de un sistema de seguridad, desde la creación de la base de datos hasta los mecanismos de seguridad para proteger los datos de ser blanco fácil de posibles ataques.

El capítulo I se menciona una introducción de las bases de datos, sus orígenes y su modo de operar.

El capítulo II habla del modelo relacional y como empezar crear bases de datos relacionales identificando las partes esenciales del modelo relacional.

El capítulo III explica la importancia de las reglas de integridad y como establecer relaciones.

En el capítulo IV se habla de los que es la seguridad de la información y de los diferentes tipos de controles que se pueden establecer en una base de datos.

El capítulo V trata de los diferentes mecanismos de los que nos podemos ayudar para tener un mejor control de la información.

El capítulo VI se analiza el estudio de un caso en donde se ve plasmado lo que se ha mencionado en los puntos anteriores.

CAPÍTULO I

ANTECEDENTES

1. GENERALIDADES SOBRE LAS BASES DE DATOS

1.1 Historia de las Bases de Datos

50's Procesamiento de datos
60's Surge COBOL
Unidad de datos (archivo)
Restricciones de archivos
Acceso secuencial
Acceso directo
Surge el conncepto DMS (Data Management System)
Evolución de apuntadores
Evolucionan los lenguajes de programación
Evolucionan los modelos jerárquicos y de red
70's Se crean estándares para bases de datos en red
La tecnologia de DBMS se empieza a comercializar
Se establecen las Reglas de Codd
80's DBMS basados en el modelo relacional de Codd
Utilización del modelo relacional en PC's y mainframes
Proceso distribuido de datos
90's Programación orientada a objetos
Bases de datos orientadas a objetos

En un principio se hacia uso de los a través de archivos planos (flat file) como lo hacen varios manejadores de datos y como también lo hace COBOL; en seguida se explicará lo que es un archivo plano.

Archivo Plano: Es aquel el cual no tiene formato alguno esta escrito de forma continua y sirve para el intercambio de información, utiliza sólo el juego de caracteres ASCII (American Standar Code). En seguida se presenta un ejemplo del procesamiento de un archivo plano de alumnos que se inscriben vía telefónica a la Facultad de Derecho de la UNAM.

89053968119066972893624891196449726593478327632843843943923732438895954743754549854909032843954905404358327342774389595432743854965490650659458438439054548549549065005443734854856432743854965490650659458438439054548549549065005443734854856432743854965490650659458438439054548549549065005443734854856

En este ejemplo vemos que los primeros 8 digitos son el número de cuenta de algún alumno de la universidad.

89053968

Los siguiente 4 caracteres pertenecen a la materia a la cual se encuentra inscrito.

1190

Los siguientes 2 caracteres pertenecen al grupo en el cual se encuentra el alumno

66

Y finalmente los últimos tres pertenecen al semestre en el que esta inscrito

972

Por lo regular estos archivos planos no poseen retornos de carro (enter's) al final de cada linea por lo que para poder trabajar con ellos es necesario utilizar un lenguaje de programación para poder darle formato alguno.

Actualmente se utilizan otras metodologías para el manejo de datos (como por ejemplo la más usada y conocida actualmente es el Modelo Relacional) actualmente casi se ha eliminado en su totalidad el uso de archivos planos.

1.2 Necesidad de las Bases de Datos.

Tradicionalmente los sistemas de información no se planean o diseña, sino que evolucionan como sistemas independientes que resuelven problemas aislados de una organización. El problema con este enfoque es que los procedimientos requeridos cambiarán de acuerdo a los cambios en el ambiente de la organización.

Al principio se tenían grandes volúmenes de archivos que formaban una gran Base de Datos, esta era manejada de manera manual de archivero en archivero hasta encontrar la información requerida. Esto funcionaba de manera ineficiente ya que para requerir un dato era una gran perdida de tiempo y consumía grandes volúmenes de espacio.

El reto es diseñar bases de datos estables que sean relativamente independientes de las aplicaciones que se construyan sobre ella. Para obtener mayor beneficio de este enfoque se debe analizar la información de la organización y planear su Base de Datos cuidadosamente. Si se trata de tomar este enfoque sin esta planeación, los resultados muy bien podrían ser desastrosos. En estos momentos ya no aplica el dicho de que el que tiene la información tiene el control, actualmente lo importante es, el que tiene la información y al momento es quien tiene el control

1.3. CONCEPTOS DE BASES DE DATOS

1.3.1 Definición de Bases de Datos

Dato:

Hechos, ideas o conceptos que pueden ser reunidos y representados electrónicamente en forma digital.

Cantidad mínima de información, hechos sin evaluar, valor sin significado.

Información:

Conjunto de datos interrelacionados entre si, que tienen un significado del cuál podemos obtener conocimientos para una futura toma de decisión.

Base de Datos

Conjunto de datos interrelacionados con independencia física y lógica, consistentes, integros y con redundancia controlada.

Una Base de Datos es una colección de datos interrelacionados, almacenados más o menos permanentemente es una computadora tal que: a) los datos son compartidos por diferentes usuarios y programas de aplicación, pero existe un mecanismo común para insertar, actualizar, borrar y consulta de los datos, b) tanto los usuarios finales como los programas de aplicación no necesitan conocer los detalles de las estructuras de almacenamiento.

1.3.2 Conceptos asociados a Bases de Datos

Redundancia. Repetición de los mismos datos o elementos a través de diferentes registros o archivos.

Integridad. Asegura que la información contenida en la Base de Datos sea correcta, que los datos sean exactos y de validez universal. La integridad implica en asegurar que lo que se trata de hacer es correcto.

Seguridad. Implica asegurar que los usuarios están *autorizados* para llevar acabo lo que tratan de hacer.

Seguridad de Objetos. Se refiere a los permisos a los diferentes usuarios para poder hacer uso de las tablas, procedimientos almacenados, triggers, etc.

Seguridad de operaciones. Aquí se manejan permisos para poder modificar (insertar, borrar, actualiza) la Base de Datos.

Independencia Lógica de los Datos. Un sistema ofrece Independencia Lógica de los datos si los usuarios y sus programas son asimismo independientes de la estructura lógica de la Base de Datos, esto es, el programa de aplicación puede cambiar sin afectar a los datos almacenados.

Independencia Física de los Datos. Los usuarios y sus programas no dependen de la estructura física de la Base de Datos almacenada. Las aplicaciones permanecen inalteradas cualquiera que sean los cambios efectuados en el almacenamiento o en los métodos de acceso.

1.3.3 Importancia de las Bases de Datos

- *Son compactas.*

No hacen falta archivos de papeles que pudieran ocupar mucho espacio.

- *Son rápidas.*

La máquina puede obtener y modificar datos con mucha mayor velocidad que un ser humano, así, es posible satisfacer con rapidez consultas de casos particulares, sin necesidad de búsquedas visuales o manuales que requieren mucho tiempo

- *Son menos laboriosas.*

Se elimina gran parte del trabajo de mantener archivos a mano, Las tareas mecánicas siempre serán mejor realizadas por las máquinas.

- *Son actuales.*

Se dispone en cualquier momento de información precisa y al día.

- *Capacidad*

El poder manejar una gran volumen de datos.

- *Tiempo de Respuesta*

Se realizan transacciones de manera más rápida y eficiente. Por ejemplo:
Ordenar un millón de registros 100 Mb.

- 1986. 3600 segundos.
- 1990. 980 segundos.
- 1992. 58 segundos.
- 1994. 8 segundos.

- *Disponibilidad*

Se garantiza el buen funcionamiento de la Base de Datos en el momento en que un usuario requiere un dato de la misma.

1.4. DEFINICIÓN DE SISTEMA MANEJADOR DE BASES DE DATOS (DBMS)

Entre la Base de Datos física (es decir, los datos tal y como están almacenados en la realidad) y los usuarios del sistema existe un nivel de programas, *manejador de bases de datos (MBD)* o, en la mayoría de los casos, el *sistema administrador de bases de datos* DBMS (Data Base Management System). El DBMS maneja todas las solicitudes de acceso a las Bases de Datos formuladas por los usuarios, para la adición y eliminación de archivos (tablas), la obtención y puesta al día de los datos de esos archivos o tablas, etc., estan todas incluidas en el DBMS. Así, una de las funciones generales del DBMS es distanciar a los usuarios de la Base de Datos de detalles al nivel del equipo (de manera muy similar a la forma como los sistemas de lenguajes de programación evitan a los programadores de aplicaciones la necesidad de ocuparse de detalles al nivel de la máquina), y hace posible sus operaciones (como por ejemplo las operaciones de SQL*[1]).

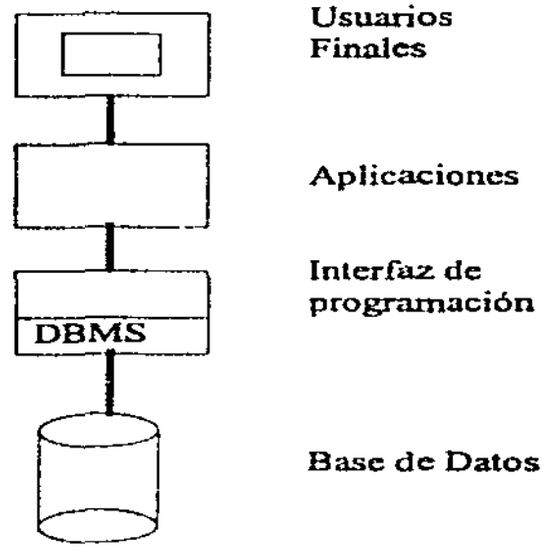
A un DBMS también se le conoce como el software que provee el mecanismo para definir, actualizar y acceder a los datos en una Base de Datos. Provee independencia de datos para programas y mantiene el control de redundancia.

Cuando un usuario pide una solicitud a la base de datos conceptualmente lo que sucede es lo siguiente:

a) un usuario solicita acceso, empleando algún sublenguaje de datos determinado (por ejemplo SQL)

b) el BDMS interpreta esa solicitud y la analiza

c) El DBMS ejecuta las operaciones necesarias sobre la Base de Datos almacenada



[1]SQL (Structure Query Language) lenguaje de consulta estructurada.

En resumen diremos que algunas de las funciones principales que tiene un DBMS es:

• **Disponibilidad de uso.** El sistema deberá dar respuesta inmediata a cualquier petición hecha por el usuario en condiciones normales.

•**Seguridad.** No cualquier usuario podrá acceder a los datos almacenados, esto dependerá de los privilegios que el DBA (administrador de las Base de Datos) le dé al usuario de acuerdo a su jerarquía.

• **Integridad.** La información contenida dentro de la Base de Datos deberá ser confiable y de valor universal.

• **Redundancia controlada.** Se deberá tratar de controlar al máximo la duplicidad de los datos.

• **Recuperación de datos.** Son mecanismos de eliminación de fallas y de la restauración de la Base de Datos, a nivel software.

1.4.1 Definición del Administrador de la Base de Datos

DBA (Data Base Administration)

Persona que tiene el control centralizado sobre el sistema de Base de Datos y que controla tanto los datos como los programas que tienen acceso a ellos.

Funciones:

- Decide el contenido de la Base de Datos
- Crea la estructura de almacenamiento y los métodos de acceso
- Modifica la Base de Datos o la descripción de la organización física
- Otorga permisos de acceso y prioridades a los diferentes usuarios
- Especifica las limitaciones de integridad
- Es el enlace con los usuarios
- Define estrategias para respaldo y recuperación
- Establece la seguridad necesaria para el manejo de la Base de Datos
- Define el nivel de acceso que tendrá un usuario sobre los datos de acuerdo a su jerarquía

1.4.2 Los enfoques en los DBMS

1.4.2.1 El enfoque Jerárquico

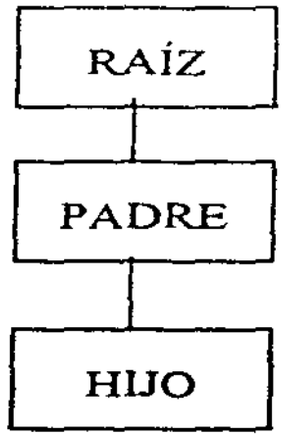
El modelo jerárquico utiliza una estructura jerárquica de árbol, que se construye de nodos y ramas; un nodo es una colección de atributos de datos que describen a la entidad de ese nodo. El nodo más alto de la estructura jerárquica de árbol se reconoce como raíz. Los nodos dependientes se encuentran en niveles más bajos del árbol. Los nodos hijos sólo pueden acceder por medio de sus padres.

Ventajas

- Relativa simplicidad y facilidad de uso
- Familiaridad de los usuarios

Desventajas:

- No modela de manea sencilla las relaciones uno a muchos
- Anomalías de inserción, borrado, actualización y consulta.

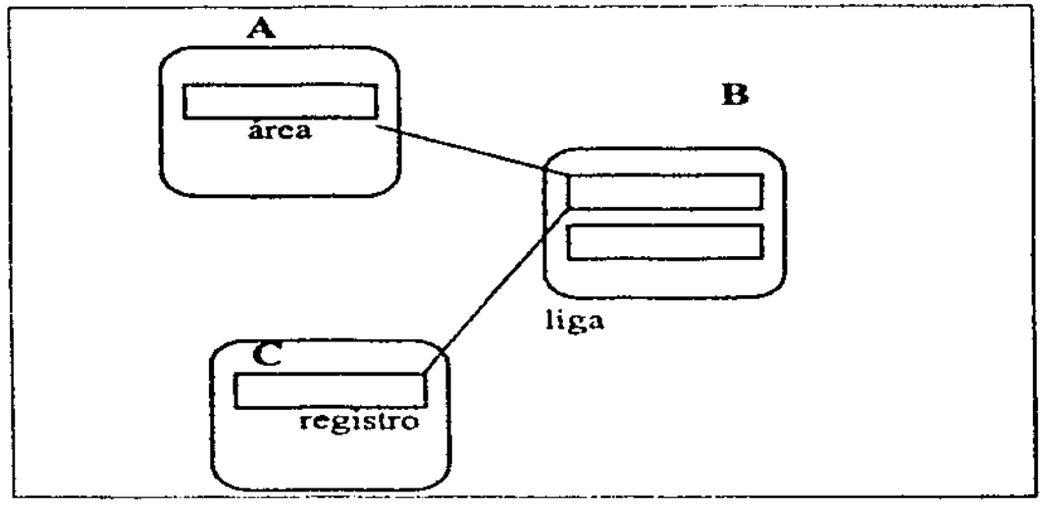


1.4.2.2 El enfoque de Red.

Está dividido en áreas, los registros se encuentran en las diferentes áreas, un mismo registro puede estar en varias áreas, las cuales se enlazan con ligas

Ventajas:

- La relación muchos a muchos se puede implantar fácilmente
- Esté modelo está respaldado por el Grupo de Tareas de Bases de Datos (dbtg) de CODASYL



Problemas en el uso de estos tipos de Manejadores

- Redundancia de datos innecesaria
- Inconsistencia (dada por redundancia no controlada)
- Falta de integridad en los datos (también por la redundancia)
- Falta de independencia lógica(aplicaciones) y física (equipo)
- Seguridad deficiente

CAPÍTULO II

SISTEMAS

RELACIONALES

CAPÍTULO II

SISTEMAS RELACIONALES

El siguiente capitulo entrará de lleno a lo que es el modelo Relacional y para hacerlo más claro de entender nos basaremos en un esquema propuesto y realizado de una Base de Datos de la Oficina de la Abogado General de la Rectoría de Ciudad Universitaria, la Base de Datos tiene por nombre CONVENIOS y su objetivo es que se encontrará la información en un DBMS relacional y que desde cualquier punto en donde una máquina tuviera red se pueda conectar a la Base de Datos y poder visualizar los datos y manipularlos. Se podrán conectar a ella todos aquellas máquinas y personas que tengan privilegios para poder acceder a los datos. La Base de Datos se constituye básicamente de tres tablas (relaciones) y con las cuales se trabajara el resto de la tesis en los siguientes capítulos.

EL ENFOQUE RELACIONAL

2.1 Introducción a los DBMS Relacionales

En 1968 en el IBM Research Laboratory en San José, California, con un modelo abstracto de información se inicio el trabajo que dio como resultado el modelo de datos relacional. El objetivo del trabajo era encontrar un fundamento teórico de los diferentes aspectos de un DBMS completamente ajeno de los aspectos de un proceso físico dentro de una máquina o CPU en particular. Este modelo es el que actualmente se conoce como Modelo Relacional o Estructura de Datos Relacionales. Una de las partes más importantes de este modelo es la introducción del concepto de tener en la estructura misma operadores lógicos. Edgar F.Codd matemático por carrera se dio cuenta que las matemáticas podían servir para introducir algunos principios sólidos y fue el álgebra relacional que ayudaba a operar estas estructuras. Estos operadores son los que se utilizan para toda la manipulación de datos en una estructura relacional.

El primer nombre que recibió el lenguaje manejador de datos fue ALPHA y posteriormente cambió a SQL (Structured Query Language). El trabajo en referencia originó que se empezará a diseñar y desarrollar sistemas basados en la teoría del modelo relacional.

Para estandarizar esto, el Instituto Americano Nacional de Normas (American National Standards Institute, ANSI) y la Organización Internacional para la Estandarización (International Organization for Standardization, ISO) han concedido adoptar el dialecto de SQL como interfaz oficial para sistemas relacionales.

SQL es una herramienta para organizar, gestiona y recuperar datos almacenados en una base de datos, usa una combinación de construcciones del álgebra relacional y del cálculo relacional. Ahora numerosos productos soportan el lenguaje SQL, aunque las versiones del producto de SQL difieren en varios detalles de los lenguajes, las diferencias son, para la mayoría sin importancia.

La potencialidad de SQL

SQL es tan poderoso en expresividad como el álgebra relacional. SQL incluye las operaciones fundamentales del álgebra relacional, por ejemplo el producto cartesiano se expresa por medio de la cláusula from de SQL, La proyección se expresa a través de la cláusula select. Los predicados de selección del álgebra se expresan a través de la cláusula where. SQL incluye también la unión y la diferencia, algo muy importante es que el resultado de cada una de las operaciones es otra relación y podemos trabajar sobre este resultado obteniendo otra relación.

¿Qué es una Base de Datos relacional?

Una Base de Datos relacional es aquella cuyos usuarios la perciben como un conjunto de tablas (y nada mas). En el siguiente ejemplo 2.1 veremos como los usuarios ven de manera abstracta una Base de Datos.

