



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

UNIVERSIDAD AUTÓNOMA
DE QUERÉTARO



Facultad de Informática

"ORACLE"
Administración y manejo de BD

TESINA

Que para obtener el título de:

LIC. EN INFORMATICA

Presenta:

Blanca Rocío Morales Medina

Santiago de Querétaro, Qro. Octubre del 2000



TS
005.75
M828o

F06979

No.	NOMBRE Y FIRMA	VENCE

TS
005.75
M828o

F06979



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

No. Adq. F06979

Clasif. TS 009.75

Cutter M828o



Universidad Autónoma de Querétaro
Facultad de Informática



CARTA DE ACEPTACIÓN

Por este medio, se otorga constancia de aceptación de tesina para obtener el título de Licenciado en Informática, que presenta la pasante **BLANCA ROCIO MORALES MEDINA** con el tema denominado *Oracle, Administración y Manejo*.

Este trabajo fue desarrollado como una investigación derivada del curso de titulación “**SISTEMA OPERATIVO UNIX I**” – Nivel **Introdutorio**, dando cumplimiento a uno de los requisitos contemplados en el artículo 34 del reglamento de titulación vigente, en lo referente a la opción de titulación por realización y aprobación de cursos de actualización.

Se extiende la presente para los fines legales a que haya lugar y para su inclusión en todos los ejemplares impresos de la tesina, a los veintidós días del mes de febrero de dos mil.

ATENTAMENTE

ING. JUAN GABRIEL FRANCO DELGADO
PROFR. CURSO DE TITULACIÓN

AGRADECIMIENTOS

Gracias, a Dios por darme
el regalo más grande que
todo ser humano pueda
tener... La vida.

A Carmen por los cuidados,
desvelos y el amor
incondicional que solo de ti
puedo recibir, gracias mamá
porque por ti, soy lo que soy.
Este logro también es tuyo!!

A mi familia, en especial a
Mamá Elena y a mis tías Sara, Elena,
Estela, a mis primos (as), sobrinos
(as); porque en todo momento bueno
y malo siempre estamos juntas,
gracias por todo su cariño y apoyo,
espero que este pequeño triunfo sirva
de motivación para los que siguen.

A Clemen, Benjamin, Blanca, Erika
y Yessica; porque me permitieron ver
que la verdadera amistad aún existe;
gracias por compartir su tiempo
conmigo.

AGRADECIMIENTOS

A todos mis compañeros de trabajo en particular a Paty, Rodolfo, Fernando y Deya; porque he recibido de ustedes, más que un apoyo de trabajo.

Mario; simplemente te puedo decir que te quiero muchísimo y te agradezco de corazón, la confianza, paciencia, dedicación y el tiempo que has compartido conmigo.

A la fam. Vázquez Cabrera y Cabrera Muñoz, que de alguna manera me motivaron siempre a seguir en el camino de la superación. Gracias de todo corazón.

A ti papá, aunque siempre estamos lejos y peleando quiero agradecerte, porque hasta eso me ha servido para madurar, te quiero.

A Antonio Torres, por ser un gran ser humano en toda la extensión de la palabra; gracias por toda la ayuda y apoyo que me has brindado, y sobre todo gracias por amarme y aceptarme sin condiciones; tal como soy... Te amo Oso!!

INDICE

1. PREFACIO	1
2. INTRODUCCION	2
3. BREVE HISTORIA DE ORACLE	3
4. VENTAJAS Y DESVENTAJAS	4
4.1 Ventajas	4
4.2 Desventajas	5
5. TIEMPOS DE IMPLANTACION RAPIDA, PROCESOS Y MEJORA CONTINUA	7
6. CONOCIENTOS Y EXITOS COMPARTIDOS	8
6.1 Conocimientos generales de una aplicación Oracle	8
6.2 El enfoque cooperativo	10
6.3 Todo el mundo tiene datos	10
6.4 Características de Oracle	12
7. ARQUITECTURA ORACLE	13
7.1 Bases de datos	13
7.2 Espacios de Tablas (Tablespaces)	14
7.3 Archivos	14
7.4 Instancias	15
8. ESTRUCTURAS INTERNAS DE LAS BD	16
8.1 Estructuras de las BD	16
8.2 Tablas y columnas	16
8.3 Usuarios	17
8.4 Esquemas	18
8.5 Indices	18
8.6 Grupos	19
8.7 Grupos Hash	19
8.8 Vistas	20
8.9 Secuencias	20
8.10 Procedimientos	20
8.11 Funciones	21
8.12 Paquetes	21
8.13 Disparadores	21

8.13.1 Tipos de disparadores	22
8.14 Sinónimos	22
8.15 Privilegios y funciones (Roles)	23
8.16 Enlaces de Bases de Datos	24
8.17 Estructuras internas de memoria	24
9. EL LENGUAJE FAMILIAR DE ORACLE	26
9.1 Ejemplo	26
9.2 Tablas de información	27
9.3 Lenguaje de Consulta estructurado	28
9.4 Ejemplo de una consulta simple con Oracle	28
9.5 Origen de la denominación "Relacional"	31
10. LOS PELIGROS DE LAS BASES RELACIONALES	33
10.1 Panorama general	33
10.2 ¿Cuáles son los riesgos?	33
11. USO DE CODIGOS FRENTE AL LENGUAJE	36
11.1 Las principales justificaciones del uso de códigos	36
11.2 Normalización	37
11.3 El método lógico	38
11.4 Primera forma normal	38
11.5 Segunda forma normal	39
11.6 Tercera forma normal	39
11.7 Caché del diccionario	39
11.8 Búfer de registro rehacer	40
11.9 Fondo común SQL compartido	41
11.10 Areas de contexto	41
11.11 Area global del programa (PGA)	41
12. ESTRUCTURAS DE LOS PROCESOS	43
12.1 Procesos	43
12.2 Supervisor del sistema: SMON	44
12.3 Supervisor del proceso: PMON	44
12.4 Escritor de Bases de Datos: DBWR	44
12.5 Escritor de registros: LGWR	45
12.6 Punto de control: CKPT	45
12.7 Archivador: ARCH	46
12.8 Recuperador (Opción distribuida): RECO	47
12.9 Bloqueo (Servidor Paralelo):LCKn	47
12.10 Despachador (Oracle, SQLNet V):Dnnn	48
12.11 Servidor: Snnn	48
12.12 Estructuras externas	48
12.13 Registros de rehacer	48
12.14 Archivos de control	49

13. IMPLEMENTACION BASICA DE LA BASE DE DATOS	51
14. COPIAS DE SEGURIDAD Y RECUPERACION	52
14.1 Export/Import (Exportar/Importar)	52
14.2 Copias de seguridad fuera de línea	53
14.3 Copias de seguridad en línea	53
15. CAPACIDADES DE SEGURIDAD	54
15.1 Seguridad de las cuentas	54
15.2 Privilegios de las Bases de Datos	54
15.3 Seguridad de los objetos	54
15.4 Auditoría	55
15.5 Comprensión de los convenios de modelado lógico	55
15.6 Configuraciones y consideraciones sobre el Hardware	55
15.7 Introducción a la arquitectura	56
15.8 Anfitriones autónomos	57
15.9 Anfitriones autónomos con ristas de discos	58
15.10 Réplica del archivo de control	59
15.11 Réplica del archivo de registro de rehacer	59
15.12 Anfitriones autónomos con duplicado de disco	60
15.13 Anfitriones autónomos con varias Bases de Datos	62
15.14 Anfitriones de red	63
15.15 Redes de Bases de Datos	64
15.16 Consultas remotas	65
15.17 Conexiones con Bases de Datos distintas a Oracle	67
15.18 Actualizaciones remotas: La opción distribuida	67
15.19 Servidores agrupados: Opción de servidor en paralelo	68
15.20 Aplicaciones de Bases de Datos Cliente-Servidor	72
16. LA UTILIZACION DEL ESPACIO DE LAS BASES DE DATOS	75
16.1 Introducción	75
16.2 Implicaciones de la cláusula Storage	77
17. SEGMENTOS	79
17.1 Segmentos, extensiones y bloques	79
17.2 Segmentos de tabla	80
17.3 Segmentos de índice	81
17.4 Segmentos de Rollback (Anulación)	81
17.5 Segmentos temporales	82
17.6 Espacio libre	83
17.7 Encaje físico	84
18. GESTION DEL PROCESO DE DESARROLLO	86
18.1 Tecnología	86
18.2 Herramientas CASE	86

18.3 Directorios compartidos	87
18.4 Bases de Datos de gestión de proyectos	87
18.5 Bases de Datos de discusión	88
18.6 Gestión del desarrollo de paquetes	88
18.7 Generación de diagramas	88
18.8 Requisitos de espacio	89
18.9 Objetivos de ajuste	89
18.10 Requisitos de seguridad	89
18.11 Requisitos de los datos	90
18.12 Requisitos de ejecución	91
18.13 Procedimientos de la prueba de aceptación	91
18.14 Entorno gestionado	92
19. PRODUCTOS ADICIONALES Y NUEVAS APLICACIONES DE ORACLE	93
19.1 Tecnología aplicada, clave de la innovación empresarial	93
19.2 Oracle Designer	93
19.3 OracleWare	94
19.3.1 Integrado con Netware 3.12, Oracle, Oracle Office y Oracle Glue	94
19.4 Oracle mobile agents	95
19.5 Manufacturing	95
19.6 Supply Chain	96
19.7 Financials	96
19.8 Projects	97
19.9 Human resources	97
19.20 Front Office	98
19.21 Desarrollo de aplicaciones multicapa	98
19.22 Aplicaciones de Oracle	99
20. COMPARATIVO CON OTRAS BD	100
20.1 El análisis	100
20.2 Alpha Five	101
20.3 CA-Visual Objets	102
20.4 DBASE 5.0	104
20.5 DB/2 para OS/2	104
20.6 Deplhi 95	106
20.7 Microsoft SQL server for windows NT	107
20.8 Oracle CDE2	108
20.9 PowerBuilder 4.0	110
20.10 Superbase 3.0	111
20.11 Visual Fox 3.0	112
20.12 Gupta	113
20.12.1 Los principales de Gupta	113
20.13 Informix	113
20.13.1 Principales productos	113
20.14 Progress	115

20.15 El concepto Cliente – Servidor	116
21. GLOSARIO	117
22. CONCLUSIONES	118
23. BIBLIOGRAFIA	120

1 PREFACIO

México, como país en vías de desarrollo, en pleno proceso de industrialización y con grandes riquezas potenciales, tiene la urgente necesidad de contar con nuevas técnicas y herramientas idóneas para coordinar los recursos de los diversos sectores productivos. Esto se puede lograr en gran parte, mediante la aplicación de la informática y más aún en la herramienta ORACLE manejador de base de datos.

Para nadie es desconocida la gran importancia de esta disciplina que capacita al profesional de cualquier área para tomar decisiones más racionales, optimizar la utilización de los recursos y lograr con mayor eficiencia los objetivos de cualquier tipo de organización.

Esta concepción de la informática específicamente en el manejador de bases de datos ORACLE, ha originado que se le considere como una panacea y consecuentemente que cada vez tenga mayor difusión tanto a nivel profesional como a nivel empresarial.

Este trabajo va dirigido a todas aquellas personas que tienen por lo menos algún vínculo con el manejo de información a través de la computadora, esperando sea de gran provecho y sirva de guía al usar esta herramienta.

2 INTRODUCCION

El presente trabajo está destinado a fomentar de una manera específica esta comprensión, ofreciendo los medios para que tanto los usuarios como los diseñadores y sobre todo los administradores puedan explotar todo el potencial que ofrece ORACLE. El usuario final conoce detalles sobre el problema que el diseñador y administrador no abarcan.

El diseñador comprende funciones internas y características de ORACLE y del entorno de la computadora que son demasiado complejas técnicamente para el usuario final; de igual forma el administrador de la base de datos ORACLE, eso es precisamente a lo que se dedica, a administrar el manejo de las bases de datos; cosa que el usuario final sólo se dedica a alimentar el sistema; por lo tanto el diseñador como el administrador de las bases de datos ORACLE deben de preparar bien la antesala, analizar todos y cada uno de los datos, así como encauzar adecuadamente la salida de los mismos, esto es, planear debidamente, formar la estructura quedando por último la presentación final para el usuario.

3 BREVE HISTORIA DE ORACLE

Liderazgo, resultado de nuestra histórica Innovación Tecnológica. Hace ya más de dos décadas, Oracle Corporation, fundada en 1977, liberó la primera base de datos relacional del mundo. Hoy en día, Oracle es el proveedor más grande de soluciones integrales para el manejo de información. Los productos y servicios Oracle están presentes en más de 140 países, lo que ha llevado a la organización a tener un crecimiento sin precedentes en la industria informática.

En catorce de sus veinte años de vida, Oracle Corporation ha tenido un crecimiento constante; convirtiéndose en el año 92 en el proveedor de software de Bases de Datos más grande del mundo y en 1994 en la segunda empresa de software más grande a nivel mundial. Dichos lugares los ha conservado hasta la fecha basados en la calidad de los productos y servicios que se ofrecen.

Oracle se distingue como empresa proveedora de soluciones que ofrecen una garantía a las necesidades cambiantes de las empresas actuales, una "Solución Integral" que contempla los siguientes aspectos: capacidad para almacenar, procesar y distribuir información, apertura hacia los estándares de la industria y portabilidad a diversas plataformas. Esta estrategia de negocios permite un mejor acceso a los datos corporativos reduciendo costos, incrementando la productividad y entregando resultados de negocios.

4 VENTAJAS Y DESVENTAJAS DE ORACLE

4.1 VENTAJAS

- ✓ Oracle es una base de datos distribuida, esto es que es una base de datos transparente que nos permite acceder a los datos sin tener que especificar dónde se encuentra la red.
- ✓ Oracle cuenta con la posibilidad de insertar, modificar y suprimir datos con una trtansparencia completa de ubicación: para insertar, modificar o borrar no necesitamos saber dónde están almacenados los datos, sino exclusivamente los nombres de las tablas y las columnas.
- ✓ Cuando la ubicación de los datos es transparente, podemos ejecutar fácilmente cualquier sentencia SQL para cualquier conjunto de datos de la base de datos distribuida.
- ✓ En Oracle se puede hacer referencia a procedimientos y triggers (disparadores) almacenados sin saber dónde se encuentran.
- ✓ Podemos introducir datos o procedimientos en el punto en que tenga más sentido para el almacenamiento, sin tener que modificar en absoluto nuestras aplicaciones y sin tener que molestar a los usuarios para que modifiquen su visión del sistema cuando se decide reubicar los datos en posiciones distintas.
- ✓ Las soluciones aportadas por Oracle incluyen un sistema de seguridad completamente nuevo, la posibilidad de trabajar con idiomas distintos del inglés estándar, los procedimientos y disparadores almacenados, los perfiles de recursos para un mayor control sobre el uso de la base de datos y un número cargador masivo de datos más rápido.
- ✓ Podemos usar procedimientos y triggers almacenados en cualquier punto de la red. Estos procedimientos almacenados pueden acceder con la misma facilidad a datos locales o remotos.

- ✓ Podemos usar las herramientas de administración de Oracle para gestionar bases de datos remotas.
- ✓ Oracle gestiona las transacciones distribuidas con una característica especial denominada confirmación en dos fases que garantiza que una transacción tendrá el mismo resultado en todos los lugares.
- ✓ Oracle permite controlar mucho mejor el uso de una base de datos utilizando perfiles que son un conjunto de limitaciones en el uso de los recursos que no pueden ser superadas por el usuario al que se le asigna el perfil. Los perfiles incluyen sesiones por usuario, tiempo de CPU, tiempo de conexión, tiempo inactivo y lecturas lógicas; así se puede limitar el impacto de las consultas y proteger la seguridad del sistema limitando los recursos asignados a un usuario.

