

8793/6



UNIVERSIDAD LASALLISTA
BENAVENTE



ESCUELA DE INGENIERÍA EN COMPUTACIÓN

Con estudios incorporados a la
Universidad Nacional Autónoma de México

CLAVE: 8793-16

**“EVALUACIÓN DE LOS MANEJADORES DE BASES
DE DATOS MÁS ACTUALES”**

TESIS

QUE PARA OBTENER EL TÍTULO DE:

INGENIERA EN COMPUTACIÓN

PRESENTA:

ROSA GONZÁLEZ GARDUÑO

Asesor: ING. MIGUEL ÁNGEL JAMAICA ARREGUÍN

Celaya, Gto.

Junio de 2004



Universidad Nacional
Autónoma de México

Dirección General de Bibliotecas de la UNAM

Biblioteca Central



UNAM – Dirección General de Bibliotecas
Tesis Digitales
Restricciones de uso

DERECHOS RESERVADOS ©
PROHIBIDA SU REPRODUCCIÓN TOTAL O PARCIAL

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

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

AGRADECIMIENTOS

A Dios por concederme la gracia de vivir y ayudarme a lograr todos mis éxitos.

A mis padres por apoyarme incondicionalmente en todo lo que me he propuesto.

A mis hermanos y hermanas por comprenderme y estimularme a seguir adelante en todos mis proyectos.

A la Universidad Lasallista Benavente por brindarme la oportunidad y el apoyo de estudiar la carrera de Ingeniería en Computación.

A mis profesores por brindarme de su conocimiento y prepararme para mi vida profesional y personal.

A mi asesor de tesis por ayudarme en la realización de ésta tesis.

A mis amigos por acompañarme por el sendero de la amistad, apoyarme, tenerme paciencia y enseñarme diferentes facetas de la vida.

INDICE

INTRODUCCION

CAPITULO I

¿QUÉ ES UN MANEJADOR DE BASES DE DATOS?	1
1.1 ¿Qué es una base de datos?	2
1.2 Definiciones básicas de las bases de datos	3
1.3 ¿Qué es un manejador de bases de datos?	6
1.4 Manejadores de bases de datos utilizados en la actualidad	9
1.5 Estadísticas de utilización	10

CAPITULO II

ARQUITECTURA Y CARACTERÍSTICAS ESENCIALES DE LOS MANEJADORES DE BASES DE DATOS	13
2.1 ORACLE	14
2.1.1 Tipos de datos	14
2.1.2 Integridad	20
2.1.3 Transacciones	21
2.1.4 Procedimientos almacenados	23
2.1.5 Triggers	27
2.1.6 Segmentos, extensiones y bloques	28
2.1.7 Concurrencia y tipos de bloqueos	30

2.1.8	Vistas	32
2.1.9	Seguridad	34
2.1.10	Replicación y comunicaciones remotas	37
2.2	SQL SERVER 2000	39
2.2.1	Tipos de datos	39
2.2.2	Integridad	43
2.2.3	Transacciones	44
2.2.4	Procedimientos almacenados	45
2.2.5	Triggers	46
2.2.6	Páginas de datos y extensiones	47
2.2.7	Concurrencia y tipos de bloqueos	48
2.2.8	Vistas	52
2.2.9	Seguridad	53
2.2.10	Replicación y comunicaciones remotas	57
2.3	ACCESS 2002	59
2.3.1	Tipos de datos	59
2.3.2	Integridad	61
2.3.3	Transacciones Procedimientos almacenados y Triggers	62
2.3.4	Almacenamiento	63
2.3.5	Concurrencia y tipos de bloqueos	64
2.3.6	Vistas	65
2.3.7	Seguridad	65

2.3.8	Replicación y comunicaciones remotas	68
2.4	MYSQL	70
2.4.1	Tipos de datos	70
2.4.2	Integridad	73
2.4.3	Transacciones	73
2.4.4	Procedimientos almacenados y Triggers	74
2.4.5	Concurrencia y tipos de bloqueos	75
2.4.6	Vistas	76
2.4.7	Seguridad	77
2.4.8	Replicación y comunicaciones remotas	81

CAPITULO III

	ASPECTOS GENERALES DE LOS MANEJADORES DE BASES DE DATOS	83
3.1	ORACLE	84
3.1.1	Características que ofrece	84
3.1.2	Recursos que utiliza	85
3.1.3	Precio en el mercado	87
3.1.4	Que lenguajes de programación lo pueden utilizar	89
3.2	SQL SERVER 2000	90
3.2.1	Características que ofrece	90
3.2.2	Recursos que utiliza	91
3.2.3	Precio en el mercado	93

3.2.4	Que lenguajes de programación lo pueden utilizar	95
3.3	ACCESS 2002	96
3.3.1	Características que ofrece	96
3.3.2	Recursos que utiliza	98
3.3.3	Precio en el mercado	99
3.3.4	Qué lenguajes de programación lo pueden utilizar	100
3.4	MYSQL	100
3.4.1	Características que ofrece	100
3.4.2	Recursos que utiliza	101
3.4.3	Precio en el mercado	102
3.4.4	Qué lenguajes de programación lo pueden utilizar	103

CAPITULO IV

	VENTAJAS Y DESVENTAJAS DE LOS MANEJADORES DE BASES DE DATOS	104
4.1	ORACLE	105
4.1.1	Ventajas de su utilización	105
4.1.2	¿Qué inconvenientes presenta?	107
4.2	SQL SERVER 2000	108
4.2.1	Beneficios que aporta	108
4.2.2	Desventajas al ser usado	109
4.3	ACCESS 2002	110
4.3.1	Ventajas de su utilización	110

4.3.2	¿Qué inconvenientes presenta?	111
4.4	MYSQL	113
4.4.1	Beneficios que aporta	113
4.4.2	Desventajas al ser usado	114
4.5	Conclusión	115

CONCLUSIÓN

BIBLIOGRAFIA

INTRODUCCIÓN

Actualmente la utilización de las bases de datos es imprescindible en cualquier institución o empresa que desee tener un control y administración de su información. Para lograr tener su información bien organizada y para poder acceder a ella necesitan tener un manejador de bases de datos que cubra todos sus requerimientos, desde la integridad de la información, la emisión de reportes, la velocidad con que lo hace, seguridad en cuanto al acceso a la información y el respaldo de la misma, control de usuarios que se conectan al base de datos, etc., hasta todos aquellos requerimientos más específicos que varían en cada empresa negocio o institución dependiendo del manejo de las mismas.

He aquí la importancia de saber cuál manejador de bases de datos debemos de implementar y si va a satisfacer en lo más posible los requerimientos que se deben de cubrir para el buen funcionamiento de la empresa.

Las personas encargadas de tomar este tipo de decisiones generalmente son personas relacionadas con el área de computación o informática que cuando se le pide implantar una base de datos deben considerar un gran número de variables de la empresa, equipo y manejadores de bases de datos para tomar la decisión más acertada, considerando, por supuesto, también el aspecto económico.

Sin embargo, no todas las personas encargadas de elegir el manejador de bases de datos conocen las diferentes opciones para elegir el adecuado, por lo que la investigación de esta tesis les será de gran ayuda para tomar una decisión independientemente de su grado de conocimiento sobre el tema, ya que les puede ayudar a reducir el tiempo de investigación para hacer una elección e

incluso, si no conocen alguno de los manejadores de bases de datos aquí mencionados, les podrá ampliar su criterio.

Esta tesis está enfocada a cuatro manejadores de bases de datos: Oracle (8i y 9i), SQL Server 2000, MySQL y Access 2002. Sólo se evalúan estos cuatro debido a que existe una gran variedad de manejadores y al ser muy extenso el hablar de ellos se optó por investigar cuáles son los más competitivos y utilizados en la actualidad dando por resultado los cuatro manejadores antes mencionados. Si usted es diestro en el tema se habrá dado cuenta que Access no es un manejador de bases de datos; sin embargo, se ha considerado en la evaluación de esta tesis debido a la gran utilización que tiene en el desarrollo de bases de datos sobretodo para los pequeños negocios.

En el primer capítulo abarco los temas de lo que es una base de datos y un manejador de bases de datos, así como los términos que conllevan y la explicación correspondiente de lo que es. Esto es con el propósito de que toda persona que lea esta tesis conforme vaya avanzando entienda el sentido de la lectura, y la pueda entender claramente si no conoce mucho del tema. También abarco en este capítulo la investigación correspondiente que me llevó a elegir los manejadores de bases de datos a desarrollar en esta tesis.

En el capítulo segundo se abarcan todas las herramientas y características específicas de cada uno de los manejadores, se hace una descripción de lo que nos ofrece para su manejo y para almacenar de forma segura nuestra información en una base de datos. Algunos de las características que aquí se mencionan son: los tipos de datos, integridad, el manejo de transacciones, procedimientos almacenados, Triggers o disparadores, vistas, seguridad, comunicaciones remotas, concurrencias y bloqueos. Con esto se pretende dar una idea, lo más clara posible, de lo que nos ofrece cada uno de los manejadores a evaluar y ayudar a tomar una decisión más apropiada al momento de su elección.

En el tercer capítulo hablo de los aspectos generales de estos manejadores como lo son sus características generales, los recursos de hardware y software que necesitan para su implantación, su precio y los diferentes lenguajes de programación con las que se pueden desarrollar aplicaciones que accedan a estos manejadores.

Por último, en el cuarto capítulo hago una evaluación de estos manejadores resaltando sus ventajas y desventajas al ser utilizado y así tratar de abarcar todos los aspectos necesarios para hacer la elección más adecuada del manejador de bases de datos con el que se va a trabajar.

Espero que esta investigación sea del agrado del lector y le sirva de referencia y/o ayuda para el desempeño de sus actividades profesionales.

CAPITULO I

¿Qué es un Manejador de Bases de Datos?

En este capítulo hablaremos de las definiciones más utilizadas para poder adentrarnos al tema. Asimismo, comprenderemos qué es una Base de Datos (BD), un Manejador de Bases de Datos (DBMS).

1.1 ¿QUÉ ES UNA BASE DE DATOS?

Una Base de Datos, gestionada o no por una computadora, no es más que la recopilación ordenada de información relacionada. Una base de datos almacena la información de forma segura y la organiza con vistas a una rápida recuperación.¹

Una base de datos relacional se define como una colección de datos cuyos elementos tienen vínculos entre sí; es decir, es una colección de datos afines.²

Las bases de datos que administran la mayoría de los DBMS son bases de datos relacionales que están conformadas por tablas y columnas relacionadas entre sí. Estas relaciones están basadas en valores claves contenidos en una columna.³

Con base en lo anterior podemos decir que: **una base de datos es un conjunto de entidades o tablas que se encuentran relacionadas entre sí con la finalidad de mantener la integridad y la seguridad de la información, por medio de valores claves ubicados en una o varias columnas de dichas tablas, con la intención de minimizar la redundancia de la información o duplicidad de datos innecesarios.**

Pero, ¿qué es una tabla, columna, relación, etc.? Muchas de estas definiciones son usadas como parte fundamental para el diseño de una base de datos, por eso es necesario definir las.

¹ Ver en BOBROWSKI, Steve, *Oracle 8i para Linux Edición de Aprendizaje*, Osborne, , España, 2001,p. 3

² Vid SMITH, Curtis y AMUNDANSEN, Michael, *Aprendiendo Programación de bases de datos con Visual Basic 6 en 21 días*, Prentice Hall, México, 1999,p.21.

³ Véase MASLAKOWSKI, Marky y BUTCHER, Tony, *Aprendiendo MySQL en 21 días*, Pearson Educación, México, 2001,p.13

1.2 DEFINICIONES BÁSICAS DE LAS BASES DE DATOS

Tabla: Conjunto de campos o columnas que describen a una entidad. Entendiéndose por entidad el objeto del cual se desea guardar información, como puede ser libros, clientes, etc.

Campos o Columnas: son los atributos que describen una entidad. Por ejemplo: de nuestra entidad Cliente, hay varios datos que nos interesan como el domicilio, el RFC, etc. Estos atributos se almacenan en las columnas y éstas deben de ser tantas como datos se deseen almacenar de la entidad.

Atributos: Son las características o cualidades que nos interesan de una entidad. Las hay de 2 tipos: descriptor e identificador.

El *descriptor* es nombrado así porque nos describe una característica de la entidad, por ejemplo: Nombre, Domicilio, Teléfono.

En cambio, el atributo *identificador* identifica de manera única los registros de una tabla. Un ejemplo de utilizar un atributo identificador es el guardar en un campo en especial algo (la mayoría de las veces un número) que identifique de manera única a los clientes que se estén registrando. Es decir, podemos asignarle a cada cliente registrado una clave que solo le pertenezca a ese cliente. De esta manera se puede tener un acceso más rápido cuando se desee buscar a alguno de ellos, así como para que haya un mejor control de la seguridad en la base de datos. Los atributos identificadores generalmente se utilizan como llaves primarias y/o como llaves foráneas.

Registro o fila: Es el conjunto de campos que guardan el contenido de una entidad en específico. Por ejemplo de una Tabla de Cliente un registro sería la información almacenada en los campos de dicha tabla de un cliente en concreto como se ve en la siguiente figura:

ClientePk	NomCliente	DomCliente	RFCCliente	TelCliente	NegocioPk
1	Gloria Gonzale; los pinos		GOGG830330	60091	1

Figura 1

Llave Primaria: Es el campo que va a identificar de manera única a los registros de una tabla.

Llave Foránea: Es campo de una tabla, pero que es una llave primaria trasladada o copiada de otra tabla para poder realizar una relación entre dichas tablas.

Relaciones: Es el vínculo existente entre dos tablas para unirlos por medio de un campo en común. Estos campos son las llaves primarias y las llaves foráneas. Hay tres tipos de relaciones:

Relación uno a uno (1:1). Se refiere que entre dos tablas relacionadas el contenido de una llave foránea sólo puede recibir un solo dato de la llave primaria de la otra tabla y ningún otro dato diferente a ese.



Figura 2

Relación uno a muchos (1:M). Hay una relación uno a muchos cuando la llave foránea de una tabla puede recibir un dato diferente de la llave primaria a la que hace referencia, por ejemplo, si en la tabla de Cliente hay una llave foránea de la tabla Negocio, y la relación es 1:M de Negocio hacia Cliente, significa que la Tabla Cliente en la llave foránea de NegocioPk puede recibir un valor diferente cada vez que se inserte un registro.

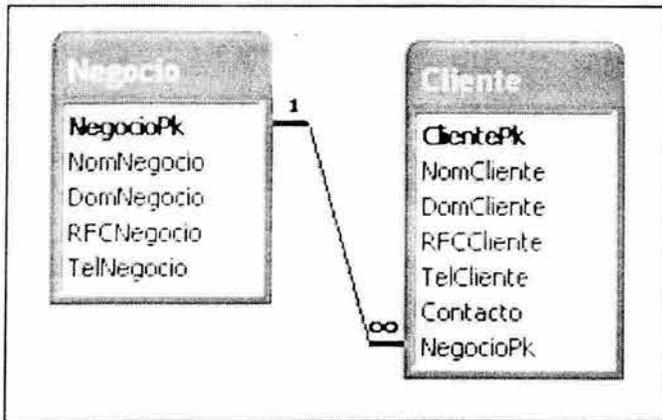


Figura 3

Relación Muchos a Muchos: Se da cuando en una tabla que está relacionada con otra puede recibir varios valores en sus llaves foráneas tanto en una tabla como en la otra.

Como este tipo de relaciones causa conflicto para el control de la base de datos, en las bases relacionales se optó por generar una tercera tabla que contenga las llaves foráneas de las dos tablas mencionadas, de tal manera que en vez de hacer referencia directa entre ellas lo hagan por medio de la tercera tabla. Ejemplo: Si en un negocio que realiza ventas y tiene un registro de las facturas y de sus productos, para una venta una factura puede contener varios productos y un producto puede pertenecer a varias ventas, por lo que se crea una tercera tabla que llamaremos Detalle Factura donde contendrá las llaves foráneas de factura y producto para que estas dos tablas hagan referencia a la de Detalle Factura y así evitar conflictos en la administración de la base de datos.

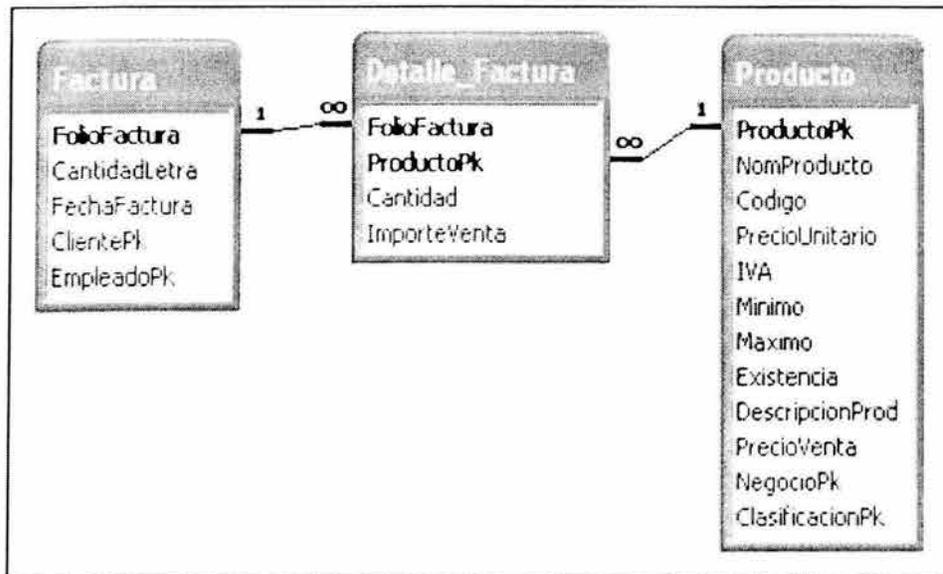


Figura 4

1.3 ¿QUÉ ES UN MANEJADOR DE BASES DE DATOS?

Un sistema Manejador de Bases de Datos (DBMS) es una colección de numerosas rutinas de software interrelacionadas, cada una de las cuales es responsable de una tarea específica. Su objetivo primordial es proporcionar un contorno que sea a la vez conveniente y eficiente para ser utilizado al extraer, almacenar y manipular la información de la base de datos.⁴

Al DBMS se le conoce también como Gestor de Bases de Datos.

La mayoría de los DBMS cuentan con un sistema de administración basado en un modelo relacional.

Las funciones principales de un DBMS son:

- Crear y organizar la Base de Datos.

⁴ Ver www.itlp.edu.mx/publica/tutoriales/basedat1/tema1_1.htm

- b) Establecer y mantener trayectorias de acceso a la base de datos de tal forma que los datos puedan ser accesados rápidamente.
- c) Manejar los datos de acuerdo a las peticiones de los usuarios.
- d) Registrar el uso de las bases de datos.
- e) Interacción entre el manejador de archivos, esto es a través de las sentencias en DML al comando de los sistemas de archivos. Así, el Manejador de la Base de Datos es el responsable del verdadero almacenamiento de las bases de datos.
- f) Respaldo y recuperación. Consiste en contar con mecanismos implantados que permitan la recuperación fácilmente de los datos en caso de ocurrir fallas en el sistema de bases de datos.
- g) Control de concurrencia. Consiste en controlar la interacción entre los usuarios concurrentes para no afectar la inconsistencia de los datos.
- h) Seguridad e integridad. Consiste en contar con los mecanismos que permitan el control de la consistencia de los datos evitando que éstos se vean perjudicados por cambios no autorizados o previstos.⁵

Componentes de un DBMS:

- *Data Definition Language* (DDL): Define elementos de los datos en la base de datos.
- *Data Manipulation Language* (DML): Manipula datos para aplicaciones.
- *Data dictionary*: Definiciones de todas las variables en la base.⁶

⁵ Ver www.itlp.edu.mx/publica/tutoriales/basedat1/tema1_9.htm

⁶ Véase <http://ict.udlap.mx/people/carlos/dbms/index.html>

Arquitectura Cliente-Servidor

Una aplicación cliente-servidor es un tipo de aplicación de procesamiento distribuido.