Catalogo_de_Dependencias

PK

IDependencia	SubDeP	[illegible]
721	01	AUDITORIA INTERNA
311	02	C. C. ATMOSFERA
211	05	C. C. Y D. DE E. LATINOAMERICANOS
612	01	C. E. LENGUAS EXTRANJERAS
211	02	C. E. SOBRE LA UNIVERSIDAD
311	09	C. ECOLOGIA

Convenios

PK PK PK

Año	IDependencia	NodoOficio	NodoEntrada	[illegible]	[illegible]	[illegible]	[illegible]	OBJETO
1995	721	7.1/0030	4143-001-	EQUIPOS PARA TEATROS,	UN AÑO	28/11/94	N	PROPORCION
1995	311	7.1/0034	4144-002-	CONSEJO NACIONAL DE			N	OTORGAR A
1995	211	7.1/0035	4145-003-	COMISION NACIONAL PARA EL	UN AÑO	6 /01/95	S	OPTIMIZAR
1995	612	7.1/0036	4146-004-	SONIA VARGAS LEON	UN AÑO	1 /01/95	N	OTORGAR EN
1995	211	7.1/0038	4147-005-	LA ASOCIACION DE	UN AÑO	26/12/94	S	RECIPROCID
1995	311	7.1/0039	4148-006-	SOCIEDAD MEXICANA DE	UN AÑO	30/01/95	S	RECIPROCID
1995	613	7.1/0050	4149-007-	LA ASOCIACION MEXICANA DE	UN AÑO	30/01/95	S	RECIPROCID

Ejemplo 2.1 Estructura de datos relacional

2.2 Elementos de un DBMS Relacional:

Lenguaje de Definición de Datos (DDL)
Lenguaje de Manipulación de Datos (DML)
Diccionario de Datos (DD)

DDL (Data Definition Lenguage). Lenguaje de definición de datos, con el cual es posible definir o declarar los objetos de la base (tablas, tipos de datos, índices, reglas, defaults, vistas, triggers, procedimientos almacenados).

Desde el punto de vista del usuario, las principales proposiciones del DDL son:

CREATE TABLE	CREATE VIEW	CREATE INDEX
ALTER TABLE	CREATE VIEW	
DROP TABLE	DROP VIEW	DROP INDEX

Se explicará brevemente a que hace referencia cada una de las proposiciones antes mencionadas:

CREATE TABLE: Sirve para crear una [2]tabla base, su formato es el siguiente:

```
CREATE TABLE  Catalogo_de_Dependencias
( IDEPENDENCIA          CHAR(2)     NOT NULL,
  SUBDEPENDENCIA        CHAR(2)     NULL,
  DEPENDENCIA           CHAR(20)    NULL,

  PRIMARY KEY     (IDEPENDENCIA));
```

El resultado de esto es crear una nueva tabla base llamada Catalogo_de_Dependencias.

ALTER TABLE.- Con altar table podemos alterar una tabla base ya existente agregando una columna nueva a la derecha su sintaxis es la siguiente:

```
ALTER TABLE       ADD UBICACION CHAR (20);
```

El resultado aquí es el que se añade una columna nueva llamada ubicación que es de tipo char de 20 caracteres. Con este comando alteramos una tabla base.

[2]Una tabla base es aquella con existencia física, de tal manera que los registros se encuentran almacenados físicamente.

DROP TABLE.- Con este comando es posible eliminar una tabla base existente.

```
DROP TABLE      Catalogo_de_Dependencias;
```

Las siguientes sentencias sirven tanto para crear un índice (CREATE INDEX índice)como para borrar un índice (DROP INDEX índice) a continuación veremos un ejemplo de cada uno:

```
CREATE INDEX     XID on Catalogo_de_Dependencias (Idependencia);
```

El resultado de este comando es que crea un índice llamado XID en la tabla Catalogo_de_Dependencias en el campo Idependencia.

Para borrar un índice la sentencia es la siguiente:

```
DROP INDEX       XID
```

El resultado de este comando es que se borra el índice

Las siguientes sentencias sirven para crear vistas; Una vista se especifica como una tabla virtual[3] que por lo regular son un conjunto de tablas algunas de ellas vistas en el sentido SQL y otras tablas base, para crear una vista se define de la siguiente manera:

```
CREATE VIEW Convenios
AS SELECT, Dependencia, Año, NodeRegistro, Vs_Parte
    FROM   Catalogo_de_Dependencias, Datos
    WHERE Dependencia = Auditoria Interna
```

En este ejemplo creamos una vista que contiene los campos, Dependencia, Año, NodeRegistro, Vs_Parte.

La cual es obtenida de las siguientes tablas: Catalogo_de_Dependencias, Datos, con la condición de que Dependencia sea igual a Auditoria Interna.

[3]Una tabla virtual es aquella que no esta almacenada físicamente aunque los usuarios la ven como si fuera una tabla reales.

Para borrar un vista se ejecuta lo siguiente:

DROP VIEW Convenios;

DML (Data Management Lenguage): Todo el software que se puede utilizar para hacer uso de los datos (manipularlos).

Un ejemplo es SQL, los siguientes comandos son propios del SQL y estos nos ayudan a poder hacer uso de los datos.

select (seleccionar)
update (actualizar)
insert (insertar)
delete (borrar)

No es intención ofrecer una descripción completa del *lenguaje de manipulación de datos (DML)*, sólo se pretende ilustrar sus características principales.

SELECT; funciona para hacer selecciones de los datos los cuales deseamos ver, a continuación veremos una consulta sencilla:

```
SELECT Convenios.NodeRegistro, Convenios.NodeOficio
FROM Convenios
WHERE ((Convenios.CveDep="112"));
```

y el resultado que esta consulta arroja es la siguiente:

[illegible]	[illegible]
4230-088-27-II-	7.1/0387
4247-105-7-III-95	7.1/0428
4264-122-15-III-	7.1/0484
4273-131-28-III-	7.1/0571
4281-139-30-III-	7.1/0860
4286-144-3-IV-	7.1/
4291-149-4-IV-	7.1/0877

En este ejemplo se muestra la forma más común de hacer una consulta (SELECT) de SQL: SELECT selecciona los campos especificados, FROM (de) la tabla especificada, WHERE (donde) para que cumpla con una condición especificada. Lo importante a resaltar es que el resultado de la consulta es otra tabla.

UPDATE (actualizar); Sirve para actualizar o modificar una tabla, por ejemplo la sintaxis para este ejemplo sería:

```
UPDATE Convenios SET Convenios.Vigencia="dos anos",
Convenios.Ingresos="s"
WHERE ((Convenios.NodeRegistro="4143-001-5-I-95"));
```

El resultado de está actualización es:

Vigencia	Ingresos
dos anos	s

DELETE (ELIMINAR) Sirve para eliminar todos los registros de una tabla especificada que cumplan con la condición dada, su sintaxis es la siguiente:

```
DELETE NodeRegistro
FROM Convenios
WHERE NodeOficio = 7.1/0387
```

El resultado de está sentencia es que borra todos aquellos registros del campo NodeRegistro de la tabla convenios que cumplan con la condición de que NodeOficio se igual a 7.1/0387.

Hay que tener especial cuidado a la hora de eliminar registros por que puede llegar a provocar una violación de la integridad referencia. (El punto de Reglas de Integridad Referencia lo trataremos con especial cuidado en el capítulo III).

NOTA

Hay destacar que DELETE no es lo mismo que DROP, ya que con el primero borramos los registros de una tabla y con el segundo borramos la tabla físicamente, por ejemplo:

```
DELETE
FROM
WHERE
```

En este ejemplo lo que estamos haciendo es borrar todos los registros que cumplen con una condición dada, pero no borramos la tabla.

DROP TABLE Convenios

En este otro ejemplo se está borrando la tabla Convenios.

***INSERT** (insertar)* Esta instrucción sirve para añadir un registro nuevo a una o varias tablas de pendiendo de la condición que se de por ejemplo:

INSET
INTO
VALUES

Hay que tener especial cuidado a la hora de insetar por que puede llegar a provocar una violación de la integridad referencial. (El punto de Reglas de Integridad Referencial lo trataremos con especial cuidado en un capitulo más adelante).

DD (Data Dicctionary)

Va a tener la información de todos los objetos de la Base de Datos, esto en ocaciones se denomina *metadatos*. Sus principales funciones son las siguientes:

- Describe todos los elementos en el sistema (flujo de datos, almacenes de datos, procesos).
- Los elementos se centran en los datos y en la forma en que están estructurados.
- Comunica los mismos significados para todos los elementos del sistema.
- Documenta las características del sistema.
- Facilita el análisis de los detalles para evaluar las características y determinar como deben realizarse los cambios.
- Localiza errores y omisiones en el sistema.

En esencia el DD (diccionario de datos), sirve para que la Base de Datos este documentada para saber exactamente en donde hay que hacer una modificación o simplemente para conocer el uso y naturaleza de los datos.

El diccionario de datos almacena información acerca de la estructura de la Base de Datos, y la información de autorización, como las restricciones de la clave.

En un sistema de bases de datos relacional se necesita saber información acerca de los datos y sus relaciones. Esta información se almacena en el *diccionario de datos, o catálogo de sistema*. Entre los tipos de información que el sistema debe contener:

- Los nombres de las relaciones.
- Los nombres de los atributos de cada relación.

- Los dominios de los atributos.

Los nombres de las vistas definidas en la Base de Datos y la definición de esas vistas.

Las restricciones de integridad de cada relación (por ejemplo, las restricciones de clave).

A demás de esto, es recomendable que en la mayoría de los sistemas se conserven los siguientes datos:

- Nombre de los usuarios autorizados.
- Información contable acerca de los usuarios.

En los sistemas que utilizan estructuras altamente sofisticadas para almacenar relaciones, pueden conservarse datos estadísticos y descriptivos acerca de las relaciones:

- Número de tuplas de cada relación.
- Método de almacenamiento utilizado para cada relación (por ejemplo, agrupado o sin agrupar).

Es importante almacenar la información de los índices de cada una de las relaciones:

- Nombre del índice.
- Nombre de la relación que se indexa.
- Atributos sobre los que está el índice.
- Tipo de indice.

Toda esta información constituye, de hecho, una Base de Datos en miniatura. Generalmente es preferible almacenar los datos acerca de la Base de Datos en la misma Base de Datos, ya que ocasionalmente en otros sistemas se almacena esta información en otro lado.

La elección precisa de cómo se van a representar los datos por medio de relaciones debe hacerla el diseñador del sistema. Una posible representación podría ser:

Esquema-catálogo-sistema = (nombre-relación, número-de-atributos)

Esquema-atributo = (nombre-atributo, nombre-relación, tipo-dominio, posición)

Esquema-usuario = (nombre-usuario, clave-codificada, grupo)

Esquema-índice =(nombre-índice nombre-relación, tipo-índice, atributos-índice)

Esquema-vista = (nombre-vista, definición).

2.3. Estructura Relacional

Tabla

Dentro del enfoque relacional una tabla es conocida como una **Relación**. Una relación es una tabla de dos dimensiones con las siguientes propiedades:

- Cada columna contiene valores relativos al mismo atributo, y cada valor de una columna de la tabla debe ser atómico (un solo valor).
- Cada columna tiene un nombre distinto (nombre del atributo), y el orden de las columnas no es importante (únicamente nos sirve para identificar al dato que contiene esa columna).
- Cada renglón es distinto, esto es, un renglón no puede duplicarse en otro para un grupo de columnas seleccionadas como llave
- Cada atributo no llave debe depender sólo de la llave de la relación no de ningún otro campo no llave.

Tupla. Conjunto de valores que componen un renglón de la relación

Grado de una tupla. Número de atributos que tiene una tupla (n de una n-tupla)

Cardinalidad. Número de tuplas de una relación

Dominio. Conjunto de todos los valores posibles para un atributo

2.3.2 Llaves Primarias y Llaves Foráneas

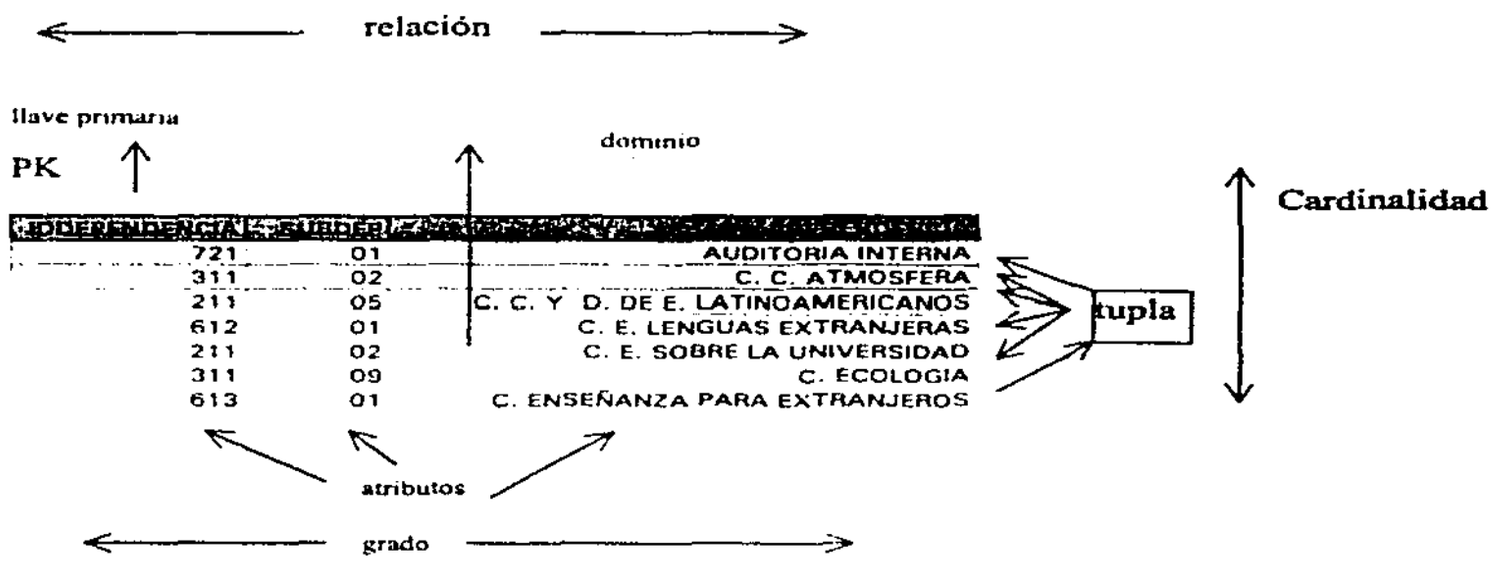
Llave primaria (PK). El atributo que identifica de manera única a un registro.

Llave extranjera o foránea (FK). Llave primaria que es la llave primaria en otra relación.

Estos términos se explicaran con mayor precisión en la parte de reglas de integridad.

ESTRUCTURA DE DATOS RELACIONAL

TABLA CATALOGO_DE_DEPENDENCIAS



TERMINO RELACIONAL	EQUIVALENTES
Relación	Tabla
Tupla	Fila o registro
Cardinalidad	Número de filas
Atributo	Columna o campo
Grado	Número de columnas
Llave Primaria	Identificador único
Dominio	Fondos de valores legales

Se ha estado haciendo mención del Modelo Relacional para el manejo de los datos, pero: ¿Cuándo se dice que una Base de Datos es Relacional?. Para responder a esta interrogativa nos basamos en las reglas de Edgar F. Codd, en la que menciona que cualquier Base de Datos que pretenda ser Relacional tendrá que cumplir con estas reglas.

2.3.3 Reglas de Codd

0. Cualquier BDMS que proclame ser relacional, deberá manejar, completamente, las bases de datos por medio de sus capacidades relacionales.

1. **Regla de información**. Toda la información dentro de una Base de Datos relacional se representa de manera explícita a nivel lógico y exactamente de una sola manera, como valores en una tabla.

2. **Regla del acceso garantizado**. Se garantiza que todos y cada uno de los datos (valor atómico) en una Base de Datos relacional pueden ser leídos recurriendo a una combinación del nombre de la tabla, valor de la llave primaria y nombre de la columna.

3. **El manejo sistemático de los valores nulos.** En un BDMS totalmente relacional se soportan los valores nulos (que son distintos de una cadena de caracteres vacías o de una cadena con caracteres en blanco o de cero o cualquier otro número), para representar información faltante o no aplicable de una forma consistente, independientemente del tipo de dato.

4. **Catálogo dinámico en línea basado en un modelo relacional.** La descripción de la Base de Datos se representa en el nivel lógico de la misma forma que los datos ordinarios, de tal suerte que los usuarios autorizados puedan aplicar el mismo lenguaje relacional para consultarla, que aquel que emplean para con sus datos habituales.

5. **Regla del sublenguaje de dato completo.** Contempla definición de datos, definición de vistas, manipulación de datos, restricciones de integridad, autorización, inicio y fin de una transacción.

6. **Regla de actualización de vistas.** Todas las vistas que teóricamente sean actualizables deberán ser actualizadas por medio del sistema.

7. **Inserción, actualización y eliminación de alto nivel.** La posibilidad de manejar una relación base o una relación derivada como un sólo operador se aplica a la lectura, inserción, modificación y eliminación de datos.

8. **Independencia física de los datos.** Los programas de aplicación y la actividad en terminales no deberán ser afectados por cambios en el almacenamiento físico de los datos o en el método de acceso.

9. **Independencia lógica de los datos.** Los programas de aplicación y la actividad en terminales no deberán ser afectados por cambios de cualquier tipo que preserven la información y que teóricamente permitan la no afectación en las tablas base.

10. **Independencia de la integridad.** Las restricciones de integridad de una Base de Datos deberán poder definirse en el mismo sublenguaje de datos relacional y deberán almacenarse en el catálogo, no en los programas aplicativos.

11. **Independencia de la distribución.** Un DBMS relacional tiene independencia de distribución.

12. **Regla de la no subversión.** Si un sistema relacional tiene un lenguaje de bajo nivel (un sólo registro cada vez), ese bajo nivel no puede ser utilizado para suprimir las reglas de integridad y las restricciones expresadas en le lenguaje relacional de nivel superior (múltiples registros a la vez).

2.3.4 Ventajas del uso de las Bases De Datos Relacionales

El uso de las Bases de Datos Relacionales tiene ventajas incomparables con respecto al funcionamiento de otras metodológicas en Bases de Datos, Así también cuenta con ciertas desventajas las cuales se mencionaran.

Ventajas.

- Acceso eficiente de datos
- Evita la redundancia
- Portabilidad
- Facilita el uso compartido de los datos
- Facilita y reduce el tiempo de desarrollo de las aplicaciones
- Acceso en línea y/o en batch (por lotes)
- Confiabilidad de los datos.
- Todos los datos en una Base de Datos Relacional se representan de una y sólo una manera.
- Combina tablas de manera general y elegante.

2.4 Modelo Entidad-Relación

El modelo entidad-relación (ER) se utiliza como una herramienta de comunicación entre los analistas y diseñadores de sistemas y los usuarios finales durante las fases de análisis de requerimientos y de diseño conceptual debido a que es simple y fácil de entender. Con esto se quiere decir que los *diagramas ER* son una técnica para representar gráficamente la estructura lógica de una Base de Datos. Como tal, ofrecen una forma sencilla y muy comprensible de comunicar el diseño de cualquier Base de Datos.

Este modelo fue introducido por Chen en 1976 y obviamente a nuestros días a cambiado la diagramación en parte, pero el concepto de la idea es la misma, representar en forma gráfica como se dará el manejo de los datos y quien interactuará con ellos.