Oracle 8 es un sistema de objetos relacional, agrega una capa de abstracción de datos sobre los métodos relacionales estándar de representación de datos y programación y también provee la habilidad para extender los mecanismos de almacenamiento para arreglárselas con complejos tipos de datos. La abstracción de objetos relacionales que Oracle provee te permite la habilidad para agregar nuevos tipos de datos para SQL estándar.

4.2 DESVENTAJAS

Así como existen ventajas, también hay desventajas como en cualquier herramienta computacional, desgraciadamente Oracle no es la excepción, pues dependiendo de lo que se quiera hacer y el equipo en donde va a trabajar, siempre habrá trabas que conlleven a una soluciones no del todo confiables y poderosas para soportar ciertos procesos. Almacenar y recuperar arreglos de valores y tablas anidadas (valores de columnas que son a su vez tablas) y métodos (subprogramas almacenados que asocias con nuevos tipos de objetos) son tareas que Oracle puede soportar; sin embargo no hay sistema de herencia para los tipos que agregas, no puedes esconder las columnas de datos en tus tipos de objetos.

Finalmente, el sistema de objetos relacionales provee un camino para mirar objetos complejos como tablas relacionales. Oracle llama a éstos en la vista de objetos.

Mientras estas características no sean dadas verdaderamente, las capacidades de Programación de Base de Datos Orientada a Objetos (OODBMS) no será lo suficientemente fuerte dentro del mundo de los objetos. La fuerza de Oracle ha sido su gran capacidad del servidor, seguridad, integridad, concurrencia y alto volumen de procesamiento de transacciones, alto desempeño, rentabilidad y la habilidad para operar en diversas plataformas han elevado el producto a la vanguardia dentro del mercado de las bases de datos.

Oracle extiende éstas ventajas al permitir las capacidades cliente-servidor y agregar la extensibilidad de objetos relacionales que las aplicaciones mundiales necesitan.

5 TIEMPOS DE IMPLANTACION, PROCESOS Y MEJORA CONTINUA

La mayoría de los sistemas de aplicaciones requieren mucho tiempo para su implantación. Una vez instalados, requieren mucho tiempo para adaptarlos a las necesidades cambiantes de la empresa.

Las Aplicaciones Oracle ofrecen una flexibilidad sin precedentes. Nuestros clientes suelen instalarlas en la mitad de tiempo y a la mitad de coste de lo que supondría instalar un sistema de la competencia.

Las Aplicaciones Oracle se pueden reconfigurar rápidamente, a medida que cambian las necesidades de la empresa, con reglas de negocio definidas por el usuario, reglas de flujo de trabajo, filtros, informes y condiciones de excepción que reflejen la mejora de los procesos de empresa. Se pueden incluso añadir campos definidos por el usuario sin tener que programar. Los cambios no sólo son fáciles de realizar, sino que se actualizan automáticamente a medida que se migra a nuevas versiones. De este modo se evita el 'bloqueo de versiones' y el coste exorbitante de las actualizaciones, característicos de otras de aplicaciones.

6 CONOCIMIENTOS Y EXITOS COMPARTIDOS

6.1 CONOCIMIENTOS GENERALES DE UNA APLICACIÓN ORACLE

Para que una aplicación Oracle se pueda desarrollar y usar rápida y efectivamente, es necesario que tanto los usuarios como los diseñadores tengan un lenguaje común y una profunda y común comprensión de la aplicación y de las herramientas de ORACLE.

Este nuevo enfoque en el desarrollo de aplicaciones. Históricamente, los analistas de sistemas estudiaban las necesidades del problema y construían una aplicación que las cubría. Los usuarios sólo intervenían en la descripción del problema, y quizás en la revisión de la aplicación una vez acabada ésta.

Con las nuevas herramientas y métodos disponibles, y especialmente con ORACLE, se pueden construir aplicaciones y métodos disponibles, y especialmente con necesidades y hábitos de trabajo de las empresas, pero sólo existe una comprensión mutua.

El usuario final conoce detalles sobre el problema que el diseñador y administrador no abarcan. El diseñador comprende funciones internas y características de ORACLE y del entorno de la computadora que son demasiado complejas técnicamente para el usuario final; de igual forma, el administrador de la base de datos ORACLE, eso es precisamente a lo que se dedica, a administrar el manejo de las bases de datos; cosa que el usuario final sólo se dedica a alimentar el sistema; por lo tanto el diseñador como el administrador de las bases de datos ORACLE deben de preparar bien la antesala, analizar todos y cada uno de los datos, así como encauzar adecuadamente la salida de los mismos, esto es,

planear debidamente, formar la estructura quedando por último la presentación final para el usuario.

No es un secreto que entre los usuarios, los analistas y administradores siempre han existido problemas. Entre las razones de esto, se encuentran las diferencias de conocimiento, de cultura, de intereses profesionales y metas, y las confusiones que la simple separación física de los grupos puede producir. Realmente, este hecho no es exclusivo del mundo informático. Lo mismo ocurre en las demás áreas de toda organización.

Las relaciones entre los miembros de estos grupos diferentes son formales, estiradas y anormales. Las barreras artificiales y procedimientos que surgen de este aislamiento también contribuyen a la aparición de este hecho.

Por su parte ORACLE como no permanece oculto por ser un lenguaje arcaico que sólo los profesionales pueden comprender, ORACLE cambia de manera fundamental las relaciones entre los administradores de bases de datos, los analistas y los usuarios. Cualquiera lo puede comprender, cualquiera lo puede utilizar, pues la información que antes estaba atrapada dentro de la computadora hasta que un analista creaba un nuevo informe y la liberaba, ahora puede ser accesible instantáneamente para el usuario, escribiendo simplemente una pregunta en inglés. Esto supone el cambio de las reglas de juego.

Desde su primera versión, ORACLE se ha basado en el fácilmente comprensible modelo relacional, así que los no programadores entienden rápidamente lo que ORACLE puede hacer y cómo. Esto lo hace próximo y no impuesto.

Además, se desarrollo de tal forma que se pudiera ejecutar de igual forma sobre prácticamente cualquier tipo de computadora, esto debido al profundo éxito del producto y la compañía.

En un mercado caracterizado por compañías con su hardware “particular”, su sistema operativo “particular”, su base de datos “particular” y sus aplicaciones particulares, ORACLE proporciona a los usuarios y a los departamentos de análisis y administración de bases de datos un nuevo control sobre su modo de trabajo y su futuro.

Los usuarios, analistas y administradores no estarán tan ligados a la base de datos proporcionada por una única compañía de hardware. ORACLE puede funcionar sobre casi todos los tipos de computadoras que pueda poseer. Esto es una revolución en el entorno de trabajo y en el desarrollo de aplicaciones, cuyas consecuencias se verán en un futuro.

6.2 EL ENFOQUE COOPERATIVO

ORACLE es una base de datos relacional. Este concepto ofrece una forma extremadamente sencilla para pensar y manejar los datos que se usan en una empresa. No es nada más que una colección de tablas de datos. Nos encontramos tablas todos los días: informes de tiempo, cotizaciones de bolsa, clasificaciones, etc. Todo esto son tablas, con encabezamientos para las columnas y filas de información presentadas de forma simple. La aproximación relacional puede ser suficientemente poderosa y sofisticada, incluso para los negocios más complejos.

6.3 TODO EL MUNDO TIENE DATOS

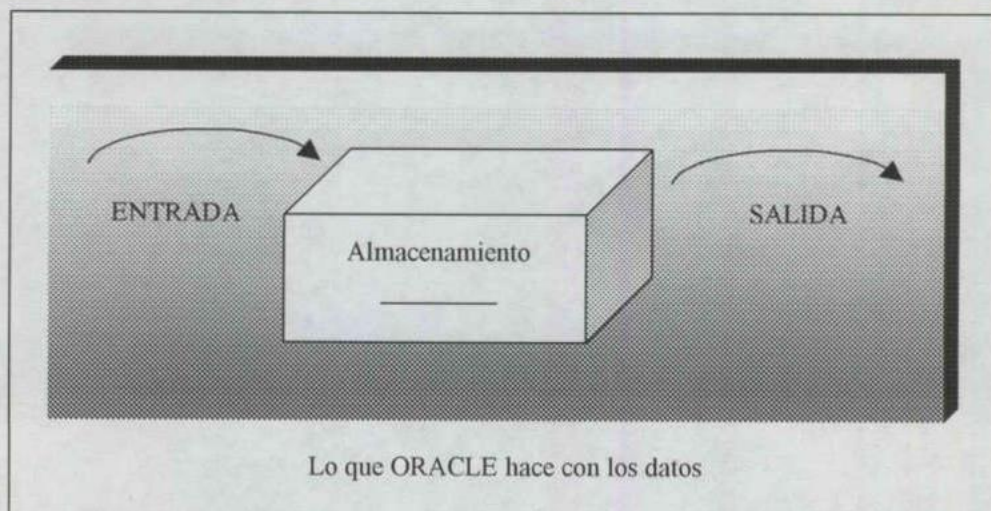
Los filósofos de la información suelen decir que un dato debe ser considerado como tal hasta el momento en que se organiza de manera significativa, en el que pasa a ser “información”. Si esto es cierto, ORACLE es una forma de ordenar y manipular los datos de forma que se pueda obtener información no visible, como totales, tendencias de venta y otras relaciones. Lo importante es disponer de los datos y manejarlos de tres maneras básicas: obtener la información, almacenarla y recuperarla.

Una vez alcanzado lo básico, se puede operar con los datos, trasladarlos de un lugar a otro o modificarlos. A esto se le llama procesamiento de datos y básicamente consta de los mismos tres pasos que cambian la forma en que la información está organizada.

Un sistema de gestión de bases de datos relacional (SGBDR), como ORACLE ofrece una forma de realizar estas tareas de forma comprensible, razonable y sencilla. ORACLE básicamente permite:

- Introducir datos.
- Almacenar datos.
- Recuperar datos y trabajar con ellos.

La siguiente figura muestra lo sencillo que es esto.



ORACLE se basa en esta aproximación de introducir-almacenar-recuperar y proporcionar potentes herramientas que permiten considerables sofisticaciones en la manera en que se capturan los datos, se editan, se modifican y se introducen; en

la forma de almacenarlos seguros y en cómo se pueden recuperar, manipular y emitir informes con ellos.

6.4 CARACTERÍSTICAS DE ORACLE

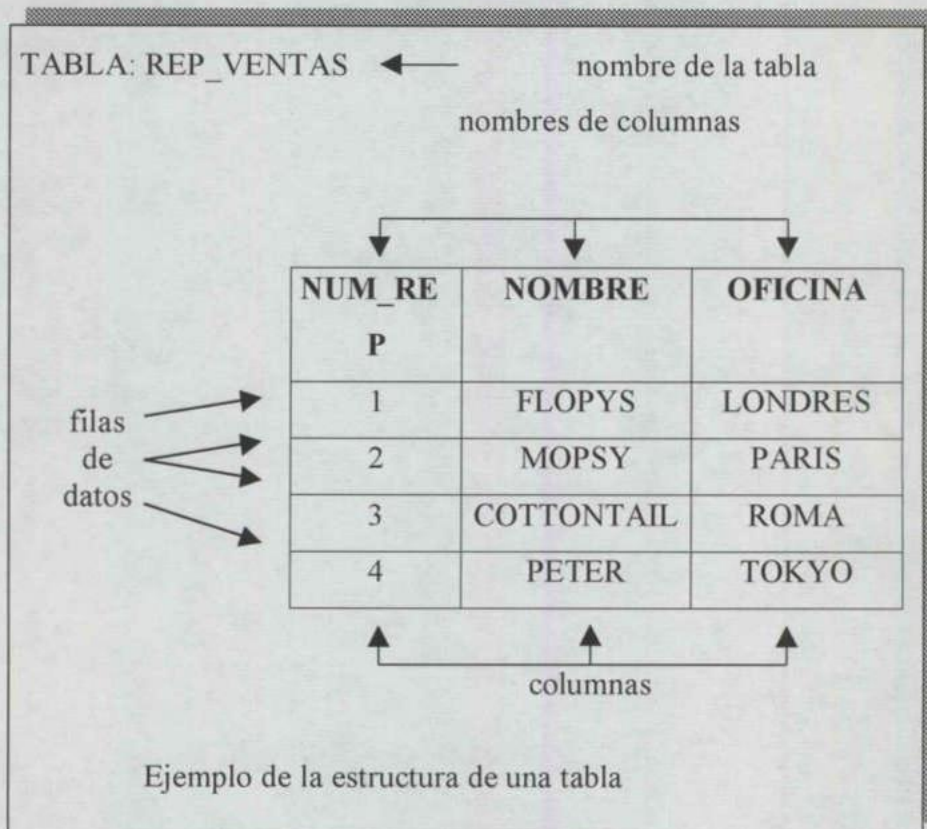
- Expansión y crecimiento
- Posibilidad de "partición" de datos
- Conexión Web
- Orientación a Objetos
- Convergencia
- Capacidad sofisticada para la toma de decisiones

7 ARQUITECTURA ORACLE

7.1 BASES DE DATOS

Oracle proporciona la capacidad de almacenar y acceder a los datos de forma consecuente con un modelo definido conocido como Modelo Racional. Por ello, Oracle se conoce como un sistema de gestión de bases de datos relacionales (RDBMS). La mayoría de las referencias a una base de datos no se refieren sólo a los datos físicos, sino también a la combinación de objetos físicos, de memoria y de proceso.

Los datos de una base de datos se almacenan en tablas. Las tablas relacionales se definen mediante sus columnas, y tienen un nombre. Los datos se almacenan como filas de la tabla. Las tablas pueden estar relacionadas entre ellas y puede utilizarse la base de datos para imponer estas relaciones.



Una base de datos Oracle almacena datos en archivos. Internamente existen estructuras de las bases de datos que proporcionan una asignación lógica de los datos con los archivos, lo que permite almacenar de forma separada diferentes tipos de datos. Estas divisiones lógicas se llaman espacios de tablas.

7.2 ESPACIOS DE TABLAS (TABLESPACES)

Cada base de datos tiene al menos un espacio de tablas (llamado espacio de tablas SYSTEM). Pueden utilizarse otros espacios de tablas para agrupar los usuarios o aplicaciones para facilitar el mantenimiento y mejorar el rendimiento, un espacio de tablas sólo puede pertenecer a una base de datos.

7.3 ARCHIVOS

Cada espacio de tablas consta de uno o más archivos, llamados archivos de datos, en un disco. Un archivo de datos sólo puede pertenecer a un único espacio de tablas. Los archivos de datos tienen un tamaño fijo en el momento de su creación; cuando se necesita más espacio hay que añadir archivos nuevos. Para crear nuevos espacios de tablas hace falta crear nuevos archivos de datos.

Así la división de los objetos de bases de datos entre varios espacios de tablas permite que dichos objetos se almacenen de forma física en archivos de datos separados, que pueden situarse en discos distintos. Esta es una herramienta importante en la planificación y ajuste de la manera en que la base de datos maneja las solicitudes de E/S que se le hacen.

7.4 INSTANCIAS

Para acceder a los datos de la base de datos, Oracle utiliza un conjunto de procesos de fondo compartidos por todos los usuarios. Además, existen ciertas estructuras de memoria que se utilizan para almacenar los datos de la base de datos más consultados recientemente. Estas áreas de memoria ayudan a mejorar el rendimiento de la base de datos al disminuir la cantidad de operaciones de E/S de acceso a los archivos de datos.