La arquitectura cliente-servidor permite diseñar y distribuir aplicaciones óptimas para una variedad de ambientes, la programación de interfaces clientes permite a la aplicación correr en diferentes computadoras y comunicarse sobre la red.

En una aplicación de cliente-servidor hay tres componentes: el cliente, el servidor y una red que conecta al cliente y al servidor.

CLIENTE: Es el primer plano de la aplicación que se utiliza para realizar el trabajo. Suele estar encargado de las siguientes operaciones:

- La presentación de una interfaz de usuario con la que se puede interaccionar, como una pantalla para la entrada de datos.
- La validación de entrada de datos, así como la comprobación de que se introduce una fecha válida en un campo de fecha.
- La solicitud de información a un servidor de base de datos, como los registros de clientes o los pedidos de ventas.
- El procesamiento de la información devuelta desde un servidor de base de datos, como llenar una pantalla con datos, calcular los totales de campos de un informe o crear gráficos.

SERVIDOR: Funciona en segundo plano para gestionar la base de datos entre todos los usuarios y las aplicaciones que lo utilizan para almacenar y recuperar datos. El servidor es responsable de las siguientes aplicaciones.

- Abrir una base de datos sin hacerla accesible para las aplicaciones.

- Impedir el acceso no autorizado a la base de datos manteniendo los controles de seguridad estrictos.
- Impedir que se interfieran de manera perjudicial las transacciones simultáneas que acceden a los mismos juegos de datos.
- Proteger una base de datos con funciones de copia de seguridad y recuperación de bases de datos.
- Mantener la integridad y la consistencia mientras muchos usuarios realizan su trabajo.

LA RED: Normalmente, los componentes cliente y servidor de una aplicación se suelen ejecutar en computadoras distintas que se comunican entre sí sobre una red. Para poder comunicarse deben utilizar un software de comunicación que les permita hablar el mismo lenguaje.⁷

1.4 MANEJADORES DE BASES DE DATOS UTILIZADOS EN LA ACTUALIDAD

Los sistemas manejadores de bases de datos más mencionados son: SQL Server, Access, Informix, Oracle, Sybase, MySQL, PostgreSQL, DB2, Interbase entre otros. Cada uno con sus propias características y con diferentes plataformas de trabajo, e incluso algunos multiplataforma. Pero, ¿cuáles realmente son los más usados y los más competitivos?

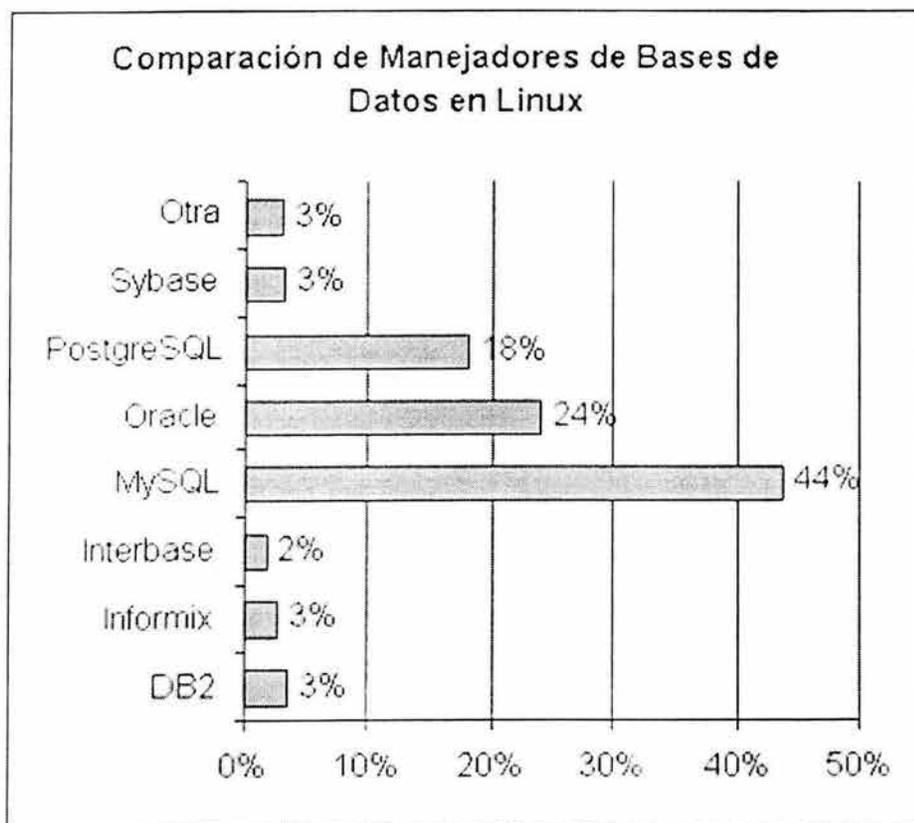
⁷ Ver en BOBROWSKI, Steve, *Oracle 8i para Linux Edición de Aprendizaje*, Osborne, España, 2001, pp. 48-49

1.5 ESTADÍSTICAS DE UTILIZACIÓN

En la actualidad existe una gran rivalidad entre los diferentes sistemas manejadores de bases de datos. Cada uno presume de ser el mejor y de presentar ventajas con respecto a sus rivales.

Pero no todos son realmente los más competitivos. Algunas páginas de Internet nos muestran estadísticas claras de cuáles DBMS son los que realmente están a la cabeza.

Por ejemplo, la página de Linux nos muestra que bajo su plataforma MySQL, Oracle y PostgreSQL –figura 5- son los DBMS más utilizados.



Fuente: linux.com

Figura 5

En cambio, la página <http://ict.udlap.mx/people/carlos/bases/dbms/index.htm> nos muestra la siguiente tabla donde nos indica cuáles manejadores de base de datos son los más utilizados a nivel mundial bajo Unix y Linux:

Top 5 Mundial

- 1 Oracle
- 2 Informix
- 3 Sybase
- 4 MySQL
- 5 PostgreSQL
- ...otros

A nivel estudiantil se tiene que el mejor Manejador de bases de datos es SQL Server 2000, según la tabla mostrada en la figura 6.

¿Cuál es el mejor RDBMS?

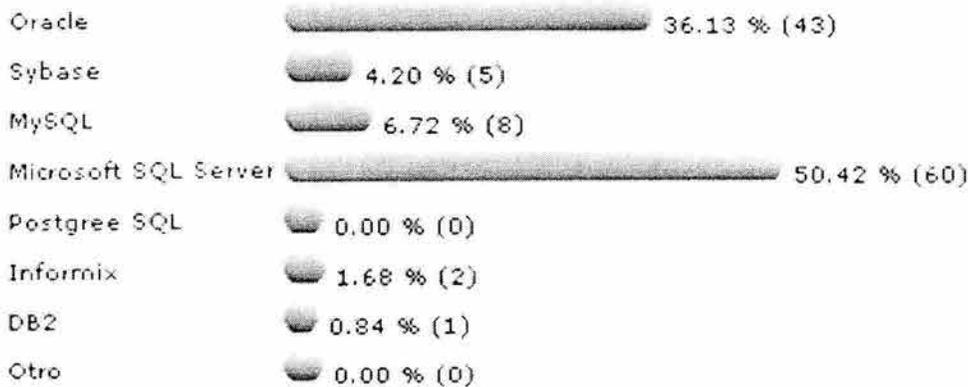


Figura 6.⁸

Estas gráficas nos muestran claramente que los DBMS más populares son Oracle, MySQL y SQL Server 2000. Es por eso que en los próximos capítulos analizaremos a

⁸ Véase <http://dei.uca.edu.sv/computacion/modules.php?name=Surveys&op=results&pollID=6>

estos manejadores de bases de datos para evaluarlos y detectar cuál es el más indicado para crear una base de datos para un negocio o empresa.

También analizaremos a Access 2002 como una alternativa más para la creación de base de datos. El motivo de esta decisión es que este software es muy utilizado en la región para la creación de bases de datos para pequeños negocios, así como estudiantes que aprenden a diseñar bases de datos por medio de Access.

CAPITULO II

Arquitectura y características específicas de los Manejadores de Bases de Datos

En el capítulo anterior comentamos las razones de por qué y cuáles DBMS elegí para el desarrollo de esta tesis. Pues bien, en este capítulo daremos una reseña de las características esenciales que tienen los manejadores de bases de datos mencionados en esta tesis como transacciones, triggers, seguridad, integridad, los tipos de datos que manejan, etc. las cuales nos proporcionan una idea general de la arquitectura de cada uno de estos DBMS así como su manera de trabajar y manipular la información de una base de datos. Al conocer estas características (que veremos en este capítulo) nos dará las herramientas necesarias para saber cuál manejador de bases de datos es el adecuado para que administre la información de un determinado negocio o empresa.

2.1 ORACLE

2.1.1 Tipos de datos

Oracle simplifica en algunos tipos de datos de origen ANSI estándar; por ejemplo, basándose en la tabla de abajo, los tipos de datos ANSI CHARACTER y CHAR se incluyen en el tipo de dato CHAR de Oracle. Los tipos de datos ANSI CHARACTER VARYING y CHAR VARYING son considerados en el tipo de datos VARCHAR2. Los tipos de datos ANSI NUMERIC, DECIMAL, DEC, INTEGER, INT y SMALLINT son reemplazados por el tipo de datos NUMBER de Oracle. También, los tipos de datos ANSI estándar FLOAT, REAL y DOUBLE PRECISION son admitidos por Oracle en el tipo de datos FLOAT.

Tipos de datos de Oracle⁹

Tipo de datos	Descripción
CHAR	Campo de caracteres de longitud fija, con un máximo de 2000 caracteres.
NCHAR	Campo de longitud fija para juegos de caracteres multibyte. El tamaño máximo es de 2000 caracteres o de 2000 bytes por fila, dependiendo del juego de caracteres, siendo la configuración predeterminada 1 byte.
VARCHAR2	Campo de caracteres de longitud variable, con una longitud máxima de 4000 caracteres.
NVARCHAR2	Campo de caracteres de longitud variable de caracteres multibyte. El tamaño máximo es de 4000 caracteres o 4000 bytes por fila, dependiendo el juego de caracteres, siendo la configuración predeterminada 1 byte.
DATE	Campo de longitud fija de 7 bytes, que se utiliza para almacenar todas las fechas. La hora se almacena como parte de la fecha. En las consultas, la fecha aparece en el formato DD-MON-YY a menos que modifique el formato de fecha configurando el parámetro de inicialización NLS_DATE_FORMAT.

⁹ LONEY, Kevin THERIAULT, Marlene TUSC, *Oracle9i Manual del Administrador*, OSBORNE, España, 2002, pp. 37-39

Tipos de datos de Oracle (continuación)

Tipo de datos	Descripción
INTERVAL DAY TO SECOND	Periodo de tiempo, almacenado en 11 bytes, representado como días, horas, minutos y segundos. Los valores de precisión especifican el número de campos DAY (día) y la parte decimal del campo SECOND (segundos) de la fecha. La configuración predeterminada de la precisión 2 es para los días y 6 para los segundos, pero se miden valores comprendidos entre 0 y 9.
INTERVAL YEAR TO MONTH	Periodo de tiempo almacenado en 5 bytes, representado como años y meses. El valor de precisión especifica el número de dígitos del campo YEAR (años) de la fecha. La configuración predeterminada de la precisión es 2 para años, pero puede ser cualquier valor entre 0 y 9.
TIMESTAMP	Valor que puede ocupar entre 7 y 11 bytes y que representa la fecha y hora, incluyendo fracciones de segundos. Se basa en el valor del reloj del Sistema Operativo. El valor de precisión especifica el número de dígitos de la parte de fracciones de segundo de la fecha SECOND (segundos). La configuración predeterminada de la precisión es 6, pero podría ser cualquier valor entre 0 y 9.

Tipos de datos de Oracle (continuación)

Tipo de datos	Descripción
TIMESTAMP WITH TIME ZONE	Un valor almacenado en 13 bytes, que representa una fecha y hora, además de una configuración de zona horaria. La zona horaria puede expresarse en relación al estándar UTC, como, por ejemplo, '5:0', o utilizando un nombre de región, como 'US/Pacific'.
TIMESTAMP WITH LOCAL TIME	Este tipo de datos, puede ocupar entre 7 y 11 bytes, es similar a TIMESTAMP WITH TIME ZONE con la salvedad de que los datos se normalizan con la zona horaria de la base de datos cuando se almacenan y se ajustan para que coincidan con la zona horaria del cliente cuando se consulten.
NUMBER	Columna numérica de longitud variable. Los valores permitidos son cero y los números positivos y negativos. El espacio de almacenamiento interno necesario para el valor NUMBER es aproximadamente la mitad del número de dígitos significativos del valor. Para calcularlo se toma la longitud total, por ejemplo 9, y se divide entre 2, se redondea el resultado a un número entero y se suma 1 para el número positivo. Por tanto, para almacenar un número de nueve dígitos se requieren 6 bytes.
LONG	Campo de longitud variable, con una longitud máxima de 2GB.
RAW	Campo de longitud variable para datos binarios, de una longitud máxima de 2000 bytes.

Tipos de datos de Oracle (continuación)

Tipo de datos	Descripción
LONG RAW	Campo de longitud variable para datos binarios, de una longitud máxima de 2GB.
BLOB	Objeto binario de gran tamaño, de hasta 4 GB de longitud.
CLOB	Objeto de caracteres de gran tamaño, de hasta 4 GB de longitud.
NCLOB	Tipo de datos CLOB para juego de caracteres multibyte, de hasta 4 GB de longitud.
BFILE	Archivo binario externo. El sistema Operativo determina el tamaño máximo.
ROWID	Datos binarios que representan un identificador de fila (RowID). Todos los RowID ocupan 6 bytes para índices normales de tablas no particionadas, índices locales en tablas particionadas y punteros de fila utilizados para filas encadenadas o migradas. El RowID ocupa 10 bytes sólo en índices globales de tablas particionadas.
UROWID	Datos binarios utilizados para direccionamiento de datos, de una longitud máxima de 4000 bytes que admiten valores de RowID tanto lógicos como físicos, además de tablas externas a las que se acceda a través de una pasarela.

TIPOS DE DATOS DEFINIDOS POR EL USUARIO

También son llamados tipos abstractos de datos. Estos tipos de datos se pueden definir a partir de que ha sido instalada la opción Object en la base de datos. Pueden ser usados para estandarizar la utilización de datos en las aplicaciones.

Por ejemplo, si se crea el tipo de datos NOM_TD con las partes de nombre, apellido paterno y apellido materno, este tipo de dato definido por el usuario puede ser utilizado en la creación de varias tablas donde el campo de nombre puede ser de tipo NOM_TD.

Los tipos abstractos de datos son el uso más sencillo de un **tipo de objeto**. Un **tipo de objeto** tiene atributos que describen el objeto correspondiente, además de métodos que permiten a las aplicaciones trabajar con objetos de clase.

Se crea un tipo de objeto para construir un tipo de datos complejo, definido por el usuario. El usuario y otros desarrolladores pueden utilizar este tipo de objeto para declarar y trabajar con las tablas y otros objetos más fácilmente.

La sintaxis en Oracle para crear un tipo de objeto, en este caso, un tipo abstracto de datos es:

```
create type NOM_TD as object  
  
Nombre      VARCHAR2(30),  
  
Apellido_P  VARCHAR2(20),  
  
Apellido_M  VARCHAR2(20));  
  
/
```

2.1.2 Integridad

Al trabajar con una base de datos relacional, nos referimos a que las tablas de esa base de datos se relacionan entre sí mediante columnas que tienen en común. Si estas relaciones están correctamente establecidas, se dice que se tiene una *integridad referencial*. La *integridad referencial* se realiza a nivel de la base de datos por medio de restricciones.

RESTRICCIONES

Cuando se crea una tabla, las columnas deben cumplir ciertas condiciones dependiendo de la restricción a la que son sometidas.

La *clave primaria* (PRIMARY KEY) es la columna o conjunto de columnas que hacen que cada registro de la tabla sea único. Al definir una clave primaria como NOT NULL se está aplicando una restricción para que todas las filas o registros de la tabla tengan un valor para esa columna, es decir, que no se debe dejar en blanco.

Existen otras restricciones como DEFAULT, CHECK, UNIQUE.

La restricción DEFAULT se utiliza para generar un valor en una columna cuando se inserta un registro en una tabla sin especificarse un valor para dicha columna.

La restricción CHECK es utilizada para asegurarse que los valores de una determinada columna cumplan con cierto criterio o condición, por ejemplo, que el valor de la columna sea mayor que cien. La restricción NOT NULL es tratada como una restricción CHECK por la base de datos.

La restricción UNIQUE asegura la unicidad de columnas que deben de ser unívocas pero que no son parte de la clave primaria.

La *clave externa* (FOREIGN KEY) es una restricción que se utiliza para especificar la naturaleza de la relación entre tablas. Una clave externa hace referencia a una clave primaria que se definió anteriormente en cualquier otra tabla de la base de datos.

2.1.3 Transacciones

Una transacción es una unidad de trabajo realizada por una o más sentencias SQL íntimamente relacionadas. Es decir, aunque una transacción esté compuesta de varias sentencias SQL, todas se validan o deshacen en una sola operación.

Para validar la transacción o unidad de trabajo se utiliza el comando COMMIT; de la misma manera, si se desea deshacer la transacción se utiliza el comando ROLLBACK.

Después de validar la transacción, Oracle realiza inmediatamente la siguiente transacción, por lo que no se puede deshacer una transacción después de validarla.

Si por algún motivo Oracle no puede validar el trabajo de una sentencia de una transacción, Oracle deshace automáticamente los efectos de todas las sentencias de la transacción, con lo que así garantiza que, en caso que se haya suscitado un error, la base de datos quedará con los datos como estaban antes de la transacción.

ADMINISTRACIÓN DE TRANSACCIONES

Cuando una base de datos es de gran tamaño, es conveniente clasificar cada una de las tablas que se crearon y se van creando en los siguientes tipos:¹⁰

¹⁰ LONEY, Kevin THERIAULT, Marlene TUSC, *Oracle9i Manual del Administrador*, OSBORNE, España, 2002, pp. 572

- **Tabla de códigos pequeña.** Las tablas código no suelen aumentar de tamaño.
- **Tabla de transacciones de gran tamaño.** Este tipo de tabla es la que se almacena la mayoría de los registros de la base de datos. Puede aumentar de tamaño con el tiempo; pero, aunque no aumente, suele registrar un gran volumen de transacciones.
- **Tabla de agregación.** Este tipo de tablas puede aumentar de tamaño con el tiempo, o puede mantener el tamaño constante, dependiendo del diseño de cada aplicación. Los datos se basan en las agregaciones de datos procedentes de tablas de transacciones de gran tamaño.
- **Tabla temporal de trabajo.** Las tablas temporales de trabajo se utilizan durante el procesamiento de la carga de datos y la manipulación de grandes cantidades de datos.

A continuación sólo se hablará de transacciones, en base a la clasificación anterior de una base de datos de gran tamaño, ya que es la finalidad de este subtítulo.

Las tablas de transacciones de negocio son las que almacenan la mayor cantidad de datos que hay en la base de datos. El dimensionamiento de las tablas se vuelve muy simple, si las tablas de transacciones de negocio almacenan siempre el mismo volumen de datos. En las bases de datos de gran tamaño, las tablas de transacciones de negocio se suelen cargar por medio de programas de procesamiento de lotes. Los sistemas por lotes, en lugar de tener muchas transacciones pequeñas, tienen pocas transacciones de gran tamaño (aunque continúa habiendo transacciones pequeñas).