2.4.1. Conceptos Básicos del Modelo Entidad-Relación

El modelo de datos entidad-relación se basa en una percepción, abstracción de un mundo real que consiste en un conjunto de objetos básicos llamados entidades, propiedades y relaciones.

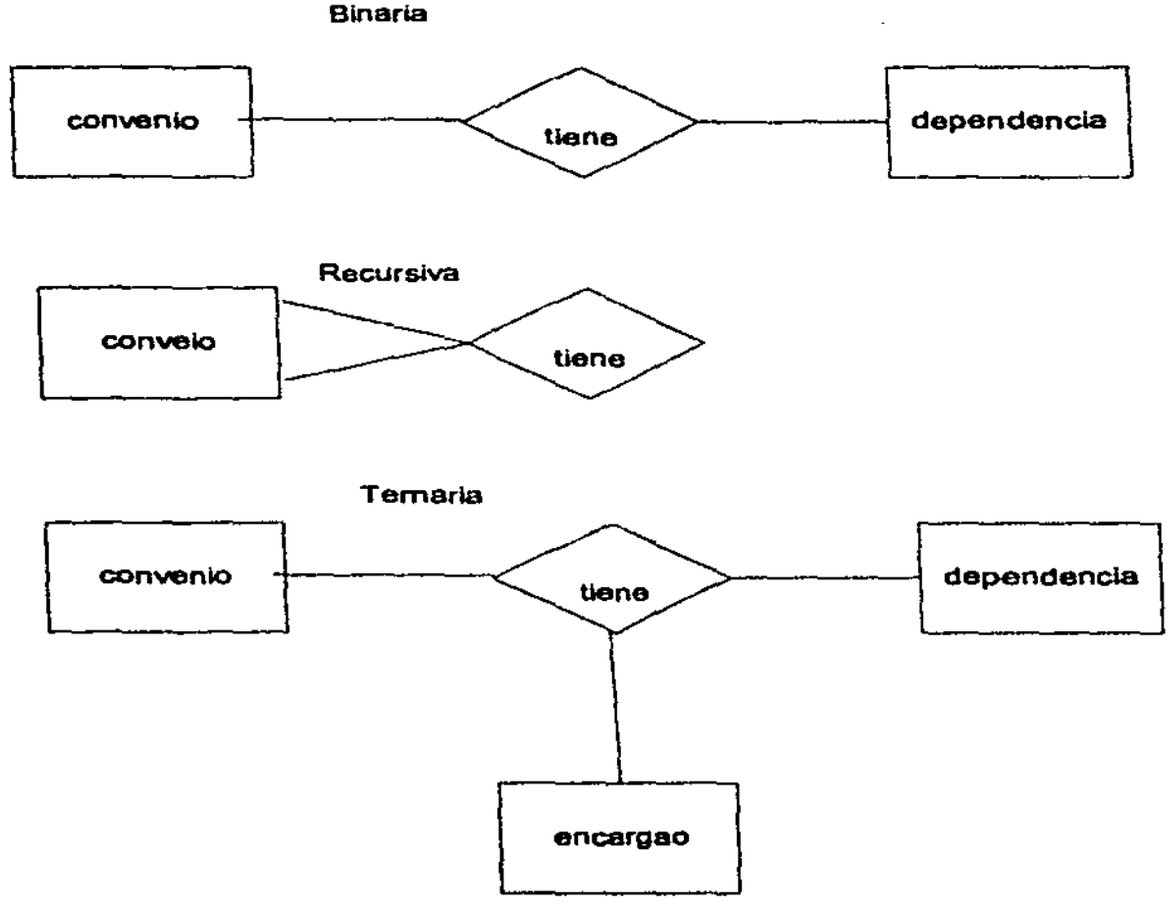
Entidad. Es el objeto principal del cual se tiene que almacenar información, normalmente denotando una persona, lugar, cosa o evento. En un diagrama entidad-relación (ER) las entidades se representan con un rectángulo; un sustantivo en español corresponde al nombre de la entidad en el DER.

Relación. Asocia una entidad de un conjunto a una o varias entidades de otro. Las relaciones se representan con un rombo con líneas conectando las entidades relacionadas; normalmente un verbo transitivo corresponde a la relación.

Propiedad. Las entidades tienen propiedades y esas son conocidas como sus atributos. Por ejemplo las propiedades de la relación Catalogo_de_Dependencias serian:

Independencia
Subdepe
Dependencia

Grado de una relación. Es el número de entidades asociadas en la relación. Una relación n-aria es de grado n. Las relaciones unarias, binarias (incluyendo las recursivas) y ternarias son casos especiales donde el grado es de 1, 2 y 3 respectivamente.

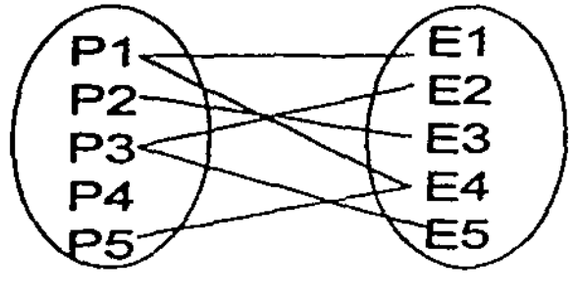

Binaria
convenio
tiene
dependencia
Recursiva
convelo
tiene
Ternaria
convenio
tiene
dependencia
encargao

2.4.2 Asociaciones

La conectividad de una relación especifica el tipo de asociación de las ocurrencias de las entidades de la relación. Los tipos básicos de conectividad son los siguientes: uno a uno, uno a muchos y muchos a muchos, la más común y usada es la asociación de uno a muchos.

EJEMPLO:

El siguiente DER se pueden ver conceptualmente como lo muestran las siguientes figuras:



Viendo la relación PROYECTO a la entidad EMPLEADO, la pregunta que se hace es ¿cuántos empleados trabajan en un solo proyecto?

El proyecto uno y el proyecto seis tienen más de un empleado trabajando en ellos. Por lo tanto, la cardinalidad es de uno a muchos.

Para cada convenio sólo hay una dependencia (relación uno a uno).

Otro caso es que una dependencia puede tener muchos convenios.

Por último, se tiene que dependencias tengan muchos convenios y viceversa

2.4.3 Conversión del Modelo Entidad-Relación al Modelo Relacional

De una manera muy simple se puede decir que las entidades del modelo ER corresponden a las tablas del modelo relacional y que las relaciones del modelo ER, si tienen campos, también corresponden a tablas del modelo relacional.

Para realizar un buen diseño es necesario tomar en cuenta consideraciones como la cardinalidad y el tipo de relación. A continuación se describen los pasos y consideraciones a seguir.

Relación de "uno a muchos".

En este caso el identificador de la entidad correspondiente a la cardinalidad "uno" para a ser la llave foránea de la tabla correspondiente a la entidad con cardinalidad "muchos".

Convenios

Año	IDepen	NoOfi	NoReg	Vsparte

CatDeps

IDepen	SubDep	Depend	....

Los atributos de la relación, en caso de que los tuviera, pasarían a ser campos de la tabla con la llave foránea.

Relación de "muchos a muchos".

En estos casos es necesario incluir una tabla que corresponda a la relación. Esta tabla contendrá los identificadores de las dos entidades asociadas y los campos propios de la relación.

Convenios

Año	IDepen	NoOfi	NoReg	Vsparte

CatDeps

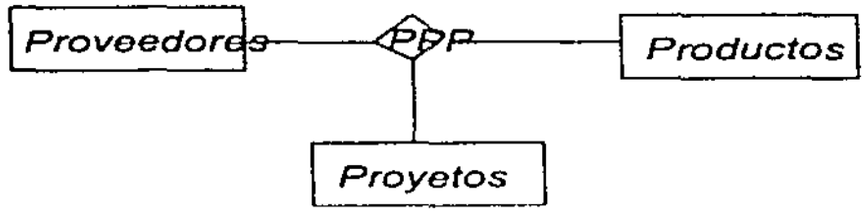
IDepen	SubDep	Depend	....

ConvDeps

NodeReg	IDepen	CveSubparte

Relaciones "n-arias".

Cuando se llega a tener una relación de grado 3, 4,... la relación se identifica con los identificadores de cada una de las entidades asociadas. Es por esto que cada entidad corresponde a una tabla, lo mismo que la relación, junto con sus campos.



PROVEEDORES

CProv	Direc	. . .

PROYECTO

CProy	Nombre	. . .

PRODUCTOS

CPar	Peso	Color	. . .

PPP

CProv	Cproy	CPar	. . .

Este último s ejemplos que se utilizó fue facilitar el entendimiento de lo que en esencia es el diagrama entidad relación DE/R.

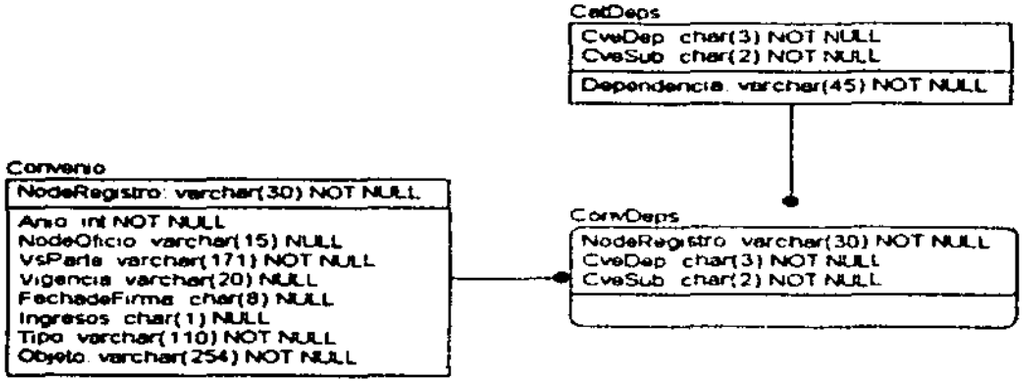
A continuación se mostrará el diagrama ER del sistema de Convenios. En este diagrama de ER contamos con tres entidades las cuales son:

ENTIDADES

CONVENIO.- Esta entidad es la que contiene toda la información de los convenios que la UNAM ha firmado con diferentes organismos. (No fue posible poner todos los campos de la que esta formado debido a espacio).

CATALOGO DE DEPENDENCIAS Contiene las Dependencias que la UNAM tiene.

DIAGRAMA ENTIDAD RELACIÓN (ER)



Ejemplo del diagrama de Entidad Relación (ER) de Convenios.

CAPÍTULO III

REGLAS DE INTEGRIDAD

REFERENCIAL

CAPÍTULO III

REGLAS DE INTEGRIDAD REFERENCIAL

En este capítulo se menciona lo que son las reglas de integridad, la gran importancia que estás tienen para las Bases de Datos Relacionales ya que estás se encargan de validar datos del mundo real, es por tal motivo que las reglas de Integridad Referencial nos ayudan a controlar el buen funcionamiento de la Base de Datos.

Asimismo se explicará con más exactitud lo que son las llaves primarias, candidatas y las llaves foráneas, de estos puntos se desprenden las restricciones de integridad que son impuestas a las Bases de Datos.

3.1 REGLAS DE INTEGRIDAD REFERENCIAL

Reglas de Integridad

El propósito de las reglas de integridad es informar al DBMS de ciertas restricciones en el mundo real para que pueda evitarse valores imposibles dentro de una tupla.

La integridad relacional previene la creación de "registros solitarios" que no tengan ninguna conexión con la tabla principal. Un ejemplo de registro solitario es que en la tabla Convenio contenga un valor nulo[1] en el campo Año de firma siendo que se tienen convenios a partir del año 1995. No existe ninguna referencia al valor nulo.

Cómo se mantiene la Integridad Relacional

La integridad relacional previene la eliminación o modificación de un registro principal del que dependen otros registros relacionados, por ejemplo:

Si se intentará eliminar un dato de una tabla relacional, primeramente el DBMS desplegará un mensaje en el que mostrará un aviso que se deberá de eliminar el campo de la tabla base y posteriormente todos los registros relacionados con el registro de la tabla principal.

Las reglas de integridad se va a mantener la Base de Datos con valores reales, relacionando con el ejemplo que se ha venido siguiendo tendríamos que:

- NodeRegistro de hasta máximo 30 caracteres.
- Los valores de FechadeFirma tendrían que se menor a la fecha actual.
- Las claves de las dependencias tendrán que venir de una lista
- La vigencia tendrá que se mayor de 0.
- La persona que hace uso de la información se identifique adecuadamente.

Estos son algunos de los puntos que con el uso adecuado de las reglas de integridad nos ayuda a mantener la Base de Datos consistente.

[1] En el capítulo 2.3.3 Reglas de Codd se hizo un estudio de lo que son valores nulos.

LLAVE

Llave, Puede ser una columna o conjunto de columnas, usado para identificar una tupla (registro).

Llave sencilla, Una llave que es un sola columna.

Llave compuesta, Una llave que es un conjunto de columnas.

Hay que tener especial cuidado al no confundir una llave con un indice,

Llave: Es una estructura lógica, me identifica una tupla (registro).

Indice: Es una estructura fisica, me sirve para buscar registros.

3.1.2 LLAVES PRIMARIAS

La llave primaria de una relación (también se le conoce como clave primaria) es un identificador único para esa relación. Por ejemplo, las llaves primarias de la Base de Datos de Convenios de la UNAM son: de Convenios es NodeRegistro, de Dependencias: CveDep CveSub y de ConvDeps: NodeRegistro CveDep CveSub. Hay que tomar en cuenta que una llave primaria puede ser compuesta, esto quiere decir que la llave primaria este formada de dos o más atributos como es el caso de las relaciones Dependencias y de ConvDeps.

Para poder obtener o seleccionar la llave primaria de una relación se tienen *llaves candidatas*, estas son: un conjunto de campos de las cuales nosotros seleccionamos una o un conjunto de llaves candidatas para hacerlas llaves primarias. Las llaves candidatas deben de reunir las siguientes propiedades:

1. *Unicidad:* En cualquier momento dado, no existen dos tuplas en R con el mismo valor de Y, siendo que Y es un atributo (posiblemente compuesto) de la relación R.

2. *Minimicidad:* Si Y es compuesto, no será posible eliminar ningún componente de Y sin destruir la propiedad de unicidad.

Como se mencionó en párrafos anteriores del conjunto de llaves candidatas se elige una y sólo una como llave primaria de esa relación, y las demás si existen, se llamarán *llaves foráneas*. Así una llave foránea es una clave candidata que no es la llave primaria, por ejemplo en la relación Dependencias CveDep y CveSub son las llaves primarias y Dependencia será la llave foránea. Con relación a esto surgen lo siguientes puntos:

• Toda relación tiene por lo menos una clave candidata, por lo tanto toda relación tendrá por fuerza una clave primaria.

• El razonamiento para elegir la llave primaria, en los casos en donde hay varias llaves candidatas, queda fuera del alcance del modelo relacional, en la práctica suele ser sencilla la elección, en ocasiones depende en gran medida de la experiencia que tenga el diseñador de la Base de Datos.

• En la práctica la llave primaria es la que tiene verdadera importancia ; las llaves candidatas y foráneas son sólo conceptos surgidos por fuerza durante el proceso de definir el concepto más importante, el de la "llave primaria".

Para resumir la importancia de las llaves primarias diremos que *constituyen un mecanismo de direccionamiento a nivel de tuplas básico en un sistema relacional*. Es decir, el único modo garantizado por el sistema, de localizar alguna tupla especifica es por su *valor de llave primaria.* Estas deben ser:

Única: No puede haber dos registros al mismo tiempo con el mismo valor de llave primaria. *Mínimo:* Todas las columnas dependen de ese valor, el valor debe de ser atómico. *No nullos:* En las claves principales no se aceptan valores nulos.

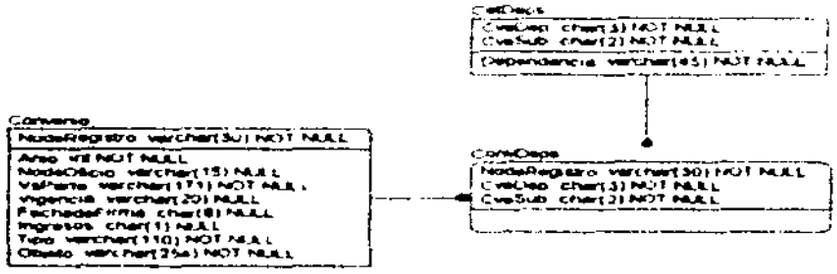


fig 3.1

3.1.3 Llaves Foráneas.

El problema de garantizar que la base de datos no contenga valores no válidos de una llave foránea se conoce como el problema de *Integridad relacional*. La restricción en la cual los valores de la llave foránea debe de concordar con los valores de la llave primaria correspondiente se conoce como *restricción referencial*, de manera formal los explicamos de la siguiente manera:

Existe una relación base R1 con llave primaria A tal que cada valor no nulo de B es idéntico al valor de A en alguna tupla de R1, como lo muestra el ejemplo3.1.

Con esto surgen los siguientes puntos:

Una llave foránea es una columna que hace referencia a una llave primaria en otra tabla.

Una llave foránea y la llave primaria correspondiente deben definirse sobre el mismo dominio.

Cualquier atributo (en una relación base) puede ser una llave foránea.

Regla de integridad de las relaciones

Dentro de estas reglas de integridad de las relaciones se desprenden dos principalmente, En la primera de ellas menciona:

Ningún componente de la llave primaria de una relación base puede aceptar nulos.

Con "nulos" (valores nulos) se hace referencia a información faltante, y no a valores extraños o blancos, la justificación de esta regla de integridad es la siguiente:

Las tuplas dentro de las relaciones base corresponden a entidades en el mundo real.

A las entidades dentro del mundo real se les puede identificar de alguna manera.

Las llaves primarias realizan la función de identificación única en el modelo relacional.

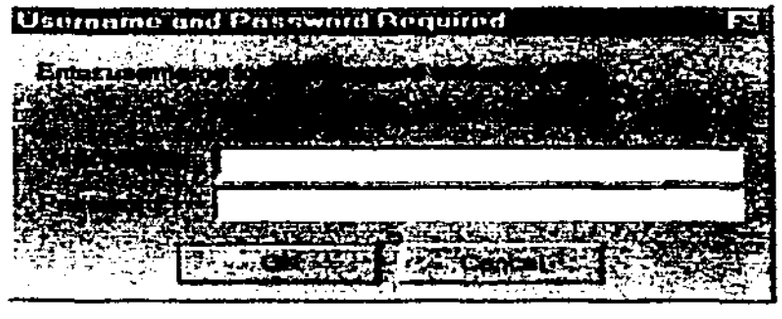
Con esto decimos que en *Una base de Datos Relacional, nunca registraremos información acerca de algo que no podamos identificar.*

La segunda regla de integridad del modelo relacional: *Las bases de datos no deben contener valores de llaves foráneas sin concordancia.*

La regla de integridad relacional dice que si B hace referencia a A, entonces A debe existir.