Una instancia de bases de datos (también conocida como un servidor) es un conjunto de estructuras de memoria y de procesos de fondo que acceden a un conjunto de archivos de base de datos. Es posible que varias instancias accedan a una única base de datos (servidor en paralelo).

Los parámetros que determinan el tamaño y composición de una instancia se almacenan en un archivo llamado `INIT.ORA`; este archivo se lee durante el arranque de la instancia y el DBA puede modificarlo. Las modificaciones que se realicen en este archivo no surten efecto hasta el siguiente arranque que lo utilice.

8 ESTRUCTURAS INTERNAS DE LAS BASES DE DATOS

8.1 ESTRUCTURAS DE LAS BD

Las estructuras de las bases de datos de Oracle pueden dividirse en tres categorías:

1. Las internas a la base de datos (tablas, columnas, usuarios, esquemas, índices, vistas, sinónimos, privilegios, segmentos, enlaces, etc.)
2. Las internas a las áreas de memoria (incluidas las áreas de memoria compartida y los procesos).
3. Las externas a las bases de datos.

8.2 TABLAS Y COLUMNAS

Las tablas son el mecanismo de almacenamiento de los datos dentro de una base de datos Oracle y constan de un conjunto fijo de columnas, que describen los atributos de la entidad que se maneja mediante la tabla. Cada columna tiene un nombre y unas características específicas.

Las características de una columna se componen de dos partes: su tipo de datos y su longitud, donde los campos numéricos pueden tener precisión que termina la el número de cifras significativas del valor de este campo. Una especificación para una columna indica que presenta un total de nueve cifras con dos decimales, la precisión por omisión es de 38 cifras, que es además la precisión máxima.

Los tipos de datos disponibles en Oracle son los siguientes:

Tipo de Datos	Descripción
HAR	ampo de longitud fija, con una longitud máxima de 255 aracteres.
ARCHAR	déntico a VARCHAR2, aunque su funcionalidad puede cambiar en ersiones futuras. Por tanto, debe utilizarse este tipo de datos para lmacenar cadenas de caracteres de longitud variable.
ARCHAR2	ampo de longitud variable, con una longitud máxima de 2.000 aracteres.
ATE	ampo con una longitud de siete bytes que se utiliza para lmacenar todas las fechas. La hora se almacena como parte de la echa. En las consultas, la fecha aparece en el formato Día-Mes-ño.
UMBER	olumna numérica de longitud variable. Los valores permitidos son ero y los números positivos y negativos desde 1.0E-130 hasta .99...E125.
ONG	ampo de longitud variable, con una longitud máxima de 2 Gb.
AW	ampo de longitud variable que se utiliza para datos binarios, con na longitud máxima de 2.000 caracteres.
LONG RAW	Campo de longitud variable que se utiliza para datos binarios, con una longitud máxima de 2 Gb.
MLSLABEL	Sólo para Trusted Oracle. Este tipo de datos se utiliza entre dos y inco bytes por fila.

8.3 USUARIOS

Una cuenta de usuario no es una estructura física de la base de datos, pero sí que tiene relaciones con los objetos de las bases de datos: los usuarios son propietarios de los objetos de las bases de datos. El usuario SYS es propietario de las tablas del diccionario de datos (data dictionary tables); en ellas se almacena información

sobre el resto de las estructuras de la base de datos. El usuario SYSTEM es propietario de las vistas que permiten acceder a estas tablas del diccionario de datos, para que las utilicen el resto de los usuarios de la base de datos.

Los objetos que se crean en la base de datos como apoyo a las aplicaciones, se crean bajo cuentas de usuarios. Cada una de estas cuentas puede personalizarse para que utilice un espacio de tablas específico como su espacio de tablas por omisión.

Las cuentas de la base de datos pueden conectarse a una cuenta del sistema operativo, lo que permite a los usuarios acceder a la base de datos desde el sistema operativo sin tener que introducir contraseñas tanto para el sistema operativo como para la base de datos. Pueden entonces acceder a los objetos de su propiedad o a aquéllos a los que se les ha concedido el acceso.

8.4 ESQUEMAS

En Oracle, es posible crear usuarios que no tengan la capacidad de acceder a la base de datos. Dichas cuentas de usuario proporcionan un esquema que puede utilizarse para mantener un conjunto de objetos de base de datos separados de otros esquemas de los usuarios.

8.5 INDICES

En una base de datos relacional, la situación física de una fila es irrelevante. Para que sea posible encontrar los datos, cada fila de cada tabla se etiqueta con un ROWID (Identificador de Fila). Este ROWID le indica a la base de datos el lugar exacto en que se encuentra la fila (mediante el archivo, bloque dentro de ese archivo y fila dentro de ese bloque).

Un índice es una estructura de la base de datos que le permite al servidor localizar rápidamente una fila de una tabla. Existen dos tipos de índices: índices de grupo e índices de tabla.

Los índices contienen una lista de entradas, cada una de las cuales consta de un valor clave y de un ROWID. El valor clave es el valor de una columna de una fila o la combinación de los valores de varias columnas en una fila.

Las entradas de los índices de tabla y de grupo de una base de datos Oracle se almacenan mediante un mecanismo de B*-tree (árbol binario), lo que garantiza un trayecto de acceso corto hasta el valor clave. Las operaciones de E/S necesarias para localizar un valor clave son mínimas, y, una vez encontrado, se utiliza el ROWID para acceder directamente a una fila. Los índices se utilizan para mejorar el rendimiento y asegurar la exclusividad de una columna, pueden crearse para una o varias columnas de una tabla.

8.6 GRUPOS

Las tablas a las que se suele acceder conjuntamente pueden almacenarse físicamente juntas. Para almacenarlas juntas, se crea un grupo (cluster) que contenga las tablas.

Los datos de las tablas se almacenan entonces juntos para minimizar el número de operaciones de E/S que deben realizarse y mejorar así el rendimiento.

Las columnas relacionadas de las tablas se denominan la clave del grupo. Esta clave del grupo se indexa mediante un índice de grupo, y su valor sólo se almacena una vez para las diversas tablas del grupo.

8.7 GRUPOS HASH

Los grupos hash (hash clusters) utilizan funciones de hash (hashing functions) sobre la clave del grupo de la fila para determinar la posición física en la que se debe almacenar la fila.

8.8 VISTAS

Una vista (view) tiene el mismo aspecto de una tabla con columnas y se consulta de la misma forma que ésta. Conceptualmente, puede pensarse en una vista como una máscara que cubre una o más tablas, de forma que las columnas de la vista se encuentran en una o más de las tablas subyacentes. Así las vistas no almacenan físicamente los datos sino en el diccionario de datos.

Cuando se consulta una vista, ésta consulta a las tablas en que se basa, y devuelve los valores en el formato y orden especificados en la definición de la vista. Las vistas no pueden indexarse ya que no hay datos físicos directamente asociados a ellas.

8.9 SECUENCIAS

Las definiciones de las secuencias (sequences) también se almacenan en el diccionario de datos.

Las secuencias proporcionan una lista consecutiva de números exclusivos que sirven para simplificar las tareas de programación.

La primera vez que se llama a una secuencia en una consulta, devuelve un valor predeterminado. En cada consulta subsiguiente a la secuencia se obtendrá un valor aumentado en el incremento especificado. Las secuencias pueden ser cíclicas o pueden seguir creciendo hasta alcanzar un valor máximo especificado.

8.10 PROCEDIMIENTOS

Los procedimientos permiten almacenar dentro de la base de datos relaciones de las aplicaciones que se utilicen con frecuencia. Cuando se ejecuta el procedimiento, sus sentencias se ejecutan con la unidad, los procedimientos no devuelven ningún valor al programa que los llama.

Los procedimientos almacenados pueden ayudar a reforzar la seguridad de los datos, lo que se consigue no concediendo a los usuarios el acceso directo a las tablas de una aplicación, sino sólo la posibilidad de ejecutar un procedimiento de acceso a las tablas; cuando se ejecuta el procedimiento, lo hará con los privilegios de su propietario, los usuarios no podrán acceder a las tablas a no ser por medio del procedimiento.

8.11 FUNCIONES

Las funciones (functions), lo mismo que los procedimientos, son bloques de código que se almacenan en la base de datos. A diferencia de éstos, las funciones pueden devolver valores al programa que las llama.

8.12 PAQUETES

Los paquetes (packages) sirven para organizar los procedimientos y las funciones en agrupamientos lógicos. Los paquetes son muy útiles en las tareas administrativas necesarias para gestionar los procedimientos y las funciones.

Distintos elementos del paquete pueden definirse como <<públicos>> o como <<privados>>. Los elementos públicos están accesibles para el usuario del paquete, mientras que los elementos privados están ocultos. Entre los elementos privados suelen encontrarse procedimientos a los que llaman otros procedimientos del paquete.

8.13 DISPARADORES (TRIGGERS)

Los disparadores son procedimientos que se ejecutan cuando se produce un evento de base de datos especificado en una tabla especificada en una base de datos Oracle; pueden utilizarse para aumentar la integridad referencial, conseguir una seguridad adicional o mejorar las opciones de auditoría disponibles.

8.13.1 Tipos de disparadores:

- Disparadores de sentencia: se activan una vez por cada sentencia de disparo.
- Disparadores de fila: se activan una vez por cada fila de una tabla afectada por las sentencias.

Para cada uno de ellos puede crearse un disparador BEFORE (ANTES) y otro AFTER (DESPUES) para cada tipo de evento de disparo. Entre los eventos de disparo se encuentran las operaciones INSERT (Insertar), UPDATE (Actualizar) y DELETE (Eliminar).

Los disparadores de sentencia resultan útiles si el código de la acción de disparo no cuenta con los datos afectados.

Los disparadores de fila son útiles si la acción de disparo cuenta con los datos afectados por la transacción.

8.14 SINÓNIMOS

Para realizar una identificación completa de un objeto de base de datos (como una tabla o una vista) es una base de datos distribuida Oracle, es necesario especificar el nombre de la máquina anfitriona, el nombre del servidor, el propietario del objeto y el nombre del objeto.

Serán necesarios entre uno y cuatro de estos parámetros en función del emplazamiento del objeto. Los desarrolladores pueden crear sinónimos que

apunten al objeto adecuado para ocultar este proceso a los usuarios, que sólo tienen que conocer el nombre del sinónimo.

Los sinónimos públicos los comparten todos los usuarios de una base de datos concreta, mientras que los sinónimos privados pertenecer a propietarios de cuentas de base de datos individuales.

Los sinónimos permiten proporcionar punteros a tablas, vistas, procedimientos, funciones, paquetes y secuencias. Pueden apuntar a objetos de la base de datos local o de base de datos remotas. El apuntar a bases de datos remotas se consigue mediante la utilización de enlaces de datos.

8.15 PRIVILEGIOS Y FUNCIONES (ROLES)

Para que una cuenta pueda acceder a un objeto propiedad de otra cuenta tiene que habersele concedido primero el privilegio (privilege) de acceso. Lo habitual es que a los no propietarios se les conceda el privilegio de INSERT (Insertar), SELECT (Seleccionar), UPDATE (Actualizar) o DELETE (Ejecutar) filas de una tabla o vista.

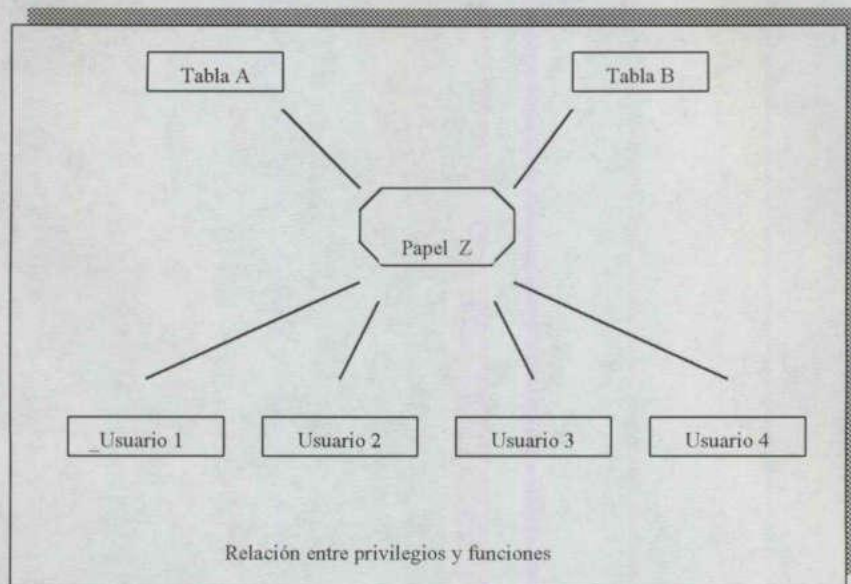
También pueden concederse privilegios para SELECT valores de secuencias o EXECUTE (Ejecutar) procedimientos y funciones. No se conceden privilegios sobre los índices o los disparadores, ya que la base de datos los utiliza durante la actividad de la tabla.

Los privilegios pueden concederse a usuarios individuales o a PUBLIC (Público), que concede los privilegios a todos los usuarios de la base de datos.

Los privilegios pueden concederse a una función (role), y la función concederse a varios usuarios.

De esta manera, el hecho de añadir nuevos usuarios a las aplicaciones se simplifica en gran medida, ya que basta con conceder o retirar funciones al usuario.

En la siguiente figura se muestra la relación entre privilegios y funciones



8.16 ENLACES DE BASES DE DATOS

Las bases de datos Oracle que utilizan SQL*Net tienen la capacidad de hacer referencia a datos almacenados fuera de la base de datos local. En estos casos debe especificarse el nombre completo del objeto remoto.

8.17 ESTRUCTURAS INTERNAS DE MEMORIA

La base de datos Oracle utiliza dos tipos distintos de estructuras de memoria: áreas globales y áreas de proceso.

La implementación de las opciones de memoria disponibles presenta un amplio abanico de posibilidades en función de la opción del servidor de bases de datos que se utilice.

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

9 EL LENGUAJE FAMILIAR DE ORACLE

9.1 EJEMPLO

La información almacenada en ORACLE se guarda en tablas, de la misma forma que la información del tiempo de un periódico como vemos en el siguiente ejemplo:

CLIMA

CIUDAD	TEMPERATURA	HUMEDAD	ESTADO
Atenas	97	89	Soleado
Chicago	66	88	Lluvia
Lima	45	79	Lluvia
Manchester	66	98	Niebla
París	81	62	Nublado
Esparta	74	63	Nublado
Sydney	29	12	Nieve

Tabla de tiempo de un periódico

Esta tabla tiene cuatro columnas verticales: Ciudad, Temperatura, Humedad y Estado. También posee una fila por cada una de las ciudades. Por último, la tabla tiene un nombre: CLIMA.

Estas son las tres características principales de las tablas: *filas*, *columnas* y *nombre*. Eso también es válido en una base de datos relacional.

Cualquiera puede comprender las palabras y las ideas que representan, ya que las palabras que se usan con ORACLE para describir las tablas que forman la base de datos son las mismas que se usan normalmente. No poseen ningún significado especial, inusual o esotérico. Lo que se ve es lo que es.