Se debe tener cuidado en la ejecución de operaciones por lotes de gran tamaño al mismo tiempo que transacciones de menor tamaño, ya que este tipo de

situaciones son las que comúnmente causan más problemas con los segmentos de anulación (que veremos en la sección 2.1.6 de este capítulo) que se forman en la base de datos.

Por ejemplo: Un proceso por lotes que ejecute una carga¹¹ directa cambiará el estado de los índices al estado de carga, y mientras que los índices que se encuentran en el estado de carga, las actualizaciones en línea que se realicen en la tabla fallarán.

Lo recomendable es que las cargas por lotes se ejecuten cuando no se esté produciendo ningún procedimiento en línea.

2.1.4 Procedimientos almacenados

En Oracle se pueden almacenar funciones y procedimientos en la base de datos de tal manera que puedan ser utilizados por cualquier aplicación.

Estas funciones y procedimientos almacenados son subprogramas con instrucciones PL/SQL que pueden asumir parámetros y utilizarse para realizar tareas específicas.

Para crear un procedimiento almacenado en una base de datos de Oracle se utiliza el comando SQL CREATE PROCEDURE. Una vez declarado el procedimiento almacenado se pueden utilizar los dominios de privilegios AUTHID CURRENT_USER o AUTHID DEFINER como opciones para ejecutar el procedimiento.

- Δ Si se crea un procedimiento con la opción AUTHID CURRENT_USER, Oracle ejecutará el procedimiento con los privilegios del usuario que llama el procedimiento, por lo que el usuario debe tener los privilegios necesarios de

¹¹ En Oracle utilizan la palabra carga, para referirse a un almacenamiento.

acceder a los objetos a los que hace referencia el procedimiento para que éste se ejecute exitosamente.

- Δ Si se crea el procedimiento con la opción `AUTHID DEFINER`, Oracle ejecutará el procedimiento con los privilegios del propietario del procedimiento, por lo que el propietario debe tener los privilegios de acceso a los objetos a los que hace referencia el procedimiento para que éste se ejecute exitosamente cuando sea llamado dicho procedimiento. Esta opción es la más adecuada para evitar otorgar privilegios a todos los usuarios que necesitan llamar el procedimiento.

En la sección de seguridad, se hablará más acerca de los diferentes privilegios de acceso a la base de datos.

Para crear una función almacenada en Oracle se utiliza el comando SQL `CREATE FUNCTION`. En este caso hay que especificar el tipo de devolución de la función, además de utilizar la sentencia `RETURN` para regresar el valor de la devolución de la función.

PAQUETES

Un paquete es un conjunto de procedimientos, funciones y otras instrucciones PL/SQL, almacenadas todas en una base de datos como una unidad.

Son útiles para agrupar una serie de funciones y procedimientos relacionados con una aplicación específica de la base de datos.

Un paquete se conforma de dos partes: una especificación y un cuerpo

- ▼ En la especificación de paquete se declaran las variables, constantes, procedimientos, funciones y otras construcciones que se quiere estén disponibles para los programas del paquete; todos éstos se declaran como

públicos. Una especificación de paquete se declara con el comando SQL CREATE PACKAGE.

- ▼ En el cuerpo de paquete se definen todas las funciones y procedimientos que se declararon en la especificación. También pueden ir otras definiciones de construcción que no fueron declaradas en la especificación del paquete y que se considerarán como privadas. Para declarar el cuerpo del paquete se utiliza el comando SQL CREATE PACKAGE BODY.

Oracle incluye varios paquetes de utilidades. En la siguiente tabla se enumeran algunos.

Tabla: Varios paquetes incorporados a partir de Oracle8i ¹²

Nombre del paquete	Descripción
DBMS_ALERT	Procedimientos y funciones que permiten a las aplicaciones asignar nombre e indicar condiciones de alerta sin sondear.
DBMS_AQ DBMS_AQADM	Procedimientos y funciones para poner en cola la ejecución de transacciones y administrar mecanismos de gestión de colas.
DBMS_DLL DBMS_UTILITY	Procedimientos que proporcionan acceso a un número limitado de sentencias DLL dentro de programas PL/SQL.
DBMS_DESCRIBE	Procedimientos que describen la API para procedimientos y funciones almacenados.

¹² Ver en BOBROWSKI, Steve, *Oracle 8i para Linux Edición de Aprendizaje*, Osborne, España, 2001, pp. 150-151.

Tabla: Varios paquetes incorporados a partir de Oracle8i (continuación)

Nombre del paquete	Descripción
DBMS_JOB	Procedimientos y funciones para gestionar los mecanismos de gestión de cola de trabajos de una base de datos.
DBMS_LOB	Procedimientos y funciones para manipular los BLOB, CLOB, NCLOB y BFILE.
DBMS_LOCK	Procedimientos y funciones que permiten a las aplicaciones coordinar el acceso a recursos compartidos.
DBMS_OUTPUT	Procedimientos y funciones que permiten a un programa PL/SQL generar salidas de terminal.
DBMS_PIPE	Procedimientos y funciones que permiten a las sesiones de base de datos comunicarse usando canales (de comunicación).
DBMS_ROWID	Procedimientos y funciones que permiten a las aplicaciones interpretar fácilmente un ROWID externo de caracteres de 64 base.
DBMS_SESSION	Procedimientos y funciones para controlar la sesión del usuario de una aplicación.
DBMS_SQL	Procedimientos y funciones para realizar SQL dinámico desde un programa PL/SQL.
DBMS_TRANSACTION	Procedimientos para realizar una cantidad limitada de control de transacciones.

Tabla: Varios paquetes incorporados a partir de Oracle8i (continuación)

Nombre del paquete	Descripción
UTL_FILE	Procedimientos y funciones que permiten a un programa PL/SQL leer y escribir archivos de texto en el sistema de archivos del servidor.

2.1.5 Triggers

Un Trigger o *disparador de base de datos* es un procedimiento almacenado que se ejecuta automáticamente cuando se satisfacen las condiciones de ejecución del disparador. Está asociado con una tabla específica.

Para crear un Trigger se utiliza el comando SQL CREATE TRIGGER.

Un Trigger está conformado por las siguientes partes:

- ❖ Listas de sentencias del disparador, incluyendo INSERT, UPDATE y DELETE, que inician el disparador.
- ❖ Un Trigger se puede accionar antes o después de la sentencia del disparador. Para que se accione antes se indica con la instrucción BEFORE. Para que se accione después se utiliza la instrucción AFTER.
- ❖ Puede ser un Trigger de sentencia o de fila. Un Trigger de sentencia se dispara sólo una vez, independientemente del número de filas afectadas. Un Trigger de fila se activa una vez por cada fila de la tabla afectada por las sentencias.

En Oracle se pueden crear Triggers INSTEAD OF. Este tipo de disparadores hace que la instrucción que hizo que se disparara –sentencias del disparador-, no se ejecute y en su lugar se ejecute el código del Trigger.

También se pueden configurar los Triggers para que se ejecuten cuando se utilicen los comandos CREATE, ALTER o DELETE.

2.1.6 Segmentos, extensiones y bloques

Los *segmentos* son la entidad física donde se almacenan los objetos de base de datos lógicos. Los *segmentos* almacenan datos. Por ejemplo, los segmentos de índice almacenan datos asociados con los índices.

Un segmento se compone de varias secciones llamadas *extensiones (conjuntos contiguos de bloques Oracle)*. Cuando las extensiones de un segmento ya no pueden contener más datos, el segmento tiene que adquirir otra extensión. Un segmento puede contener tantas extensiones necesite tener, o bien hasta un número máximo de extensiones.

Cuando se elimina un segmento se liberan las extensiones que éste utilizaba. A estas extensiones liberadas se les denomina Extensiones libres. Estas extensiones pueden ser utilizadas por nuevos segmentos, o bien para ampliar otras extensiones de segmentos ya existentes.

A continuación se mencionan los diferentes tipos de segmentos que existen en Oracle:

- ❖ TABLE
- ❖ INDEX
- ❖ ROLLBACK o ANULACION

✧ TEMPORARY

✧ PARTITION

✧ CLUSTER

SEGMENTOS DE TABLA. También denominados *segmentos de datos*, son los que guardan las filas de los datos asociadas a las tablas o clusters. Cada segmento de datos contiene un bloque de cabecera, que sirve como directorio del espacio ocupado por el segmento.

SEGMENTOS DE INDICE. Los segmentos índice conservan el espacio que se les ha asignado hasta que los eliminen. Si están guardados junto a las tablas o clusters que indexan, al eliminar estas tablas o clusters, se eliminan indirectamente.

SEGMENTOS DE ANULACIÓN. Cuando se inicia una transacción, Oracle empieza escribiendo una entrada en un segmento de anulación. La entrada empieza escribiendo en el siguiente bloque disponible dentro de la actual extensión del segmento de anulación al que está asignada la transacción. Cada bloque situado dentro de esa extensión solamente debe contener información para una transacción activa. Cuando el usuario ejecuta un COMMIT, confirmando la transacción, Oracle comprueba que haya al menos 400 bytes de espacio libre disponible en el bloque. Si es así, el bloque se sitúa en el área de bloques libres del segmento de anulación.

SEGMENTOS TEMPORALES. Sirven para guardar los datos temporales durante las operaciones de ordenación (por ejemplo consultas). Cada usuario tiene un espacio de tablas temporal que se especifica al crear su cuenta.

2.1.7 Concurrencias y tipos de bloqueos

En Oracle para controlar la concurrencia en las bases de datos multiusuario se utilizan dos niveles de bloqueo:

- Bloqueo exclusivo. Cuando se realiza una transacción con este tipo de bloqueo, no permite compartir los recursos que utiliza.
- Bloqueo compartido. Muchas transacciones pueden tener este tipo de bloqueo sobre un mismo recurso.

Oracle también utiliza **multiversión** para permitir un acceso concurrente, es decir, proporciona automáticamente **consistencia de lectura a nivel sentencia o a nivel transacción** con lo cual se garantiza que los datos devueltos son consistentes con respecto al tiempo en que fue solicitada la consulta o transacción.

Cuando se realiza una transacción de solo lectura se debe indicar con la sentencia READ-ONLY a excepción de que se vayan a modificar, borrar o insertar datos, ya que en este caso se debe utilizar bloqueos.

Para esto Oracle permite tres niveles de aislamiento:

- ♦ READ-COMMITTED: Este nivel es el utilizado por defecto. Cada consulta de una transacción sólo ve los datos que fueron confirmados antes de que comenzara la consulta. Es apropiado para la mayoría de las transacciones. Es poco probable que se ejecute una consulta más de una vez.
- ♦ SERIALIZABLE TRANSACTIONS: se ven solamente los cambios realizados por las transacciones confirmadas y también los cambios efectuados por ella misma. Es adecuado cuando es poco probable que dos transacciones concurrentes modifiquen las mismas filas y cuando las transacciones de larga duración son principalmente de solo lectura.

- ♦ READ-ONLY: este tipo de transacciones ven los datos confirmados antes de empezar pero no permiten modificaciones a los datos.

En cuanto a los bloqueos, Oracle define de manera automática los bloqueos necesarios dependiendo de la sentencia que se valla a ejecutar.

Utiliza el nivel menos restrictivo basándose en las siguientes reglas:

- » Las operaciones de lectura no esperan a las de escritura sobre los mismos datos.
- » Las operaciones de escritura no esperan a las de lectura sobre los mismos datos.
- » Las operaciones de escritura solamente esperan a otras operaciones de escritura si intentan modificar los mismos datos.

También se utilizan bloqueos a nivel fila que consiste en que una transacción tendrá que esperar para modificar un registro cuando una transacción diferente lo haya bloqueado y no ha confirmado la transacción.

Bajo ciertas circunstancias, un usuario puede omitir el bloqueo por defecto y hacer un bloqueo manual con la instrucción LOCK TABLE con la cual se permite bloquear una o varias tablas.

Si se llega a dar un íterbloqueo (cuando dos o más usuarios están esperando datos bloqueados por otros) Oracle lo detecta automáticamente y lo resuelve abortando una de las transacciones.

2.1.8 Vistas

Una vista es muy similar a una tabla, e incluso se consulta de la misma forma que una tabla. Una vista puede mostrar columnas de más de una tabla. A las tablas de las cuales es sacada la información para formar una vista se le conoce como tabla subyacente. Las vistas sólo muestran la información de las tablas subyacentes, pero no almacenan físicamente los datos. La definición de una vista, en Oracle, se almacena en el diccionario de datos.

Cuando se consulta una vista, en realidad la vista hace una consulta a las tablas en que se basa y devuelve los datos de las columnas (y en el mismo orden) especificadas por ésta.

Las vistas se utilizan como un mecanismo de seguridad, Por ejemplo, se podría permitir a un usuario acceder a una vista con información que sólo se le quiera mostrar a ese usuario, sin tener que concederle el acceso a las tablas que conforman la vista y se corra el riesgo de que vea registros de los cuales no se desea sean del conocimiento de dicho usuario.

Además de las vistas normales, en Oracle reutilizan las **vistas objeto** y las **vistas materializadas**.

Se pueden definir *vistas objeto* para crear un nivel de orientado a objetos. Las *vistas objeto* pueden simular tipos abstractos de datos, identificadores de objeto (OID)¹³ y referencias. Con ellas se puede proporcionar una presentación de tipo objeto-relación a los datos relacionales.

En Oracle también se pueden utilizar las *vistas materializadas*, las cuales se utilizan para proporcionar copias locales de datos remotos, es decir, se basa en una consulta que se enlaza a una base de datos remota para seleccionar los datos y

¹³ Cuando se crean bases de datos orientadas a objetos, se utilizan tipos de objeto para crear tablas de objeto, que se definen utilizando un solo tipo de objeto. Cada objeto en una tabla de objetos tiene un identificador de objetos (OID) que Oracle indexa para garantizar que es único entre los demás OID.

los muestra donde fue solicitada dicha vista. También se utilizan para resumir, precalcular, duplicar o distribuir datos. Las *vistas materializadas* se pueden manipular como sólo de lectura, o bien que se pueda actualizar.

La utilización de las *vistas materializadas* en las bases de datos de gran tamaño, incrementan la velocidad de respuesta a las consultas que requieren agregaciones como sumas, recuentos, medias, varianzas, desviaciones típicas, máximos y mínimos, o bien, en la combinación de tablas.

A diferencia de las vistas normales, las *vistas materializadas* almacenan datos y ocupan un espacio físico (tabla maestra) en la base de datos. En las *vistas materializadas* se insertan datos a partir de las consultas base, por lo que siempre que se modifiquen los datos a los que acceden las consultas base, se deberán refrescar las *vistas materializadas* para que reflejen los cambios.

Si la consulta base de la *vista materializada* es muy compleja, ésta podría utilizar un *REGISTRO DE VISTAS MATERIALIZADAS*, el cual debe ser creado antes que la vista materializada. Un registro de vistas materializadas es una tabla que almacena las modificaciones realizadas a la tabla maestra de la *vista materializada*. Se almacena en la misma base de datos que la tabla maestra.

La integridad referencial en las vistas materializadas no puede ser garantizada, si las tablas subyacentes se refrescan a diferentes instantes, o si se producen transacciones en la tabla maestra durante los refrescos. Para resolver esto existen varias soluciones:

- ☆ Realizar el refresco cuando las tablas no estén en uso.
- ☆ Inmediatamente después bloquear las tablas maestras o de interrumpir el funcionamiento de la base de datos, realizar el refresco manualmente mediante el procedimiento REFRESH del paquete DBMS_REFRESH.

- ☆ Combinar las tablas de la vista materializada. En base a las tablas maestras, relacionadas entre sí apropiadamente, crear una vista materializada más compleja.
- ☆ Usar los grupos de refresco cuyo propósito es coordinar los planes de refresco de sus miembros.

2.1.9 Seguridad

Oracle maneja toda una gamma de seguridad para tener un control preciso referente a la seguridad de acceso a los datos, bases de datos y todo lo relacionado con su creación, control y administración. A continuación se mencionan algunas características de seguridad que trabaja este DBMS.

Seguridad de cuentas

Las cuentas de las bases de datos son protegidas mediante contraseñas. Se pueden crear cuentas que sean capaces de tener un inicio de sesión automático que permita a los usuarios (que accedieron con una cuenta a un host) acceder automáticamente a una base de datos sin que se tenga que introducir otra contraseña para esa base de datos. Cabe aclarar que si se posee una cuenta para acceder a una base de datos, esa base de datos será la única que podrá acceder, no podrá tener acceso a ninguna otra.

Privilegios de nivel de sistema

Oracle posee ciertos privilegios a nivel de sistema (CREATE TABLE, ALTER TABLE, CREATE INDEX, SELECT ANY TABLE, etc.) que dependiendo de la cuenta que se tenga, un usuario los puede poseer. Se pueden crear nuevos roles a parte de los ya definidos (CONNECT, RESOURCE y DBA; usuarios de aplicaciones,

desarrolladores y administradores, respectivamente), para otorgar diferentes privilegios de nivel de sistema a los usuarios.

Seguridad de los objetos

Si un usuario ha creado un objeto, dicho usuario puede otorgar privilegios sobre el objeto a otros usuarios por medio del comando GRANT. También puede conceder a un usuario acceso a las tablas con la opción de concesión.

Auditoría

Las actividades de los usuarios que afectan a objetos de la base de datos como accesos a tablas, intentos de conexión a la base de datos, entre otros, se pueden auditar a través del comando AUDIT. Los resultados de esta auditoría se almacenan en una tabla auditora en la base de datos.

Auditoria de granularidad fina

Oracle9i ofrece un paquete PL/SQL para poder tener un seguimiento de la información a la que se ha accedido y de la forma en que se ha modificado dicha información.

Base de datos privada virtual

Proporciona un control de acceso de granularidad fina. Si una organización cuenta con una base de datos privada virtual, sólo necesita crear una estructura de seguridad una vez en el servidor de datos. Esto se debe a que las directrices de seguridad se añaden a los datos en vez de la aplicación. De esta manera los datos presentados en una consulta son idénticos independientemente del modo de conexión (desde una aplicación SQL*Plus o un controlador ODBC).

COPIAS DE SEGURIDAD

La base de datos de Oracle incorpora varias opciones de **copia de seguridad** y recuperación:¹⁴

Exportar e importar (Export/Import). Export realiza una lectura lógica de la base de datos, para esto lee las tablas completas del diccionario de datos. Con Import se puede elegir los objetos o usuarios del archivo de volcado para insertarlos en la base de datos.

Copias de seguridad fuera de línea. Para poder realizarla, se cierra la base de datos para hacer la copia de los archivos que componen la base de datos en un dispositivo de almacenamiento.

Copias de seguridad en línea. Permite realizar copias de seguridad físicas mientras la base de datos está abierta.

Recovery Manager (RMAN). Realiza el seguimiento de copias de seguridad, bien a través de un catálogo de recuperación, o bien colocando la información requerida en el archivo de control de la base de datos a la que se le realiza la copia de seguridad.

Oracle SQL*Net admite conexiones de red entre servidores de bases de datos de Oracle y sus clientes. Se comunica con el protocolo de flujo de datos de sustrato de red transparente (TNS, *Transparent Network Substrate*) y permite a los usuarios ejecutar diferentes protocolos de red sin necesidad de escribir código especializado.

¹⁴ LONEY, Kevin THERIAULT, Marlene TUSC, *Oracle9i Manual del Administrador*, OSBORNE, España, 2002, pp. 56-57

2.1.10 Replicación y comunicaciones remotas

Para que las bases de datos de Oracle se COMUNIQUEN con otras bases de datos en una red se utiliza Oracle Net.¹⁵ Todos los servidores implicados deben estar ejecutando Oracle Net para que las bases de datos puedan comunicarse.

En cada uno de los host implicados en las comunicaciones de datos se ejecuta un proceso llamado *listener* (*proceso de escucha*), para que Oracle Net reciba y procese las comunicaciones.