CAPÍTULO IV

SEGURIDAD



CAPÍTULO IV

SEGURIDAD EN LA INFORMACIÓN

El desarrollo de los sistemas electrónicos de información con el paso de los años se ha ido difundiendo y se ha hecho muy cotidiano, es así que en nuestros días, grandes corporaciones como lo son bancos, universidades, industrias de servicio, hospitales, organizaciones gubernamentales y privadas, manejan su información a través de los medios electrónicos con los que contamos actualmente.

La evolucióno en el desarrollo de tecnologías tanto de hardware como de software ha incrementado la participación profesional en el manejo de la información esto requiere de herramientas capaces de poder ayudar al hombre en el manejo de la información y es por eso que se ha difundido el uso de los servicios de la computación. Esto significa un mejor aprovechamiento de los datos almacenados en grandes volúmenes, lo cual no sería posible sin el uso de las computadoras, en donde toda la información almacenada o concurrente la podemos manejar fácilmente a través de lo DBMS relacionales.

La complejidad del diseño e implementación de sistemas de información seguros, depende en gran medida de varios factores que se deben de tomar en cuenta tales como la heterogeneidad de los usuarios que hacen uso de los datos, la extensión de los sistemas de información, lo incontrolable e impredecible en pérdida de información debido a diferentes factores como pueden ser: temblores, inundaciones, incendios, desastres naturales, etc.

"Los datos son el componente más valioso de un sistema de cómputo. "*Para mí, los datos de los usuarios son de importancia inigualada. Cualquier otra cosa es generalmente reemplazable. Se pueden comprar más discos, más computadoras, más energía eléctrica. Pero si se pierden los datos, por un incidente de seguridad o por cualquier otra cosa, se van para siempre.*" Russell Brand

4.1 ¿Qué es seguridad en Cómputo?

Términos como "seguridad", "protección", "privacidad", pueden tener más de un significado, dependiendo de quién lo aplica, y en que ámbitos. Incluso los profesionales que trabajan en el área de seguridad no siempre coinciden en lo que estos términos significan. Una definición bastante práctica de seguridad es:

"Un sistema, es seguro si se puede *confiar* en él y su software se comporta como los usuarios esperan que lo haga.

4.1.2 Seguridad En Bases De Datos.

La seguridad de la información en las Bases de Datos incluye tres principales aspectos: seguridad, integridad y disponibilidad.

Seguridad significa el prevenir y/o detectar la difamación de información. En general seguridad se refiere a la protección de los datos en diferentes ambientes, tanto en ambientes militares como en ambientes comerciales. Otro termino importante en el estudio de la seguridad de la información es el de la privacidad de los datos que se refiere a la información individual de cada individuo, grupo o institución para determinar cuando o que información le concierne para su propio propósito, y está a su vez puede ser almacenada o liberada para otras personas o entidades. Privacidad se refiere a los datos de las personas que están protegidos por las leyes o reglas dependiendo del país donde se encuentre. Por ejemplo las coordenadas de un blanco de un misil no deberían revelarse impropiamente, o en el ejemplo que hemos venido siguiendo de CONVENIOS DE LA UNAM, no sería conveniente que el sindicato sé de cuenta del presupuesto que la UNAM asigna a determinadas facultades, direcciones, investigaciones, o que si un convenio firmado con X entidad va a generar ingresos o no. En los ambientes comerciales se encuentra información secreta estrictamente, acompañada por sistemas de seguridad (llamados policías de seguridad) que aseguran que la información denominada como crítica por la organización tendrá una alta seguridad en donde los empleados no tendrán acceso a información que no les corresponde, por ejemplo un empleado no tendrá acceso a saber de cuanto es el salario de los gerentes. Estos son sólo algunos ejemplos en donde ponemos en claro que una Bases de Datos deberá tener ciertos niveles de seguridad.

Integridad de la información; significa el prevenir y/o detectar modificaciones impropias de la información, por ejemplo en un ambiente militar las coordenadas de un

blanco de un misil no deben ser modificadas impropiamente, en el caso de nuestro ejemplo de convenios no podemos permitir que modifiquen un convenio que tiene la UNAM con X empresa de duración de 5 años de mantenimiento de PC's que lo modifiquen y lo reduzcan a 3 años, en el ambiente comercial no se puede permitir que los trabajadores modifiquen su salario.

Disponibilidad del Sistema; significa la prevención y/o detección del mal funcionamiento al acceso de los servicios que el sistema brinda, por ejemplo en el ambiente militar, cuando se haga la petición de disparar el misil, el misil deberá ser disparado, en el caso de los convenios de la UNAM cuando hagamos la petición de un convenio este estará dispuesto a darnos la petición que le solicitamos.

En muchos ambientes necesitamos de estos tres aspectos que deben de manejar las Bases de Datos (Seguridad, Integridad de la Información y Disponibilidad del Sistema) tanto en ambientes de hospitales, aerolíneas, compañías de crédito, etc., estos aspectos son esenciales, un mal dato podría causarnos grandes problemas como pérdidas financieras y hasta pérdidas de vidas.

4.2 PROBLEMAS DE SEGURIDAD EN LAS BASES DE DATOS

En los ambientes de bases de datos, las diferentes aplicaciones y usuarios de una organización acceden a los datos a través del DBMS. Una de los principales problemas a los que se enfrenta un DBMS y debe resolver a través de una buena administración es el controlar que no halla datos duplicados, inconsistencia de los datos, y el manejo de amenazas de posibles usuarios no autorizados al acceso de los datos.

Una *amenaza* contra los datos la podemos definir de la siguiente manera: como una persona hostil que de manera casual o usando alguna técnica especial (usualmente denominamos como fuerza bruta[2], hace uso de técnicas especiales para poder acceder a los datos), para difamar o modificar los datos manejados por un sistema.

Las violaciones a la seguridad de las Bases de Datos consisten en, la lectura de datos y difamación de esos datos, modificaciones de los datos, y borrado de los datos, por personas no autorizadas. Las consecuencias de las violaciones de los datos están agrupadas dentro de 3 categorías:

[2] Se le denomina fuerza bruta al uso de varias computadoras ó estaciones de trabajo muy potentes con varios procesadores trabajando en paralelo y en tiempo real para poder romper una llave de acceso [password] y así poder entrar a un sistema y acceder los datos).

- *Difamación de los datos:* Esto es causado por lectura de datos de manera intencionalmente así como casualmente por usuarios no autorizados, incluyendo en esta categoría las violaciones de claves secretas de datos autorizados únicamente a ciertas personas.

- *Impropia modificación de los datos:* Esto en vuelve todas las violaciones a la integridad de los datos a través del manejo impropio de los datos o modificaciones de los mismos.

- *Mal funcionamiento del servicio:* Envuelve todas aquellas acciones que niegan el servicio a los usuarios al acceso de los datos o uso de los recursos.

Las amenazas contra la seguridad de los datos están clasificadas dentro de los siguientes factores en los que puede ocurrir de manera accidental o intencional:

- *Desastres naturales o accidentales:* tales como temblores, inundaciones o fuego. Estos accidentes pueden dañar los sistemas de hardware como los de almacenamiento de los datos, estos accidentes siempre causan violaciones de integridad y negación del servicio.

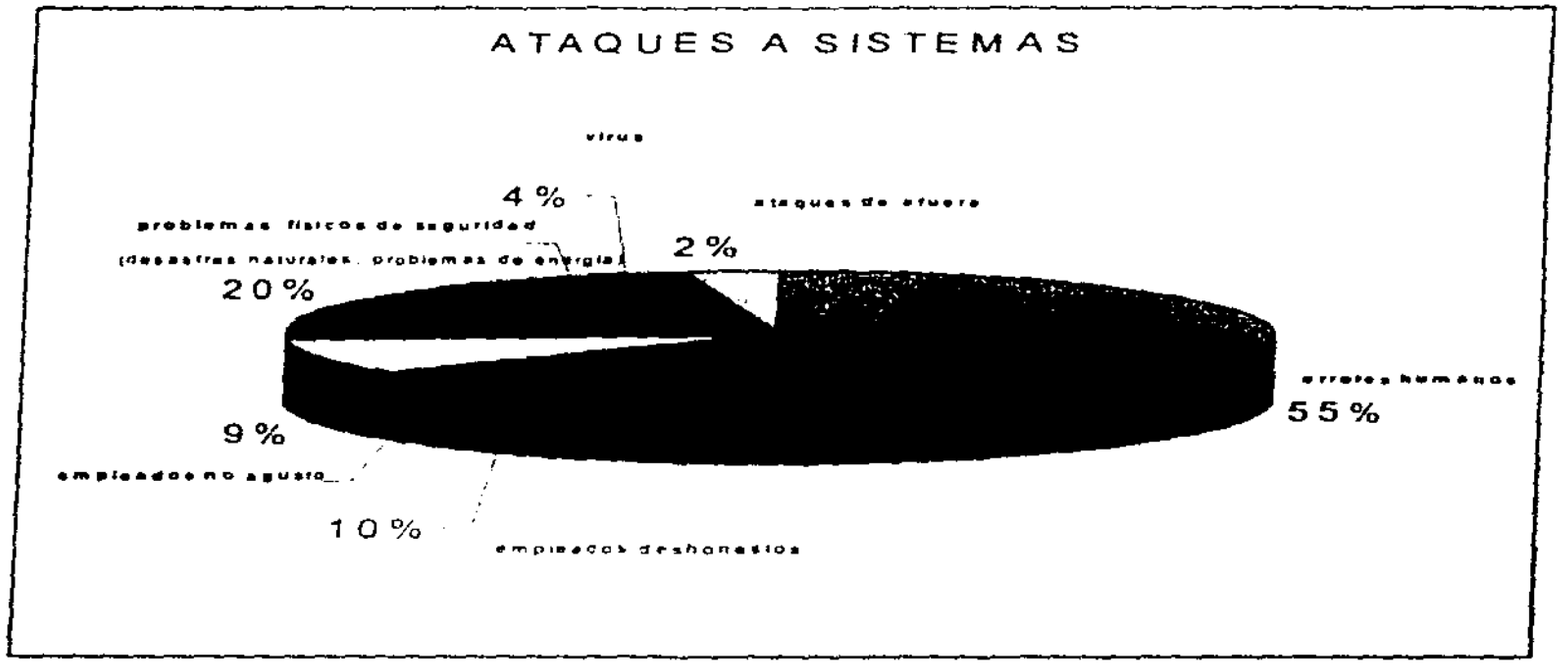
- *Errores o problemas en hardware o software:* Esto puede permitir la incorrecta aplicación de policías de seguridad y por consiguiente el acceso fácil para la lectura de datos no autorizados y modificación de los mismos o la negación del servicios a las personas autorizadas al uso y manejo de los datos.

- *Errores humanos:* violaciones al acceso del sistema no intencionadas como al dar una clave de acceso incorrecta o un mal uso de las aplicaciones las consecuencias son las mismas a las anteriores causando problemas en el sistema o falta de integridad de los datos.

Factores intencionales denotan explícita y determinadamente fraude que causa problemas en los accesos de los datos, las violaciones a los sistemas envuelven dos tipos de usuarios los cuales son:

1. *Usuarios autorizados:* Aquellos quienes abusan de sus privilegios y autorizaciones para hacer de los datos lo que a ellos mejor le convenga, como es el vender información dar accesos a ciertos datos o eliminar información.

2. *Agentes hostiles:* Este tipo de personas realizan acciones de vandalismo hacia el software y/o hardware, leyendo, modificando datos privados. En ambos casos "legal" o "ilegal", el uso que estas personas le dan a los datos puede ser para sus propósitos fraudulentos. Usualmente las personas hostiles (denominadas también *hackers*) suelen usar cierto tipos de técnicas para hacerse de información tales como: virus, caballos de troya y trampas de puerta, estás son algunas de las técnicas que las personas hostiles usan para hacerse de la información.



Tipos de violaciones a los sistemas

4.2.1 Requerimientos de Protección de una Base de Datos

Proteger una base de datos de posibles amenazas significa proteger sus recursos particularmente los datos almacenados. La protección de la Base de Datos se refiere á accidentes así como de accesos no autorizados para la lectura y/o modificación de los datos.

Protección de accesos no autorizados.

Esto consiste en garantizar el acceso a la base de datos sólo a usuarios autorizados. El acceso a la base de datos debe ser revisado por el DBMS dependiendo de los recursos y aplicaciones que el usuario va a ejecutar. Los controles de acceso son un punto muy complejo para las Bases de Datos debido a los archivos que son manejados por el sistema operativo. El cargado de dar los privilegios necesarios para que los usuarios puedan tener acceso a los recursos de Base de Datos es el DBA de acuerdo a los siguientes aspectos:

Lo necesario que necesite el usuario para llevar acabo sus actividades y

De acuerdo a las políticas que se manejen dentro de la organización.

Integridad de la Base de Datos.

Este requerimiento compete a la protección de la base de datos de accesos no autorizados que pueden modificar el contenido de la base, tanto de virus, sabotajes, o fallas en el sistema que pueden dañar los datos almacenados. Este tipo de protección es atendido por el DBMS a través de controles adecuados del sistema, y de tener respaldos, y procedimientos automáticos de recuperación de datos.[3]

Los respaldos y los procedimientos de recuperación de datos son procesos que constantemente se están estudiando para implementar un mejor funcionamiento en el desarrollo de los DBMS. Para perseverar la consistencia de los datos se requiere que cada transacción sea atómica.

[3] (cron: es el nombre que se le da en UNIX a un conjunto de instrucciones para que ejecuten una actividad cuando se presenta algún factor que lo origina, en este caso este cron podría ejecutar un procedimiento de recuperación.)

Las transacciones atómicas se refieren a:

- El término de una transacción correcta, modificando el acceso de los datos.
- El término de una transacción no exitosa, sin modificar los datos al hacer el acceso a estos.

Después que se ha realizado una transacción correctamente los datos modificados son de manera permanente, hasta que otro proceso los vuelve a modificar.

El sistema de recuperación de datos usa un log diario, normalmente es un archivo que contiene toda la secuencia de las operaciones realizadas sobre los registros dentro de una tabla (relación) de almacenamiento. Por cada transacción realizada, el log diario graba las operaciones que se han ejecutado sobre los datos, ya sea para lectura, escritura, inserciones y borrado, como también las operaciones de control como son las de: begin transaction, commit, abort, todas aquellas operaciones que se ejecutaron sobre los datos y no tuvieron éxito.

Auditoria y Contabilidad del sistema

Estos puntos consisten en la posibilidad de recobrar y verificar todos los accesos hechos a los datos en operaciones de lectura y escritura. Tanto la auditoría como la contabilidad son herramientas útiles para llevar un control de la integridad física de los datos así como también el análisis de acceso a la base de datos. En ocasiones puede ser que el elevado número de accesos a la base de datos haga que la revisión de los registros almacenados de todas las entradas sea impráctica desde el punto de vista de tiempo y dinero, pero es tarea del administrador de la base de datos tener el control pleno del sistema.

Uso de autentificación

Este requerimiento determina la necesidad de identificar únicamente la base de datos del usuario y al usuario con la base de datos. La identificación del usuario debe ser pedida por mecanismos del propio sistema, cada vez que el usuario intente hacer uso de los datos, los usuarios tendrán permitido el acceso a los datos y con ciertos privilegios que el administrador les dé, siempre y cuando se identifiquen con el sistema

Protección de Multinivel

La protección de multinivel significa diferentes niveles hacia los datos dependiendo de su importancia de los mismos, como se menciono anteriormente habrá datos de dominio público y otros más que de uso exclusivo, es por eso la necesidad de clasificar los datos dependiendo de su importancia. Por ejemplo en un ambiente militar los datos que se tengan será de uso exclusivo del ejercito, en un ambiente comercial, tendrán datos públicos y también tendrán datos privados, es por eso que acudimos a los diferentes niveles de seguridad.

A continuación se muestran dos diferentes tipos de clasificación que bien empleadas ofrecen un eficaz sistema de seguridad, una clasificación es según el libro naranja, y la segunda es el modelo *Bell-La Padula*:

EL LIBRO NARANJA

Llamado oficialmente "Department of Defense Trusted Computer System Evaluation Criteria. Este documento establece los requerimientos que debe cumplir un sistema para poder ser calificado formalmente como confiable. El concepto central del libro Naranja es que es posible medir la confianza que se pone en un sistema. El libro Naranja define cuatro divisiones jerárquicas de seguridad, y cada una de ellas se divide en clases:

División	Clase	Descripción
D		Protección mínima
C		Protección discrecional
	C1	Protección de acceso controlado
B		Protección obligatoria
	B1	Protección por etiquetas
	B2	Protección estructurada
	B3	Dominios de seguridad
A		Protección verificada
	A1	Diseño verificado

El modelo Bell-La Padula de seguridad

- La información se clasifica de manera jerárquica.
- Cada usuario tiene prioridades de acceso bien definidas
- Ningún usuario puede obtener información en una clasificación más alta de la que él tiene. (llamada "no read up")
- Ningún usuario puede reclasificar información es decir, escribirla en una clasificación menor (llamada "no write down")

4.3 Policías de Seguridad en la Base De Datos

Los policías de seguridad son sistemas de alto nivel en la cual su filosofía es principal es el diseño y manejo de sistemas de autorización de los datos. Generalmente las organizaciones escogen sus datos básicos que van hacer puestos a disposición de los policías de seguridad. La definición de policías de seguridad permite la explícita formulación de estrategias de seguridad, con esto se menciona que los Policías de Seguridad definen que datos serán utilizados o denegados. Las reglas de autorización son expresadas a través de los policías de seguridad, ellos determinan la conducta y administran los datos a la hora de la corrida de los mismos (inserciones y modificaciones). Ahora surge una pregunta que tanta información le corresponde a un sujeto, la respuesta a esto esta relacionado con la limitación de acceso, y para esto existen dos tipos de policías básicos:

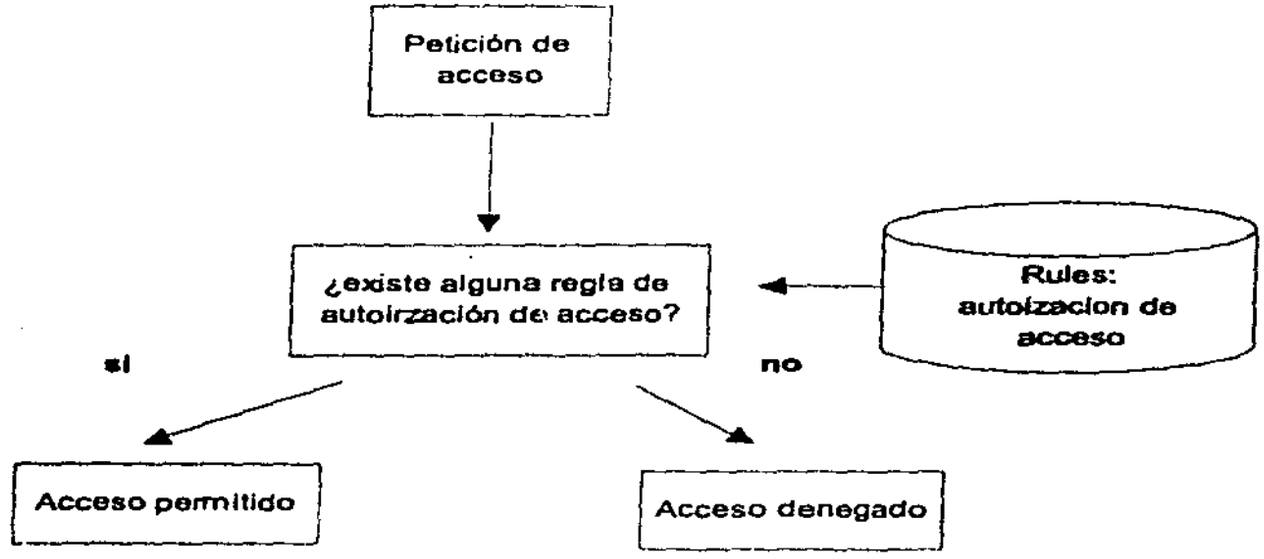
Privilegio mínimo. También conocido como "need-to-know" (el necesario para saber). Acordando con este policía de seguridad, el sujeto sólo tendrá acceso necesario a una mínima cantidad de información para realizar su trabajo.