9.2 TABLAS DE INFORMACION

Oracle almacena la información en tablas, tal como se muestra a continuación. Cada una de estas tablas tiene una o más columnas. Las cabeceras de las columnas, tales como Ciudad, Temperatura, humedad y Estado, describen el tipo de información que se guarda en la columna. La información se almacena fila por fila, ciudad por ciudad. Cada conjunto de datos únicos, como es la temperatura, la humedad y el estado del tiempo de la ciudad de Atenas, se encuentra en su propia fila.

CLIMA			
<u>Ciudad</u>	<u>Temperatura</u>	<u>Humedad</u>	<u>Estado</u>
Atenas	97	89	Soleado
Chicago	66	88	Soleado
Lima	45	79	Nieve
Manchester	66	98	Nublado
París	81	62	Lluvia
Esparta	74	63	Nieve
Sydney	29	12	Soleado

Tabla del tiempo en ORACLE

ORACLE evita terminología académica especializada para lograr un producto mucho más comprensible. En la teoría relacional a una columna se le llama "atributo", a una fila se le llama "tupla" y a una tabla se le llama "entidad".

Pero para un usuario estos términos son confusos. Fundamentalmente no es más que haber renombrado de forma innecesaria cosas cuyos nombres normales se entienden y se utilizan en el lenguaje diario, y los diseñadores también deberían aprovecharse de él.

Es necesario reconocer la barrera de equivocaciones y desconfianza que produce la utilización de una jerga técnica.

9.3 LENGUAJE DE CONSULTA ESTRUCTURADO

Oracle fue el primer fabricante que ofreció un producto que usaba el lenguaje de consulta estructurado, SQL, basado en el inglés. Esto permite a los usuarios finales extraer por sí mismos sin tener que recurrir al grupo de sistemas para cada pequeño informe.

El lenguaje de consultas ORACLE es un lenguaje estructurado, más que el inglés hablado o que cualquier otro lenguaje. Tiene reglas de gramática y de sintaxis.

9.4 EJEMPLO DE UNA CONSULTA SIMPLE CON ORACLE

Si ORACLE tuviera en su base de datos la tabla CLIMA, la primera pregunta sería tan sencilla como:

```
Select Ciudad from CLIMA where Humedad = 89
```

ORACLE respondería:

```
Ciudad  
-----
```

```
Atenas
```

La segunda pregunta sería:

Select Ciudad from CLIMA where Temperatura = 66

A esta pregunta ORACLE respondería:

```
Ciudad
-----
Manchester
Chicago
```

Para utilizar order by (ordenar según), hacemos la suposición de que se quieren ver todas las ciudades ordenadas según su temperatura. Simplemente habría que escribir:

```
Select Ciudad, Temperatura from CLIMA
Order by Temperatura
```

Y ORACLE instantáneamente respondería:

Ciudad	Temperatura
-----	-----
Sydney	29
Lima	45
Manchester	66
Chicago	66
Esparta	74
París	81
Atenas	97

ORACLE ha reordenado la tabla rápidamente por temperaturas. (Esta tabla presenta primero las temperaturas más bajas.

Existen muchas otras preguntas que se pueden plantear con la facilidad de consultas de ORACLE, pero estos ejemplos sirven para mostrar lo sencillo que es obtener la información que se necesita de la base de datos de ORACLE de la forma que sea más útil. Se pueden construir complicadas preguntas a partir de los elementos sencillos, pero el método usado para hacerlo siempre será comprensible.

Por ejemplo, se pueden combinar **where** (donde) y **order by** (ordenado según) de una manera sencilla, e indicarle a ORACLE que seleccione las ciudades que tengan una temperatura mayor de 80 y que las muestre en orden creciente de temperatura.

```
Select Ciudad, Temperatura from CLIMA
      Where Temperatura > 80
      Order by Temperatura
```

Y ORACLE respondería inmediatamente:

Ciudad	Temperatura
París	81
Atenas	97

9.5 ORIGEN DE LA DENOMINACION "RELACIONAL"

Siguiendo el mismo ejemplo, vemos que en la tabla CLIMA hay ciudades de varios países y que varios países tienen más de una ciudad en la lista.

Supongamos que se necesita saber en qué país se encuentra una determinada ciudad. Se podría crear una tabla aparte llamada LOCALIZACION con las ciudades y sus países, como se muestra en la siguiente figura.

CLIMA				LOCALIZACION	
<i>Ciudad</i>	<i>Temp.</i>	<i>Humedad</i>	<i>Estado</i>	<i>Ciudad</i>	<i>País</i>
Atenas	97	89	Soleado	Atenas	Grecia
Chicago	66	88	Lluvia	Chicago	E.U.
Lima	45	79	Lluvia	Conakry	Guinea
Manchester	66	98	Niebla	Madras	India
París	81	62	Nublado	Madrid	España
Esparta	74	63	Nublado	Manchester	Inglaterra
Sydney	29	12	Nieve	Moscú	Rusia
				París	Francia
				Roma	Italia
				Shenyang	China
				Esparta	Grecia
				Sydney	Australia
				Tokyo	Japón

TABLA DE CLIMA Y LOCALIZACION

Para cualquier ciudad de la tabla CLIMA, basta con buscar en la tabla LOCALIZACION el nombre de la columna Ciudad, mirar en la columna país de la misma fila y se tiene el nombre del país.

Estas dos tablas son completamente independientes y separadas. Cada una contiene su propia información en filas y columnas; y además tienen en común la columna Ciudad. Para cada nombre de ciudad de la tabla CLIMA hay un nombre de ciudad idéntico en la tabla LOCALIZACION.

10 LOS PELIGROS DE LAS BASES DE DATOS RELACIONALES

10.1 PANORAMA GENERAL

Como con cualquier nueva tecnología o empresa, conviene analizar no sólo los beneficios y oportunidades que se presentan, sino también los costes y riesgos. En una tecnología relativamente nueva como son las bases de datos relacionales, no ha pasado todavía el tiempo suficiente en la mayoría de las empresas de cara a disponer de “veteranos” que sepan lo que hay que evitar y cómo hay que evitarlo.

La combinación de bases de datos relacionales con herramientas poderosas y fáciles de usar, tal y como hace ORACLE, hace que la posibilidad de llevar al diseñador al desastre llegue a ser real. Es por tal motivo necesario cubrir estos problemas en mayor profundidad, especialmente aquellos que interesan a los diseñadores en la tarea de construir una aplicación productiva y conveniente.

10.2 ¿CUÁLES SON LOS RIESGOS?

El primer riesgo es que es tan fácil como dicen. No existe dificultad en comprender las tablas, las filas y las columnas. La relación entre dos tablas es conceptualmente sencilla. Incluso la *normalización*, que es el proceso de analizar las relaciones inherentes o “normales” entre los diferentes elementos de datos de una empresa, es francamente sencilla de aprender.

Desafortunadamente, esto produce expertos instantáneamente, llenos de confianza e ingenuidad, pero con poca experiencia en el diseño de aplicaciones relacionales

reales para un uso productivo. Para una pequeña base de datos de mercado, o para un pequeño inventario casero, esto no tiene mucha importancia.

Los errores se harán presentes por sí mismos en poco tiempo, y se aprenderá la lección para que puedan evitarse posteriormente. Sin embargo, en una aplicación importante ésta es una fórmula que asegura el desastre. Esta falta de experiencia es la que está normalmente detrás de las historias de la prensa que cuenten fracasos de importantes proyectos.

Los métodos de desarrollo más antiguos son normalmente más lentos. Hay que imponer métodos de control para revisar y asegurar la calidad, pero las tareas de los viejos métodos (codificar, cometer un trabajo a compilación, enlazar y comprobar) resultan lentas. El ciclo es a menudo tan tedioso, particularmente sobre un computador mainframe, que los programadores se pasan una gran cantidad de tiempo haciendo comprobaciones sobre el papel para evitar el retraso de otro ciclo completo debido a un error de código.

Las herramientas de cuarta generación seducen a los diseñadores a “lanzarse a la producción”. Los cambios se pueden hacer e implementar tan rápidamente que las pruebas alcanzan rápidamente la absolución. Esto se completa con la virtual eliminación de todas las pruebas sobre el papel (el largo ciclo) ha desaparecido, y las pruebas con él.

El ciclo de prueba en un gran proyecto de ORACLE debería ser más largo y más completo que el de un proyecto tradicional. Esto es cierto, incluso si los controles de proyecto se producen en el tiempo apropiado, e incluso el proyecto está guiado por diseñadores experimentados, debido a que tendrá menos pruebas sobre el papel y un exceso de confianza inherente. Esta prueba debe comprobar la exactitud de las pantallas de entrada de datos, y de los informes, de la carga de datos y de su actualización, de la integridad y concurrencia de los datos y,

particularmente, de las transacciones y volúmenes de almacenamiento durante los picos de carga.

Como esto es tan sencillo, el desarrollo de aplicaciones con herramientas ORACLE puede ser sorprendentemente rápido. Sin embargo, esto reduce automáticamente la cantidad de comprobaciones que se hacen en ser incrementados conscientemente la cantidad de comprobaciones que se hacen en un desarrollo normal, y las pruebas planeadas y los controles de calidad deben ser incrementados conscientemente para compensar. Esto no es previsto normalmente por los diseñadores de ORACLE o de las herramientas de la cuarta generación, pero se debería incluir el tiempo y presupuesto correspondiente en los planes de los proyectos.

11 USO DE CODIGOS FRENTE AL LENGUAJE

11.1 LAS PRINCIPALES JUSTIFICACIONES DEL USO DE CÓDIGOS

- Una categoría tiene artículos que ni se pueden representar o recordar todos de una manera razonable.
- Ahorrar espacio en la computadora.

El segundo punto es una anacronía, un residuo de los tiempos en que 4 MIPS costaban 1,000 dólares por hora (y más). La memoria (diseñada en ferrita y montada sobre parrillas de hilos) y el almacenamiento permanente (cintas o grandes tambores magnéticos) eran demasiado caros, y las CPU tan lentas (con menos potencia que una calculadora de mano) que los programadores tenían que reducir cualquier tipo de información lo máximo posible.

Los números, carácter a carácter, ocupan la mitad de espacio de almacenamiento que las letras, y los códigos reducen la ocupación de la memoria mucho más. Debido a que las máquinas eran caras, los diseñadores tenían que usar códigos para todo si querían que algo funcionara. Esta era una solución técnica para un problema económico

Existe un beneficio adicional inmediato: los errores de entrada de claves se reducen prácticamente a cero, debido a que el usuario ve inmediatamente, en su idioma, la información que está introduciendo. Los dígitos no se intercambian, los códigos no se recordarán incorrectamente, y en aplicaciones financieras el dinero rara vez se pierde en cuentas poco claras debido a errores de entrada, suponiendo un ahorro importante.

Para reducir la confusión, se cuenta con otra justificación del argumento de “demasiados elementos por categoría” y es que el número de productos, clientes o tipos de transacción son demasiado grandes para diferenciar cada nombre, o que haya demasiados elementos en una categoría que son iguales o muy parecidos.

Una categoría puede tener demasiadas entradas para hacer las opciones fáciles de recordar o diferenciar, pero muy a menudo esto se debe a un trabajo incompleto de clasificación de información: demasiados objetos diferentes se meten en una categoría demasiado amplia. Desarrollar una aplicación con una fuerte orientación basada en inglés, como oposición al desarrollo basado en código, requiere gastar tiempo con los diferentes diseñadores y usuarios, tomando información del problema, comprendiendo las relaciones naturales y las categorías y construyendo cuidadosamente una base de datos y un esquema de nombres que refleje simplemente y con exactitud estos hallazgos.

Existen tres etapas básicas para conseguir esto:

1. normalizar los datos.
2. Escoger los nombres significativos para las tablas y columnas.
3. Escoger los nombres de los datos.

11.2 NORMALIZACION

Las relaciones entre los países o departamentos de una compañía, o entre usuarios y diseñadores son el resultado de circunstancias históricas particulares, que pueden definir las actuales relaciones aun cuando las circunstancias iniciales hayan pasado ya. El efecto producido son a menudo relaciones anormales, o en un lenguaje más asequible, disfuncionales. La historia y las circunstancias tienen a menudo los mismos efectos sobre los datos: cómo son recogidos, organizados y presentados. Por esto, los datos también pueden llegar a ser anormales y disfuncionales.

La normalización es un proceso que pone las cosas en su sitio, haciéndolas normales. El significado en una base de datos, es matemático, realizando una separación de elementos de datos (tales como nombres, direcciones u oficios) en grupos afines y definiendo las relaciones normales o “correctas” entre ello.

Los conceptos básicos de normalización se introducen aquí para que los usuarios puedan contribuir a diseñar una aplicación que van a usar posteriormente, o para comprender una aplicación que ya ha sido construida. Sin embargo, es un error pensar que este proceso sólo es aplicable al diseño de bases de datos o aplicaciones de computadoras. La normalización entra con profundidad en la información usada en un negocio y en cómo están relacionados entre sí los diferentes elementos de información. Esto resultará útil en áreas diferentes de las bases de datos y de las computadoras.

11.3 EL METODO LOGICO

Un primer paso en el proceso de análisis es la construcción de un *modelo lógico*, que es implemente un diagrama normalizado de datos usados en la empresa. Conocer el cómo y el porqué los datos se separan y segregan es fundamental para comprender el modelo, y este modelo es fundamental para construir la aplicación que usará la empresa por mucho tiempo, sin necesitar un soporte extraordinario.

La normalización se discute habitualmente en términos de normas: la primera, la segunda y la tercera forma normal, son las más comunes, donde la tercera representa el estado más normalizado.

11.4 PRIMERA FORMA NORMAL

El primer paso es poner los datos en la primera forma normal. Esto se hace poniendo los datos en tablas separadas, de manera que los datos de cada tabla sean

de un tipo similar y dando a cada tabla una *clave primaria* y un identificador o etiqueta única. Esto elimina los grupos repetidos de datos.

Lo siguiente que se debe hacer es definir una clave primaria para cada tabla: aquella que identificará de manera única y servirá para obtener una fila de información.

11.5 SEGUNDA FORMA NORMAL

El segundo paso, que es la segunda forma normal, se centra en sacar los datos que sólo dependen de una parte de la clave. Para obtener la segunda forma normal se deben sacar las columnas de la tabla donde se encuentra la primera forma normal.

11.6 TERCERA FORMA NORMAL

El tercer paso, la tercera forma normal, implica deshacerse de cualquier cosa de las tablas que no dependan únicamente de la clave primaria. Una vez que los datos están en la tercera forma normal, están ya automáticamente en la primera y en la segunda forma normal. Por tanto el proceso total se puede acabar de una forma menos tediosa usando directamente la tercera forma normal, que haciéndolo forma a forma, basta con organizar los datos para que las columnas de cada tabla, aparte de la clave primaria, dependan únicamente de toda la clave primaria. La tercera forma normal se describe algunas veces como “la clave, toda la clave y nada más que la clave”.

11.7 CACHÉ DEL DICCIONARIO

La información sobre los objetos de la base de datos se almacena en las tablas del diccionario de datos. En esta información se incluyen los datos de las cuentas de usuario, los nombres de los archivos de datos, los nombres de los segmentos, las ubicaciones de las extensiones, las descripciones de las tablas y los privilegios.

Cuando la base de datos necesita esta información se leen las tablas del diccionario de datos y los datos se almacenan en el SGA, en el caché del diccionario (dictionary cache).

Este caché también se gestiona mediante un algoritmo LRU. En Oracle la base de datos es la encargada de gestionar internamente el tamaño del caché del diccionario; forma parte del fondo común SQL compartido, cuyo tamaño se establece mediante el parámetro `SHARED_POOL_SIZE` del archivo `INIT.ORA` de la base de datos.