Las consultas a bases de datos remotas utilizan *enlaces de base de datos*, que identifican el host, la base de datos y la cuenta que debe utilizarse, para identificar la ruta que debe seguir la consulta para localizar los datos. Los enlaces de la base de datos pueden ser públicos, disponible para todos los usuarios de la base de datos local, o privados: disponible únicamente para el usuario que haya creado el enlace.

Los enlaces de datos permiten que las consultas accedan a bases de datos remotas y permiten también que la información referente a la ubicación física de los datos sea transparente para el usuario.

Además de poder consultar datos almacenados en bases de datos remotas, también se puede actualizar utilizando transacciones que pueden modificar las bases de datos remotas, o también pueden combinarse con actualizaciones a la base de datos local en una misma unidad lógica de trabajo.

Un host que no contiene una base de datos puede acceder a una base de datos remota por medio de una aplicación que tenga la capacidad de acceder a la base de datos. Si se realiza este tipo de conexión al host que contiene la aplicación se le llama *cliente* y al que contiene la base de datos *servidor*. El cliente debe ejecutar Oracle Net.

¹⁵ Software de Oracle que se emplea para transferir datos entre bases de datos.

En esta configuración Cliente-servidor la base de datos se utiliza principalmente para operaciones de entrada y salida. Los procesamientos provocados por la ejecución de programas de aplicación pasan a realizarse en el host cliente, por lo que ayuda a reducir la cantidad de trabajo que realiza el servidor.

Se puede configurar una base de datos para que se pueda acceder a ella desde la Web. En cuanto a arquitectura, se basan en un modelo de tres niveles: cliente, servidor de aplicaciones y servidor de bases de datos. En el nivel cliente debe de ejecutarse un explorador de Web. El explorador se comunica con el software de aplicación (Oracle Net u otro) del nivel intermedio, servidor de aplicaciones, por medio del protocolo HTTP. A través del software de aplicaciones se pasan las solicitudes al servidor de bases de datos.

REPLICACION

Una replicación se refiere a utilizar la técnica de duplicación de una base de datos. Esta técnica es apropiada cuando se requiere introducir datos en un gran volumen de ubicaciones remotas. A este tipo de duplicación se le conoce como *Replicación multimaestro*. También se puede usar para crear bases de datos solo de lectura o informes y almacenes de datos y este tipo de duplicado se le conoce como *Replicación sólo de lectura*.

En la *Replicación multimaestro* se pueden enviar transacciones a las distintas bases de datos para que el contenido de todas las bases de datos esté relativamente actualizado. Es decir, si se modifican datos en cualquier replica de la base de datos, estos se deben enviar a las otras bases de datos de la red.

La *Replicación sólo de lectura* se utiliza principalmente para separar la base de datos origen y utilizar otras bases de datos para la generación de informes o bien, como bases de datos de prueba. Es más fácil de administrar y manipular que la *Replicación multimaestro*.

2.2 SQL SERVER 2000

2.2.1 Tipos de datos

SQL Server 2000 proporciona los tipos de datos que definen cada columna, variable, expresión o parámetro. Todos los tipos de datos tienen una longitud fija o variable. Los datos que maneja SQL Server son de tipo numérico, binario, decimal, monetario, de fecha, cadenas de caracteres, o bien definidos por el usuario.

A continuación se describen estos tipos de datos en las siguientes tablas:¹⁶

TIPOS DE DATOS PARA NÚMEROS Y UNIDADES MONETARIAS

Nombre	Rango-Descripción
ENTEROS	
Bit	0 ó 1.
Bigint	Números enteros de -2^{63} a 2^{63} .
Int	Números enteros de -2^{31} ($-2,147,483,648$) a $2^{31}-1$ ($2,147,483,647$).
Smallint	Números enteros de -2^{15} ($-32,768$) a $2^{15}-1$ ($32,767$).
Tinyint	Números enteros de 0 a 255.

¹⁶ STANEK, R. William, *Microsoft SQL Server 2000 Manual del administrador*, MC GRAW HILL INTERAMERICANA, España, 2001, pp. 247-249

TIPOS DE DATOS PARA NÚMEROS Y UNIDADES MONETARIAS (continuación)

Nombre	Rango-Descripción
MONETARIOS	
Money	Valores de moneda comprendidos entre -922,337,203,685,477,5808 y 922,337,203,685,477,5807
Smallmoney	Valores de moneda comprendidos entre -214,748.3648 y 214,748.3647
NÚMEROS EXACTOS	
Decimal	de -10^{38} a $10^{38}-1$.
Numeric	de -10^{38} a $10^{38}-1$.
NÚMEROS APROXIMADOS	
Float	de $-1.79E+308$ a $1.79E+308$.
Real	de $3.40E+38$ a $3.40E+38$.
NUMÉRICOS	
Cursor	Una referencia al cursor.
Rowversion	Un número unívoco para toda la base de datos, que indica la secuencia en que las modificaciones tienen lugar en la base de datos. Antes se le llamaba timestamp (marca temporal).
Sql_variant	Un tipo especial de datos que permite a una misma columna almacenar múltiples tipos de datos (excepto text, ntext, rowversion, y sql_variant).

TIPOS DE DATOS PARA NÚMEROS Y UNIDADES MONETARIAS (continuación)

Nombre	Rango-Descripción
NUMÉRICOS	
Table	Un tipo especial de datos usado para almacenar un conjunto de resultados temporalmente para su procesamiento. Puede usarse sólo para definir variables locales y como tipo de retorno para las funciones definidas por el usuario.
Uniqueidentifier	Un identificador globalmente unívoco (GUID).

TIPOS DE DATOS PARA VALORES DE FECHA, CARACTERES Y BINARIOS

Nombre	Rango-Descripción
FECHA	
Datetime	del 1 enero de 1753 al 31 de diciembre del 9999, con una precisión de tres centésimas de segundo.
Smalldatetime	del 1 enero de 1900 al 6 de junio del 2079, con una precisión de un minuto.
CARACTERES	
Char	Datos de caracteres de longitud fija no Unicode, con una longitud máxima de 8000 caracteres.

TIPOS DE DATOS PARA VALORES DE FECHA, CARACTERES Y BINARIOS

Nombre	Rango-Descripción	(continuación)
CARACTERES		
Varchar	Datos de longitud variable no Unicode, con una longitud máxima de 8000 caracteres.	
Text	Datos de longitud variable no Unicode, con una longitud máxima de $2^{31}-1$ (2,147,483,647) caracteres.	
Nchar	Datos Unicode de longitud fija, con una longitud máxima de 4000 caracteres.	
Nvarchar	Datos Unicode de longitud variable, con una longitud máxima de $2^{30}-1$ (1,073,741,823) caracteres.	

BINARIOS

Binary	Datos binarios de longitud fija, con una longitud máxima de 8000 bytes.	
Varbinary	Datos binarios de longitud variable, con una longitud máxima de 8000 bytes.	
Image	Datos binarios de longitud variable, con una longitud máxima de $2^{31}-1$ (2,147,483,647) caracteres.	

TIPOS DE DATOS DEFINIDOS POR EL USUARIO

Los tipos de datos definidos por el usuario son tipos de datos especiales basados en un tipo de datos nativo (del sistema). Se crean a nivel base de datos y sólo se

aplican a dicha base de datos, a menos que el tipo de datos definido por el usuario sea creado en la base de datos *model*, con lo cual existirá en todas las bases de datos.

Hay dos maneras de crear un tipo de datos definido por el usuario: una es por medio del administrador corporativo (Enterprise Manager) y la otra es por medio del procedimiento almacenado `sp_addtype`.

2.2.2 Integridad

SQL Server 2000 maneja 4 tipos de integridad:

- † INTEGRIDAD DE ENTIDAD. Establece a una fila como entidad única para una tabla definida. Fuerza la integridad de la(s) columna(s) de los identificadores mediante las restricciones PRIMARY KEY, UNIQUE o propiedades IDENTITY.
- † INTEGRIDAD DE DOMINIO. Está dada por la validación de las entradas para una columna determinada. Esto se puede hacer definiendo el tipo de dato para la columna, restringiendo el formato (con las reglas RULE o la restricción CHECK) o restringiendo el intervalo de valores posibles utilizando las definiciones DEFAULT, NOT NULL, la restricciones FOREIGN KEY y CHECK, o bien utilizando reglas.
- † INTEGRIDAD REFERENCIAL. Se basa en las relaciones entre las columnas que funcionan como llaves primarias y las que funcionan como llaves foráneas, así se garantiza que los valores de estas columnas sean coherentes en las diferentes tablas en las que se encuentran.

- † INTEGRIDAD DEFINIDA POR EL USUARIO. Aquí SQL Server permite al usuario definir reglas del negocio o compañía que no pertenecen a ningún tipo de integridad mencionado anteriormente, pero que será compatible con ellas.

SQL Server tiene el comando DBCC (*data-base consistency check*, comprobación de coherencia de la base de datos) que sirve para realizar comprobaciones de coherencia de manera manual. Aunque rara vez se necesita realizar este tipo de comprobaciones, pues es muy poco común que la base de datos se corrompa.

2.2.3 Transacciones

Como hemos visto anteriormente, una transacción es una unidad única de trabajo si tiene éxito todas las modificaciones de los datos realizados durante la transacción se confirman y se almacenan de forma permanente en la base de datos; por lo contrario, si encuentra errores debe deshacerse o cancelarse (se borran todas las modificaciones de los datos).

SQL Server para asegurar la integridad de la base de datos durante la transacción proporciona tres servicios:

- * Servicio de bloqueo para preservar el aislamiento de la transacción
- * Registro de transacciones. En este registro se almacena la información necesaria para restaurar la base de datos, es decir, que en caso de que haya una falla ya sea del servidor, sistema operativo o el propio SQL Server al momento de reiniciarse deshacerá automáticamente las transacciones incompletas en el momento e que se produjo el error en el sistema.

- * Administración de transacciones. Una vez iniciada la transacción debe concluirse correctamente, pues de lo contrario SQL Server cancelará las modificaciones de datos realizados desde que se inició la transacción.

SQL Server trabaja tres modos de transacción:

- ☞ Transacciones de confirmación automática. Cada instrucción individual es una transacción.
- ☞ Transacciones explícitas. Se especifica el inicio y término de la transacción con las instrucciones BEGIN TRANSACTION y COMMIT o ROLL EBACK, respectivamente.
- ☞ Transacciones implícitas. Se inicia implícitamente cuando ha terminado una transacción anterior, pero se tiene que completar con la instrucción COMMIT o ROLL EBACK.

2.2.4 Procedimientos Almacenados

Los procedimientos almacenados facilitan la administración de la base de datos. Son una colección de instrucciones SQL y de control de flujo almacenadas bajo un solo nombre y procesadas como una unidad. Estos se guardan en una base de datos y se pueden ejecutar desde una aplicación; puede aceptar parámetros proporcionar resultados aceptar variables declaradas por el usuario y de volver conjunto de resultados individuales o múltiples, todo dependiendo de cómo haya sido definido en el momento de su creación.

El usar procedimientos almacenados en vez de programas Transact_SQL,¹⁷ nos da ventajas como: permitir una ejecución más rápida, facilitar su mantenimiento, pues se le puede modificar independientemente del código del programa;

¹⁷ Versión mejorada del lenguaje estándar SQL. Es utilizado por SQL Server

reducir el tráfico de la red, pues no tardará tanto en ejecutarse un procedimiento almacenado que su equivalente en instrucciones Transact-SQL, usarse como mecanismo de seguridad, pues es posible crear procedimientos almacenados para conceder permisos a los usuarios.

Un procedimiento almacenado se define con el comando `CREATE PROCEDURE`, y se ejecuta con el comando `EXECUTE` o `EXEC` NomProcedimiento.

2.2.5 Triggers

Los desencadenadores o Triggers son procedimientos almacenados especiales que tienen la peculiaridad de ejecutarse automáticamente cuando se modifican los datos de una tabla a la que hace referencia.

Las instrucciones que pueden hacer ejecutar automáticamente un Trigger al modificar los datos de una tabla son `INSERT`, `UPDATE` y `DELETE`. La instrucción `CREATE TABLE` se puede definir con las cláusulas `FOR INSERT`, `FOR UPDATE` o `FOR DELETE` dependiendo de con cual de las estas tres instrucciones se quiere que se active el Trigger.

En SQL Server se puede definir cuando se quiere que se ejecute el desencadenador.

AFTER (después de). Se ejecuta tras la finalización de la instrucción que lo desencadena. No se puede especificar desencadenadores **AFTER** para vistas, sólo para tablas.

INSTEAD OF (en lugar de). Se ejecuta el desencadenador en lugar de la instrucción que lo desencadena. Se puede especificar desencadenadores **STEAD OF** para tablas y vistas.

Los Triggers pueden utilizar instrucciones SELECT y regresar un conjunto de resultados generado por ese SELECT. Pero no es muy utilizado este tipo de desencadenadores, a excepción de que se desee llenar algunos parámetros o para comparar datos.

2.2.6 Páginas de datos y extensiones

En las bases de datos SQL Server, las tablas se definen como objetos. Las tablas disponen de dos tipos de almacenamiento de datos: las páginas de datos y las extensiones.

PAGINAS DE DATOS

Son las unidades fundamentales de almacenamiento. Almacenan todos los tipos de datos exceptuando los tipos *text*, *ntext* e *image*. Tienen un tamaño fijo de 8 KB (8192 bytes). Cada página de datos tiene una cabecera de página, filas de datos y espacio libre que puede contener indicadores de desplazamiento de fila. La cabecera utiliza los primeros 96 bytes de cada página. Los desplazamientos de fila indican el orden lógico de las filas en una página, lo que significa que un desplazamiento 0 se refiere a la primera fila, un desplazamiento 1 corresponde a la segunda fila y así sucesivamente.

EXTENSIONES

Una extensión es un conjunto de ocho páginas de datos contiguas lo que significa que se encuentran en bloques de 64 KB y que hay 16 extensiones por MB. SQL Server maneja dos tipos de extensiones:

- ◊ Extensiones Mixtas. Objetos diferentes pueden tener páginas en la extensión.

- ◊ Extensiones Uniformes. Un único dato posee todas las páginas de la extensión.

En una extensión mixta SQL Server asigna páginas a la nueva tabla o índice, cuando estos se crean. La tabla o índice continúa utilizando la extensión mixta hasta que su crecimiento hace que se utilice las 8 páginas. Cuando esto sucede, cambia la tabla o el índice a una extensión uniforme.

2.2.7 Concurrencias y tipos de bloqueos

Los bloqueos son utilizados para asegurar la integridad de las transacciones y la coherencia de la base de datos. El bloqueo impide que los usuarios lean los datos que otros están modificando y que varios usuarios modifiquen los mismos datos de forma simultánea o concurrente.

A continuación se presenta una tabla con la información referente a los bloqueos usados en la administración de bases de datos.¹⁸

Información relativa a los bloqueos usada en la administración de las bases de datos

Categoría	Tipo	Descripción
Tipo de Bloqueo	RID	Identificador de fila. Utilizado para bloquear una sola fila de una tabla.
	KEY	Bloqueo de una fila en un índice. Se utiliza para proteger intervalos de claves.

¹⁸ STANEK, R. William, *Microsoft SQL Server 2000 Manual del administrador*, MC GRAW HILL INTERAMERICANA, España, 2001, pp. 130-131

Información relativa a los bloqueos usada en la administración de las bases de datos

Categoría	Tipo	Descripción	(continuación)
	PAG	Un bloque en una página de datos o en una página de índice.	
	EXT	Grupo contiguo de ocho páginas de datos de páginas índice.	
	TAB	Un bloqueo sobre una tabla completa, incluyendo todos los datos e índices.	
	DB	Un bloque sobre una base de datos.	
Modo de bloqueo	S	Compartido. Se usa para operaciones de sólo lectura, tal como una instrucción SELECT.	
	U	Actualizar. Se utiliza para leer o bloquear un recurso actualizable. Evita algunas situaciones de interbloqueo.	
	X	Exclusivo. Sólo permite que una sesión actualice los datos. Utilizado para operaciones de modificación de datos, como INSERT, DELETE y UPDATE.	
	Intención Intent	Se usa para establecer una jerarquía de bloqueos.	

Información relativa a los bloqueos usada en la administración de las bases de datos

Categoría	Tipo	Descripción	(continuación)
	SchS	Estabilidad del esquema (Schema Stability). Se usa al comprobar el esquema de una tabla.	
	BU	Actualización masiva (bulk update). Se usa al realizar copias masivas de datos.	
Estado	GRANT	El bloque ha sido concedido.	
	WAIT	El bloqueo está siendo bloqueado por otro proceso.	
	CNVT	El bloqueo está siendo convertido, es decir, se mantiene en un modo pero de espera de adquirir un modo de bloqueo más fuerte.	
Propietario	curs	El propietario del bloqueo es un cursor.	
	sess	El propietario del bloqueo es una sesión de usuario.	
	xact	El propietario del bloqueo es una transacción.	
Índice	Nombre del Índice	El índice asociado con el recurso. Si el índice está agrupado, en su lugar verá el nombre de la tabla.	

Información relativa a los bloqueos usada en la administración de las bases de datos

Categoría	Tipo	Descripción	(continuación)
Recurso	RID	Identificador de la fila bloqueada en la tabla. La fila se identifica mediante <IdArchivo>:<página>:<rid>, donde rid es el identificador de fila de la página especificada.	
	CLAVE KEY	Una clave que presenta como un número hexadecimal, y SQL lo utiliza internamente.	
	PAG	Un número de página, que se identifica mediante <idArchivo>:<página>, donde IdArciovo es el identificador del archivo en la tabla sysfiles y página es el número lógico de la página dentro de dicho archivo.	
	EXT	El primer numero de la página en la extensión que está bloqueando. La página se identifica mediante <IdArchivo>:<página>.	
	TAB	El identificador de tabla.	
	DB	El identificador de base de datos.	

Los problemas más comunes por el uso de bloqueos son:

- ∇ Las conexiones bloqueantes. Se producen cuando una conexión mantiene el bloqueo y una segunda conexión establece un tipo de bloqueo que entra en conflicto. Esto fuerza a la segunda conexión a que espere o que bloquee a la primera.
- ∇ Los Interbloqueos. Se producen cuando dos bloqueos aplican bloqueos sobre objetos distintos y cada uno de ellos desea bloquear el objeto del otro. Cada uno de los usuarios espera que el otro libere el bloqueo y eso no ocurre.

2.2.8 Vistas

Una vista se considera como una tabla virtual o una consulta almacenada. Los datos accesibles a través de la vista no están almacenados en ella, sin que continúen estando almacenados en la tabla a la que pertenecen. De hecho, lo único que normalmente se almacena de la vista en la base de datos es una instrucción SELECT.

Algunas funciones de las vistas son:

- ✕ Restringir el acceso al usuario a filas y columnas específicas de una tabla.
- ✕ Combinar columnas de varias tablas de tal manera que parezcan una.
- ✕ Agregar información en lugar de presentar los detalles. Por ejemplo, presentar la suma de una columna o su multiplicación.

Las vistas se crean con el comando CREATE VIEW seguido del nombre de la vista y de una instrucción SELECT. Las vistas pueden ser *actualizables* (aceptar modificar los datos con instrucciones INSERT, UPDATE y DELETE) mientras la modificación sólo afecte a una tabla base.

Las vistas indexadas mejoran considerablemente el rendimiento de las vistas complejas. Las consultas complejas pueden hacer referencia a un gran número de filas de tablas base y agregar mucha información en agregados concisos. Cuando se ejecuta una instrucción CREATE INDEX, el conjunto de resultados de la vista SELECT se almacena en la base de datos de forma permanente. De esta manera, cuando se haga referencia a la vista, tendrán tiempo de respuesta considerablemente mejor.

2.2.9 Seguridad

El acceso a las bases de datos de SQL Server se controla mediante los siguientes componentes.