Privilegio máximo. Basada en el principio de máxima disponibilidad en la base de datos, este tipo de policías es adecuado en ambientes tales como universidades y centros de investigación, donde no es necesaria una estricta protección.

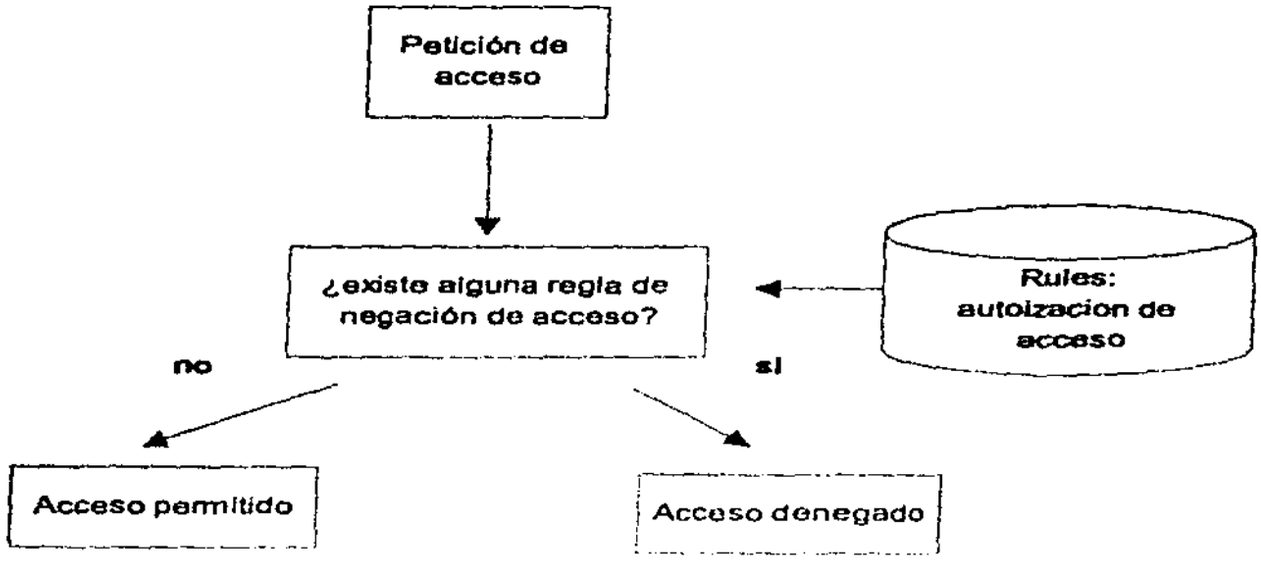
4.3.1. Sistemas cerrados y Sistemas abiertos

En *sistemas cerrados* sólo explícitas autorizaciones de acceso son permitidas.. Un policía en un sistema abierto se encuentra en un estado que por cada individuo hay una autorización estas autorizaciones trabajan sobre reglas (rules) específicas acerca de los privilegios de acceso que cada individuo tiene sobre los datos. Las reglas dan derechos de acceso a los usuarios a través de mecanismos de control.

En *sistemas abierto* el acceso no es prohibido Un policía en sistema abierto a diferencia del anterior encuentra en un estado que por cada sujeto, existe una regla (rule) de autorización acerca de los privilegios que el sujeto no tiene sobre los datos. En este caso los derechos del sujeto que van a ser denegados por mecanismos de control.



Control de acceso de un sistema cerrado



Control de acceso en un sistema abierto

En un sistema de seguridad quien es el encargado de dar los permisos de acceso (grant) o las limitaciones (no accesos "revoke") es el administrador de la base de datos.

4.4 *Diseño de la seguridad en una Base de Datos Relacional.*

El nivel lógico de la base de datos aborda el problema de la seguridad (seguridad e integridad) a través de un conjunto de reglas que establecen la autorización de acceso a la información de la base de datos y a sus recursos. Estas reglas tienen que ser propiamente definidas y aplicadas básicamente a los requerimientos de seguridad, evitando inconsistencias y errores que podrían exponer al sistema de posibles ataques. La seguridad lógica tiene que ser parte integral de la seguridad global del sistema de la organización.

El diseño lógico del sistema seguridad significa diseñar el software de seguridad, las reglas de seguridad. El software de seguridad comprende la seguridad de los paquetes, como la seguridad operativa del sistema, la seguridad del DBMS.

Las reglas de seguridad tienen que ser definidas correctamente y consistentemente, tomando en cuenta los diferentes requerimientos de seguridad de los usuarios y tratando de balancear los aspectos de integridad y seguridad del sistema.

Diseño de la seguridad del DBMS

La seguridad de una base de datos es llevada acabo a través de un conjunto de mecanismos en donde involucra al SO (sistema operativo) y al DBMS.

La implementación de los niveles de seguridad y de las funciones de seguridad es impuesta por el nivel del SO que puede ser explotado por el DBMS. En particular el manejo de funciones de I/O (de entrada y salida) y el manejo compartido de los recursos que proveen utilidad para el manejo de la seguridad por el DBMS es compartida por el SO. Por lo tanto las funciones de seguridad del DBMS no deben considerarse una simple extensión de las funciones del SO. Hasta hoy la seguridad entre el SO y DBMS tienen un manejo diferente que puede ser listado de la siguiente manera:

Seguridad de objeto: El nivel de seguridad de objeto en el SO es a nivel de archivo, mientras que la seguridad de objeto en un DBMS es refinado (por ejemplo, renglones, columnas, campos, con esto decimos que es más elegante el uso de los datos). La refinada seguridad para compartir datos es algo requerido por los niveles del DBMS Relacional.

Correlación semántica entre datos: En una base de datos los datos tienen semántica y a su vez esta es relacionada con otros datos. Consecuentemente, diferentes tipos de controles de acceso tienen que ser impuestos, dependiendo del contenido del objeto y contexto, esto en orden para asegurar la correcta implementación de los requerimientos de seguridad, y por consiguiente evitar las violaciones de seguridad relacionando con las correlaciones semántica entre los datos.

Metadatos: En los DBMS existen los metadatos, los metadatos proveen información acerca de la estructura de los datos en la base de datos. Los metadatos son usualmente almacenados en el diccionario de datos separado de los datos. Por ejemplo, en las Bases de Datos relacionales, los metadatos describen atributos, dominios de atributos relaciones entre atributos, y particiones locales de la base de datos. Los metadatos requieren protección tanto como los datos mismos la requieren. De hecho los metadatos pueden proveer de información acerca del contenido de la base de datos (tipos de datos y relaciones) y puede ser usado como método para controlar accesos para ocultar datos (por ejemplo, en SQL la creación de una "vista" para proteger datos que no se desea que los usuarios vean). Por el contrario no existe ningún nivel de metadato en SO.

Objetos físicos y lógicos: Los objetos en el SO son objetos físicos (por ejemplo archivos memoria, periféricos, procesador). Los objetos en el DBMS son objetos lógicos (por ejemplo relaciones vistas). Los objetos lógicos en el DBMS son independientes de los objetos físicos, y estos requieren mecanismos de seguridad específicamente orientados a la protección de las bases de datos.

Múltiple tipo de datos: Las Bases de Datos son caracterizadas por una variedad de tipos de datos para cualquier modo de acceso que sea requerido (por ejemplo para un modo estadístico o un modo administrativo). En el nivel físico de acceso al SO sólo puede ser para lectura, escritura y operaciones de ejecución.

Objetos estáticos y dinámicos: Los objetos manejados por el SO son estáticos y corresponden a objetos actuales. En Bases de Datos los objetos son creados de manera dinámica (por ejemplo, el resultado de una consulta). En Bases de Datos relacionales las vistas son creadas de manera dinámica, como relaciones virtuales derivada de relaciones base almacenadas en la base de datos.

Transacciones de multinivel: En los DBMS es a menudo necesario perfeccionar las transacciones que envuelven datos de diferentes niveles de seguridad. El DBMS debe asegurar que las transacciones multinivel son ejecutadas de una manera segura. En los niveles del SO sólo cuenta con las operaciones básicas (por ejemplo leer, escribir y ejecución) sobre los datos con el mismo nivel de seguridad.

4.5 El modelo de autorización del sistema R

Para ilustrar lo que se ha mencionado con anterioridad se demostrará como se diseña la seguridad en los sistemas DBMS relacionales, este modelo fue desarrollado en los laboratorios de IBM en San Jose California. (IBM Research Laboratory).

La autorización del sistema R considera: la protección de objetos, en esencia las tablas de la base de datos. Estas pueden ser tablas base o vistas, y estas se tienen que proteger contra los usuarios quienes tienen acceso a las bases. Los privilegios considerados por el modelo son aplicables a las tablas de la base de datos. En particular, siguiendo los modos de acceso se considera:

Read : Lectura de tuplas (registros) de una tabla. La autorización para la lectura

perteneciente a un usuario es algo definido en la vista de una tabla.

Insert: Agregar tuplas a una tabla.

Delete: Borra tuplas de una tabla.

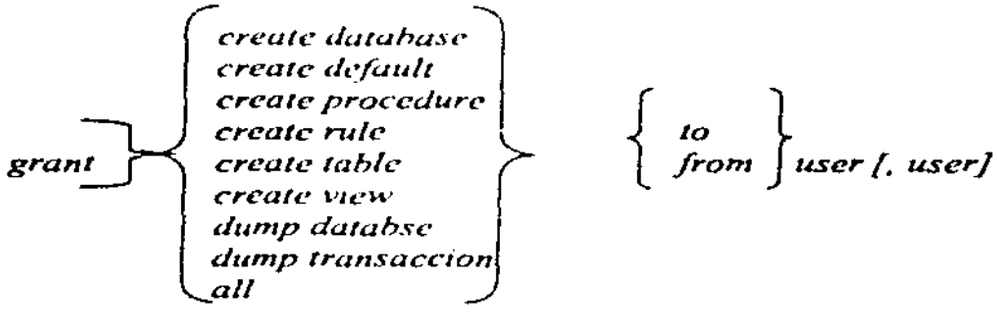
Update: Modifica tuplas existentes en una tabla.

Drop: Borra una tabla completa del sistema.

Este modelo acepta una administración descentralizada de autorización. En particular cualquier base de datos un usuario puede ser autorizado para crear una nueva tabla. Cuando un usuario crea una tabla el es único y completamente autorizado para ejecutar privilegios en la tabla. Como dueño, el usuario es el único titular en borrar una tabla o vista. El dueño puede autorizar a otros usuarios privilegios en la tabla. Cuando los dueños dan privilegios a otros usuarios en una tabla, un conjunto de autorizaciones son insertadas en el sistema. Los privilegios pueden ser otorgados con el comando grant, que significa *el objeto tiene permisos para realizar ciertas actividades sobre otros usuarios.*

Por ejemplo:

Control de acceso a comandos



Ejemplo:

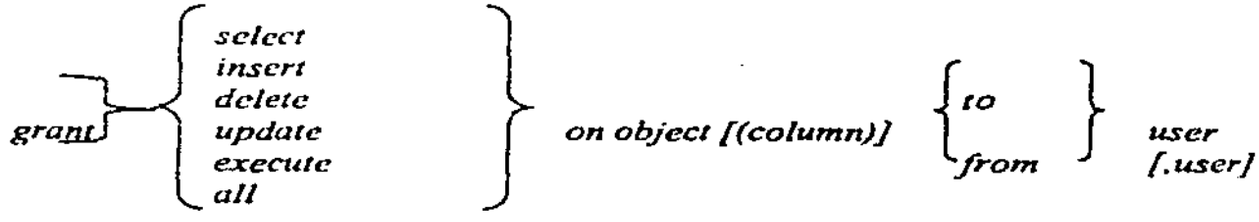
grant all to fred , Significa que fred tiene todos los derechos sobre la base.

grant create procedure to public , Significa que cualquier usuario tiene derecho para crear un procedimiento, siempre y cuando el usuario pertenezca a la base de datos.

Es importante resaltar que:

- Todos los usuarios pueden crear tablas temporales y
- Sólo el administrador del sistema puede conceder permisos (grant) o quitar (revoke) para la creación de Bases de Datos.

Control de accesos a objetos



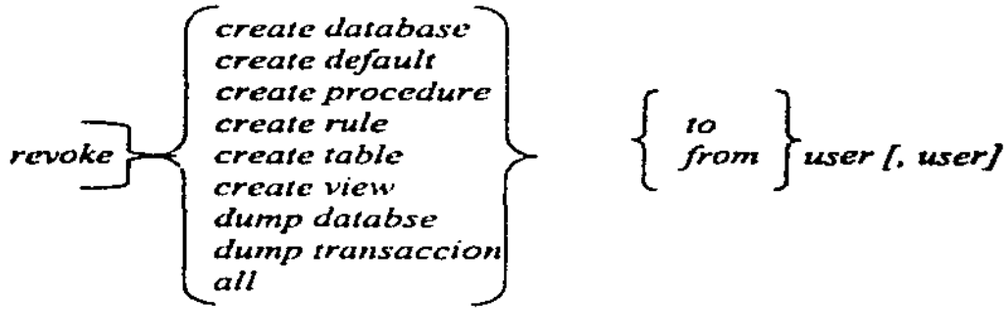
ejemplo:

grant update on titles (title_id) to fred, mary , Aqui estamos dando permisos a fred y a mary para que puedan actualizar el campo title_id de la tabla titles.

Revocación de Autorizaciones.

Revocación de autorizaciones significa quitar, eliminar ciertos privilegios de ver o modificar datos no disponibles para un usuario x, en otras palabras más formales diríamos; revocar el privilegio *p* en *t* del usuario *x* dado por *y* En el modelo R la revocación trabaja de manera recursiva de la siguiente manera:

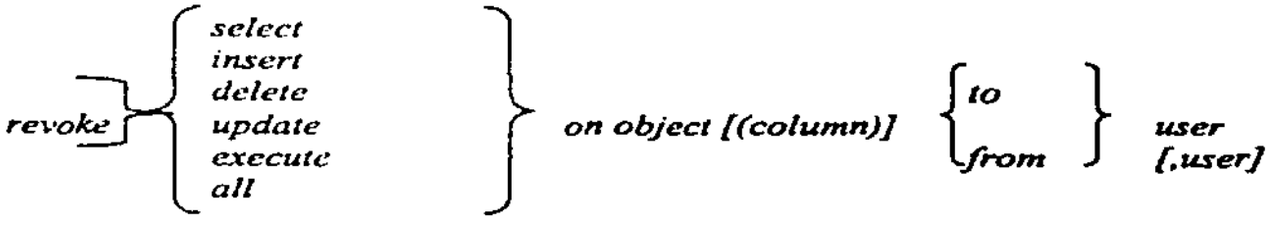
Revoke en acceso a comandos



ejemplo:

Revoke create view , create table from mary, john **en este ejemplo explica que no hay permisos para crear una vista de mary, o john.**

Revoke en accesos a objetos



ejemplo:

revoke select on convenio from clerk, **no hay permisos para que clerk seleccione convenios.**

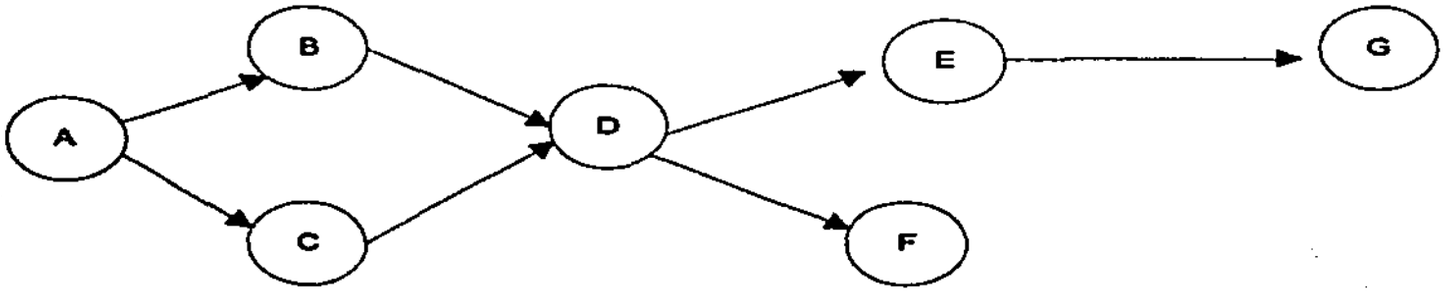


Fig.1

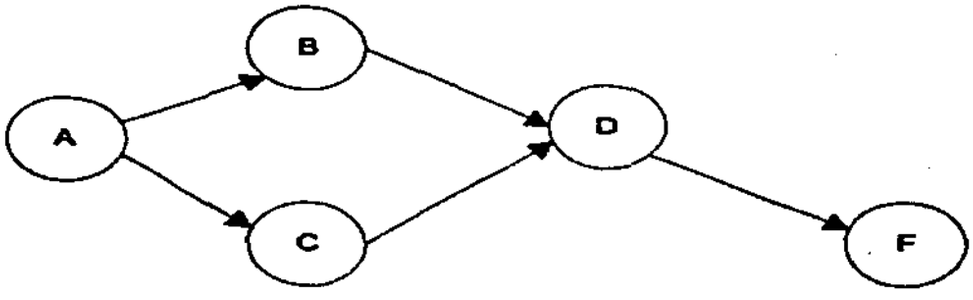


Fig.2

El algoritmo para revocar trabaja de la siguiente manera:

Cada nodo nos muestra un usuario, el figura 1 se ve claramente que tanto B como C tienen acceso a D y este a su vez tiene acceso a E y F y sólo E puede ejecutar o tener acceso a G.