Si el caché del diccionario es demasiado pequeño, la base de datos tendrá que consultar reiteradamente la información que necesita en las tablas del diccionario de datos.

Estas consultas se llaman *aciertos recursivos*, y se resuelven más lentamente que las consultas que pueden manejarse sólo mediante el diccionario que está en memoria.

11.8 BÚFER DE REGISTRO REHACER

En las entradas de rehacer se describen los cambios realizados en la base de datos, se escriben en los archivos de registro de rehacer en línea de forma que puedan utilizarse en las operaciones de reconstrucción durante las recuperaciones de las bases de datos; sin embargo, antes de escribirse en los archivos de registro de rehacer en línea, se almacenan primero en el SGA, en un área llamada *búfer de registro de rehacer*.

La base de datos escribe después de forma periódica series de entradas de rehacer a los archivos de registro de rehacer en línea, optimizando así operación. El

tamaño (en bytes) de los búferes de registro de rehacer se define mediante el parámetro LOG_BUFFER del archivo INIT.ORA.

11.9 FONDO COMÚN SQL COMPARTIDO

En las bases de datos Oracle, el caché del diccionario se almacena en un área denominada *fondo común SQL compartido (shared SQL pool)*. En esta área de memoria también se incluye información sobre sentencias ejecutadas sobre la base de datos. Así, mientras que el búfer del bloque de datos y el caché del diccionario permiten la compartición estructural y de los datos entre los usuarios, el fondo común SQL compartido permite compartir las sentencias SQL más utilizadas.

El fondo común SQL compartido contiene el plan de ejecución y el árbol de análisis de las sentencias SQL que se ejecutan sobre la base de datos. La segunda vez que un usuario ejecuta una sentencia SQL idéntica, puede aprovecharse la información de análisis disponible en el fondo común SQL compartido para acelerar su ejecución.

11.10 AREAS DE CONTEXTO

Dentro del área SQL compartida de Oracle existen tanto áreas públicas como privadas, todas las sentencias SQL emitidas por un usuario necesitan un área SQL privada, que sigue existiendo hasta que se cierra el cursor correspondiente a la sentencia.

11.11 AREA GLOBAL DEL PROGRAMA (Program Global Area: PGA)

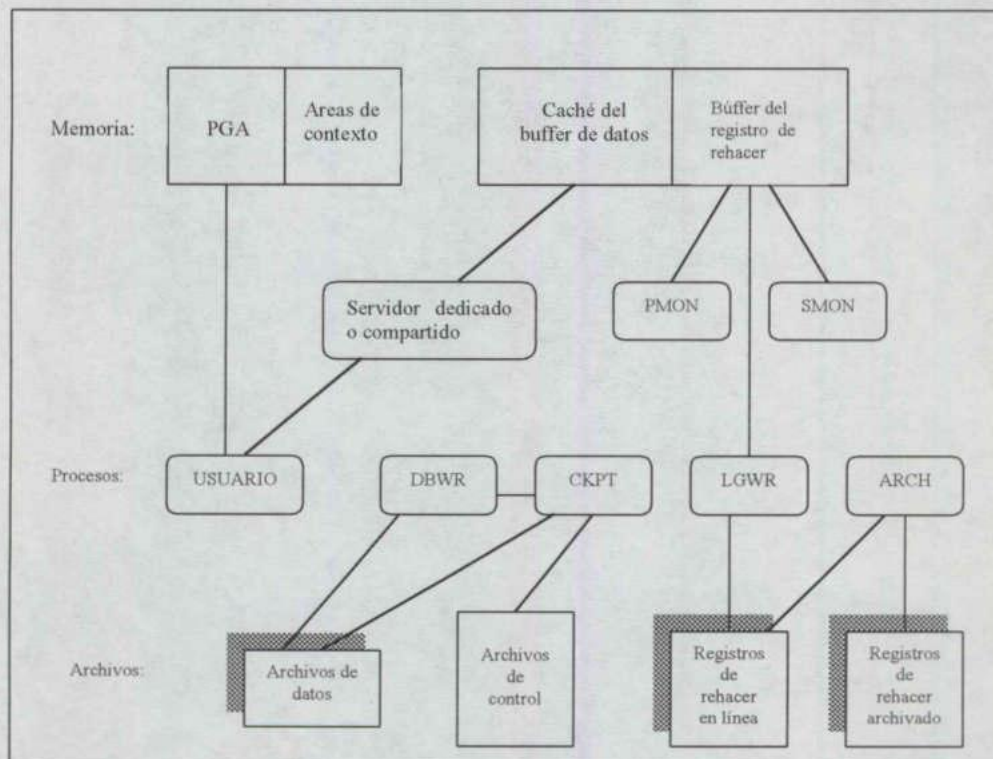
El área global del programa (PGA) es un área de la memoria utilizada por un único proceso de usuarios de Oracle, dependiendo de la versión puede contener áreas de contexto de usuario, además de la información del control de procesos. La memoria del PGA no se comparte.

Parte del PGA puede almacenarse en el SGA; esto ocurre cuando se utiliza el servidor multiproceso. La arquitectura multiproceso del servidor permite que varios procesos de usuario utilicen el mismo proceso del servidor, reduciendo así los requisitos de memoria de la base de datos. Si se utiliza esta opción, la información sobre la sesión del usuario se almacena en el SGA en lugar de en el PGA.

12 ESTRUCTURAS DE LOS PROCESOS

12.1 PROCESOS

Las relaciones entre las estructuras físicas y de la memoria de la base de datos se mantienen y aplican mediante *estructuras de proceso*, que son los procesos de fondo propios de la base de datos; estos procesos los gestiona la base de datos y apenas necesitan trabajo administrativo. En la siguiente figura se muestran las relaciones existentes entre las estructuras físicas y las estructuras de la memoria en la base de datos junto con los procesos principales.



Estructura física, de la memoria y de los procesos en Oracle.

12.2 SUPERVISOR DEL SISTEMA: SMON

El proceso de fondo SMON sirve para las mismas funciones de supervisión. Cuando se arranca la base de datos, SMON realiza la recuperación de la instancia según se necesite (utilizando para ello los archivos de registro de rehacer en línea). También limpia la base de datos, eliminando los objetos de las transacciones que el sistema ya no necesita. SMON tiene una función adicional: agrupa extensiones libres contiguas en extensiones libres mayores.

12.3 SUPERVISOR DEL PROCESO: PMON

El proceso de fondo PMON realiza una limpieza al terminar la ejecución de los procesos; PMON libera los recursos que estuviera utilizando el usuario, sus efectos se hacen patentes cuando se elimina un proceso bloqueado: PMON es responsable de liberar el bloqueo y hacer que esté disponible para otros usuarios.

Al igual que SMON, PMON se activa de forma periódica para comprobar si es necesaria su intervención.

12.4 ESCRITOR DE BASES DE DATOS: DBWR

El proceso de fondo DBWR es responsable de gestionar el contenido del caché del búfer del bloque de datos y del caché del diccionario. Lee los bloques de los archivos de datos y los almacena en el SGA. Realiza series de escrituras de los bloques modificados de vuelta en los archivos de datos.

El ajuste de las actividades del proceso DBWR (Eficiencia de las lecturas de los archivos de datos, Eficiencias de las lecturas de la memoria) es una buena parte del trabajo de ajuste que deben realizar los DBA (Administrador de bases de datos).

Aunque solo existe un proceso SMON y otro PMON en ejecución para cada instancia de base de datos, es posible tener varios procesos DBWR ejecutándose a la vez, en función de la plataforma y del sistema operativo.

La utilización de varios procesos DBWR ayuda a minimizar la contienda dentro del DBWR durante consultas largas que afectan a los archivos de datos; el número de procesos DBWR en ejecución se configura mediante el parámetro DB_WRITERS del archivo INIT.ORA de la base de datos.

12.5 ESCRITOR DE REGISTROS: LGWR

El proceso de fondo LGWR gestiona la escritura del contenido del búfer del registro de rehacer a los archivos del registro de rehacer en línea. Dicha escritura la realiza por series; las entradas del búfer de registro de rehacer siempre contienen el estado más actualizado de la base de datos, ya que el proceso DBWR debe esperar antes de escribir los bloques modificados desde los búferes del bloque de datos a los archivos de datos.

Se observa que el LGWR es el único proceso que escribe en los archivos de datos de rehacer en línea y el único que lee directamente los búferes de registro de rehacer durante el funcionamiento normal de la base de datos. En los archivos del registro de rehacer en línea están replicados, LGWR escribe simultáneamente en los conjuntos replicados de registros.

12.6 PUNTO DE CONTROL: CKPT

Los puntos de control (checkpoints) ayudan a reducir la cantidad de tiempo necesaria para realizar la recuperación de las instancias; los puntos de control provocan que el DBWR escriba en los archivos de datos todos los bloques que se hayan modificado desde el último punto de control y que actualice las cabeceras de los archivos de datos y los archivos de control para registrar el punto de control.

Se producen de forma automática cuando se llena un archivo de registro de rehacer en línea; puede utilizarse el parámetro `LOG_CHECKPOINT_INTERVAL` del archivo `INIT.ORA` de la instancia de la base de datos para configurar el punto de control más frecuentemente.

Puede crearse un proceso de fondo adicional, *CKPT* para separar las dos funciones de *LGWR* (Señalar puntos de control y copiar la entrada de rehacer) en dos procesos de fondo. El proceso de fondo *CKPT* se habilita definiendo como `TRUE` (Verdadero) el parámetro `CHECKPOINT_PROCESS` del archivo `INIT.ORA` de la instancia de la base de datos. Este proceso no es necesario a menos que la base de datos experimente un elevado volumen de transacciones, lo que provoca retardos en las conmutaciones de registro.

12.7 ARCHIVADOR : ARCH

El proceso de fondo *LGWR* escribe en los archivos de registro de rehacer en línea de forma cíclica; después de rellenar el primer archivo empieza a escribir en el segundo, hasta que lo rellena, y empieza entonces a escribir en el tercero. Cuando se rellena el último archivo de registro de rehacer en línea, *LGWR* empieza a sobrescribir el contenido del primero.

Cuando Oracle se ejecuta en modo `ARCHIVELOG`, la base de datos realiza una copia de cada uno de los archivos de registro de rehacer antes de sobrescribirlo. Dichos archivos suelen escribirse en un dispositivo de disco, aunque también pueden escribirse directamente en un dispositivo de cinta magnética, aunque esto último suele ocupar mucho tiempo de operador.

La función de archivado la realiza el proceso de fondo *ARCH*. Las bases de datos que utilizan esta opción experimentarán problemas de contienda en su disco de registro de rehacer durante los periodos con altas tasas de transacción de datos, ya

que LGWR intentará escribir en un archivo de registro de rehacer mientras ARCH intenta leer de otro. También pueden producirse bloqueos en la base de datos si se llena el disco de destino de registro del archivo.

En este caso, ARCH se detiene, lo que impide que LGWR siga escribiendo, lo que a su vez impide que se produzcan más transacciones en la base de datos hasta que se libere espacio para los archivos de registro de rehacer archivados.

12.8 RECUPERADOR (OPCIÓN DISTRIBUIDA): RECO

El proceso de fondo RECO sirve para corregir fallos en las bases de datos distribuidas de Oracle. Este proceso intenta acceder a las bases de datos implicadas en transacciones dudosas y resolver dichas transacciones; Este proceso sólo se crea si la plataforma de trabajo admite la opción distribuida de Oracle y el parámetro `DISTRIBUTED_TRANSACTIONS` del archivo `INIT.ORA` está definido con un valor superior a cero.

12.9 BLOQUEO (SERVIDOR PARALELO): LCKn

Existen varios procesos LCK, denominados LCK0 a LCK9, que sirve para el bloqueo entre instancias cuando se utilizan las opciones de Servidor Paralelo de Oracle, el número de procesos LCK se define por medio del parámetro `MI_BG_PROCS` del archivo `INIT.ORA` de la base de datos o bien mediante el parámetro `GC_LCK_PROCS`.

12.10 DESPACHADOR (ORACLE, SQLNET V): Dnnn

Los procesos *despachadores* forman parte de la arquitectura de servidor multiproceso (MTS) de SQL*Net V2; al manejar varias conexiones ayudan a minimizar las necesidades de recursos.

Debe crearse al menos un proceso despachador por cada protocolo que admita el servidor de la base de datos. Los procesos despachadores se crean durante el arranque de la base de datos, basándose en la configuración de SQL*Net, y pueden crearse o eliminarse mientras la base de datos permanezca abierta.

12.11 SERVIDOR: Snnn

Los *procesos del servidor* se crean para gestionar las conexiones con la base de datos que necesiten un servidor dedicado. Son típicos en las conexiones anteriores a SQL*Net V2.

12.12 ESTRUCTURAS EXTERNAS

Los archivos de datos de la base de datos, proporcionan el almacenamiento físico de los datos de la base de datos. Por tanto, son estructuras tanto <<internas>>, por estar ligadas directamente a los espacios de tablas, como <<externas>>, por tratarse de archivos físicos.

Los siguientes tipos de archivos, aunque están relacionados con la base de datos, están separados de los archivos de datos:

- ◆ Registros de rehacer
- ◆ Archivos de control

12.13 REGISTROS DE REHACER

Oracle conserva registros de todas las transacciones que afectan a la base de datos; estas transacciones se registran en un archivo denominado *archivo de registro de rehacer en línea*, y se utilizan para recuperar las transacciones de la base de datos en el orden adecuado en caso de que produzca un fallo de la base de datos. La información de registro de rehacer se almacena externamente a los archivos de datos de la base de datos; los archivos de registro de rehacer también permiten que Oracle coordine la forma en que escribe los datos al disco.

Cuando se produce una transacción en la base de datos, se introduce en los búferes de registro de rehacer, mientras que los bloques de datos afectados por la transacción no se escriben de manera inmediata al disco; esto permite que la base de datos realice series de escrituras al disco, optimizando así el rendimiento de esta función.

Todas las bases de datos Oracle tienen dos o más archivos de registro de rehacer en línea, Oracle escribe en ellos de manera cíclica: después de rellenar el primer archivo de registro escribe en el segundo, hasta rellenoarlo. Cuando se rellenan todos los archivos de registro de rehacer en línea, vuelve al primero y empieza a sobrescribir su contenido con nuevos datos de transacciones. Si la base de datos está trabajando en modo ARCHIVELOG, se hará una copia de los archivos de registro de rehacer en línea antes de sobrescribirlos. Esta copia puede utilizarse para recuperar parte de la base de datos al estado que tuviera en cualquier instante.

Los archivos de registro de rehacer pueden replicarse dentro de la base de datos, lo que permite que el DBA replique los archivos de registro de rehacer sin confiar en el sistema operativo o en las posibilidades hardware del entorno operativo.

12.14 ARCHIVOS DE CONTROL

La arquitectura física general de una base de datos se mantiene por medio de sus archivos de control, en los que se registra la información de control sobre todos

los archivos de la base de datos. Se utilizan para conservar la consistencia interna y servir de guía en las operaciones de recuperación.

Como los archivos de control son fundamentales para las bases de datos, se guardan varias copias en línea; estos archivos suelen almacenarse en discos separados para minimizar las consecuencias perjudiciales de un posible fallo del disco.

La base de datos creará y mantendrá los archivos de control especificados en los guiones de creación de una base de datos.