- » Modos de autenticación de SQL Server
- » Inicios de sesión de servidor
- » Permisos
- » Funciones.

MODOS DE AUTENTICACIÓN DE SQL SERVER

SQL Server dispone de dos modos de conexión:

- ≈ **Autenticación de Windows exclusivamente.** Se utilizan las cuentas de usuario y de grupo de usuario del dominio de Windows para acceder a las bases de datos sin que el administrador les proporcione un identificador de inicio de sesión. Esta forma de autenticación es recomendable para usuarios que acceden a la base de datos desde dentro de la organización.

- ≈ **Seguridad mixta.** Se utilizan tanto los inicios de sesión de Windows como los inicios de sesión de SQL Server. Los inicios de sesión de SQL Server son recomendables cuando se quiere acceder a la base de datos desde fuera de la organización, por ejemplo: Internet.

INICIOS DE SESION DE SERVIDOR

El acceso a SQL Server se configura utilizando inicios de sesión de servidor o las funciones a las que dichos inicios de sesión pertenecen, o ambas cosas.

Hay varios inicios de sesión configurados de manera predeterminada, como son:

- ♣ El grupo local de administradores (Administrators). Entre los miembros de este grupo se incluyen la cuenta del Administrador (administrador) y cualquier otro usuario que necesite administrar el sistema localmente. A este grupo se le concede por default la función del servidor System Administrators (sysadmin).
- ♣ La cuenta local administrador (Administrador). Proporciona privilegios de administración sobre el sistema local. Se le concede de manera predeterminada la función System Administrators.
- ♣ El inicio de sesión sa. Es la cuenta del administrador para el sistema SQL Server. Se le concede la función System Administrators.
- ♣ El inicio de sesión guest. Es un inicio de sesión que se puede añadir a una base de datos (pues originalmente no existe en ellas) para permitir que cualquiera disponga de un inicio de sesión válido. Quienes acceden a la base de datos por medio de esta cuenta asumen la identidad de usuario invitado y heredan los privilegios y permisos de la cuenta guest.

- ♣ El usuario dbo. Es un tipo especial de usuario de base de datos al que se le conceden privilegios especiales. El propietario de una base de datos es quien crea esa base de datos. A este propietario se le conceden implícitamente los permisos sobre la base de datos y puede conceder dichos permisos a otros usuarios.

PERMISOS

Los permisos determinan las acciones que los usuarios pueden realizar. Estos permisos deben de ser apropiados y definidos antes de poder realizar ninguna acción.

En SQL Server se utilizan tres permisos:

- ✓ *Permisos de Objeto.* Controlan el acceso a tablas, listas, columnas y procedimientos almacenados.
- ✓ *Permisos sobre instrucciones.* Controlan acciones de administración, tal como crear una base de datos o añadir objetos a una base de datos. Sólo a los miembros de función sysadmin y a los propietarios de las bases de datos pueden asignar permisos sobre instrucciones.
- ✓ *Permisos implícitos.* Sólo los miembros de las funciones del sistema predefinidas y los propietarios de las bases de datos u objetos pueden tener permisos implícitos. Estos permisos los autorizan para llevar a cabo las actividades con la base de datos u objeto que posean. Los permisos implícitos no pueden ser modificados.

FUNCIONES

Permiten asignar fácilmente permisos a un grupo de usuarios y también pueden tener permisos predefinidos.

Hay disponibles dos tipos de funciones:

- **Funciones del servidor.** Se utilizan para conceder capacidades de administración del servidor. Puede llevar a cabo cualquier tarea que la función permita.
- **Funciones de base de datos.** Estas se configuran por separado para cada base de datos ya que cada base de datos tiene su propio conjunto de funciones. SQL Server admite tres tipos de funciones de bases de datos:
 - ▲ Funciones estándar definidas por el usuario. Permiten crear funciones con privilegios y permisos exclusivos. Pueden utilizarse para agrupar a los usuarios y luego asignarles un único permiso la función en lugar de tener permisos por separado.
 - ▲ Funciones de aplicación definidas por el usuario. Permite crear funciones protegidas mediante contraseña para aplicaciones específicas.
 - ▲ Funciones de la base de datos predefinidas. Están ya integradas al sistema y disponen de permisos que no pueden ser modificados. Se emplean para asignar privilegios de administración de bases de datos.

COPIAS DE SEGURIDAD

SQL Server proporciona varias formas para efectuar **copias de seguridad**:¹⁹

Copias de seguridad de bases de datos completa. Realiza una copia de la base de datos completa. SQL Server copia todo lo que contiene la base de datos

¹⁹ STANEK, R. William, *Microsoft SQL Server 2000 Manual del administrador*, MC GRAW HILL INTERAMERICANA, España, 2001, pp. 478-480

incluyendo las porciones necesarias del registro de transacciones a medida que se realiza la copia de seguridad.

Copia de seguridad diferenciales. Realiza una copia de seguridad de sólo aquellos datos que hayan cambiado desde la última copia de seguridad completa.

Copia de seguridad del registro de transacciones. Almacena los cambios que hayan tenido lugar desde la última copia de seguridad del registro de transacciones, y luego trunca el registro, es decir, borra todas las transacciones que hayan sido confirmadas o anuladas.

Copia de seguridad de archivos y grupos de archivos. Permiten realizar copias de seguridad de archivos de la base de datos en vez de la base de datos completa. Cuando se usa este tipo de copia de seguridad, es necesario realizar también una copia de seguridad del registro de transacciones. Además, si hay objetos en la base de datos que abarquen múltiples archivos o grupo de archivos, será necesario realizar una copia de seguridad de todos los archivos y grupos de archivos relacionados al mismo tiempo.

Con Microsoft SQL Server, las Net-Libraries (librerías de red) admiten conexiones de red entre los clientes y el servidor mediante el uso del protocolo Secuencia de datos tabulares (TDS). Éstas permiten conexiones simultáneas entre clientes que utilizan Canalizaciones con nombre, sockets TCP/IP u otros mecanismos de comunicación entre procesos (IPC).

2.2.10 Replicación y comunicaciones remotas

COMUNICACIONES REMOTAS

Para poder establecer una comunicación remota SQL Server se hace valer de servidores vinculados configurados con OLE DB. Con OLE DB se pueden vincular

instancias de SQL Server a otras instancias de SQL Server, así como a otras fuentes de datos.

Los servidores vinculados se usan para manejar consultas y transacciones distribuidas, llamadas remotas a procedimientos almacenados y para duplicación.

CONSULTAS Y TRANSACCIONES DISTRIBUIDAS

Cuando se ejecuta una *consulta distribuida*, SQL Server interpreta el comando y lo divide, para el proveedor OLE DB de destino, utilizando peticiones de conjunto de filas. El proveedor OLE DB llama al origen de datos, abre los archivos necesarios y devuelve la información solicitada como conjuntos de filas. Después, SQL Server da formato a los conjuntos de filas como conjuntos de resultados.

Las Transacciones distribuidas son las que usan consultas distribuidas o llamadas a procedimientos remotos (Remote Procedure Calls, RPC). Si comienza una aplicación que actualice bases de datos en tres instancias de servidor diferentes, hay que asegurarse de que la transacción se confirma cuando se completa correctamente, o que la transacción se deshace si ocurre un error. De esta manera se asegura la integridad de las bases de datos involucradas en la transacción distribuida.

Existen tres componentes para que las transacciones distribuidas funcionen adecuadamente:

- a) Administración de recursos. Se debe configurar administradores de recursos, que son los servidores vinculados que se usan en las transacciones distribuidas.
- b) Servicio Coordinador de Transacciones distribuidas (Distributed Transaction coordinator). Este servicio debe de estar ejecutándose en todos los

servidores que vayan a manejar transacciones distribuidas. Ya que si no es así, las transacciones no funcionarán adecuadamente.

- c) Administrador de transacciones. Es quien coordina y administra las transacciones distribuidas. El administrador de transacciones de SQL Server es el coordinador de las transacciones distribuidas.

DUPLICACIÓN O REPLICACION

Se utiliza la duplicación para hacer la copia de una base de datos de un servidor y distribuirla a otros servidores. También se puede usar para copiar datos, transformarlos y luego distribuir los datos personalizados a múltiples servidores.

Razones para emplear la replicación:

- ♦ Sincronizar los cambios en bases de datos remotas con la base de datos central.
- ♦ Crear múltiples instancias de trabajo para distribuir la carga de trabajo.
- ♦ Mover conjuntos de bases de datos desde un servidor central y distribuirlo a otros servidores.
- ♦ Personalizar los datos y distribuirlos a múltiples suscriptores.

2.3 ACCESS 2002

2.3.1 Tipos de datos

Access maneja diez tipos de datos los cuales se mencionan a continuación;

Tabla de tipos de datos de Access

Tipo de dato	Descripción
Texto	Contiene cualquier combinación de hasta 255 caracteres y números.
Memo	Se utiliza para almacenar un tipo largo, pero de longitud variable. El tamaño del texto podría variar de unas cuantas palabras hasta 65,535 caracteres.
Número	Se utiliza para definir valores u ordenar los datos o hacer cálculos.
Moneda	Se utiliza para valores monetarios. Pueden utilizarse en cálculos aritméticos.
Autonumérico	Access genera un valor único de tipo entero para cada registro.
Fecha/Hora	Se utiliza para valores de fecha y horas con una variedad de formas para desplegarlo, así como hacer cálculos para determinar un periodo cronológico.
Si/No	Es útil cuando se quiere una opción equivalente a una marca de verificación. Es equivalente a decir Verdadero/Falso.
Objeto OLE	Se podrá obtener datos desde varios tipos de objetos como una hoja de cálculo de Excel, un documento en Word, un gráfico, un sonido.
Hipervínculo	Contiene cuatro partes: el texto que se mostrará, la dirección destino, una dirección secundaria en caso de que sea necesaria y una información de pantalla.

Tabla de tipos de datos de Access

(continuación)

Tipo de dato	Descripción
Asistente para Búsquedas	Crea un campo que está validado para una lista de valores válidos.

2.3.2 Integridad

Para Access la integridad referencial es un conjunto de reglas que mantiene completa la base de datos, evitando la redundancia de datos. Para poder realizar esto utiliza las relaciones (uno a uno y uno a muchos). Cuando se crea en la *ventana de relaciones* de Access una relación entre una clave principal y la clave foránea, se puede habilitar la opción de Exigir integridad referencial. Al habilitarse ofrece dos opciones para mantener la integridad:

- Actualizar en cascada los campos relacionados.** Permite que al cambiar un valor en el campo de la clave principal de la tabla principal, se actualice automáticamente el valor en los campos de la clave foránea de la tabla secundaria con el fin de que éstos coincidan.
- Eliminar en cascada los registros relacionados.** Cuando se elimina un registro principal, Access elimina de forma automática los registros secundarios relacionados.

Para poder crear la relación entre dos campos (clave primaria y foránea) éstos deben de ser del mismo tipo de datos y contener el mismo tipo de información. A menos que la clave principal sea de tipo autonumérico, ya que en este caso se puede relacionar con claves foráneas de tipo numérico.

2.3.3 Transacciones, Procedimientos almacenados y Triggers

Access no maneja transacciones debido a que su capacidad de almacenamiento no es muy grande y, por lo tanto, no puede soportar bases de datos grandes.

Las transacciones comúnmente se utilizan en bases de datos grandes, donde la cantidad de registros que se tienen que modificar son tantos que el proceso de actualización de la base de datos puede tardar mucho tiempo, además de que son muy utilizadas en bases de datos distribuidas.

Como en Access las bases de datos no son tan robustas, las operaciones que se pueden llevar a cabo con una transacción se realizan individualmente, cada vez que lo requiera el usuario, por lo que es necesario tener cuidado al modificar la base de datos para que conserve su coherencia e integridad.

De la misma manera, y en consecuencia, Access no trabaja **Procedimientos Almacenados** ni **Triggers**. Pues recordemos que un procedimiento almacenado puede servir para ejecutar transacciones y que los Triggers son un tipo de procedimiento almacenado.

En vez de esto Access utiliza **macros**.

Una **macro** es un conjunto de una o más acciones²⁰ que realiza una operación determinada, como abrir un formulario o imprimir un informe. Las macros pueden ayudar a automatizar las tareas comunes.²¹

Las **macros** pueden estar compuestas por una secuencia de acciones o comandos por un grupo de macros. Pueden utilizar condiciones dependiendo de en qué casos se desea que se realice una acción cuando se ejecute la macro.

²⁰ Instrucción independiente que se puede combinar con otras para automatizar tareas.

²¹ Véase macros en Ayuda de Microsoft Access.

Sí se tienen muchas macros, y reunimos las que estén relacionadas, estamos formando grupos de macros que pueden simplificar las bases de datos.

Las condiciones son utilizadas, solamente si se quiere que se lleve a cabo una acción o una serie de acciones. La macro sigue trayectorias diferentes dependiendo si la condición usada es verdadera o falsa.

Access evalúa la primera expresión condicional, si es verdadera se lleva a cabo la acción o acciones controladas por esa condición, si es falsa se omite la acción o acciones.

Después de agregar una acción a una macro se establecen los argumentos. Estos argumentos dan a Microsoft Access información adicional sobre cómo llevar a cabo la acción. Si se desea modificar los argumentos que establece Access por default, una forma es usar una expresión precedida por el signo igual.

Los macros controlan tareas simples. Permiten vincular fácil y rápidamente los objetos de la base de datos (Tablas, formularios, informes etc.) creados. También se pueden utilizar para realizar asignaciones globales de teclas o ejecutar una acción o serie de acciones cuando se abre por primera vez una base de datos.

2.3.4 Almacenamiento

Access almacena la información por medio de archivos. Un archivo de Microsoft Access es un archivo de base de datos o de proyecto de Access.

Una base de datos de Access almacena objetos de base de datos y datos en un archivo .mdb.

Una Base de datos de Access contiene **objetos** como tablas, consultas, formularios, informes, páginas macros y módulos. Un proyecto de Access contiene objetos como formularios, informes, páginas, macros y módulos.

Una **tabla** en Access puede contener registros de un máximo de 2,048 bytes y hasta 255 campos.

2.3.5 Concurrencias y tipos de bloqueos

Access utiliza el bloqueo por registros para garantizar la integridad de la base de datos cuando dos usuarios desean modificar el mismo registro y al mismo tiempo.

El bloqueo por registro consiste en que sólo un usuario tiene acceso exclusivo en forma temporal a un registro. Cuando un objeto de la base de datos queda bloqueado, será de solo lectura para todos los usuarios, excepto para aquel que está introduciendo o modificando los datos en el registro.

Access ofrece tres niveles de bloqueo de registros:

- ⊗ Sin bloquear. Con esta opción se garantiza que todos los registros se podrán actualizar en cualquier momento, pero se podría crear conflictos si más de un usuario lo hace al mismo tiempo.
- ⊗ Todos los registros. En este nivel se bloquea todos los registros en un formulario o en una hoja de datos y en las tablas subyacentes hasta que el formulario o la hoja de datos se cierre.
- ⊗ Registro modificado. En cuanto un usuario comience a modificar un registro, ningún otro podrá hacerle cambios hasta que el primer usuario termine de modificar los datos y guarde los cambios.

2.3.6 Vistas

Access no trabaja las vistas como en Oracle o SQL Server, sin embargo la utilización de formularios, consultas y páginas de acceso a datos, proporcionan beneficios similares a las vistas.

Los formularios pueden mostrarnos datos de una o varias tablas o consultas, que contengan todos o varios campos. De esta manera se puede mostrar al usuario los campos que se desea que él vea y de las tablas que sean necesarias, desafortunadamente si se quiere insertar nuevos datos sólo se podrá hacer si cubre todos los requerimientos de integridad necesarios, de lo contrario sólo se verá como sólo lectura.

En Access, al trabajar consultas, se pueden hacer actualizaciones, borrados e inserciones de registros en tablas, incluso crear tablas, dependiendo del diseño de su creación. También por medio de las consultas podemos ver las columnas de determinadas tablas que están trabajando de forma distribuida en la red.

Algo similar ocurre con las páginas de acceso a datos, sólo que éste nos muestra sus resultados como una página Web.

Como podemos ver, aunque Access no utiliza vistas, podemos utilizar los recursos antes mencionados para utilizarlos simulando este servicio, en caso de que sea necesario.

2.3.7 Seguridad

Access utiliza varias formas de proteger la base de datos:

- ⌘ Proteger la base de datos con una contraseña
- ⌘ Grupos de trabajos

⌘ Cifrar y descifrar una base de datos

⌘ Proteger las páginas de acceso a datos

PROTEGER LA BASE DE DATOS CON UNA CONTRASEÑA

Access ofrece 2 niveles de *seguridad* para las bases de datos:²²

1. Protección global por medio de contraseñas para la base de datos.
2. Seguridad a nivel usuario basada en el modelo de seguridad de Access.

El primer caso es más apropiado para bases de datos de un solo usuario. Este método sólo se aplica al abrir la base de datos, una vez abierta la información estará disponible para el usuario.

El segundo caso se aplica para múltiples usuarios. Se basa en el concepto de grupos de trabajo que consiste en usuarios que comparten información, los usuarios aparecen listados en cuentas de usuarios y de grupos. Cada grupo de trabajo cuenta con un grado de libertad dentro de la base de datos.

El modelo de seguridad de Access se compone de cuatro elementos:

- * Usuario. Es la persona que utiliza la base de datos. Si se está controlando la seguridad en la base de datos, debe introducir su nombre y contraseña.
- * Grupo. Conjunto de usuarios que tiene el mismo nivel de seguridad y que además accedan a las mismas áreas de la base de datos.
- * Permisos. Se le otorga a un usuario o a un grupo el derecho de ejecutar una acción específica sobre un objeto de la base de datos.
- * Objeto. Se refiere a cualquiera de las tablas, formularios, informes, macros o módulos de Access, así como la misma base de datos.

²² ANDERSEN, Virginia, *Access 2002 a su alcance*, Mc Graw Hill, México, 2002, pp. 606-608.

- * Propietario. Todos los objetos de la base de datos le pertenecen a un usuario o propietario. Él es responsable del objeto y tiene la autoridad de la utilización del objeto por él y por otros usuarios si así lo desea.

GRUPOS DE TRABAJOS

Grupo de Administradores. Todos los miembros son administradores de la base de datos y cuentan con todos los permisos sobre los objetos de la base de datos.

Grupo de Usuarios. Incluye las cuentas individuales de usuarios de la base de datos. Tienen permisos sobre todos los objetos de la base de datos de nueva creación de la base de datos.

Usuario Integrado. Access incluye un usuario integrado que pertenece a los grupos anteriores llamado **Administrador**. Cuenta con permisos para abrir, ver y modificar todos los datos.

Con la cuenta de Administrador se pueden crear nuevos grupos de trabajo en donde se pueden restringir varios tipos de permisos, según lo que vayan a trabajar los usuarios que pertenezcan a ese grupo.

CIFRAR Y DESCIFRAR UNA BASE DE DATOS

El cifrado es una forma menos rigurosa de proteger la base de datos. Cuando se cifre la base de datos será compactada y se volverá completamente ilegible para cualquier procesador de palabras o cualquier otro programa. El cifrado no restringe el acceso a los objetos de datos.

El descifrado de la base de datos invierte el proceso del cifrado y lo restaura a su forma original.

PROTEGER LAS PAGINAS DE ACCESO A DATOS

Una página de acceso de datos consta de dos partes: el acceso directo que se almacena en la base de datos y el archivo HTML, que se almacena por separado. Para proteger el acceso directo se recomienda hacer que la base de datos sea sólo de lectura. Para proteger el archivo HTML se utiliza el sistema de seguridad de Windows y que haga el archivo también de sólo lectura.

2.3.8 Replicación y comunicaciones remotas

La *replicación* permite hacer copias de una base de datos para emplearla en diferentes ubicaciones pero permitiendo que todas las copias estén totalmente sincronizadas.