Mientras que en la figura 2, únicamente C tiene acceso a D y por consiguiente podrá ejecutar o tener acceso a G.

4.6 Vistas (Views)

Las vistas son una palabra reservada del SQL (y en general de todos los sistemas relacionales) y se refiere a una tabla virtual derivada, con nombre propio, y en si, una vista es un conglomerado de varios atributos de una o varias relaciones (tablas) en la tabla virtual, está es creada en el nivel interno por el DDL. La finalidad principal es poder juntar varios atributos de varias relaciones para poder observar únicamente los datos que sean utilizados frecuentemente o para proteger datos que no queremos que sean del dominio público, por ejemplo el sueldo de un trabajador no sería conveniente que lo viera y lo comparara con el de su superior, ya que podría causar serios problemas.

A continuación mostramos la relación EMPLEADOS,

EMPLEADOS

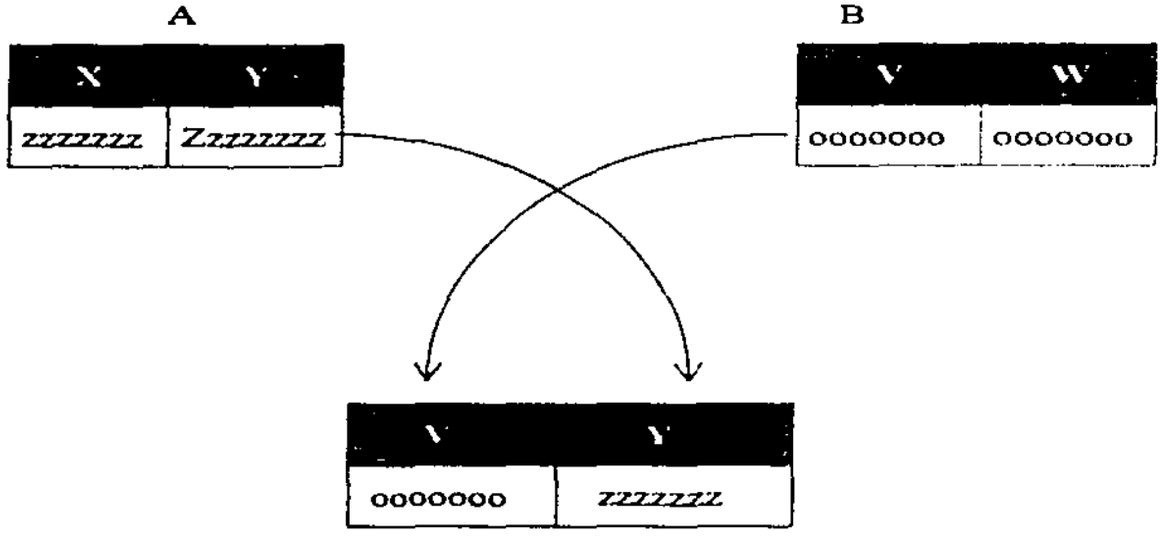
ID de empleado	Nombre	Teléfono	Depto	ID de categoría
Dcaa12	Hugo Alonso	606-8808	integración	C
Dcaa13	Silvia	781-9900	Nomina	A
Dcaa14	Luis Antonio	657-8934	Contabilidad	B
Dcaa15	Héctor	762-9327	Analista	C

Y también tenemos la relación sueldos que esta conformada de la siguiente manera:

SUELDOS

ID de categoría	Importe del pago
A	$2,000.00
B	$2,500.00
C	$3,000.00

A continuación, veremos la utilidad de manejar vistas, cuando surge la necesidad de tener información permanente para tomas de decisiones y por supuesto esconder información importante, todo esto es hecho por el DBA. Supongamos que queremos ver información referente a los empleados, en donde nos interesa únicamente el ID de empleado, nombre y a que ID categoría pertenece, el resultado de la creación de una vista sería:



De manera general esto es lo que sucede cuando creamos una vista, **el resultado de la** creación de la vista aplicando la interrogativa anterior es:

Emplea_catego

dcaa12	Hugo Alonso	A
dcaa13	Silvia	B
dcaa14	Luis Antonio	C
dcaa15	Hector	

Se puede ver que únicamente tenemos la ID_de_empleado, **nombre, ID_de_categoria. La** forma sintactica de crear la vista es:

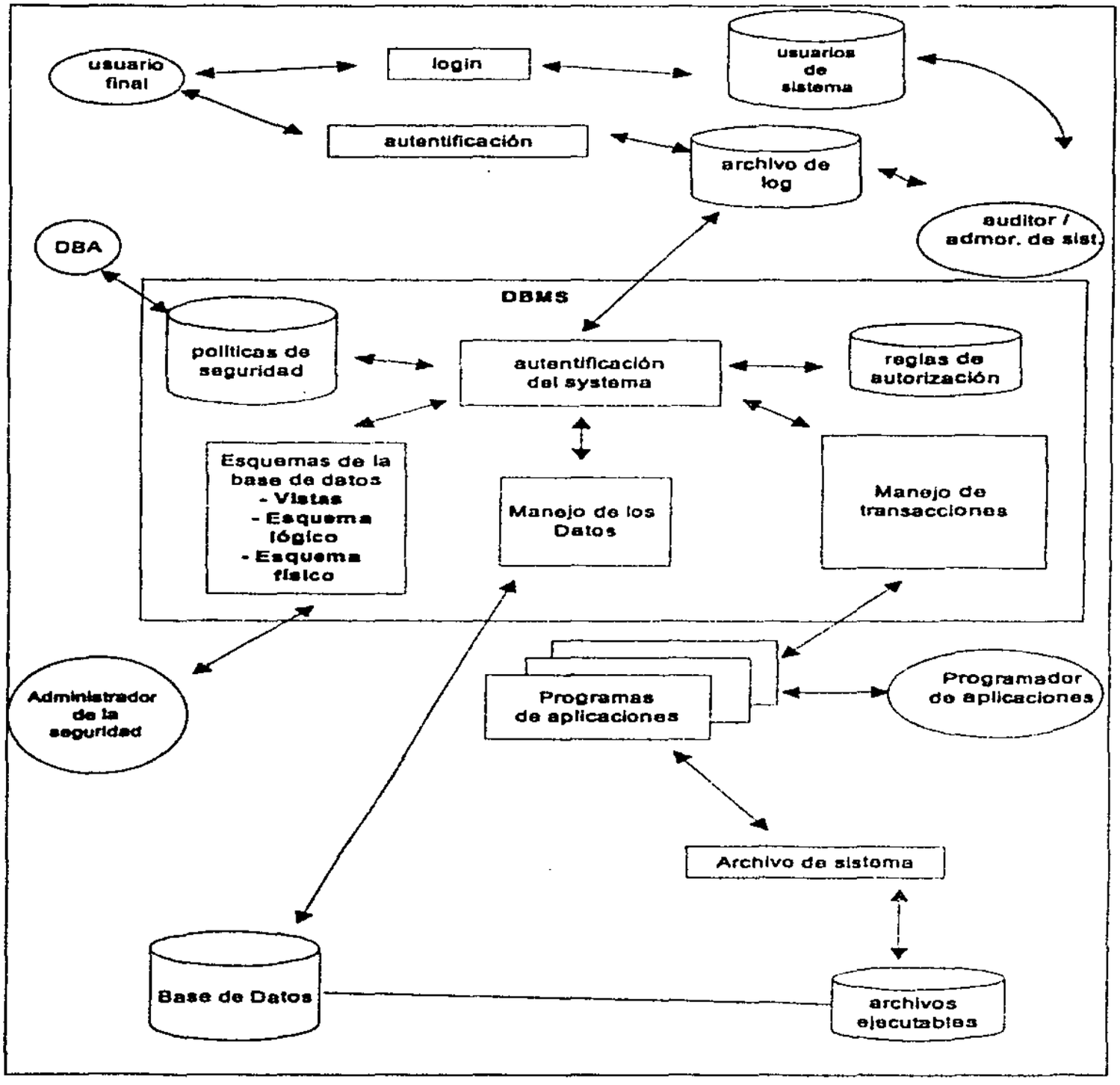
```
create view Emplea_catego
as
select ID_de_empleado, Nombre, ID_de_categoria
from EMPLEADOS, SUELDOS.
```

Y posteriormente se crean los permisos o restricciones para uso de esta vista:

revoke select on Emplea_catego to public

La creación de vistas es de gran utilidad ya que nos brindan un mayor control sobre la seguridad de los datos que se tiene almacenado en la Base de Datos, ya que de esta manera se restringen datos que no son de dominio público.

Diseño de la seguridad en una Base de Datos Relacional.



CAPÍTULO V

OTROS MECANISMOS DE SEGURIDAD

ΧΡΙΠΤΟΓΡΑΦΙΑ ΧΙΦΡΑΔΟ ΔΕ ΔΑΤΟΣ

5.1. CRIPTOGRAFÍA

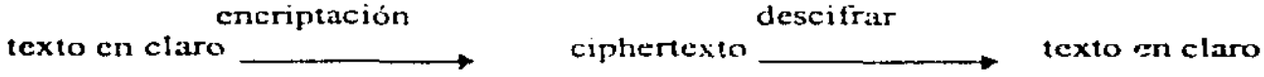
¿Que es criptografía?

La palabra criptografía proviene de la palabra griega χριπτο: secreto y γραφια: escritura. Es una palabra vieja, en donde criptografía se refiere al arte de escribir en secreto. Usualmente, se piensa que criptografía es el arte de manejar la información de manera inelegible y teniendo un método secreto que permite descifrar el código secreto. El objetivo básico de la criptografía es la habilidad de mandar información entre dos personas sobre un medio donde los demás no lo puedan leer. La criptografía provee servicios tales como:

•Chequeo de integridad: válida que el mensaje no haya sido alterado desde la última vez que se genero el código fuente.

•Autentificación: verifica la identidad de la persona.

Pero regresando a la tradicional criptografía ¿Cómo se usa esta?. El mensaje original es llamado como **texto plano** o **texto en claro.** Luego se le aplica la encriptación y el resultado de esta información es conocida como **ciphertexto.** El proceso de trabajar el ciphertexto para obtenerlo en claro se llama **desincriptación.**



Mientras los criptografos inventan claves para generar códigos secretos, los criptoanalistas se encargan de romper estos códigos. Estas dos disciplinas van trabajando paralelamente por llevar la cabeza.

Cabe mencionar un punto relevante a este término, en nuestra lengua española la palabra "encriptar" no existe, los conceptos que se han dado provienen del inglés, pero por tal motivo en vez de usar la palabra criptografía, utilizaremos la palabra "cifrar", de aquí en adelante utilizaremos la palabra "cifrar". Por lo anterior surge una pregunta:

¿Qué es el cifrado?

Al igual que el concepto de criptografía, consiste en convertir un mensaje (llamado texto en claro) en otro (llamado texto cifrado), mediante la utilización de algoritmos matemáticos determinados y una llave de cifrado secreta, esta última la llave de cifrado determina el resultado de la función.

Descifrar es el proceso inverso, convertir el texto cifrado en el texto claro.

Normalmente se necesita la misma llave que se utilizo para el cifrado.

Fortaleza del cifrado.

Es una medida de seguridad del mensaje cifrado, depende de:

- Que tan secreta sea la llave
- La dificultad de probar todas las llaves posibles
- La dificultad de invertir el algoritmo sin conocer la llave
- La existencia o no de puertas "traseras"
- La facilidad de descifrar un mensaje si se conoce una parte de él en su forma original
- Las propiedades del mensaje original y si el atacante las conoce o no

5.2 Tipos de sistemas de cifrado

De *llave privada*: Son los que utilizan la misma llave para cifrar y descifrar el mensaje. Para este tipo de llave se utiliza el algoritmo "cript", que esta basado en el Estándar de Encriptacion de Datos (DES).

CRIPT

El algoritmo cript utiliza una variante del algoritmo Enigma, que fue utilizado durante la Segunda Guerra Mundial por los alemanes.

cript funciona de la siguiente manera:

cript key < nomina > fuente

El comando anterior lee el archivo nomina, cifra el archivo mediante la clave de contraseña (key)y guarda el texto cifrado resultante en el archivo fuente. Los archivos cifrados pueden verse o descifrar mediante una linea de comando similar, como se muestra a continuación:

crypt key < fuente > nomina

El texto cifrado de nomina se descifra mediante key y se guarda en el archivo nomina.

El algoritmo utilizado por cript es sumamente fácil de romper, e incluso existe un programa llamado Crypt Breakers's Workbench, que descifra automáticamente los mensajes con cript.

Estándar de Encriptacion de Datos (DES)

Durante mucho tiempo DES fue considerado prácticamente "irrompible". Actualmente, se está comenzando a cuestionar su fortaleza, sobre todo debido a que es posible construir chips que implementen el algoritmo en hardware, y que puedan probar millones de llaves por segundo, para realizar un ataque de "fuerza bruta". DES fue establecido por el Instituto de Estándares y Tecnologia (NITS) de estados Unidos. En la operación regular, de acuerdo con el estándar DES, se usa una clave de 56 bits, por ejemplo ocho caracteres de siete bits para cifrar el texto original, al que se le llama comúnmente texto en claro. El texto cifrado resultante no puede descifrarse fácilmente si no se tiene la clave original.

Sistemas de llave pública: utilizan una llave para cifrar el mensaje y otra para descifrarlo. Un ejemplo de estos, algoritmos Rivest-Shamir-Adleman (RSA).

Una de las llaves, llamadas llave publica. Puede divulgarse públicamente sin comprometer la seguridad del algoritmo.

La otra llave, llamada llave privada, debe de ser del conocimiento únicamente para el dueño de la llave.

Fortaleza de RSA

Se basa en la dificultad de factorizar un número muy grande (más de 200 dígitos), que es la base del algoritmo. Es posible variar la longitud de la llave, con la cual se puede incrementar su seguridad, aunque también se hace mas lento. Se estima que para factorizar un número de 200 dígitos decimales se necesitarían aproximadamente 380 267 años de cómputo, trabajando a razón de 10 a la décima potencia de operaciones por segundo.

Si se encuentran mejores algoritmos de factorización, la seguridad de RSA se vería gravemente comprometida.

Privacidad Bastante Buena (PGP)

La Privacidad Bastante Buena (PGP, Pretty Good Privacy), fue elaborado por Phil Zimmerman, es un sistema de dominio público que utiliza una combinación de sistemas de llave pública y privada para proporcionar privacidad y autentificación de mensajes. PGP constituye una forma sólida de protección de cifrado de datos de la que antes no se disponía y que se utiliza para proteger correo electrónico, archivos y documentos con firmas digitales; está disponible en forma comercial y no comercial.

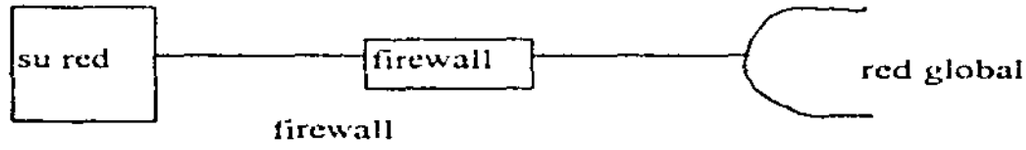
La clave de cifrado utilizada es el factor limitante para determinar el grado de esfuerzo para descifrar los datos. Mientras más grande sea la contraseña, más complejo es el patrón de cifrado, y más tiempo se requiere para transformar la clave.

En consecuencia, como cualquier otro sistema de seguridad, la contraseña usada para cifrar datos es el componente decisivo.

5.3 FIREWALLS

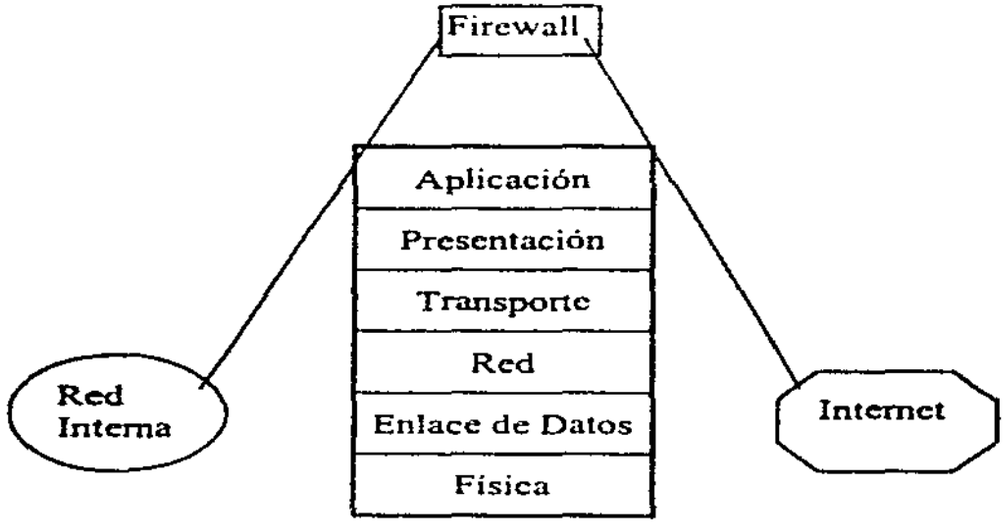
Literalmente la palabra firewall significa muralla de fuego. Antiguamente se le denominaba a sí a las murallas que construían los imperios para protegerse y salvaguardarse de los ataques hostiles de los vándalos, estas murallas detenían las flechas candescentes y los ataques.

Actualmente en el mundo de la computación la palabra firewall sirve para proteger una red de otra, deteniendo los ataques de intrusos y controlando muy estrechamente el paso de usuarios e información hacia y desde la red.



firewall

Los firewall operan en las capas superiores del modelo OSI, tienen información completa sobre las funciones de la aplicación en la que basan sus decisiones. En general, un firewall se coloca entre la red interna confiable y la red externa no confiable. El firewall actúa como un punto de cierre que monitorea y rechaza el tráfico de red a nivel de aplicación. Los firewall también pueden operar en las capas de red y transporte, en cuyo caso examinan los encabezados de IP y de TCP de paquetes entrantes y salientes, y rechazan o pasan paquetes con base en reglas de filtración de paquetes programadas.



Operación de un firewall.

Hay dos tipos de firewall, *internos y externos*.

Firewall internos. Consiste en hacer las subredes internas lo más independiente posible, haciendo que se comuniquen entre ellas a través de ruteadores.

Firewall externo. Son máquinas que aíslan la red local del exterior, toda la comunicación tiene que pasar a través del firewall, y este puede ser configurado.

Recomendaciones para la creación de firewall.

- Ninguna máquina debe confiar en una máquina que esté fuera de su red.
- Usuarios con cuentas en más de una subred deben usar diferentes password.
- Los ruteadores deben tener máximo nivel de login, y la seguridad más eficiente.
- De ser posible no debe haber cuentas en los ruteadores.
- No montar sistemas de archivos de NFS entre una subred y otra.

Partes de un firewall.