13 IMPLEMENTACION BASICA DE LA BASE DE DATOS

En su forma más elemental, una base de datos Oracle consta de:

1. Uno o más archivos de datos.
2. Uno o más archivos de control.
3. Dos o más registros de rehacer en línea.

Internamente, dicha base de datos contiene:

1. Varios usuarios/esquemas.
2. Uno o más segmentos de rollback (anulación).
3. Uno o más espacios de tablas.
4. Tablas del diccionario de datos.
5. Objetos de usuario (tablas, índices, vistas, etc).

El servidor que accede a esa base de datos consta como mínimo un SGA (que incluye el caché del búfer del bloque de datos, el caché del diccionario, el caché del búfer de registro de rehacer y, en Oracle, el fondo común SQL compartido).

- El proceso de fondo SMON.
- El proceso de fondo PMON.
- El proceso de fondo DBWR.
- El proceso de fondo LGWR.
- Procesos de usuarios con PGA asociados.

Esta configuración es la básica, las demás son opcionales dependiendo de la versión de Oracle y de las opciones que se utilicen.

14 COPIAS DE SEGURIDAD Y RECUPERACION

14.1 EXPORT/IMPORT (EXPORTAR/IMPORTAR)

La utilidad *Export (Exportar)* consulta la base de datos y almacena su respuesta en un archivo binario. Es posible personalizar las partes de la base de datos completa, un usuario o un conjunto de esquemas de usuario o un conjunto de tablas específico; también dispone de opciones que permiten exportar sólo las tablas que se ha modificado desde la última exportación (exportación incremental) o desde la última exportación del sistema completo (exportación acumulativa).

Las *exportaciones del sistema completo* leen también las tablas del diccionario de datos completas. Por lo tanto, pueden utilizarse para recrear por entero una base de datos, ya que en el diccionario de datos se realiza el seguimiento de los usuarios, archivos de datos, y objetos de la base de datos.

Export realiza una lectura lógica de la base de datos; para leer la información contenida en el archivo de volcado binario creado por la exportación debe utilizarse la utilidad *Import (Importar)*.

Import puede elegir de forma selectiva los objetos o usuarios del archivo volcado que se deseen importar, intentará entonces introducir esos datos en la base de datos (en lugar de sobrescribir los registros ya existentes).

14.2 COPIAS DE SEGURIDAD FUERA DE LINEA

Se pueden realizar copias de seguridad físicas de los archivos. Se dispone de dos opciones para llevar a cabo una copia de seguridad física de la base de datos: *copia de seguridad en línea (online backups)* y *copias de seguridad fuera de líneas (offline backups)*.

Para realizar copias de seguridad fuera de línea debe apagarse primero la base de datos; puede realizarse entonces una copia de seguridad de los archivos que componen la base de datos en un dispositivo de almacenamiento (copias de disco a disco o escrituras en cinta magnética). La base de datos puede volver a abrirse una vez terminada la copia de seguridad, no es posible realizar copias de seguridad físicas incrementales de la base de datos Oracle.

14.3 COPIAS DE SEGURIDAD EN LINEA

Las bases de datos que trabajan en modo ARCHIVELOG dispone de copias de seguridad en línea, que permiten realizar copias de seguridad físicas de la base de datos mientras la base de datos se encuentra abierta; esto se consigue poniendo los espacios de tablas en un estado de copia de seguridad de forma temporal u restableciéndolos a continuación a su estado normal, una vez realizada la copia de seguridad de los archivos.

15 CAPACIDADES DE SEGURIDAD

15.1 SEGURIDAD DE LAS CUENTAS

Las cuentas de la base de datos pueden protegerse mediante una contraseña; esta protección es independiente de la protección mediante contraseña del sistema operativo. También se pueden crear cuentas con capacidad de acceso automático, lo que permite a los usuarios que han accedido a una cuenta del anfitrión acceder a una cuenta de base de datos. Disponer de una cuenta o privilegios en otra base de datos.

15.2 PRIVILEGIOS DE LA BASE DE DATOS

En Oracle existen tres niveles de privilegios de sistema en la base de datos: CONNECT (Conexión), RESOURCE (Recursos) y DBA (Administrador de base de datos); todos los usuarios necesitan tener el privilegio CONNECT y los que van a crear segmentos necesitan tener el privilegio (RESOURCE).

En versiones más recientes de Oracle pueden crearse funciones (roles) de nivel de sistema a partir de todo el conjunto de privilegios de nivel de sistema –como CREATE TABLE (Crear tabla), CREATE INDEX (Crear índices), SELECT ANY TABLE (Seleccionar cualquier tabla)- para aumentar el conjunto básico de funciones de nivel de sistema.

15.3 SEGURIDAD DE LOS OBJETOS

Los usuarios que han creado objetos pueden conceder privilegios sobre ellos a otros usuarios por medio de la orden GRANT (Conceder).

También pueden conceder a otros usuarios la posibilidad de realizar concesiones sobre sus objetos; el privilegio GRANT ANY TABLE le da a las cuentas la capacidad de conceder privilegios sobre cualquier objeto a cualquier usuario.

15.4 AUDITORIA

Las actividades de los usuarios que afectan a objetos de la base de datos pueden supervisarse mediante la orden AUDIT (Auditoria). Entre dichas actividades se incluyen los accesos a tablas, los intentos de iniciar una sesión y las actividades que son privilegio del DBA. El resultado de estas auditorías se almacena en un archivo de auditoría dentro de la base de datos; Oracle ofrece además de opciones adicionales como la utilización de disparadores (*triggers*) de base de datos para complementar las auditorías ordinarias.

15.5 COMPRENSIÓN DE LOS CONVENIOS DE MODELADO LÓGICO

Una clave primaria (primary key; PK) es la columna o conjunto de columnas que hacen que cada registro de una tabla sea exclusivo. Una clave externa (foreign key, FK) es un conjunto de columnas que hacen referencia a una clave primaria existente. Las tablas pueden estar relacionadas entre ellas por medio de tres tipos de relaciones: de una a una, de una a muchas y de muchas a muchas.

15.6 CONFIGURACIONES Y CONSIDERACIONES SOBRE EL HARDWARE

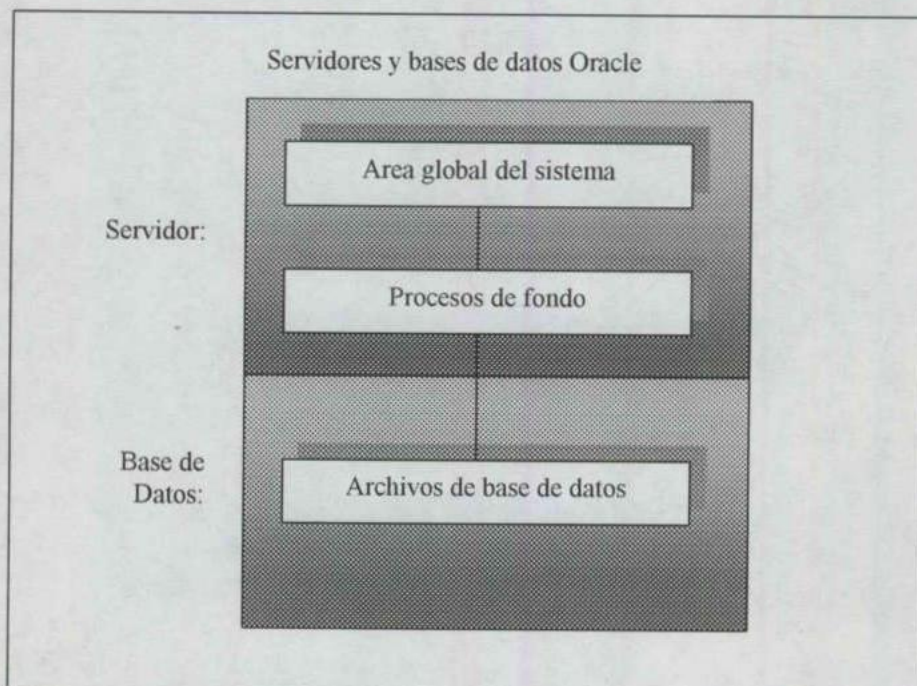
Aunque todas las bases de datos Oracle se construyen a partir de las mismas piezas básicas, las opciones de que se dispone dependerán de la plataforma hardware y del sistema operativo. En la mayoría de las plataformas, se dispondrá de varias opciones para elegir.

15.7 INTRODUCCIÓN A LA ARQUITECTURA

Una base de datos Oracle consta de archivos físicos, áreas de memoria y procesos. La distribución de estos componentes varía en función de la arquitectura de base de datos que se elija.

Los datos de la base de datos se almacenan en archivos físicos –llamados archivos de datos (datafiles) en un disco. Los datos se van almacenando en memoria según se va utilizando. Oracle utiliza áreas de memoria para mejorar el rendimiento y para gestionar la compartición de datos entre usuarios. El área de la memoria principal de una base de datos se denomina *Area global del sistema (SGA)*. Oracle utiliza un conjunto de procesos de fondo, compartidos por todos los usuarios, para leer y escribir datos entre el SGA y los archivos de datos.

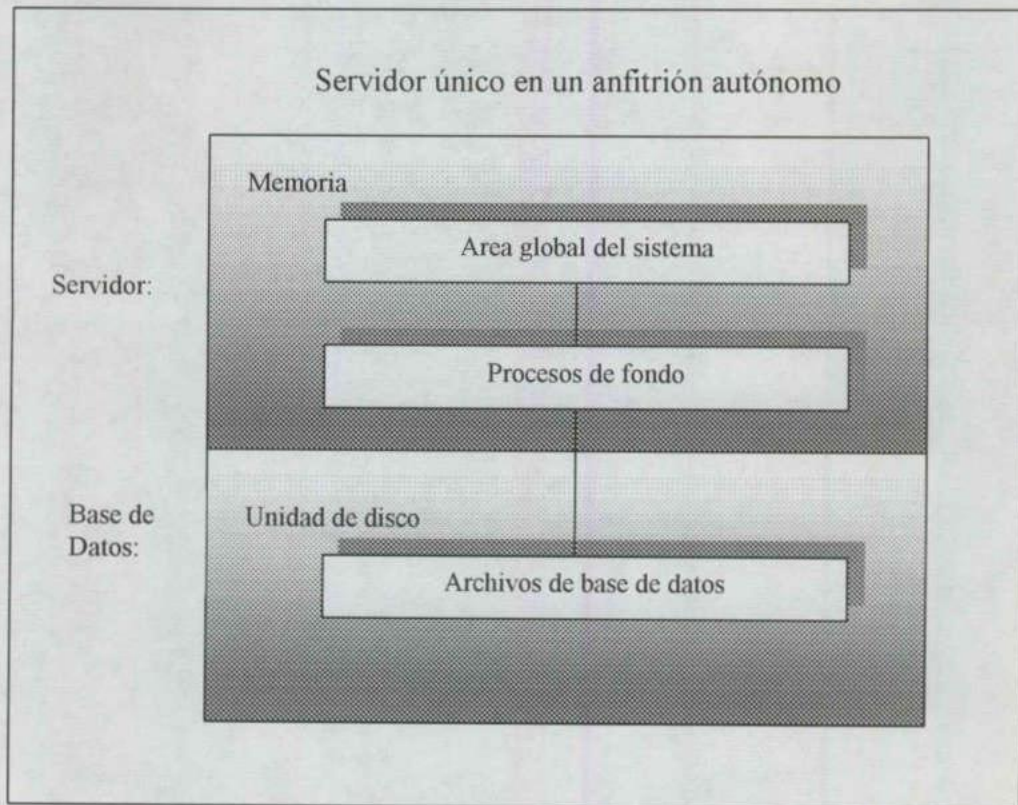
Un servidor de bases de datos es un conjunto de estructuras de memoria y procesos de fondo que acceden a un conjunto de archivos de base de datos. La relación entre servidores y bases de datos se ilustra en la sig. figura.



15.8 ANFITRIONES AUTONOMOS

La configuración más sencilla conceptualmente de una base de datos, es un único servidor que accede a una única base de datos en un anfitrión autónomo con un único disco.

En la sig. configuración se muestran todos los archivos que están almacenados en el único dispositivo del servidor, y sólo existe un SGA y un conjunto de procesos de fondo de Oracle en el servidor.



La arquitectura que se muestra en la figura anterior representa la configuración mínima. Todas las demás configuraciones de base de datos son modificaciones a esta estructura básica. También se muestra la existencia de dos puntos de interfaz principales en la base de datos:

- 1.- Entre los archivos de la base de datos y los procesos de fondo.
- 2.- Entre los procesos de fondo y el SGA.

Las labores de ajuste suelen concentrarse en la mejora del rendimiento de estos puntos de interfaz. Si el área de memoria dedicada a la base de datos es lo suficientemente grande, se realizarán pocas lecturas repetitivas de los archivos de la base de datos. Si la distribución está bien pensada, las operaciones de lectura y escritura de dichos archivos por parte de los procesos de fondo estarán optimizadas.

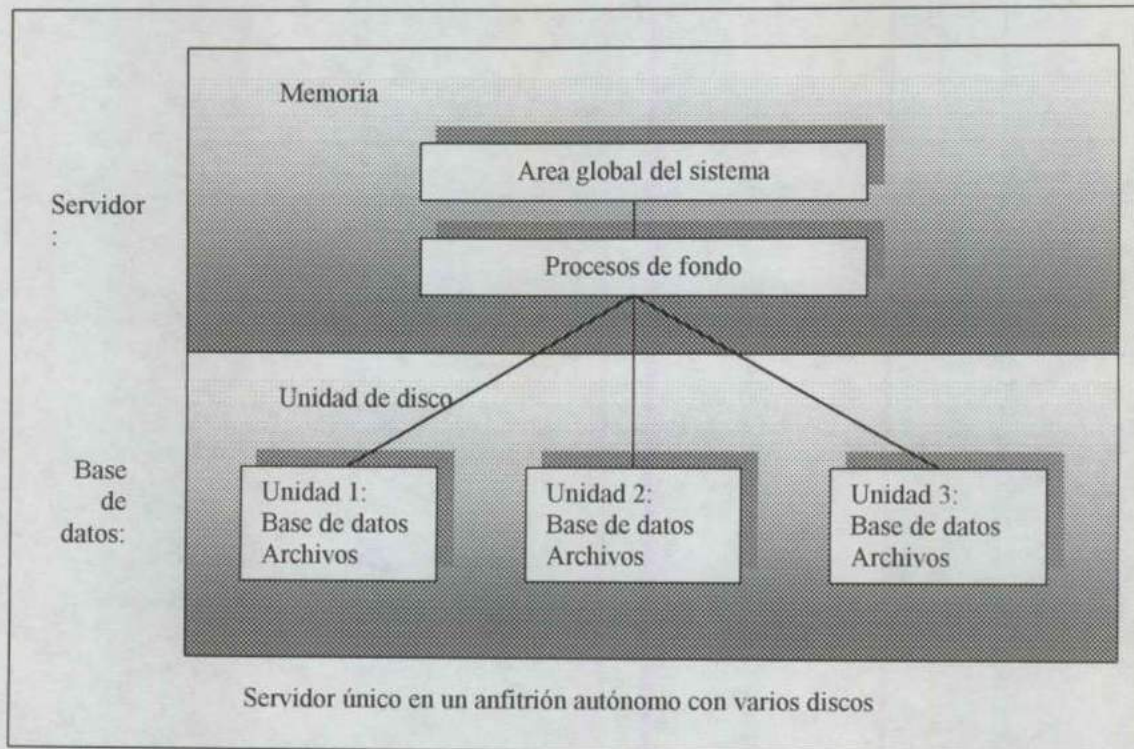
Entre los archivos almacenados en el disco se encuentran los archivos de la base de datos y el archivo INIT.ORA del anfitrión

15.9 ANFITRIONES AUTONOMOS CON RISTRAS DE DISCOS

Si se dispone de varios discos, los archivos de la base de datos podrían separarse en dispositivos distintos. Así se consigue mejorar el rendimiento de la base de datos al reducir la contienda entre los archivos de la base de datos.