A la base de datos en la que se implementan todos los cambios se conoce como *diseño principal* y a las copias se les conoce como *réplicas*.

En Access no es bueno utilizar la replicación si se va a manejar una gran cantidad de introducción y modificación de datos en las réplicas, ya que puede haber conflictos cuando dos o más usuarios actualicen los mismos datos. Tampoco es bueno utilizar la replicación si los datos se van a manejar en tiempo real y tienen que estar actualizados en todo momento.

Existen otras formas de compartir una base de datos de forma remota además de la creación de réplicas. Algunas de estas formas son:²³

- ✎ Colocar la base de datos en una ubicación central en donde todos los usuarios tengan acceso a todos los objetos dentro de ella.

²³ Ibidem. pp. 597.

- ✎ Dividir la base de datos de tal forma que los usuarios sólo compartan los datos de la tabla.
- ✎ Colocar la base de datos completa o una parte en Internet.
- ✎ Crear una aplicación basada en el modelo cliente-servidor.

COMPARTIR UNA BASE DE DATOS COMPLETA. Consiste en colocar la base de datos completa en un servidor de la red o en una carpeta que se pueda compartir. Para que los usuarios puedan acceder a la base de datos es necesario que en cada computadora de la red esté instalado Access.

DIVIDIR LA BASE DE DATOS. Es un método más rápido que consiste en colocar las tablas en el servidor de la red y permitir que los usuarios guarden los objetos en sus propias estaciones de trabajo. Esta estrategia es útil cuando los usuarios realizan actividades diferentes.

COLOCAR LA BASE DE DATOS EN INTERNET. Consiste en crear páginas (Web) de acceso a datos por medio de las cuales un usuario de la base de datos se podrá conectar y utilizar dicha base de datos.

CREAR UNA APLICACIÓN BASADA EN EL MODELO CLIENTE –SERVIDOR. Se puede simular el modelo cliente servidor por medio de programación. Para poder desarrollar este tipo de aplicaciones se puede utilizar el lenguaje de programación Visual Basic.

2.4 MYSQL

2.4.1 Tipos de datos

MySQL, como muchos otros manejadores de bases de datos trabaja con los tipos de datos que se enumeran en las siguientes tablas: Tipos numéricos y Tipos de datos de cadena.

Tipos numéricos

Nombre	Espacio en memoria	Rango de valores	Sin signo
TINYINT	1 byte	-128 a 127	0-125
SMALL INT	2 bytes	-32768 a 32767	0-65535
MEDIUMINT	3 bytes	-8388608 a 8388607	0-16777215
INT	4 bytes	-2147483648 a 2147483647	0-4294967295
BIGINT	8 bytes	-9223372036854775808 a 9223372036854775807	0-18,446,744, 073,709,550,615
FLOAT (M, D)	4 bytes	Varía según los valores	
DOUBLE (M, D)	8 bytes	Varía según los valores	
DECIMAL (M, D)	M+2 bytes	Varía según los valores	

Los tipos FLOAT, DOUBLE y DECIMAL pueden manejar fracciones, así como definir el número de dígitos a la derecha del punto decimal.

MySQL utiliza modificadores que sólo pueden utilizarse con datos de tipo numérico. Estos modificadores son: AUTO_INCREMENT, UNSIGNED y ZEROFILL. UNSIGNED hace columnas positivas, AUTO_INCREMENT incrementa en 1 el valor de la columna y no reutiliza el valor máximo si éste fue borrado; de la misma manera, si se desea insertar un número mayor al valor máximo actual continuará incrementando a partir del valor mayor insertado. ZEROFILL despliega de ceros a la izquierda de un número de acuerdo con el ancho de columna.

Tipos de datos de Cadena

Nombre	Tamaño máximo	Espacio en disco
CHAR(X)	255 bytes	X bytes
VARCHAR(X)	255 bytes	X+1 byte
TINYTEXT	255 bytes	X+1 byte
TINYBLOB	255 bytes	X+2 bytes
TEXT	65535 bytes	X+2 bytes
BLOB	65535 bytes	X+2 bytes
MEDIUMTEXT	1.6 MB	X+3 bytes
MEDIUMBLOB	1.6 MB	X+3 bytes
LONGTEXT	4.2 GB	X+4 bytes
LOB	4.2 GB	X+4 bytes

TIPOS MISCELANEOS

MySQL trabaja tres tipos misceláneos: ENUM, SET y DATE/TIME.

Tipo ENUM. Este tipo de datos es una lista enumerada, es decir, que la columna con este tipo de datos sólo podrá almacenar uno de los valores declarados en la lista dada. La lista puede estar conformada hasta con 65,535 elementos.

Tipo SET. Es similar al tipo ENUM, la diferencia es que en SET se puede escoger más de una opción para almacenar. Puede contener hasta 64 elementos.

Tipo DATA/TIME. Este tipo de datos se utiliza para manejar información de fecha y hora. En la siguiente tabla, se muestran los formatos que utilizan los tipos de datos que pertenecen al tipo misceláneo DATA/TIME.

Tipos de datos de fecha y hora

Tipo de datos	Formato estándar
DATETIME	AAAA-MM-DD HH:MM:SS
DATE	AAAA-MM-DD
TIME	HH:MM:SS
YEAR	AAAA
TIMESTAMP(14)	AAAAMMDDHHMMSS
TIMESTAMP(12)	AAMMDDHHMMSS
TIMESTAMP(10)	AAMMDDHHMM
TIMESTAMP(8)	AAAAMMDD
TIMESTAMP(6)	AAMMDD

Tipos de datos de fecha y hora

(continuación)

Tipo de datos	Formato estándar
TIMESTAMP(4)	AAMM
TIMESTAMP(2)	AA

2.4.2 Integridad

Para asegurar la integridad de los datos muchas bases de datos utilizan restricciones como lo son: CHECK, FOREIGN KEY y UNIQUE, entre otras.

Las restricciones presentan beneficios como no romper las reglas de una relación entre tablas y conservar la integridad de los datos. El mayor problema de las restricciones es que agregan mucha carga sobretodo al insertar y actualizar los registros, lo que provoca que el sistema tenga que disminuir su velocidad.

Esa es la razón principal por lo que MySQL no soporta restricciones, ya que tienden hacer más lento el procesamiento. El control de la integridad de datos queda entonces, en manos del desarrollador y el DBA (Administrador de la base de datos).

2.4.3 Transacciones

Recordemos que una transacción es la ejecución de uno o varios INSERT, UPDATE o DELETE de forma continua e individual pero en un solo bloque, que se confirman con un COMMIT, o bien se deshace cualquier cambio con un ROLLBACK.

El problema más claro que presenta MySQL en lo que se refiere a las transacciones, es la realización de un ROLLBACK, por lo que la única manera de enfrentar esa limitación es utilizar bloqueos.

Aunque la última versión estable de MySQL (3.23.32) ya soporta transacciones, estas son muy limitadas y simples y por lo tanto deficientes. Por lo que aún no es recomendable su uso para editar una gran cantidad de registros.

2.4.4 Procedimientos almacenados y triggers

Un procedimiento almacenado, como ya lo hemos visto, es una serie de comandos que ya han sido compilados y almacenados dentro de una base de datos.

Los disparadores o Triggers son procedimientos almacenados que se ejecutan cuando se realiza una acción específica en una tabla. Las acciones que se realizan normalmente son instrucciones INSERT, UPDATE y DELETE.

Los procedimientos almacenados tienen muchas ventajas, entre ellas se encuentran: se ajustan a la programación modular, son reutilizables, pueden tomar o transferir parámetros, creando una especie de programación por lotes. También se ejecutan con mayor rapidez que las instrucciones normales. Los Triggers presentan ventajas como la integridad de la base de datos al mantener las relaciones entre las tablas.

Sin embargo, como para MySQL lo primordial es la velocidad, no utiliza ninguna de estas dos herramientas ya que esto significaría hacer más lento el sistema.

2.4.5 Concurrencias y tipos de bloqueos

La concurrencia se da cuando una base de datos está tan ocupada que los accesos se están haciendo simultáneamente. Mientras algunos subprocesos²⁴ están tratando de leer datos, otros necesitan leerlos, calcularlos y escribirlos.

El bloquear permite tener acceso exclusivo a cierto número de tablas, asegurándose que se pueden hacer varios movimientos sin tener interferencia con otros subprocesos.

UTILIZACIÓN DE BLOQUEOS

Al comando `LOCK TABLES` se le puede asignar una lista de tablas, algunas para bloquearse para lectura (`READ`) y otras para escritura (`WRITE`).

Si un subproceso hace un bloqueo de lectura (`READ`) en algunas tablas, este subproceso y los demás sólo podrán leer esas tablas.

Si un subproceso hace un bloqueo de escritura (`WRITE`), será el único que podrá acceder a esas tablas. Podrá leer y escribir en ellas sin que ningún otro subproceso pueda acceder a las tablas hasta que el subproceso que las bloqueo las libere.

Cuando un subproceso solicita un bloqueo, tiene que esperar en la cola hasta que los bloqueos de las tablas sean liberados. Existe una cola de espera para `WRITE` y otra para `READ`.

En la solicitud de bloqueos `WRITE`, se otorga al subproceso que lo solicite si no hay cola de espera; de lo contrario, la solicitud se coloca en la cola de espera.

En la solicitud de bloqueos `READ` se otorga al subproceso que lo solicite si no hay cola de espera y si la tabla no tiene bloqueos `WRITE`, de lo contrario se coloca en la cola de espera.

²⁴ Es un proceso limitado en tiempo de ejecución. Nace cada vez que se hace una consulta y muere cuando ésta termina.

Cuando un bloqueo es liberado, los subprocesos de la cola de bloqueos WRITE tienen prioridad sobre los de las colas READ, ya que de esta manera se asegura que cualquier actualización de la base de datos se procesará de lo más rápido posible.

Si es más urgente conceder bloqueos READ que WRITE se puede utilizar la instrucción `LOW_PRIORITY WRITE`, de esta manera el manejo de las colas WRITE y READ se comportará de forma inversa a la explicada anteriormente.

Otra forma de influir en la política de listas de espera es utilizar la instrucción `SELECT HIGH_PRIORITY`, la cual es conveniente utilizar sólo para consultas con `SELECT` que puedan completarse rápidamente. Esta instrucción permite al `SELECT` leer la tabla, incluso si hay un bloqueo WRITE en la cola.

Para desbloquear las tablas se utiliza la instrucción `UNLOCK TABLES`. Las tablas también serán desbloqueadas si el mismo subproceso envía otro comando `LOCK`, o si está cerrada la conexión al servidor.

2.4.6 Vistas

MySQL no maneja vistas, por lo tanto no ofrece la opción de limitar las filas y/o columnas a las que puede acceder un usuario. Y se corre el riesgo de que el usuario vea información que no se deseaba que él conociera. Si un usuario desea acceder a los registros de una tabla para verlos o eliminarlos o modificarlos, solamente lo podrá hacer directamente en dicha tabla.

2.4.7 Seguridad

Todo el sistema de permisos MySQL lo guarda en una base de datos llamada **mysql**.

La base de datos de MySQL, **mysql**, se compone de cinco tablas denominadas tablas de acceso: **host**, **user**, **db**, **tables_priv**, **columns_priv**.

La tabla **user** contiene información sobre los usuarios, desde qué máquinas pueden acceder al servidor MySQL, su clave y de sus diferentes permisos. La tabla **host** nos informa sobre qué máquinas podrán acceder a nuestro sistema, así como a las bases de datos que tendrán acceso y sus diferentes permisos. Finalmente, las tablas **db**, **tables_priv**, **columns_priv** nos proveen de un control individual de las bases de datos, tablas y columnas (campos).²⁵

Tabla **user**

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
User	char(16)	
Password	char(16)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
Reload_priv	enum('N','Y')	N
Shutdown_priv	enum('N','Y')	N
Process_priv	enum('N','Y')	N
File_priv	enum('N','Y')	N
Grant_priv	enum('N','Y')	N

²⁵ Ver <http://otri.us.es/recursosPHP/manual/mysql/mysql.htm>

References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

Tabla **host**

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
Db	char(32)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N
Drop_priv	enum('N','Y')	N
Grant_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

Tabla **db**

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
Db	char(32)	
User	char(16)	
Select_priv	enum('N','Y')	N
Insert_priv	enum('N','Y')	N
Update_priv	enum('N','Y')	N
Delete_priv	enum('N','Y')	N
Create_priv	enum('N','Y')	N

Drop_priv	enum('N','Y')	N
References_priv	enum('N','Y')	N
Index_priv	enum('N','Y')	N
Alter_priv	enum('N','Y')	N

Tabla **columns_priv** y **tables_priv**

Estas dos tablas comparten las siguientes columnas:

CAMPO	TIPO	POR DEFECTO
Host	char(60)	
Db	char(64)	
User	char(16)	
Table_name	char (64)	
Column_name	char (64)	
Column_priv	Set('Selct','Insert','Update', 'References')	
Timestamp	Timestamp(14)	

Tabla **tables_priv** no contiene la columna **Column_name** pero si contiene otras dos columnas:

CAMPO	TIPO	POR DEFECTO
Grantor	char(77)	
Table_priv	Set('Selct','Insert','Update', 'Delete','Create','Drop','Grant', 'References', 'Index','alter')	

La columna **Grantor** contiene el nombre de la persona que concede los permisos.

Cada columna de estas tablas muestra los permisos que tiene una persona. En la asignación de permisos, una **Y** significa que puede realizar la operación y una **N** significa lo contrario.

A continuación se enumera la descripción de los diferentes permisos:²⁶

Select_priv: Permite utilizar la sentencia SELECT.

Insert_priv: Permite utilizar la sentencia INSERT.

Update_priv: Permite utilizar la sentencia UPDATE.

Delete_priv: Permite utilizar la sentencia DELETE.

Create_priv: Permite utilizar la sentencia CREATE o crear bases de datos.

Drop_priv: Permite utilizar la sentencia DROP o eliminar bases de datos.

Reload_priv: Permite recargar el sistema mediante mysqladmin reload.

Shutdown_priv: Permite parar el servidor mediante mysqladmin => Permite parar el servidor mediante mysqladmin shutdown.

Process_priv: Permite manejar procesos del servidor.

File_priv: Permite leer y escribir ficheros usando comando como SELECT INTO OUTFILE y LOAD DATA INFILE.

Grant_priv: Permite otorgar permisos a otros usuarios.

Index_priv: Permite crear o borrar índices.

Alter_priv: Permite utilizar la sentencia ALTER TABLE.

²⁶ ver <http://webmaster.bankhacker.com/ayuda-linux/consulta/Preguntas+acerca+de+Mysql/>

ETAPAS DE CONTROL

Cuando un usuario hace una conexión a la base de datos, MySQL realiza dos medidas de control:

- **Verificación de la conexión.** Ocurre en que un usuario intenta conectarse a MySQL. Para realizar la conexión se requiere de un nombre de usuario, una contraseña y un nombre de host. MySQL verifica la solicitud con la información de la tabla de permisos de acceso **user** en busca de una coincidencia con el nombre de usuario, contraseña y el nombre del host. Si MySQL no encuentra una coincidencia, niega el acceso. MySQL no sólo puede limitar la conexión de una persona, puede restringir el acceso de una máquina e incluso de dominios completos.
- **Verificación de la solicitud.** Ocurre cada vez que un usuario envía una consulta a la base de datos. Primero se verifica los permisos que tiene el usuario en el nivel **user**. Si tiene permisos aquí, el usuario puede hacer lo que desee en cualquier base de datos contenida en el DBMS MySQL. Si no los tiene, consulta la tabla **db** que es la siguiente línea de seguridad. Los permisos especificados aquí son aplicables sólo a la base de datos especificada. Si en esta tabla no tiene permisos, MySQL checa la tabla **tables_priv**, si los privilegios necesarios se encuentran aquí, MySQL ejecuta la consulta; si no, busca en la última tabla: **columns_priv**. Si el usuario no tiene permiso aquí, MySQL devuelve un error indicando que la solicitud es denegada.

2.4.8 Replicación y comunicaciones remotas

MySQL no utiliza la replicación para trabajar de forma distribuida en una red. MySQL se comunica utilizando la configuración cliente-servidor. Es decir, se

pueden crear aplicaciones que permitan el acceso a la base de datos creadas y administradas en MySQL. De esta manera en las máquinas cliente se trabaja con la aplicación que accede a una base de datos y en el servidor se administra y procesa la base de datos.

Para que una aplicación se comuniquen con el servidor utiliza el diseño de tres capas, donde la interfaz²⁷ es la capa intermedia entre el cliente y la base de datos en el servidor.

La interfaz consta de dos partes: la llamada a funciones y variables, y el intérprete o controlador. Este último toma las llamadas a funciones de la primera parte, las traduce e interactúa con la base de datos para producir los resultados esperados.

Para poder hacer una conexión al servidor se debe tener instalado el controlador adecuado, dependiendo de la aplicación.

También se requiere que en la conexión se indique el nombre o la dirección del servidor donde está localizada la base de datos, el nombre de la base de datos, el nombre del usuario y su contraseña.

En MySQL existe la función *mysql_connect* que sirve para hacer la conexión al servidor de la base de datos, Para seleccionar la base de datos se utiliza la función *mysql_select-db* y para cerrar la conexión se utiliza la función *mysql_close*.

²⁷ Colección de funciones que utiliza el programador para, por ejemplo, realizar ciertas tareas de la base de datos como conectarse a la base de datos o enviar una consulta.

CAPITULO III

**Aspectos Generales de los Manejadores
de Bases de Datos**

En este capítulo se hablará de los aspectos generales que rodean a cada uno de los siguientes Manejadores de Bases de Datos: como sus características generales, los recursos de hardware y software que necesita para trabajar de manera óptima, su costo y qué lenguajes de programación pueden acceder a las bases de datos que sean creadas en ellos. Esta información nos servirá para complementar nuestro conocimiento sobre estos Manejadores y así tomar una elección más adecuada cuando se desee implantar un DBMS.

3.1 ORACLE

3.1.1 Características que ofrece

El DBMS líder actual del mercado es Oracle Data Server, mejor conocido como Oracle a secas. La última versión de Oracle es Oracle9i, que salió el 14 de junio del 2001, pero actualmente se sigue utilizando todavía la versión 8i.

A partir de Oracle8i, estos DBMS funcionan como un servidor de bases de datos que ofrece las capacidades de los sistemas relacionales y de los sistemas orientados a objetos.

Estos DBMS ejecutan aplicaciones como:²⁸

- **Procesamiento de Transacciones en línea (OLTP, online Transaction Processing)**. Son aplicaciones que procesan muchas transacciones de actualización pequeñas, como los sistemas bancarios, de reservas y de entrada de pedidos.

²⁸ BOBROWSKI, Steve, *Oracle 8i para Linux Edición de Aprendizaje*, Osborne, España, 2001, pp.5-6

- **Sistema de Toma de decisiones (DSS, Decisión Support Systems)**. Son aplicaciones que consultan la información buscada desde una base de datos para fines de análisis de datos.
- **Grandes almacenes de datos**. Son aplicaciones que acceden a grandes bases de datos de solo lectura. Optimizadas específicamente para un acceso rápido, incluso para la información más impenetrable.
- Particionamiento de datos, los tipos de objetos y los métodos, tipos de datos Objetos grandes (ILOB, Large Objects), La gestión de contraseñas, la unidad Recovery Manager, etc.
- Grandes almacenes de datos y el desarrollo de aplicaciones basadas en web.
- Incluyen una VM (Maquina virtual, Vitual Machine) de Java.
- Sistema de archivos de internet (IFS, Internet File Sistem) de Oracle.

3.1.2 Recursos que utiliza

Oracle8i para Linux: ²⁹

Componente del Sistema	Requisitos
Sistema operativo	Núcleo de linux 2.2 o superior, Paquete GLIBC 2.1, gestor de ventanas que soporta Motif 1.2.