Compuerta (gate). Pasa información entre las dos redes (externa e interna). Es un sistema Unix que recibe toda la información proveniente del exterior, así como del interior de la red. Su función es asegurar la validez de los datos, autentificar a los usuarios, y dejar pasar hacia su destino final solamente los paquetes autorizados. La compuerta es el único sistema al que se tiene acceso desde el exterior de la red, por eso debe tener activados todos los mecanismos de seguridad pertinentes, no debe tener cuentas de usuarios, y debe ser el servidor de todos los servicios que se otorguen hacia el exterior (como ftp anónimo).

Ruteador. Bloquea todos los paquetes que no vayan o vengan de la compuerta como destino o como origen. También se puede configurar para que solamente deje pasar los paquetes correspondientes a ciertos servicios. Por ejemplo, se puede dejar pasar a los paquetes de correo electrónico, pero no a los de "telnet" o "rlogin". El ruteador puede ser un dispositivo especial, o una máquina Unix que tenga dos interfaces de red.

5.4 POLÍTICAS DE SEGURIDAD

Las políticas de seguridad, es un documento que describe los intereses de seguridad de los datos en una organización.

Una política de seguridad adecuada puede ser tan importante en la protección del sistema como todos los mecanismos de protección. Vale la pena implementar una política de seguridad si los recursos y la información que la organización tiene en sus redes merecen protección. La mayoría de las organizaciones tienen en sus redes información delicada y secretos importantes, esto debe protegerse del vandalismo, del mismo modo que se protegen bienes valiosos, como una propiedad o las oficinas de un edificio.

Una política de seguridad es un código de conducta para la utilización del sistema. Especifica que actividades no son permitidas, los pasos a seguir para lograr la protección adecuada, los pasos a seguir en caso de un incidente de seguridad, establece responsabilidades y derechos, etc. Es importante recalcar que no es suficiente escribir una política de seguridad "genérica", es necesario hacer un cuidados análisis de:

- El tipo de trabajo que se realiza
- El tipo de usuario que se tiene.
- El tipo de información que se utiliza.
- Las políticas de la organización en otros aspectos.
- La Base de Datos que se va a utilizar.
- Las tablas que se van ocupar.
- Los grupos y privilegios que se tiene sobre esos grupos.

En el punto pasado se hizo referencia al uso del sistema, de las tablas, aplicaciones, etc., pero antes de esto debe de haber políticas de acceso al sistema, tales como:

- Quiénes son los usuarios de cada sistema.
- Quién está autorizado a proporcionar el acceso.
- Mecanismos de creación de cuentas.
- Mecanismos para la eliminación de cuentas.
- Sanciones por no atenerse a los mecanismos establecidos.

De igual manera es importante hacerle tomar conciencia al usuario de que en gran parte, el que funcione correctamente el sistema y la seguridad de sus datos dependerá de él, esto es, que haga un buen uso de su clave secreta de acceso al sistema y a la base de datos, que no comparta su clave secreta.

Recomendaciones para elegir passwords seguros:

No usar nombres propios, ni el de su fecha de nacimiento, ni el de su equipo favorito ni el número de camiseta de su estrella de fútbol, tampoco usar la fecha de su nacimiento, ni el nombre de su novia, ni, el de su mascota, ya que estos son blancos fáciles para un hacker y este se podrá adueñar de la clave y de todos sus datos, por ejemplo poner passwords como el siguiente: *Ga#toRate69, o así por el estilo, combinar números*, mayúsculas, minúsculas, signos especiales, etc. Es muy importante no dejar apuntado el password en un papel.

- Establecer claramente el carácter secreto de los passwords.
- No de el password a nadie, ni compartirlo.
- Cambiar el password periódicamente, pero sin llegar a confundirse.

Se calcula que el 80% de todos los problemas de seguridad en introducción a la base de datos o a la red son creados por contraseñas inseguras.

5.4.1. Derechos y responsabilidades de los administradores.

Es importante que los administradores tengan alto grado de responsabilidad hacia el sistema y hacia los usuarios, entre los derechos y responsabilidades éticas que debe reunir un administrador son:

- Respetar la privacidad de los usuarios, esto quiere decir que el administrador no deberá estar de viendo los archivos de los usuarios sin previa justificación que lo amerite.
- Derecho de monitorear a los usuarios si la situación lo amerita.
- Comportamiento adecuado de los superusuarios.
- Exhortar el uso apropiado del sistema.
- Explicar los puntos del comportamiento éticamente correcto.
- Reglas de "etiqueta", esto se refiere a etiquetar todos los respaldos e información que se tenga en dispositivos externos.

Para llevar un control adecuado de los recursos del sistema y de red se recomienda llenar una hoja de trabajo similar a la que se presenta a continuación, que dará información acerca del usuario y de la red con la que se esta trabajando:

RECURSOS DE LA RED		USUARIO	PROYECTO	TIPO DE ACCESO	PERMISOS DEL SISTEMA OPERATIVO UNIX: RWX
Número	Nombre				

Para llevar un control adecuado de la base de datos y de la red se llena una hoja de trabajo, que me dará información acerca del usuario, la base de datos que ocupa, permisos o restricciones que se tiene el tamaño de la base de datos, de su log de transacciones, la red con la que se esta trabajando:

RECURSOS DE LA RED		USUARIO	TAMAÑO	LOG	DUEÑO	PERMISOS Y/O RESTRICCIONES EN LA BASE DE DATOS
Número	Nombre					

Es importante que el administrador establezca dentro de sus políticas:

- Qué actividades son permitidas y cuales no.
- Los diferentes tipos de sanciones que se aplicarán a los usuarios que violen las políticas de seguridad establecidas por la organización.

CAPÍTULO VI

ESTUDIO DE CASO

CAPÍTULO VI

ESTUDIO DE CASO

En este capítulo se estudiará el sistema del que se ha mencionado, se analizará el sistema a partir del surgimiento del problema, su solución y la liberación del mismo. El sistema del que se hará mención es el de Sistema de Convenios de la Oficina del abogado General de la UNAM a través de la red.

Los principales puntos que se enfocará al: análisis del sistema, diseño de las tablas, reglas de validación, y seguridad del mismo.

Presentación del problema:

El departamento de informática de la Oficina del Abogado General de la UNAM desea procesar de manera eficiente los convenios que se celebran en la UNAM con universidades o dependencia externas a ella. Se requiere el tener información en línea en cualquier parte del mundo que se tenga acceso al red, en donde se encuentre, con sólo una PC conectada a la red y un visualizador de Netscape, con esto se podrá obtener un convenio deseado.

Otro punto a resolver en la creación de este nuevo sistema, es eliminar discrepancias entre dos entidades, por ejemplo, usualmente se tienen reuniones de trabajos con otras universidades o dependencias para la revisión de los puntos que anteriormente se habían trabajado sobre algún convenio entre ambas universidades, y resultaba que gente de la UNAM o la gente externa contaba con un contrato diferente al que tenia la contra parte, y por consiguiente no se llegaba a ningún acuerdo. Los datos se encontraban en una "PC" de escritorio, por eso surge la necesidad de contar con un sistema en línea a través de la red y tener disponible en cualquier momento los convenios que se han firmado con la Oficina del Abogado General de la UNAM.

Puntos relevantes para la creación del nuevo sistema.

La solicitud de la creación del Sistema de Convenios para la Oficina del Abogado General de la UNAM fue hecha a la Dirección de Computo para la Administración Académica (DCAA) a través de la Dirección General de Servicios Cómputo Académico (DGSCA). El sistema tendría que reunir los puntos anteriormente señalados a través de la red, pero detrás de esto, el sistema debe reunir lo siguientes:

- El sistema funcionará en una plataforma a nivel multiusuario y estará disponible en la red
- El sistema sólo soportará de cinco a siete usuarios (dependencias pertenecientes a la UNAM, por ejemplo: Difusión Cultural, CONACYT, Intercambio Académico, y otras más), que únicamente tendrán derecho de ver la información más no de modificarla
- Los únicos que tendrán permisos para realizar los cambios, será el departamento de Informática de la Oficina del Abogado General de la UNAM
- Por tales motivos es indispensable que se cuente con un nivel alto en seguridad, tanto a nivel base de datos, como de acceso al sistema

Situación en la que se encontraba el departamento de informática de la Oficina del Abogado General.

La información de los convenios se encontraba almacenada en una base de datos relacional para ambiente Windows como lo es Microsoft Access 2.0, corriendo en dos PC 486 SX, de manera centralizado.

Propuesta para la creación del sistema del nuevo sistema.

Hardware. Para la creación del sistema se recomendó la compra de una estación de trabajo, ya que estas debido a su arquitectura (RISC) que permite gran robustez en la creación de aplicaciones de emisión crítica, alta confiabilidad y seguridad en la ejecución de programas. Utilizará el back-bone de la UNAM que es de FDDI a 100mbps.

Software. El equipo trabajará bajo un sistema operativo multiusuario como lo es Unix, ya que proporciona mayor estabilidad y seguridad. Como manejador de base de datos (DBMS) se tendrá a Sybase que es una base de datos relacional, y el visualizador será Netscape, debido a la seguridad que maneja, ya que cifra información con llave de 40 bits, (lo permitido por las políticas Americanas).

Recursos humanos. Se requiere la capacitación de personal para la administración del servidor, y para la administración de la Base de Datos. Este es un punto que le corresponde a la gente de la Oficina del Abogado General de la UNAM, y que la Dirección de Cómputo para la Administración Académica (DCAA) los apoyará con la capacitación de su personal en el momento en que ellos lo decidan.

Descripción del sistema

El Sistema de Consultas de Convenios de la UNAM a través de la RED es un sistema basado en una arquitectura cliente/servidor por medio del cual se puede compartir información de una manera fácil, sencilla y eficaz.

Tal Sistema de Consultas de Convenios de la UNAM a través de la RED utiliza como Servidor al Manejador de Datos (DBMS) SYBASE System XI y como cliente al visualizador Netscape Versión 3.0.

¿Qué es el Sistema de Consultas de Convenios de la UNAM a través de la RED?

El Sistema de Consultas de Convenios de la UNAM a través de la RED es una aplicación que muestra los extractos más importantes de los convenios que la UNAM a firmado con diversas instituciones, este trabajo ha sido realizado por la subdirección de Sistemas de la DGSCA-DCAA para la oficina de la Abogado General de Rectoría de Ciudad Universitaria.

El Sistema de Consultas de Convenios de la UNAM a través de la RED va a permitir el manejo en línea de consultas sobre determinados convenios requeridos, así mismo se podrán actualizar, e insertar nuevos convenios, solicitados por los administradores del equipo. Cabe destacar que los únicos que tienen permisos para modificar la Base de Datos de los Convenios de la UNAM será la gente que trabaje en la oficina del Abogado General, las demás dependencias tendrán permiso exclusivamente para consultar información.

El Sistema de Consultas de Convenios de la UNAM a través de la RED surge como una herramienta de apoyo para incorporarse a los sistemas automatizados de la Rectoría de la UNAM al sistema cliente/servidor, por lo que dicho sistema trabaja bajo esta arquitectura. Al ser una arquitectura abierta, el manejo, control, seguridad y distribución de la información se vuelve más eficiente.

Cualquier sistema desarrollado bajo el esquema cliente/servidor consta de dos partes: un cliente y un programa servidor. En este caso el Sistema de Consultas de Convenios de la UNAM a través de la RED que fue desarrollado bajo un ambiente UNIX con una interfaz gráfica como lo es al programar cgi's con un visualizador Netscape, y el Manejador de Bases de Datos Sybase versión System X conforman el servidor.

Los clientes lo conforman las distintas dependencias de la UNAM que le solicitan un servicio a la Oficina del abogado General que es donde se encuentra el servidor.

Objetivo del Sistema de Consultas de Convenios de la UNAM a través de la RED.

El objetivo del Sistema de Consultas de Convenios de la UNAM a través de la RED es poder consultar los convenios que la Universidad ha firmado con cualquier otra institución perteneciente al sector público, privado y/o extranjero, estas consultas pueden ser hechas ya sea localmente o remotamente desde cualquier punto en donde la máquina tenga acceso a Internet y esté dada de alta (por seguridad misma de los datos.).

Antecedentes del Desarrollo.

A partir del objetivo antes mencionado y con idea de obtener los requerimientos de la Oficina del Abogado General de Rectoría de la UNAM se realizaron diferentes sesiones de trabajo con la gente encargada del departamento de informática, obteniéndose lo siguiente:

- Su información se encuentra almacenada en el Manejador de Bases de Datos Access, la cual se desea migrar a una plataforma cliente/servidor como lo es SYBASE.
- Selección de los datos requeridos para el Sistema de Consultas de Convenios de la UNAM a través de la RED.
- Diseño y creación de las bases de datos del Sistema de Consultas de Convenios de la UNAM a través de la RED en SYBASE.
- Migración de los datos requeridos para las actualizaciones y/o consultas.

Esto permitió detallar los requerimientos del nuevo sistema.

Ambiente de trabajo del sistema

El Sistema de Consultas de Convenios de la UNAM a través de la RED trabaja bajo ambiente Windows 3.x. Además de Windows se necesita tener el visualizador Netscape para poder acceder a las hojas correspondientes, y las Net - Library de Sybase correspondientes. (Éstas librerías las suministra la misma compañía de Sybase y son independientes del DBMS). Por otra parte es necesario instalar y configurar el ODBC.

Interacción con el sistema a través de la red

Para manejar el Sistema de Consultas de Convenios de la UNAM a través de la RED, se presentan dos modos de acceso, con base en los permisos que un usuario tenga para ejecutar ciertos procesos del sistema, esto es, con base en la relación sistema-suario:

- *Superusuario*

Si el usuario tiene una clave de acceso al sistema tal que le permita llevar a cabo cualquier proceso, desde una consulta hasta modificaciones en la base de datos (altas, bajas o cambios propiamente dichos).

- *Usuario normal*

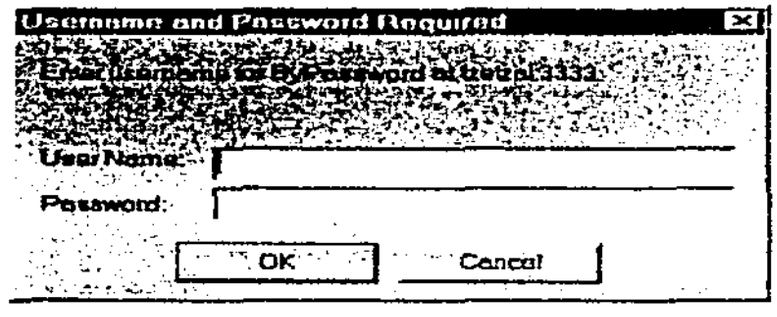
Si los privilegios con los que cuenta el usuario le permiten efectuar consultas al sistema sobre los registros ya existentes, de tal manera que no tendrá acceso al proceso de modificaciones a la base de datos.

En cuanto al acceso al sistema, siendo un sistema cuya interfaz es la RED, es necesario indicar a un navegador (por ejemplo *Netscape*) la dirección correspondiente al proceso específico que se requiera ejecutar en el sistema. La razón por la cual se cuentan con diferentes direcciones (una dirección asociada a un proceso), es la que se ha establecido anteriormente puesto que no todos los usuarios del sistema podrán ejecutar todos los procesos.

Acceso Restringido al Sistema

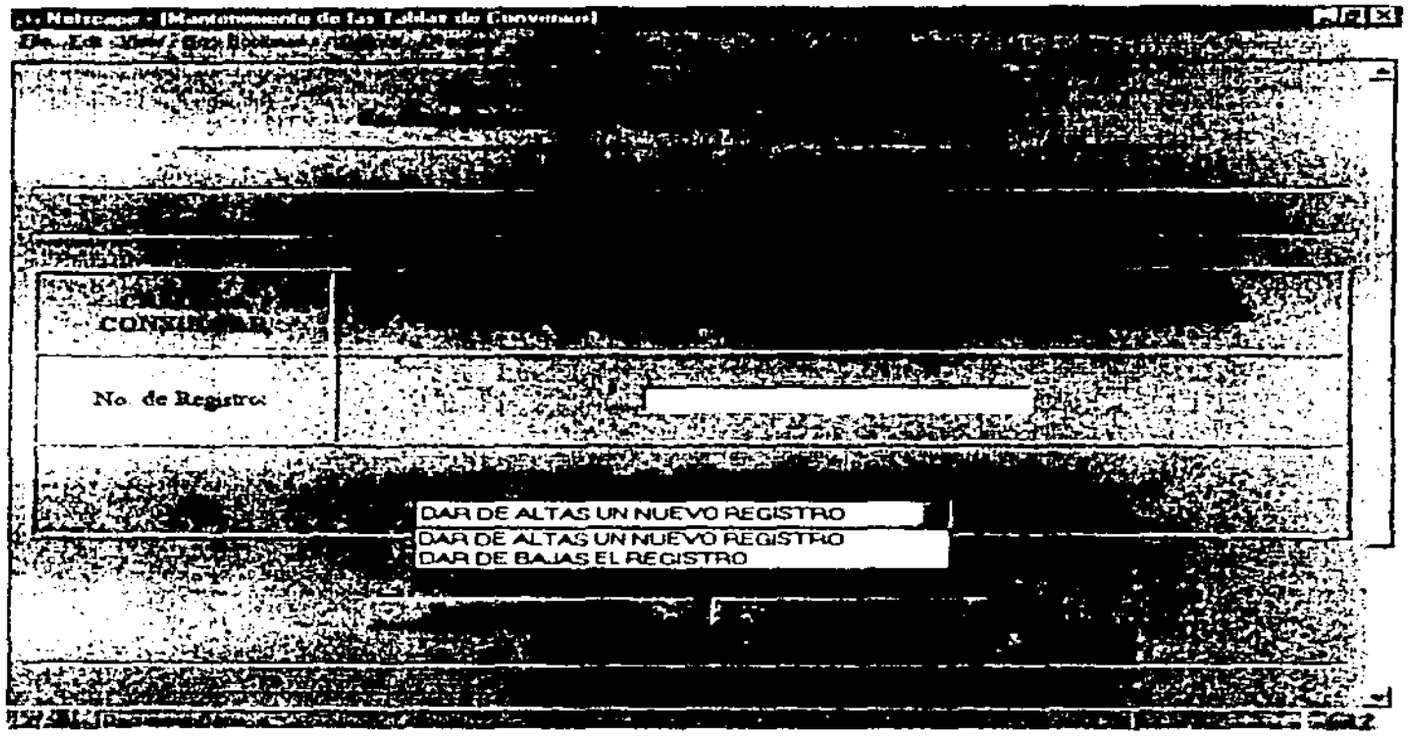
El acceso restringido, como se había establecido, implica que se podrá tener acceso al sistema con sólo introducir la clave de **Usuario** y su correspondiente **Password,** cabe aclarar que la máquina de donde se quiera conectar deberá estar dada de alta, si no es de esta manera, no se podrá tener acceso al sistema.