Durante el funcionamiento de la base de datos es habitual que se necesite información de varios archivos para manejar una transacción o consulta.

Si los archivos no están distribuidos en varios discos, el sistema tendrá que leer de varios archivos en el mismo disco al mismo tiempo. Este proceso se muestra en la figura siguiente.



15.10 REPLICA DEL ARCHIVO DE CONTROL

El archivo INIT.ORA del servidor que accede a la base de datos se almacena en los directorios del software Oracle, generalmente en ORACLE_HOME.dbs.

15.11 REPLICA DEL ARCHIVO DE REGISTRO DE REHACER

La base de datos replica de forma automática los archivos de control, En Oracle, la base de datos también puede replicar los archivos de registro de rehacer en línea. Esto se consigue por medio de los *grupos de registro de rehacer*.

Si se utilizan estos grupos de registros de rehacer, el sistema operativo no tendrá que realizar la réplica de los archivos de registro de rehacer en línea, ya que será la base de datos la que lo haga de forma automática.

Al utilizar esta posibilidad, el proceso de fondo LGWR (*Log Writer, Escritor de registros*) escribe simultáneamente en todos los miembros del grupo de registros de rehacer en línea actual. Así, en lugar de pasar de forma cíclica por los archivos de registro de rehacer, pasa de forma cíclica por los grupos de archivos de registro de rehacer. Cuando los miembros de un grupo suelen estar situados en unidades de disco diferentes, no existe contienda por los discos entre los archivos, y LGWR apenas experimenta modificaciones en su rendimiento.

Los grupos de registro de rehacer pueden crearse mediante la orden CREATE DATABASE (Crear base de datos). También pueden añadirse a la base de datos después de creada, usando para ello la orden ALTER DATABASE (Modificar base de datos).

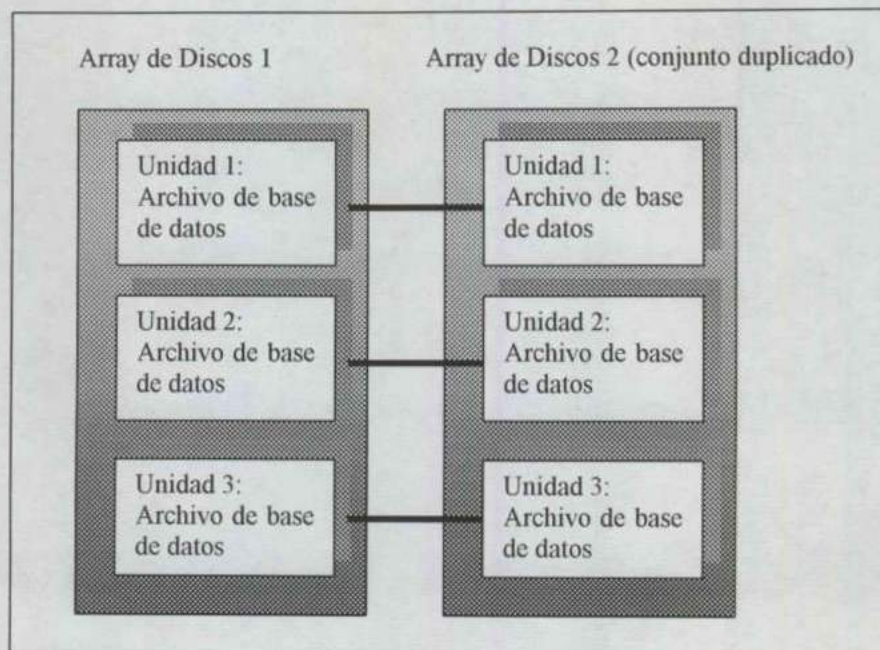
15.12 ANFITRIONES AUTONOMOS CON DUPLICACION DE DISCO

Muchos sistemas operativos proporcionan la posibilidad de mantener duplicados sincronizados de los archivos, mediante un proceso conocido como *duplicación de disco o duplicación de volumen* (esta práctica se denomina también *réplica*).

La utilización de la duplicación de los discos presenta dos beneficios. En primer lugar el conjunto de discos duplicados sirve de copia de seguridad en caso de que se produzca un fallo del disco. En la mayoría de los sistemas operativos, un fallo del disco provoca que el disco correspondiente del conjunto duplicado ocupe de forma automática el lugar del disco que ha fallado. El segundo beneficio es de una mejora del rendimiento.

La mayoría de los sistemas operativos que admiten duplicación de volumen pueden indicarles a las solicitudes de E/S en archivos que utilicen el conjunto de archivos duplicando el lugar del conjunto de archivos principal.

Así se reduce la carga de E/S del conjunto de discos principal y se obtiene un mejor rendimiento en las operaciones de E/S en archivo. Dicha duplicación del disco se muestra a continuación:



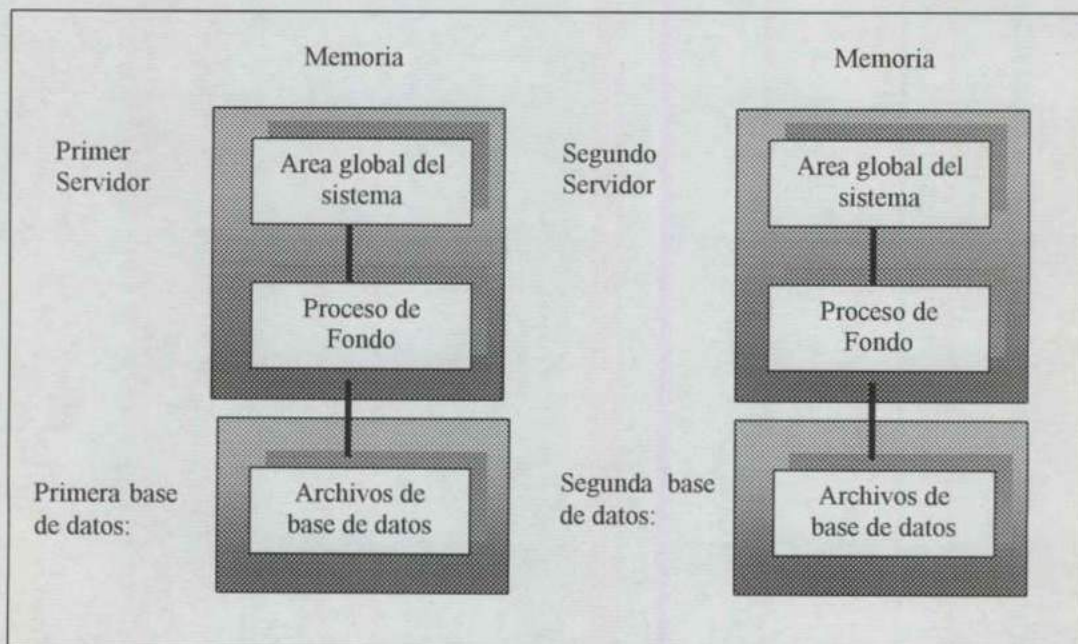
El tipo de duplicación que se muestra en la figura se denomina duplicación RAID-1. En este tipo de duplicación, cada disco del conjunto de discos principal está apareado, uno a uno con un disco del conjunto duplicado. Esta función del sistema operativo es posible que se disponga de otras opciones de duplicación. En la duplicación RAID-3 y RAID-5, por ejemplo, un conjunto de discos se trata como una sola unidad lógica, y cada archivo se distribuye automáticamente por cada disco. Se utiliza entonces un sistema de comprobación de paridad, que proporciona un medio de recuperar un miembro deteriorado o defectuoso del conjunto de discos. El método de duplicación que se utilice afectará a la distribución de los archivos por los dispositivos. Por ejemplo, los archivos de datos en los que se almacenan tablas suelen estar almacenados en un disco distinto al de los archivos de datos en los que se almacenan los índices de dichas tablas. No obstante, si se utilizan RAID-3 o RAID-5, la distinción entre los discos se

difumina. Para acceder a un archivo de datos cuando se utilizan esas opciones casi siempre hará falta acceder a todos los discos del conjunto, lo que aumenta la probabilidad de contienda entre los discos.

15.13 ANFITRIONES AUTONOMOS CON VARIAS BASES DE DATOS

Es posible crear varias bases de datos en un único anfitrión. Cada base de datos dispondrá de un conjunto independiente de archivos y se accederá a ella mediante un servidor distinto.

En la siguiente figura se muestra un único anfitrión que admite dos bases de datos. Como cada servidor necesita un SGA y procesos de fondo, el anfitrión debe ser capaz de admitir los requisitos de memoria y de proceso que esta configuración le impone.



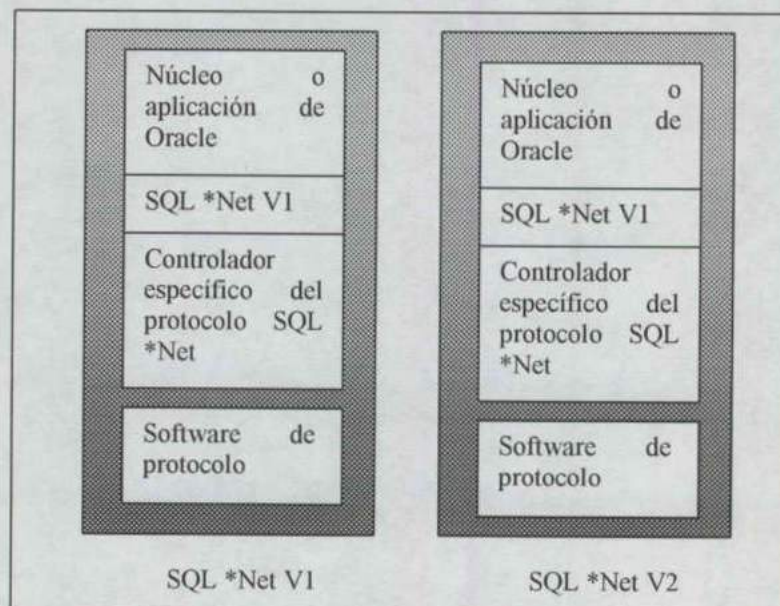
Varias bases de datos en el mismo servidor suelen compartir los mismos directorios del código fuente de Oracle. Si los archivos INIT.ORA de las dos bases de datos están ambos almacenados en el mismo directorio

(ORACLE_HOME/dbs), sus archivos INIT.ORA deberán contener los nombres de los servidores. Así se consigue que los nombres de los archivos sean exclusivos.

15.14 ANFITRIONES EN RED

Cuando anfitriones que admiten bases de datos Oracle se conectan por medio de una red, dichas bases de datos pueden comunicarse mediante SQL* Net. La integración de SQL*Net con el software de comunicaciones varía en función de la versión de SQL*Net que se utilice.

En la figura siguiente se ve como los controladores de SQL*Net confían al protocolo local de trabajo en red la tarea de lograr la conectividad entre dos servidores. La parte SQL*Net se encarga de las comunicaciones entre los niveles de aplicación de los dos servidores. En SQL*Net V2, el controlador de SQL*Net se divide en dos secciones: un adaptador de protocolo y el Sustrato de red transparente (TNS: *Transparent Network Sustrate*).

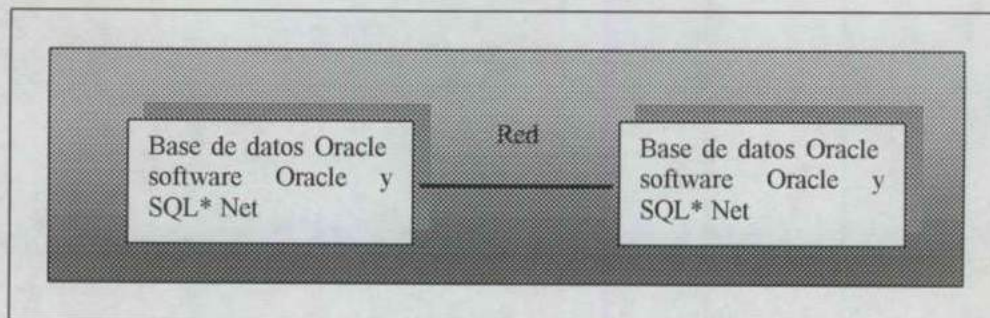


Las opciones de configuración de la base de datos de que se dispone en un entorno de red dependen de la configuración y opciones de la red. En las siguientes secciones se describen las principales arquitecturas:

- Redes de bases de datos (incluidas bases de datos distintas a Oracle), utilizadas para consultas remotas.
- Bases de datos distribuidas, utilizadas para transacciones remotas.
- Bases de datos de servidor paralelo, en las que varios servidores acceden a la misma base de datos.
- Bases de datos cliente-servidor.

15.15 REDES DE BASES DE DATOS

SQL*Net permite que las bases de datos Oracle se comuniquen con otras bases de datos que estén accesibles por medio de una red. Todos los servidores implicados deben estar ejecutando SQL*Net, como se ilustra en la siguiente figura:



En esta figura se muestran dos anfitriones. Cada uno de ellos puede manejar una base de datos en modo autónomo. Cada uno de los anfitriones de este ejemplo mantienen una copia del software Oracle y una o más bases de datos Oracle.

Para que las bases de datos puedan comunicarse, es necesario que sus servidores respectivos puedan comunicarse entre ellos, así los niveles de comunicación de las bases de datos confían en el software y el hardware de trabajo en red la tarea de

establecer el enlace de comunicaciones entre los servidores. Una vez creado dicho enlace de comunicaciones, el software de la base de datos puede utilizarlo para transportar paquetes de datos entre bases de datos remotas.

El software de Oracle que se encarga de la transferencia de datos entre bases de datos se denomina SQL*Net. En su configuración más sencilla, consta de un proceso anfitrión que espera las conexiones que se produzcan por medio de una vía de conexión específica.

Cuando detecta dichas conexiones, sigue las instrucciones que se han transferido por la conexión y devuelve los datos solicitados.

En SQL*Net V1, el proceso anfitrión se denomina *orasrv*, y es el proceso que debe estar ejecutándose en cada anfitrión implicado en las comunicaciones de las bases de datos. Cada servidor debe configurarse para asignar este proceso a un puerto de comunicaciones específico. En SQL*Net V2, el proceso anfitrión se denomina el *Escucha (Listener)*.

15.16 CONSULTAS REMOTAS

Las consultas a bases de datos Oracle remotas utilizan *enlaces de bases de datos (database links)* para identificar el trayecto que la consulta debe seguir para localizar los datos. Las opciones disponibles para manejar los enlaces de bases de datos han cambiado de SQL*Net V1 a V2 aunque su funcionalidad y sintaxis apenas han cambiado.

Un enlace de base de datos específica, directa e indirectamente, el anfitrión, base de datos y cuenta que debe utilizarse para acceder a un objeto específico. Para ello, hace referencia a la cadena de conexión (SQL*Net V1) o nombre de servicio (SQL*Net V2) de SQL*Net que va a utilizarse. Cuando una sentencia SQL hace referencia a un enlace de base de datos, abre una sesión en la base de datos especificada y ejecuta la sentencia SQL allí. Se devuelven los datos y la base de

datos remota permanece abierta por si vuelve a ser necesaria. Los enlaces de bases de datos pueden crearse como públicos (por la DBA, haciendo que el enlace esté disponible para todos los usuarios de la base de datos local) o como privados.