²⁹ Ib idem, p. 16

Explorador web	Soporte Java y Macros (para la documentación)
Memoria física y virtual	128 MB de RAM mínimo; 256 MB de espacio de intercambio mínimo.
Almacenamiento de disco	Una partición de disco de 750 MB para la instalación típica, incluyendo la base de datos inicial (recomendado).
Unidad de CD-ROM	Para la instalación de Oracle8i.

Oracle9i:³⁰

Componente del Sistema	Requisitos
Procesador	Pentium o equivalente (500 MHz recomendado) o SPARC (200 MHz recommended) o HP PA-RISC (200 MHz recomendado)
Explorador web	Soporte Java y Macros (para la documentación)
Memoria física y virtual	256 MB de RAM.
Almacenamiento	J2EE Development

³⁰ Véase http://download-west.oracle.com/docs/html/A95828_02/chap2.htm#1017305

de disco	<p>704 MB (Windows)</p> <p>850 MB (UNIX)</p> <p>Business Intelligence (solo Windows)</p> <p>1.7 GB</p> <p>Rapid Application Development (solo Windows)</p> <p>2.13 GB</p> <p>Completo</p> <p>Windows -2.13 GB</p> <p>UNIX - 1.71 GB</p>
Total Pagefile Size, TMP, o Swap Space	<p>Windows - 384 MB</p> <p>UNIX - 500 MB</p>

3.1.3 Precio en el mercado.

Este manejador de Base de Datos es el más caro en el mercado.

Para Oracle9i:³¹

Para un procesador

³¹ Ver www.oraclestore.oracle.com

Oracle Database Enterprise Edition

- Para siempre: US\$ 40,000.00
 - Por cada usuario más: US\$ 800.00
- Licencia para 4 años: US\$ 24,000.00
 - Por cada usuario más: US\$ 480.00
- Licencia para 2 años: US\$ 14,000.00
 - Por cada usuario más: US\$ 280.00
- Licencia para 1 año: US\$ 8,000.00
 - Por cada usuario más: US\$ 1600.00

Oracle Database Standard Edition

- Para siempre: US\$ 15,000.00
 - Por cada usuario más: US\$ 300.00
- Licencia para 4 años: US\$ 9,000.00
 - Por cada usuario más: US\$ 180.00
- Licencia para 2 años: US\$ 5,250.00
 - Por cada usuario más: US\$ 105.00
- Licencia para 1 año: US\$ 3,000.00
 - Por cada usuario más: US\$ 60.00

Oracle Internet Application Server Edition

→ Para siempre: US\$ 20,000.00

Por cada usuario más: US\$ 400.00

Oracle Database Personal Edition

→ Licencia para 4 años por cada usuario: US\$ 240.00

→ Licencia para 2 años por cada usuario: US\$ 140.00

→ Licencia para 1 año por cada usuario: US\$ 180.00

3.1.4 Que lenguajes de Programación lo pueden utilizar

- Java
- Visual Basic 6.0

De hecho, cualquier lenguaje de programación que se utilice para el desarrollo de sistemas de información, puede tener acceso a Oracle; pero, para esto, es necesario que cuente con el API adecuado para la conexión entre la base de datos y el lenguaje. Comúnmente Java es el lenguaje que se utiliza con mayor frecuencia para crear sistemas que accesan a bases de datos hechas en Oracle.

3.2 SQL Server 2000

3.2.1 Características que ofrece

SQL Server tiene las siguientes características:³²

- **Escalabilidad y Disponibilidad.** SQL Server 2000 admite características como servidores federados, vistas indexadas y soporte para memorias grandes, que le permiten ampliarse a los niveles de rendimiento requeridos por los mayores sitios web.
- **Es un DBMS Multiusuarios.** Es decir que muchos usuarios pueden utilizar la misma aplicación al mismo tiempo.
- **Características de bases de datos corporativas.** El motor de base de datos protege la integridad de los datos a la vez que minimiza la carga de trabajo que supone la administración de miles de usuarios modificando la base de datos simultáneamente. El soporte para transacciones distribuidas protege la integridad de las actualizaciones de los datos distribuidos. La duplicación permite también mantener varias copias de datos a la vez que garantiza que las distintas copias permanezcan sincronizadas.
- **Facilidad de instalación, distribución y utilización.** SQL Server 2000 incluye un conjunto de herramientas administrativas y de desarrollo que mejora el proceso de instalación, distribución, administración y uso de SQL Server en varios sitios. Estas características permiten entregar con rapidez aplicaciones

³² GUZMÁN ZAZUETA, Alejandro, *Microsoft SQL Server 2000*, México, 2002, p.4

de SQL Server que los clientes pueden implementar con un trabajo de instalación y administración mínimo.

- **Almacenamiento de datos.** SQL Server 2000 incluye herramientas para extraer y analizar datos de resumen para el procesamiento analítico en línea. SQL Server incluye también herramientas para diseñar gráficamente las bases de datos y analizar los datos mediante queries.

3.2.2 Recursos que utiliza

Para poder utilizar Microsoft SQL Server 2000 mínimo requiere que tenga su equipo con la siguiente configuración³³.

Procesador	Intel Pentum o compatible 166 MHz en adelante
Sistema Operativo	<p>SQL Server 2000 Edición empresarial y Edición Estándar corren bajo Windows 2000 Server, Windows 2000 Advancer Server, Windows 2000 Datacenter Server operating sistem, Microsoft Windows NT Server versión 4.0 Servis Pack 5 (SP5) o posterior, y Windows NT Server Edición Empresarial con SP5 o posterior.</p> <p>SQL Server 2000 Desktop Engine corre en los Sistemas Operativos mencionados para las ediciones Empresarial y Estándar y además Windows XP Profesional, Windows XP Home Edition,</p>

³³ Ver www.microsoft.com/sql/evaluation/sysreqs/2000/default.asp

	<p>Windows Profesional and Windows NT Workstation 4.0 con SP5 o posterior.</p> <p>SQL Server 2000 Personal Edition and SQL Server 2000 Desktop Engine corre en los sistemas listados para las ediciones Estándar y Empresarial así como, Windows 98, Windows Millennium Edition (Windows Me), Windows XP Profesional, Windows XP Home Edition, Windows 2000 Profesional y Windows Workstation 4.0 con SP5 o posterior</p>
Memoria	<p>Edición Empresarial: 64 MB en RAM; 128 MB de preferencia</p> <p>Edición Estándar: 64 MB; 128 MB de preferencia</p> <p>Developer Edition: 64 MB;</p> <p>Edición Personal: 128 MB para Windows XP; 64 MB para Windows 2000; 32 MB para otro sistema operativo.</p> <p>Desktop Engine: 128 MB para Windows XP; 64 MB para Windows 2000; 32 MB para otro sistema operativo.</p>
Disco Duro	<p>Las ediciones Empresarial, Estándar, Evaluation y personal requieren:</p> <p>De 95 a 270 MB libre en el disco duro para el servidor; 250 MB para la instalación Typical.</p> <p>50 MB libre en disco duro para una mínima</p>

	<p>instalación de análisis de servicios*; 130MB para una instalación Typical.</p> <p>80 MB libres de disco duro para English Query (soportado en Windows 2000).</p> <p>Desktop Engine requiere 44 MB o más de espacio en disco duro.</p>
Drive	CD-ROM drive
Monitor	VGA o mayor resolución.
Otros recursos	<p>Microsoft Internet Explorer versión 5.0 o posterior</p> <p>Soporte de Clientes:</p> <p>Windows 95**, Windows 98, Windows Me, Windows NT Workstation 4.0, Windows 2000 Professional, Windows XP professional y Windows XP Home edition.</p> <p>Unix, Apple Macintosh, and OS/2 requieren Open Database Connectivity (ODBC).</p>

* Análisis de Servicios incluye Online Analytical Edition (OLAP), Data mining, y otras características de almacenamiento de datos (data warehousing features). Son incluidos en la edición personal pero no pueden ser instalados en Windows 98 o Windows Me.

** Soportado solo como cliente conectado, pero no incluye las herramientas graficas de soporte.

3.2.3 Precio en el mercado

SQL Server maneja dos tipos de licencias:

Licencia por Procesador: Se requiere una licencia simple por cada CPU en la computadora que se corra SQL Server. Esta licencia incluye un ilimitado recursos a clientes.

Server/per-seat client access license (CAL). Requiere una licencia para la computadora corriendo SQL Server así como los recursos para un determinado número de CALs. El número de CALs viene incluido en la licencia del servidor y en la licencia del software.

Licencia	Precio*
Licencia por Procesador	Enterprise edition: 19,000 dólares por procesador. Standard Edition: 4,999 dólares por procesador
Server/Per-seat CAL	Enterprise Edition: Con 25 CALs – 11,099 dólares Standard Edition : Con 5 CALs – 1,489 dólares Con 10 CALs – 2,249 dólares
Si se necesita comprar el S. O.	Precio*

Microsoft Windows 2000 Server	1,199 dólares por servidor
Windows 2000 Advanced Server (con 25 licencias para acceso a clientes)	3,999 dólares por servidor
Windows 2000 Datacenter Server	OEM only

* Los precios anunciados en este recuadro pueden variar, para mayor información consultar la página de SQL Server 2000 en www.microsoft.com, o bien en

www.microsoft.com/sql/howtobuy/production.asp

3.2.4 Qué lenguajes de programación lo pueden utilizar

Hay una gran cantidad de lenguajes de programación que pueden tener acceso a SQL Server 2000 algunos de ellos son:

- Visual Basic 6.0
- Visual Fox Pro 6.0
- Visual C++
- Java

De todos estos Lenguajes de Programación el predilecto para conectarse a SQL Server 2000 es Visual Basic 6.0

3.3 ACCESS 2002

Access en realidad es una base de datos de escritorio y no realmente un DBMS. Sin embargo comparte sus archivos con muchos usuarios, y aunque no existe una interfaz que controle la conexión o las respuestas de las solicitudes, es ideal para pequeños negocios donde una o algunas personas van a acceder a la base de datos.

3.3.1 Características que ofrece

Además de las herramientas utilizadas en Access 2000, Access 2002 innova las siguientes características:³⁴

- **Ejecuta sofisticadamente el análisis de datos.** Transforma la dirección del analizador de datos. Fácilmente crea y publica interactivamente distribuyéndola y usando Microsoft PivotTable y Microsoft PívoChar views para presentar importante información dinámica en diferentes direcciones.
- **Fáciles retornos en para el trabajo -o cambiar hacia atrás** (regresar hasta 12 veces)-. Ahora se puede regresar e ir hacia delante en muchos cambios y acciones para una mejor productividad cuando se creen formas, reportes, acceso a páginas de datos, macros y módulos.
- **Convertir Formas y reportes a páginas web.**

³⁴ www.microsoft.com/office/access/evaluation/tour/default.asp

- **Utilizar herramientas disponibles en la web.** En Access 2002 Se puede visitar el sitio de Office Tools por medio de un link a esta página, para bajar ejemplos, herramientas, tips y actualizaciones para trabajar más rápido.
- **Interactuar con tus datos en la Web.** Se puede construir poderosas páginas de acceso a datos que se activen y se abran, vean y actualicen fácilmente datos dentro de la web.
- **Acelerar el desarrollo de acceso a la web.** La poderosa línea de designar páginas de acceso de datos (Data Access Page Designer) hace que éste sea más fácil de publicar las formas y reportes en la web. Tiene la ventaja de enaltecerse con múltiples select, múltiples correcciones hacia atrás y hacia delante e improvisa con colas de control.
- **Usar el poder de XML³⁵ para publicar Reportes.** Ahora se puede publicar rápidamente reportes de Access en la Web usando Internet-standard XML y Extensible Stylesheet Language (XLS), permitiendo ver reporte dentro de la Web navegador que es soportado por HTML 4.0
- **Integración SQL Server Data.** La introducción de esta nueva herramienta da la habilidad de tener acceso a la información nivel corporativo, regresando al final de bases de datos como SQL Server. Access Database Projects le permite crear verdaderas aplicaciones Cliente-servidor usando el familiar interfaz de Access.
- **Fácil manejo de SQL Server Objects.** Creación de SQL Server-specific objects para sus soluciones usando el Query gráfico Designer y Stored Procedure Designer – fuera del inicio de SQL Server expert.

³⁵ XML(Lenguaje de marcado extensible) es un estándar de World Wide Web Consortium (W3C) para la representación de información en forma de documentos estructurados. XML se utiliza cada vez más para el transporte de datos entre sistemas heterogéneos.

- **Tomar ventaja del poder de XML.** Se puede importar fácilmente dentro de Access (Jet) o SQL Server database con unos pocos y simples comandos. Publicando datos a XML es muy fácil: se exporta la tabla, reporte o query e incluye una asociación al archivo XLS para la presentación.

3.3.2 Recursos que utiliza

Access 2002 utiliza los requerimientos mínimos:³⁶

Procesador	Pentium 133MHz o superior, se recomienda Pentium III
Memoria RAM	Depende del sistema operativo: Windows 98 : 32MB Windows Me y Windows NT: 40 MB Windows 2000: 40MB Windows XP: 136MB
Disco Duro	170 MB de espacio del disco duro disponible.
Sistema Operativo	Windows 98, Windows Me, Windows NT, Windows 2000 o Windows XP
Drive	CD_ROM
Monitor	Super VGA (800x600) o mas alta resolución con

³⁶ www.microsoft.com/office/access/evaluation/sysreq.asp

	256 colores
Periféricos	Mause

3.3.3 Precio en el mercado

Para poder adquirir Access 2002 se puede hacer mediante el paquete de Office XP o bien por separado, Debido a esto resulta más barato utilizar Access 2002 para el desarrollo de Bases de Datos pequeñas, ya que no se tiene que invertir en grandes cantidades de dinero, y además en la región la mayoría de las PC's ya cuentan con este tipo de software.

	Professional	Standard	Standard for Students and Teachers	Developer
Retail Upgrade Price ³⁷	\$329 US ²	\$239 US	\$149 US ³	\$549 US
Order Suite Online	Order	Order	Order	Order
<u>Microsoft Word</u> Office XP Word Processor	•	•	•	•
<u>Microsoft Excel</u> Office XP Spreadsheet	•	•	•	•
<u>Microsoft Outlook®</u> Office XP Personal Information Manager and Communication Solution	•	•	•	•
<u>Microsoft PowerPoint®</u> Office XP Presentation Graphics Program	•	•	•	•
<u>Microsoft Access®</u> Office XP Database Solution	•			•
<u>Microsoft FrontPage®</u> Office Web Site Creation and Management Solution				•
<u>SharePoint™ Team Services from Microsoft</u> Team Web Site Solution				•
<u>Developer Tools</u> Developer Tools				•

Figura 7³⁷

³⁷ Ver www.microsoft.com/office/access/howtobuy/default.asp

3.3.4 Que lenguajes de programación lo pueden utilizar

- Visual Basic 6.0
- Visual Fox Pro 6.0
- Visual C++
- Java

Si se desea acceder a una base de datos de Access, comúnmente el lenguaje que más se utiliza es Visual Basic 6.0. Como Access no es multiusuario, los programadores controlan el acceso a la base de datos por medio de este lenguaje de programación, es decir, ellos programan la validación de usuarios del sistema.

3.4 MySQL

3.4.1 Características que ofrece

- **Multiusuarios.** Más de una persona puede acceder a la información de un negocio al mismo tiempo. Pudiendo tener hasta 101 conexiones simultáneas. Ya que MySQL es un servidor multiprocesos.
- **Seguridad.** El acceso a la base de datos en MySQL puede ser determinado desde una maquina remota que controle que usuarios pueden ver una tabla. El acceso a una base de datos puede restringirse aún más aprovechando la seguridad que pueda aportar el sistema operativo.

- **Flexibilidad.** MySQL es extremadamente flexible, y fácil de usar. Puede ejecutarse casi en cualquier plataforma. Si un nuevo CIO quiere cambiar de Windows NT a Linux, MySQL se puede adaptar. MySQL viene con el código fuente, de tal manera que si necesita cambios profundos usted mismo puede editar los cambios.
- **Conectividad con los lenguajes.** MySQL también cuenta con diferentes interfases a nivel aplicación en una variedad de lenguajes. Si el que usted utiliza es principalmente de Microsoft, puede utilizar ODBC para ejecutar MySQL. Si su empresa utiliza Unix puede usar C, Perl o JDBC.
- **Servidor de bases de datos en internet.** MySQL trabaja muy bien como servidor de Bases de datos en internet, ya que ha sido probado y se ha concluido que es una de las bases de datos preferida de los proveedores de servicios de internet.
- **Ayuda en línea.** Para ver qué tipo de argumentos acompañan a los comandos o ver que función desempeña cada comando, todo lo que tiene que hacer es teclear el comando, seguido de un guión y la palabra help o -?. Esto desplegará información acerca del comando.

3.4.2 Recursos que utiliza

Procesador

- 486DX o el más actual

Sistema Operativo:

- Windows 95/98/NT7200/XP

- Linux (x86 libc6),(s7390),(IA64),(Alpha),(SPARC)
- Solaris 2.9/2.8/2.7 (SPARC)
- MacOSXServer 1.x (Power PC)

Memoria RAM:

- 32 MB mínimo

Disco Duro:

- 97.5 MB mínimo

Tarjeta de video:

- 8 MB o superior

Optativo:

- Conexión a Internet

3.4.3 Precio en el mercado

MySQL es muy barato. La versión completa no tiene costo. Para pagar una licencia por una copia se pagaba 200 dólares hasta julio del 2000 cuando se cambio la licencia de MySQL a GPL (General Public License) y ahora sólo para la versión de Windows, cuando se desea comerciar un producto que sólo trabaje con MySQL o cuando se hacen copias de MySQL sin su código fuente se tiene que solicitar la licencia comercial.

Precios para MySQL con licencia comercial: ³⁸

MySQL Pro: US\$ 395.00

MySQL Classic US\$ 200.00

1 año de soporte y el software MySQL: US\$ 1500.00

MySQL con soporte de instalación US\$ 645.00

3.4.4 Qué lenguajes de programación lo pueden utilizar

- Java
- PHP
- C/C++
- Visual Basic 6.0

Lenguaje predilecto para su utilización: PHP, aunque también puede hacer sus aplicaciones para Internet con Java u otro programa similar que le permita conectarse con MySQL y realizar su página Web.

³⁸ www.order.mysql.com/?infopage=?

CAPITULO IV

Ventajas y Desventajas que ofrecen los Manejadores de Bases de Datos

En este cuarto capítulo se señalarán las ventajas y desventajas que presenta cada uno de estos manejadores de datos de acuerdo a la utilización que es ideal para cualquier administración de bases de datos.

4.1 ORACLE

4.1.1 Ventajas de su utilización

Se puede instalar en varias plataformas como Linux, Windows, Unix ...

Tiene más capacidades que los otros DBMS en cuanto a la administración de **grandes bases de datos**.

Oracle se utiliza donde hay un **enorme número de transacciones**, es decir, es costeable para grandes empresas como bancos, compañías de seguros, líneas aéreas, etc.

Como SQL no es en sí un lenguaje de programación, sino de acceso de datos, Oracle implemento **PL/SQL**, que es un *lenguaje de procedimientos* que incluye el lenguaje SQL para construir una aplicación de base de datos. Con esto cualquier desarrollador de Oracle puede realizar diferentes procedimientos con menos código y más fácilmente. PL/SQL se puede utilizar para programar un servidor de bases de datos Oracle y aplicaciones asociadas.

Es muy **seguro**, proporciona toda la seguridad que una empresa necesita.

Permite el acceso a la base de datos tanto de inicios de sesión del sistema operativo como inicios de sesión de Oracle.

Tiene un rasgo llamado Dataguard que permite la creación y dirección de una base de datos de reserva. Se usa típicamente para el apoyo de emergencia.