Por motivos de seguridad la dirección correspondiente a procesos de altas, bajas y cambios, cuenta con acceso restringido por lo que previamente al ingresar al contenido de la página correspondiente al proceso de altas, bajas y cambios, será necesario introducir datos de usuario solicitados, mismos que serán requeridos mediante la ventana que se muestra enseguida.



En ella, el primer campo, en el que se sitúa el cursor de manera automática, corresponde a nombre del usuario o *login*, en tanto que el segundo campo corresponde a la clave de acceso o *password* respectivo que lo acredita como super usuario. En tanto no introduzca correctamente tales datos, no podrá acceder a la página ya mencionada.

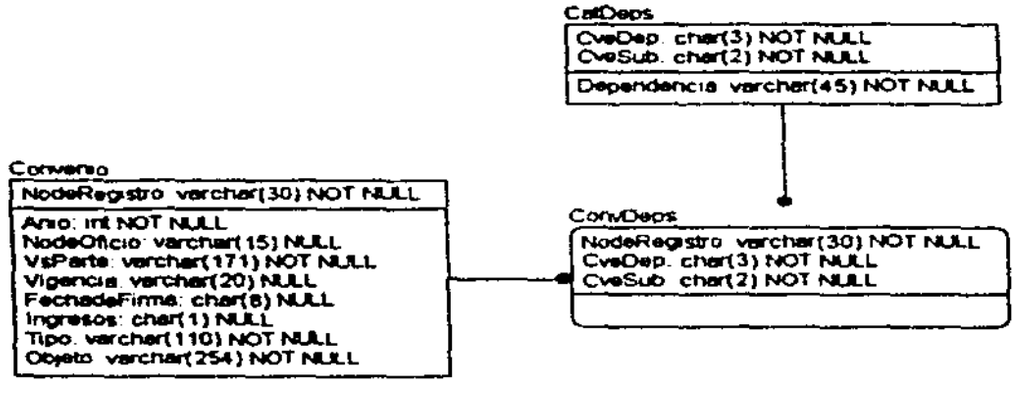
Una vez que se inserto correctamente el login y el password de una máquina reconocida por el sistema y nos mostrará la siguiente pantalla.



A partir de esta pantalla podemos hacer la selección deseada, ya se para dar de alta un nuevo registro, una baja, o un cambio.

Para tener disponible esta página, se tuvo que crear primeramente un diseño relacional para establecer las relaciones de las entidades, y esto queda ilustrado en el Diagrama Entidad Relación.

Diagrama Entidad Relación de Convenios



TABLAS

La creación de las tablas se hizo siguiendo el diagrama Entidad-Relación, resultando como se muestra a continuación:

CatDeps

PK		
CveDep	SiglasDep	Dependencia

ConvDeps

PK	PK	PK
NodoRegistro	CveDep	CveDep

Convenios

PK	PK		PK					
Año	Dependencia	NodoOficio	NodoRegistro	Vs. Parte	VIGENCIA	FECHAFIRMA	INGRESOS	TIPO

Permisos y Restricciones

La creación de los permisos y restricciones se hizo sobre la base de las peticiones de la Oficina del Abogado General en donde ellos controlaran y darán administración al equipo de manera centralizada.

```
/*********************************************
  Objetivo:
        Proporcionar los permisos necesarios a cada uno de los
        Usuarios de la Base de Datos CONVENIOS.

  Todos los Usuarios de Convenios que pertenecen al Grupo llamado: abcambios
  podrán únicamente hacer Cambios y los de abconsultas solo harán consultas.

*********************************************/

/*
 1) Grupo grpcambios:
     Usuario:
     abcambios


 2) Grupo grpconsultas:
     Usuario:
     abconsultas

*/


CREATE PROCEDURE spPermisos
AS

/*Permisos para los Usuarios de las Tablas de CONVENIOS*/

/* grpcambios: */

grant execute on spVaciaLog to grpcambios
grant insert, delete, update on Convenio to grpcambios
grant insert, delete, update on ConvDeps to grpcambios
grant insert, delete, update on CatDeps to grpcambios

grant select on Convenio to grpcambios
grant select on ConvDeps to grpcambios
grant select on CatDeps to grpcambios
```

```
/***Permisos Los Usuarios de las Tablas de CONVENIOS***/

/* grpconsultas: */

revoke execute on spVaciaLog to grpconsultas
revoke insert, delete, update on Convenio to grpconsultas
revoke insert, delete, update on ConvDeps to grpconsultas
revoke insert, delete, update on CatDeps to grpconsultas

grant select on Convenio to grpconsultas
grant select on ConvDeps to grpconsultas
grant select on CatDeps to grpconsultas

Return
```

La dirección con la cual se puede ejecutar la consulta normal es la siguiente:

http://132.248.27.106.92/cgi-bin/oag/Exes/fquery

CONCLUSIONES

CONCLUSIONES

La seguridad en las Bases de Datos representa un punto crucial y de gran relevancia para los administradores de Bases de datos, pues de su correcto trabajo depende en gran medida el buen funcionamiento de un ente social. Desde el principio las Bases de Datos se han visto amenazadas por personas que buscan el hacerse de los datos para sus diferentes fines personales.

En este sentido el control requerido por el Administrador de la Bases de Datos plantea la necesidad de establecer rigurosas políticas y sistemas de seguridad para garantizar que los datos que están alojados en la Base de Datos, son verídicos y de gran relevancia para la organización. No es posible proteger de manera absoluta la Base de Datos contra un manejo indebido, pero puede hacerse que el costo para el criminal sea tan alto que frene prácticamente todos los intentos de lograr el acceso a la Base de Datos sin la autorización debida, por lo tanto, podemos decir: El término *seguridad de la bases de datos, normalmente se refiere a la protección contra el acceso mal intencionado.*

Para proteger a la Base de Datos es necesario adoptar medidas de seguridad en varios niveles:

- Físico: La localidad en donde se encuentran el DBMS.
- Humano: Hay que tener especial cuidado en conceder privilegios y restricciones a los usuarios.
- Sistema Operativo. Tener sistemas de autentificación para reducir la probabilidad de ataques.
- DBMS. Las restricciones necesarias que me permita el DBMS para tener el control centralizado de los datos.

Es importante contar con seguridad e integridad en un sistema de Base de Datos, distinguiremos los dos términos de la siguiente manera: Seguridad: Significa proteger la Base de Datos contra usuarios no autorizados. Integridad: Significa protegerla de usuarios autorizados.

Tanto la seguridad como la integridad implica: La definición de restricciones apropiadas, políticas que se han de tomar si se violan esas restricciones y Una supervisión por parte del DBMS de las operaciones de los usuarios.

Imponer un sistema de seguridad sobre una Base de Datos dependerá de la importancia del valor de los datos, la tarea más difícil del DBA será el identificar los mejores medios para salvaguardar la integridad de los datos. Aquí entrará el juicio del administrador al implantar un sistema de seguridad según sea el valor de sus datos.

La dificultad para interarctuar con la información que se tiene en una base de datos (esto es, consultar información o actualizar), es directamente proporcional a su seguridad. ver figura. Esto es, entre mayor seguridad: se vuelve más complejo es el desarrollo de aplicaciones, se incrementan costos se requiere gente más capacitada en el área para la correcta utilización y administración de la Base de Datos, pero de un punto podrá estar seguro, se tendrá una Base de Datos*: estable y confiable*

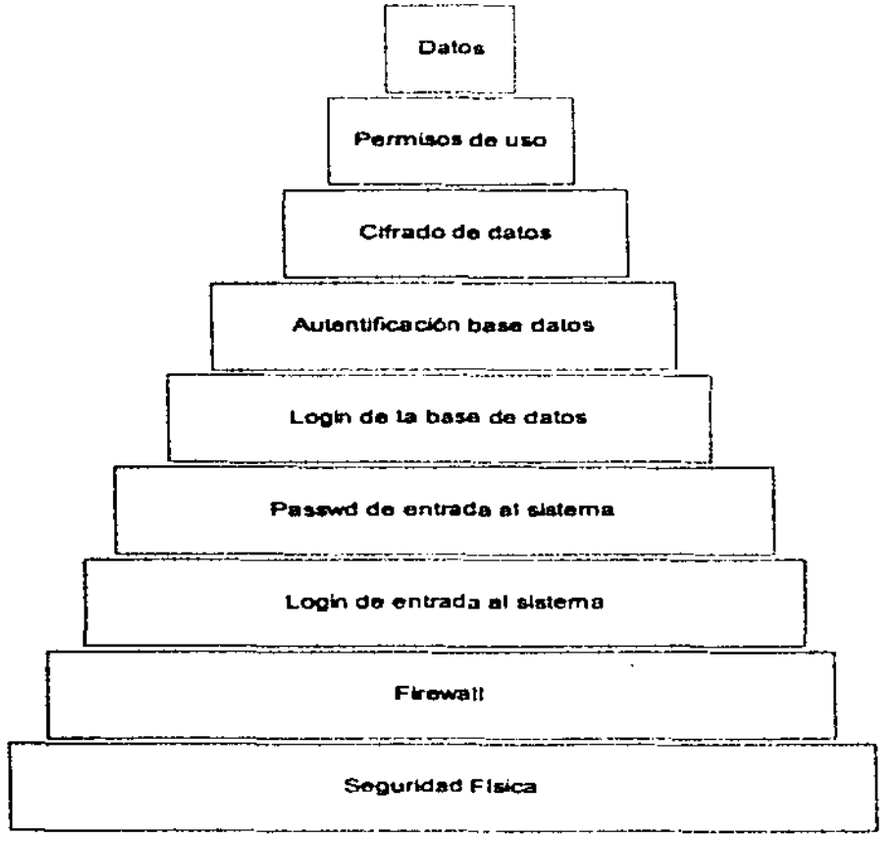


Fig. grado de complejidad de uso, implementando diferentes niveles de seguridad.

APÉNDICE

APÉNDICE

SEGURIDAD DE SYBASE

El servidor SQL de seguridad de sybase (SYSSS) es un servidor para bases de datos en ambientes de red. Sybase es un DBMS comercial el cual ofrece las siguientes funciones a nivel de seguridad:

- Identificación y autentificación del usuario
- Control de acceso
- Una Bitácora de eventos, relacionado con la seguridad
- Hace una diferencia de responsabilidad de usuarios, y maneja los diferentes roles que puede tener un usuario en la Base de Datos como son: el oficial de seguridad, el administrador de la Base de Datos, dueño de la base, y el resto de los usuarios.

La interfaz que tiene para usuarios confiables es:

- Sistema de administración, que permite a los usuarios almacenar operaciones en un dispositivo.
- Para dueños de la base les permite definir y modificar la estructura de la base, para grupos de usuarios les permite definir y actualizar las tablas, a si también permite habilitar o deshabilitar el acceso a la Base de Datos de usuario, forzando de esta manera un registro de control en la bitácora de la Base de Datos.
- Los dueños de las tablas pueden definir autorizaciones de acceso a sus tablas y permitirles el uso de operaciones sobre ellas.

En el siguiente cuadro se muestra las características que tienen diferentes RDBMS comerciales para el control de la seguridad de su producto ver tabla API

SEGURIDAD Y REGLAS DE INTEGRIDAD EN BASES DE DATOS RELACIONALES

DBMS server	Autenficación por passwd	Autenficación por OS	Usuarios crean tipos de objetos CREATE ALTER, DROP	Control por Superusuario	SLECT INSERT UPDATE DELETE tablas y vistas	ALTER INDEX tablas	Referencias	Nivel de Control de Columna	Execución	Opcion grant	Opción Admon	Diferencia entre grupos	Roles	Auditoria
Ingres		•	•	•	•			Update	•			•	•	•
Oracle	•	•	•	•	•	•	•	Select, Update Refer	•	•	•		•	•
Sybase	•		•		•			Select, Update	•			•		
Informix		•	•		•	•		Select, Update Refer	•	•				
SQLBase	•		•		•	•		Update						

API Diferentes niveles de seguridad en RDBMS comerciales

BIBLIOGRAFÍA

BIBLIOGRAFÍA

Silvano Castano, Mariagrazia fugini, Giancarlo Martella, Pierangela Samarati
Database security, Milán Roma
Addison-Wesley
Primera edición 1994

Karanjit Siyan, Chris Hare
Firewalls y la seguridad en internet, USA
Prentice Hall
Segunda edición 1996

Charlie Kauman, Radia Perlman, Mike Speciner
Network Security, Private Communication in a Public World, USA
Prentice Hall 1995

C. J. Date
Introducción a los sistemas de bases de datos, USA
Addison-Wesley
Quinta edición 1990

Hnery F. Korth, Abraham Silberschatz
Fundamentos de bases de datos, USA
McGraw Hill 1990

Aeleen Frish
Essentia System Administration,
O'reilly & Associates
1992

Michell Shiner
Dictionary of pc, hardware an data comunications terms
O'reilly & Associates 1996

César A. Galindo Legaria
VI Escuela Internacional en Temas Selectos de Computación
Nuevas Lineas de Investigación en Bases de Datos, San Luis Potosí México
23 al 27 octubre 1995

Andrew Tanenbaum
Redes de Ordenadores, USA
Prentice Hall
Segunda edición 1991

Umberto Eco
Como se hace una tesis, Barcelona España
Gedisa editorial
Diecinueve edición 1996

REVISTAS

Kim Won
"Bringing Object/Relational"
Database

Tober Bruce
"Same security for all"
Byte
Diciembre 1996

REFERENCIAS ELECTRÓNICAS

http://www.yahoo.com/Computers_and_internet/Software/Databses

http://wwwgocsi.com/db_area/24th_97/a.htm

http://wombat.doc.ic.ac.uk/foldoc/foldoc.cgi?**Advanced+Program-to-Program+Communications**

GLOSARIO

DE

TÉRMINOS

GLOSARIO DE TÉRMINOS

ANSI (American National Standards Institute)

Instituto Nacional Estadounidense de estándares. Es una organización que define los estándares para diferentes industrias de información. ANSI participó en la definición estándar de protocolos de red.

API (Aplicación Program interface)

Interfaz de Programa de Aplicación API define una manera estándar en donde los programas son más sencillos de usar debido a que son aplicaciones gráficas que funcionan con menús, cuadros de diálogo y ventanas. Microsoft, Windows, OS/2 son ejemplos de aplicaciones API.

ATM (Asynchronos Transfer Mode)

Modo de Transferencia Asincrona

ARPA (Advanced Research Projects Agency)

Agencia del departamento de los Estados Unidos de América en cargado del desarrollo de nuevas tecnologías para uso militar. Anteriormente llamada DARPA por (Defensa), patrocinó el proyecto que se conoció como ARPANET.

ARPANET

Fue la red pionera en proporcionar conectividad a través de ancho de banda, proporcionaba a los usuarios la capacidad de transferir correo electrónico y archivos de un sitio a otro.

CABALLOS DE TROYA

Son programas que introducen a una computadora de manera oculta y que tienen como propósito el hacer cosa mal a un sistema.

CASE (Computed-Aided Software Engineering)

Ingenieria de Software Asistida por Computadora. Técnica que es usada para ayudarse en un o más etapas en las del ciclo de vida de los sistemas, incluye: análisis, diseño, implementación y mantenimiento de los sistemas.

CLIENTE/SERVIDOR

Es un modelo de interacción entre sistemas distribuidos en donde un programa en un sitio pide una pregunta, y otro programa en otro sitio responde a esa respuesta. El que pregunta es llamado cliente, y el que satisface la respuesta es llamado servidor. Este

modelo involucra tres personajes 1)servidor, 2) cliente 3) el medio de comunicación entre los dos primeros.

CRON (Clock daemon)

Sistema automático de administración. Un cron ejecuta un comando a una fecha y hora específica. Esto se encuentra en el sistema UNIX. /usr/etc/cron

FDDI (Fiber Distribution Data Interface)

Es un estándar para tecnologías de redes basadas en fibra óptica que estableció el American National Standards Institute (ANSI). FDDI trasmite datos a 100 mbps.

FLAT-FILE (Archivos planos)

Archivos planos sin formato alguno, que se utilizan para almacenar información.

INTERNET

Colección de redes conectadas entre si a través del protocolo TCP/IP, cuyo objetivo es compartir recursos. Internet provee conectividad universal, en donde sus usuarios pueden utilizar sus servicios como son: correo electrónico, copiar archivos, transferencia de datos, etc. Internet actualmente se encuentra en: universidades, gobierno, laboratorios de investigación, instalaciones militares y en una gran mayoría de países.

INTEROPERABILIDAD

La habilidad del software y hardware para comunicarse correctamente en múltiples máquinas y de múltiples vendedores.

IP dirección

Se utiliza para configurar las interfaces de red. Las direcciones son asignadas de 32-bits para cada hosts bajo el protocolo de TCP/IP. Estas direcciones de red son únicas.

LOGIN

Sirve para iniciar identificar a un usuario al inicio de una sesión.

PASSWORD

Es una arbitraria cadena de caracteres escogida por el usuario o que se la da el administrador del sistema y le sirve al usuario para identificarse para el sistema.

PROCEDIMIENTO ALMACENADO

Es una colección de instrucciones en SQL almacenadas en la Base de Datos que pueden ser ejecutadas llamadas por su nombre.

PROCESO EN PARALELO

El uso de una o más computadoras trabajando simultáneamente para resolver un problema.

REGLAS

Lista de valores, rango o un patrón. Las reglas definen a un más las características que deben tener los datos en una tabla.

TCP/IP (Transfer Control Protocol/Internet Protocol)

Transferencia del protocolo de control de internet. El protocolo estándar de Ethernet incorporado a Unix. TCP/IP fue desarrollado por DARPA para trabajo en red, realizando la comunicación entre protocolos de la capa de red y la de transporte. TCP (Transfer Control Protocol) se aplica sobre IP (Internet Protocol) para ofrecer una comunicación confiable entre terminales internet.

TRANSACCIÓN

Unidad lógica de trabajo.

TRIGGERS

Tipo especial de procedimiento almacenado el cual se encuentra ligado a una tabla. Los triggers son activados automáticamente por el servidor SQL cuando en una tabla se inserta y/o modifica datos. (insert, update, delete).

RISC (Reduced Instruction Set Computer)

Es un procesador cuyo diseño es basado en la rápida ejecución de simple instrucciones más que en ofrecer instrucciones complejas.

SPARC (Scalable Processor ARChitecture)

La arquitectura del procesador escalable de 32 bits de Sun. SPARC basada en estaciones de trabajo que trabajan en Unix.

UNIX

Sistema operativo multitarea desarrollado por Ken Thompson en los laboratorios Bell después de que Bell abandonó el proyecto (Multics), Dennis Ritchie creador de C, es considerado como coautor del sistema. Un sistema operativo es el programa de control que administra los recursos del hardware y software. Una de las principales ventajas de Unix es su portabilidad (Habilidad para correr aplicaciones Unix en una variedad de plataformas de hardware). Unix nos permite la multitarea, la cual nos permite llevar acabo tareas en un mismo tiempo, y multiusuario que nos permite a tender a varios usuarios simultáneamente.