Oracle además de procedimientos almacenados, permite funciones almacenadas con lo que se hace más fácil la programación para la obtención de datos específicos. También cuenta con paquetes que, como lo vimos anteriormente, es un conjunto de instrucciones PL/SQL, procedimientos y funciones almacenados que son de gran ayuda para la administración de la base de datos.

Cuenta con Triggers que se pueden accionar antes o después de una sentencia. También se puede disparar sólo una vez independientemente de las filas afectadas, o bien una vez por cada fila afectada.

Oracle cuenta además con el trigger INSTEAD OF que tiene la peculiaridad de que se ejecuta el código del trigger en lugar de la instrucción que hizo que se disparara.

Cuenta con tres diferentes tipos de vistas: las normales, las vistas objeto y las vistas materializadas.³⁹

Se pueden realizar copias de seguridad de completas o parciales de una base de datos cuando se ha cerrado la base de datos o cuando está abierta. También puede importar bases de datos ajenas a Oracle, o bien comunicarse con ellas.

Se puede trabajar con Oracle de manera remota por medio de Oracle Net utilizando las configuraciones Servidor-Servidor, Cliente-Servidor, o bien desde la Web.

Permite la replicación multimaestro y de sólo lectura.

Permite diferentes tipos de concurrencia y emplea bloqueos para las transacciones de manera automática, pero también permite que el usuario defina el tipo de bloqueo que desee, aunque normalmente es a nivel tabla.

³⁹ La descripción de estos tipos de vistas se puede ver en la página 31.

4.1.2 ¿Qué inconvenientes presenta?

Se debe tener cuidado al instalar oracle, ya que si no se cumplen con los requerimientos necesarios en cuanto a hardware y software, la instalación puede fracasar o no funcionar adecuadamente, por lo que se recomienda leer detenidamente la documentación que viene incluida en los discos de instalación de Oracle o bien conectarse a la página <http://metalink.oracle.com> o <http://technet.oracle.com> donde debe de estar registrado previamente y podrá obtener información relacionada con la instalación de las últimas versiones de Oracle.

Oracle es más lento que otros DBMS, tanto al cargar el sistema, como ante la conexión de usuarios.

Requiere de equipos con mucho más capacidad que la de los demás manejadores de bases de datos.

Oracle es grande, caro, complejo usar.

Para que la empresa goce de una seguridad óptima debe de adquirir el software necesario como opciones del recargo y sólo para la edición empresarial de Oracle.

Los programadores de Oracle exigen salarios altos debido a que existen pocas personas especializadas en su utilización.

Los contratos de manutención de Oracle son bastante caros.

4.2 SQL SERVER 2000

4.2.1 Beneficios que aporta

Es más veloz que Oracle. En 2 minutos puede realizar más de un millón de transacciones aunque puede variar dependiendo del tamaño de las transacciones y del sistema operativo.

Es el manejador de bases de datos que soporta más usuarios: SQL Server 2000 sobre Windows 2000 proporciona el mayor rendimiento en las pruebas hechas por SAP R/3 Sales and Distribution frente a cualquier plataforma, soportando 26,000 usuarios concurrentes.⁴⁰

Es muy seguro, ya que en su instalación incluye encriptación de paquetes de red (network packet encryption), PKI, o Kerberos.

Soporta bases de datos grandes.

No requiere de equipos con mucha capacidad en comparación con Oracle.

Utiliza TRANSACT-SQL que es un lenguaje de programación creado para SQL Server y contiene un conjunto de instrucciones y programas que facilitan el mantenimiento y la administración de la base de datos.

Permite el uso de procedimientos almacenados que permiten ejecuciones más rápidas y pueden ser ejecutadas desde una aplicación.

Contiene Triggers que se ejecutan después de la instrucción que lo desencadena o en vez de ella (instead_of).

⁴⁰ Véase www.microsoft.com/spain/servidores/sql/productinfo/sql2000_metas.asp

Utiliza bloqueos automáticos en la realización de transacciones o bien permite que el usuario defina el tipo de bloqueo que se desea los cuales pueden ser desde una fila hasta una base de datos.

Maneja vistas simples, actualizables e indexadas.

Permite utilizar cuentas de creadas tanto en un sistema operativo Windows como en SQL Server.

Ofrece copias de seguridad para una base de datos completa o de los cambios generados a partir de la última copia de seguridad hasta de un grupo de archivos.

SQL Server puede trabajar de manera remota por medio de servidores vinculados configurados con OLE DB, configuraciones Cliente-Servidor o bien desde Internet.

Permite la replicación para distribuir las cargas de trabajo o como copia de una base de datos.

Todo lo que se necesita para su instalación y administración de las bases de datos viene incluida en un solo software, por lo que a diferencia de Oracle, no necesita comprar otro tipo de software que complementa al comprado originalmente.

Tiene un precio más accesible y los contratos de mantenimiento no son muy caros.

4.2.2 Desventajas al ser usado

Sólo puede trabajar bajo plataforma Windows.

La OLAP (*On Line Analytical Processing*, procesamiento analítico en línea) no se integra. Por lo que se tiene que realizar la partición OLAP y después acceder a la

base de datos, en vez de hacerlo en un solo paso.⁴¹ Esto es una desventaja ante Oracle.

Ofrece menos recursos en cuanto a código y programas predefinidos para la administración y mantenimiento de una base de datos.

No tiene la capacidad suficiente de administrar bases de datos demasiado grandes en comparación con Oracle.

4.3 ACCESS 2002

4.3.1 Ventajas de su utilización

Access 2002 se caracteriza por un rápido desarrollo y creación de bases de datos que no son muy complejas.

Si trabaja bajo plataforma Windows y tiene Microsoft Office XP, no se tiene que invertir un costo adicional para obtener Access 2002 ya que ya viene incluido. De la misma manera, si usted no cuenta con Microsoft Office XP, no es necesario comprar este software para adquirir Access 2002, pues se puede comprar por separado.

No tiene un costo muy elevado.

Se puede comunicar con otros programas de Office como Excel.

Es un programa amigable con el usuario. Tiene asistentes que le ayudaran a crear una base de datos, informes, consultas, formularios y páginas Web con acceso a datos de Access con mayor facilidad.

⁴¹ ver http://searchwindowsmanageability.techtarget.com/originalContent/0,289142,sid33_gci835650,00.html

Puede generar graficas personalizadas.

En comparación con las versiones anteriores, Access 2002 soporta varios usuarios y se puede publicar datos seguros en Internet.

Access 2002 crea una tabla que especifica la información sobre cada error, haciendo así más fácil la identificación y solución de problemas.

Utiliza macros para realizar ciertas operaciones como la automatización de tareas simples como imprimir un informe.

Tiene una seguridad aceptable pues se puede proteger la base de datos con una contraseña y cifrar la base de datos así como crear grupos de trabajo o usuarios individuales que gocen de ciertos permisos.

Permite la replicación de la base de datos para poder acceder a la base de datos de forma remota, o bien por medio de aplicaciones basadas en el modelo cliente-servidor. También se puede tener acceso a la base de datos por Internet y por diferentes maquinas de una red, si la base de datos es colocada en el servidor.

Cualquier equipo que soporte Windows XP es suficiente para trabajar con Access 2002 pues no requiere de demasiados recursos.

4.3.2 ¿Qué inconvenientes presenta?

Sólo puede trabajar bajo plataforma Windows.

Microsoft Access 2002 no es un manejador de base de datos, por lo que no puede realizar funciones de servidor.

No maneja bases de datos grandes en comparación con SQL Server 2000 y Oracle9i.

Solo maneja 10 tipos de datos por lo que en ocasiones puede causar problemas el almacenar un tipo de dato no contemplado por Access.

No maneja transacciones, procedimientos almacenados ni triggers.

Cuando más de un usuario desean modificar el mismo registro y al mismo tiempo, Access sólo utiliza bloqueo por registro y estos se hacen en tres posibles niveles: sin bloquear, para todos los registros y por registro modificado. Por lo que corre más riesgo la integridad de los datos que con los otros manejadores de bases de datos.

Cuando se trabaja en ambiente multiusuario, se corre el riesgo de que se generen errores debido a que, como Access utiliza la replicación, puedan surgir conflictos cuando 2 usuarios actualicen los mismos datos; o si se tiene que trabajar en tiempo real, pues la base de datos principal no estará actualizada al cien por ciento entre actualizaciones, es decir, es posible que no contemos con todo los datos reales hasta que se actualice la base de datos principal, y como se está trabajando en tiempo real, este problema se vuelve cíclico.

No maneja el tipo de vistas utilizadas en Oracle o en SQL Server y aunque estas pueden ser suplidas con formularios, consultas y páginas de acceso de datos, estos últimos no proporcionan los mismos beneficios.

De los cuatro manejadores de bases de datos evaluados en esta tesis es el que más oportunidades da de violar su seguridad y permitir que personas no deseadas vean la información almacenada en la base de datos.

4.4 MYSQL

4.4.1 Beneficios que aporta

MySQL es pequeño, libre, fácil usar, apoya muchas declaraciones de SQL, las más populares.

Este manejador de bases de datos es el más barato de los mencionados en esta tesis.

Es el manejador de bases de datos más rápido.

Está disponible casi para cualquier sistema operativo.

Soporta muchas interfases disponibles como Perl, Python, PHP, C++, JDBC y ODBC que sirven para crear aplicaciones que se comuniquen con la base de datos.

Soporta hasta 101 usuarios que acceden a la base de datos de manera concurrente.

Tiene muy buena seguridad tanto para un acceso local como para un acceso remoto.

Permite el manejo de bloqueos para procesos de lectura, de escritura con o sin prioridad dependiendo de la urgencia para conceder el bloqueo.

Para recibir una comunicación remota solo lo permite mediante aplicaciones bajo el modelo cliente-servidor o bien en aplicaciones que accedan a la base de datos por medio de Internet.

No requiere de maquinas con gran capacidad para su instalación y para el almacenamiento de bases de datos.

4.4.2 Desventajas al ser usado.

Soporta bases de datos de funcionalidad limitada, es decir, aplicaciones de tamaño de medio.

No contiene un ambiente gráfico, o al menos no para muchos sistemas operativos.

Aunque la última versión estable de MySQL (3.23.32) ya soporta **transacciones** éstas son muy limitadas debido a que no soporta transacciones grandes ni demasiadas.

Tampoco soporta las **vistas, procedimientos almacenados y triggers**.

No tiene una fuerte Integridad de Referencial:

Llaves foráneas: La sintaxis de llaves foráneas en MySQL sólo existe para la compatibilidad con otros vendedores de SQL. La llave foránea a veces se usa como un pase obligatorio pero en realidad no verifica si los datos introducidos en el campo designado como llave foránea existen en el campo correspondiente a la tabla de la cual se hizo referencia.

No evita la concurrencia en datos que son modificados al mismo tiempo y por más de un usuario si estos últimos no tienen el cuidado de hacer la solicitud con el tipo de bloqueo apropiado; es decir, MySQL no maneja la concesión de bloqueos de manera automática.

No utiliza la replicación para trabajar de forma distribuida en una red.

4.5 Conclusión

Cada manejador de bases de datos tiene su ventaja pero no debemos tomar una decisión tan sólo por las herramientas que nos brindan o no, sino por las características que necesitamos para desarrollar una base de datos así como por los recursos económicos de la empresa para la que se va a construir la base de datos y además, por las características del hardware, ya sea el que tiene la empresa, o por como se va a adquirir.

Debemos evaluar las ventajas y desventajas de estos manejadores de base de datos a conciencia, para que al tomar la decisión de cual manejador de bases de datos utilizar, sea la acertada y podamos cubrir la mayor cantidad de necesidades del cliente en lo posible.

CONCLUSIÓN

A lo largo del estudio de esta tesis hemos visto y analizado los cuatro manejadores de bases de datos más utilizados en la actualidad: Oracle SQL Server, Access y MySQL. Me he enfocado a sus versiones más actuales procurando que esta investigación no sólo sea de ayuda a personas del área en estos momentos sino también más adelante.

Se trató de abarcar todos los aspectos de estos manejadores: desde las herramientas que ofrece, sus capacidades, los requerimientos de equipo, su precio, lenguajes que desarrollan aplicaciones para conectarse a sus bases de datos, entre otras características que considero necesarias para tomar la decisión más acertada al momento de elegir un DBMS.

El siguiente cuadro muestra de manera resumida algunas de estas características:

	Oracle	SQL Server	Access	MySQL
Tipos de datos	Abarca todos los diferentes tipos que hay	Abarca todos los diferentes tipos que hay	Solo 10 tipos diferentes	Abarca todos los diferentes tipos que hay
Integridad	La maneja	La maneja	La maneja	La simula
Transacciones	Las trabaja	Las trabaja	No las utiliza	No las utiliza
Procedimientos Almacenados	Maneja Procedimientos y funciones almacenadas y paquetes	Sólo procedimientos almacenados	No las maneja	No las maneja

Triggers	Se ejecuta antes y después de la instrucción que lo dispara o en ves de ella	Se ejecuta después de la instrucción que lo dispara o en ves de ella	No lo utiliza	No lo utiliza
Concurrencia y tipos de bloqueo	Permite la concurrencia y los bloqueos de manera automática (default) o manual. Los bloqueos más utilizados son para el bloqueo de registros	Permite la concurrencia sólo para lectura. Utiliza bloqueos automáticos o manuales desde un registro hasta una base de datos	No restringe para la concurrencia. Utiliza bloqueos a nivel registro los cuales pueden ser nulos, de sólo lectura o totales	No restringe la concurrencia ni establece los tipos de bloqueo de manera automática. Utiliza bloqueos a nivel registro y a nivel tabla
Vistas	Maneja las vistas normales, de objetos y materializadas	Maneja vistas simples, actualizables e indexadas	No maneja vistas pero si formularios, consultas y paginas de acceso a datos	No maneja vistas
Seguridad	Excelente	Excelente	Regular	Excelente
Comunicaciones remotas	Servidor-Servidor Cliente-servidor Páginas Web	Servidor-Servidor Cliente-servidor Páginas Web	Cliente-servidor Páginas Web (páginas de acceso a datos)	Cliente-servidor Páginas Web
Requerimientos de equipo	256MB en RAM para 9i	De 64 a 128 Mb en RAM	Hasta 133Mb en RAM	32 MB en RAM

	2.13Gb de disco duro para instalación y memoria virtual para 9i Procesador a 500Mhz de velocidad	500Mb como mínimo de disco duro Procesador a 166Mhz en adelante	170MB de espacio en disco duro como mínimo Procesador a 133Mhz o superior	97.5 MB como mínimo en disco duro Procesador 486DX en adelante
Costo	40,000.00 dólares + 800.00 por cada usuario más	19,000.00 dólares	Hasta 549 dólares	Licencia GLP gratis Comercial: 395 dólares
Sistema Operativo	Windows Linux Unix	Windows	Windows	Windows Linux Solaris ...

Estas y otras características son necesarias para decidir cual manejador de bases de datos es el ideal para implantar en cualquier institución, negocio o empresa.

También es importante evaluar las necesidades del cliente para en base a esas necesidades, a las características que nos ofrecen los DBMS y su costo, podamos elegir cuál es el más adecuado.

Por ejemplo, Oracle es más ideal para empresas muy grandes donde manejan bases de datos de gran tamaño y con muchas relaciones, ya que le puede

soportar todos estos requerimientos de la empresa además de toda la seguridad que necesita y todos los reportes estadísticos o de cualquier otro tipo que se necesiten.

SQL Server puede soportar bases de datos muy grandes y con muchas relaciones por lo que es recomendable para empresas grandes a las cuales les puede proporcionar todos los servicios de seguridad, de reportes, etc. que éstas requieran. En realidad no hay mucha diferencia en cuanto a capacidad entre Oracle y SQL Server.

Access es más apropiado para negocios pequeños donde solo requieren de una base de datos normal y bien relacionada, la cual tendrá uno o varios usuarios. Como Access no es un servidor de bases de datos algunas cuestiones de seguridad y de acceso a usuarios tendrán que ser controladas por la aplicación que acceda a la base de datos.

Con MySQL se pueden crear bases de datos que no requieran de muchas relaciones por lo que no pueden ser bases de datos muy grandes. Permite trabajar con varios usuarios de forma simultánea por lo que es ideal para negocios o instituciones que necesitan una base de datos del estilo a la que acabo de describir. Las bases de datos creadas en MySQL son utilizadas con mayor frecuencia para aplicaciones de Internet (sobretudo para ventas) debido a la cantidad de usuarios que permite acceder y a que la base de datos no es muy compleja, pero hay que aclarar que para este tipo de aplicaciones el programador es el que tiene que cuidar muchos aspectos de integridad que es de donde más cojea este DBMS.

Como hemos visto a lo largo de la tesis, podemos tener un criterio más amplio para elegir un manejador de bases de datos adecuado a las necesidades propias o de nuestro cliente basándonos en las ventajas y desventajas que cada uno de estos DBMS presenta.

Esta tesis también nos puede servir de referencia para analizar otro tipo de manejadores de bases de datos y evaluar qué tan conveniente es su utilización o no de acuerdo a lo que necesitamos.

Así que considero que mi tesis representa una gran ayuda para las personas que tengan que tomar el tipo de decisiones descritas anteriormente, así como a estudiantes y catedráticos que estén involucrados con el tema de bases de datos.

BIBLIOGRAFÍA

ANDERSEN, Virginia, **Access 2002 a su alcance**, Mc Graw Hill, México, 2002, 662 pp.

BOBROWSKI, Steve, **Oracle 8i para Linux Edición de Aprendizaje**, Osborne, España, 2001, 489 pp.

GUZMÁN ZAZUETA, Alejandro, **Microsoft SQL Server 2000**, México, 2002, 117 pp.

LONEY, Kevin THERIAULT, Marlene TUSC, **Oracle9i Manual del Administrador**, OSBORNE, España, 2002, 996 pp.

MASLAKOWSKI, Marky y BUTCHER, Tony, **Aprendiendo MySQL en 21 días**, Pearson Educación, México, 2001, 534 pp.

SMITH, Curtis y AMUNDANSEN, Michael, **Aprendiendo Programación de bases de datos con Visual Basic 6 en 21 días**, Prentice Hall, México, 1999, 883 pp.

STANEK, R. William, **Microsoft SQL Server 2000 Manual del administrador**, MC GRAW HILL INTERAMERICANA, España, 2001, 646 pp.

OTRAS FUENTES:

<http://dei.uca.edu.sv/computacion/modules.php?name=Surveys&op=results&pollID=6>

<http://dis.um.es/~mjortin/ibd.html#temasteo>

http://download-west.oracle.com/docs/html/A95828_02/chap2.htm#1017305

<http://ict.udlap.mx/people/carlos/dbms/index.html>

<http://informatica.uv.es/iiguia/1993/GBD/arquitectura.doc>

<http://otri.us.es/recursosPHP/manual/mysql/mysql.htm>

http://searchwindowsmanageability.techtarget.com/originalContent/0,289142,sid33_gci835650,00.htm

<http://webmaster.bankhacker.com/ayuda-linux/consulta/Preguntas+acerca+de+Mysql/>

www.faqs.org/faqs/databases/free-databases/

www.itlp.edu.mx/publica/tutoriales/basedat1/tema1_1.htm

www.itlp.edu.mx/publica/tutoriales/basedat1/tema1_9.htm

www.microsoft.com/latam/technet/articulos/200202/art15/

www.microsoft.com/office/access/evaluation/sysreq.asp

www.microsoft.com/office/access/evaluation/tour/default.asp

www.microsoft.com/office/access/howtobuy/default.asp

www.microsoft.com/spain/servidores/sql/productinfo/sql2000_metas.asp

www.microsoft.com/sql/evaluation/sysreqs/2000/default.asp

www.microsoft.com/sql/howtobuy/production.asp

www.oraclestore.oracle.com

www.order.mysql.com/?infopage=?