En el siguiente ejemplo SQL*Net V2 se crea un enlace de base de datos público denominados HR_ENLACE.

```
CREATE PUBLIC DATABASE LINK hr_enlace  
CONNECT TO hr IDENTIFIED BY puffinstuff  
USING 'hg';
```

La orden CREATE DATABASE LINK (Crear enlace de base de datos), como se muestra en el ejemplo, tiene varios parámetros:

- La palabra clave opcional PUBLIC, que permite que los DBA creen enlaces para todos los usuarios de una base de datos.
- El nombre del enlace (hr_enlace, en este ejemplo).
- La cuenta a la que conectarse (si no se especifica ninguna, en la base de datos remota se utilizarán el nombre de usuario y contraseñas locales).
- El nombre del servicio (<<hq>>). En SQL*Net V1 se utiliza la cadena de conexión en su lugar.

Para utilizar este enlace, basta con añadirlo como sufijo a los nombres de las tablas en las órdenes. En el siguiente ejemplo se consulta una tabla remota mediante el enlace de base de datos HR_ENLACE.

```
SELECT * FROM empleados@hr_enlace  
WHERE oficina='ANNAPOLIS';
```

Los enlaces de bases de datos no pueden utilizarse para devolver valores de campos con un tipo de datos LONG.

Los enlaces de bases de datos permiten que las consultas accedan a bases de datos remotas. También permiten que la información relativa a la ubicación física de los datos (su anfitrión, base de datos y esquema) sea transparente para el usuario.

Por ejemplo, si un usuario de la base de datos local crea una vista basada en un enlace de base de datos, cualquier acceso a la vista local consultará de forma automática la base de datos remota. El usuario que realiza la consulta no tiene por qué saber en qué lugar se encuentran los datos.

15.17 CONEXIONES CON BASES DE DATOS DISTINTAS A ORACLE

Oracle permite conexiones con bases de datos distintas de Oracle por medio de su producto *SQL*Connect*. La base de datos distinta de Oracle a la que se accede debe admitir *SQL*Connect* para que esta opción esté disponible. La sintaxis y utilización de *SQL*Connect* dependen de la base de datos que resida en el servidor remoto.

Al realizar consultas a bases de datos remotas distintas de Oracle, se está limitando a las capacidades de la base de datos remota.

Las extensiones al SQL, tales como la opción *CONNECT BY* de Oracle pueden utilizarse en la base de datos local en lugar de en la base de datos remota. Este suele implicar el envío por la red de un conjunto de datos mucho mayor del que sería necesario en una comunicación de Oracle.

15.18 ACTUALIZACIONES REMOTAS: LA OPCION DISTRIBUIDA

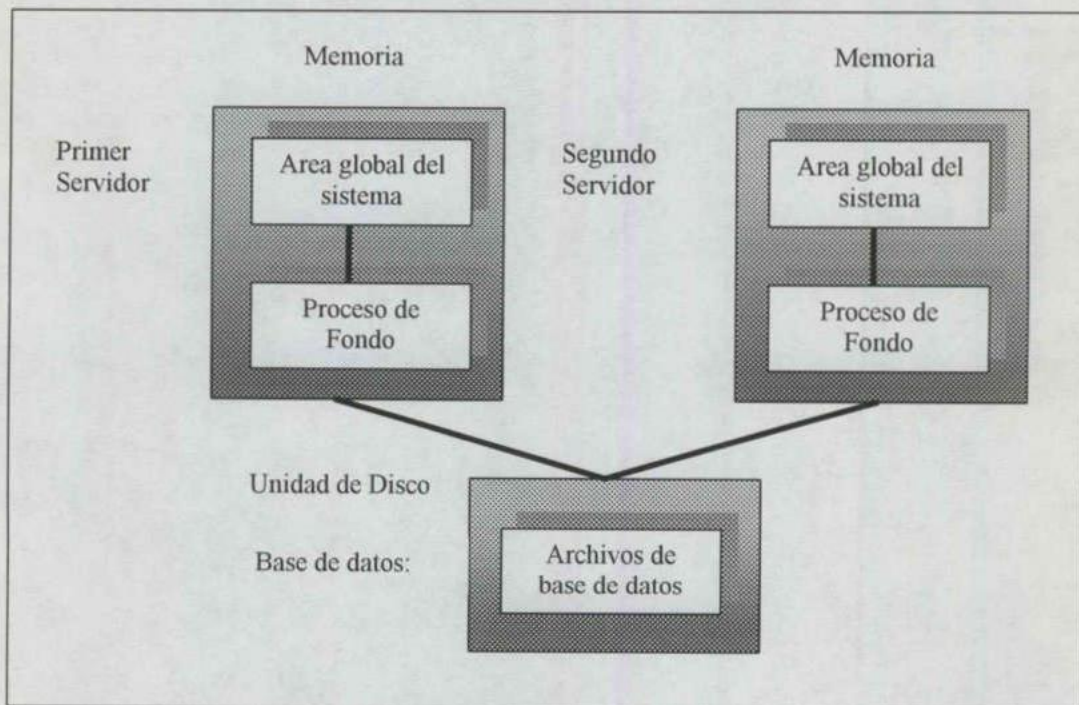
Además de consultar bases de datos remotas, las bases de datos Oracle que utilicen la Opción distribuida pueden actualizar bases de datos situadas en anfitriones remotos.

Las actualizaciones de estas bases de datos remotas pueden combinarse con actualizaciones de la base de datos local en una única unidad lógica de trabajo: se confirman todas o se anulan todas.

15.19 SERVIDORES AGRUPADOS: OPCION DE SERVIDOR PARALELO

Como se ha visto, todas las configuraciones que se han presentado se han caracterizado por tratarse de bases de datos a las que se accede desde un único servidor.

No obstante, en función de la configuración del hardware, puede ser posible utilizar varios servidores para acceder a una única base de datos. Esta configuración, denominada *Opción de servidor paralelo (Parallel Server Option)* se muestra en el siguiente esquema:



Esta figura representa dos servidores independientes que comparten el mismo conjunto de archivos de datos. Generalmente, estos servidores están situados en anfitriones independientes de un grupo hardware como un grupo VAX (VAX *cluster*). La utilización de esta configuración proporciona los siguientes beneficios:

- Se dispone de más recursos de memoria, ya que se utilizan dos máquinas.
- Si uno de los anfitriones deja de funcionar, el otro todavía puede acceder a los archivos de datos, lo que proporciona un medio de recuperación frente a los desastres.
- Los usuarios pueden separarse según el tipo de procesamiento que realizan, y los usuarios que ocupan mucha CPU se mantienen en un anfitrión aparte de las transacciones de procesamiento en líneas normales.

A pesar de estas ventajas, existe un importante problema potencial en esta configuración cuando dos servidores intentan actualizar los mismos registros, para esto al acabar una transacción Oracle no escribe de manera inmediata los bloques modificados del SGA de vuelta en los archivos de datos. Mientras dichos bloques se encuentran en el SGA, otra instancia puede solicitarlos. Para admitir esa solicitud, Oracle escribe los bloques al disco y a continuación los lee en el segundo SGA. El resultado es una consulta a una base de datos que utiliza muchos recursos de E/S.

La mejor forma de resolver el problema consiste en planificar la distribución de los usuarios no por su utilización de la CPU sino por su utilización de los datos, es decir, los usuarios que actualizan la misma tabla deben utilizar la misma instancia para acceder a la base de datos.

Al configurar un conjunto de servidores para que se utilicen la Opción de servidor en paralelo, deben especificarse varias estructuras y parámetros de base de datos.

En primer lugar, la base de datos central debe configurarse para manejar servidores independientes. Su requisito primario es un conjunto de segmentos de rollback (anulación) que pueden utilizar todos los servidores.

Lo mejor que puede hacerse para gestionar esto es crear un espacio de tablas de segmento de rollback aparte para cada servidor, utilizando el nombre de servidor como parte del nombre del espacio de tablas. Para especificar qué segmentos de rollback deben utilizarse, cada servidor debe nombrarlos en su archivo INIT.ORA. Es necesario configurar otros parámetros de inicialización de la base de datos para los servidores paralelos.

Como la mayoría de ellos serán los mismos para todos los servidores, deberá utilizarse el parámetro <<IFILE>> de INIT.ORA, que especifica un <<archivo para incluir>> (<<include file>>) en el que se enumeran los valores de los parámetros adicionales. Si ambas instancias se refieren al mismo IFILE, no será necesario preocuparse por los valores comunes.

En la siguiente tabla se enumeran los parámetros de INIT.ORA que deben ser exclusivos para cada servidor de un entorno de Servidor paralelo.

Parámetros de INIT. ORA específicos del anfitrión para la Opción de Servidor paralelo.

Parámetro	Descripción
Instance_number	Identifica la instancia a la base de datos
Rollback_segments	Segmentos de rollback (anulación) que va a utilizar la instancia.
Thread	Identifica el proceso del registro de rehacer de la instancia. Cada instancia dispone de su propio conjunto de grupos de registros de rehacer, en los que se mantiene el <i>proceso</i> de la instancia, o historia de rehacer.

En la tabla siguiente se enumeran los parámetros de INIT.ORA que deben ser idénticos para todos los servidores de un entorno de Servidor paralelo. El parámetro IFILE de los archivos INIT.ORA específicos de los servidores debe hacer referencia al archivo que contiene estos parámetros.

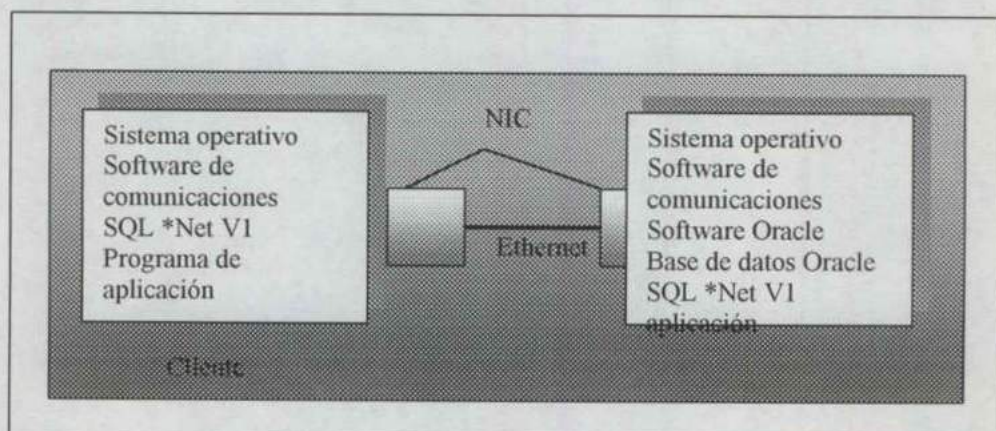
Parámetros comunes de INIT.ORA para la Opción de servidor paralelo

Parámetro	Descripción
Control_files	Nombres de los archivos de control en la base de datos, separados por comas.
Db_block_size	Tamaño del bloque para la base de datos.
Db_files	Número máximo de archivos de datos abiertos en la base de datos. El valor MAXDATAFILES establecido durante la creación de la base de datos anula este valor.
Db_name	Nombre de la base de datos.
Gc_db_locks	Número total de bloqueos de instancia.
Gc-files_to_locks	Controla los bloqueos de archivos de la base de datos.
Gc_lck_procs	Número total de bloqueos distribuidos.
gc_rollback_locks	Número de bloqueos de rollback (anulación); es aconsejable utilizar el valor por omisión para dos instancias, o diez por instancia para más de dos instancias.
gc_rollback_segments	Número total de segmentos de rollback utilizados por todas las instancias.
gs_save_rollback_locks	Número de bloqueos de almacenamiento de rollback; es aconsejable utilizar el valor por omisión para dos instancias, o diez por instancia para más de dos instancias.
gc_segments	Número máximo de segmentos en los que diferentes instancias pueden realizar a la vez actividades de gestión del espacio. Es aconsejable utilizar el valor por
Log_files	

Row_locking	omisión para dos instancias, o diez por instancia para más de dos instancias.
Serializable	Número máximo de grupos de registro. Tiene un valor por omisión de <<ALWAYS>> (<<SIEMPRE>>), lo que significa que sólo se obtienen bloqueos de fila al actualizar una tabla.
Single_process	Tiene un valor por omisión de <<FALSE>> (<<FALSO>>), que permite que las actualizaciones de una tabla se realicen mientras que realiza una consulta a dicha tabla. Debe establecerse como <<FALSE>> para la Opción de servidor paralelo.

15.20 APLICACIONES DE BASES DE DATOS CLIENTE-SERVIDOR

En una configuración anfitrión-anfitrión existe una base de datos Oracle en cada anfitrión y las bases de datos se comunican mediante SQL*Net. No obstante, es posible que un anfitrión sin una base de datos acceda a una base de datos remota. Esto suele conseguirse haciendo que los programas de aplicación de un anfitrión accedan a una base de datos en un segundo anfitrión. En dicha configuración, el anfitrión que ejecuta la aplicación se denomina *cliente* y el otro se denomina *servidor*. Esta configuración se ilustra en la siguiente figura.



En esta figura se puede apreciar que el cliente debe tener la capacidad de comunicarse con el servidor por la red. Los programas de aplicación se ejecutan del lado del cliente; por lo tanto, la base de datos se utiliza principalmente para las operaciones de E/S. De esta manera, los costes de CPU derivados de la ejecución de los programas de aplicación se cargan al PC cliente en lugar de al servidor.

Para que esta configuración funcione, el cliente debe estar ejecutando SQL*Net. Cuando el programa de aplicación del cliente le solicita al usuario la información de conexión a la base de datos; éste debe especificar la cadena de conexión (para SQL*Net VI) o el nombre de servicio (para SQL*Net V2) de SQL*Net. La aplicación abrirá entonces una sesión en la base de datos remota.

La utilización de una configuración cliente-servidor ayuda así a reducir la cantidad de trabajo que realiza el servidor. No obstante, llevar una aplicación a una configuración cliente-servidor no mejora de forma automática el rendimiento del sistema, por dos puntos principales:

- Puede que los recursos del CPU no sean un problema. Por lo general, los recursos de la CPU se utilizan a menudo durante las horas no laborables. Una programación eficiente de los programas de proceso por lotes puede ayudar a resolver esta situación.
- Puede que la aplicación no se haya rediseñado. El diseño para un entorno cliente-servidor requiere que se tengan en cuenta los volúmenes de datos que se envían por la red en cada acceso a la base de datos, mientras que en las aplicaciones cliente-servidor, es necesario tener en cuenta el tráfico en la red durante la planificación y el ajuste.

Existen muchas formas distintas de implementar la configuración cliente-servidor, en función del hardware disponible. La configuración que se muestra en la figura anterior es de las más comunes; puede utilizarla una herramienta de

consulta `ad hoc` que se ejecute en una PC para acceder a una base de datos Oracle que se ejecute en un servidor.

Si en el entorno se dispone de servidores de archivos, o si los clientes son estaciones de trabajo en lugar de PC, se tendrá una enorme flexibilidad con respecto a la arquitectura del sistema.

16 UTILIZACION DEL ESPACIO DE LAS BASES DE DATOS.

16.1 INTRODUCCION

Para entender la forma en que debe asignarse el espacio dentro de la base de datos, es necesario conocer primero cómo se utiliza este espacio.

Cuando se crea una base de datos, se divide en varias secciones lógicas denominadas espacios de tablas. El primero que se crea es el espacio de tablas SYSTEM. A continuación se crean espacios de tablas adicionales que contienen distintos tipos de datos.

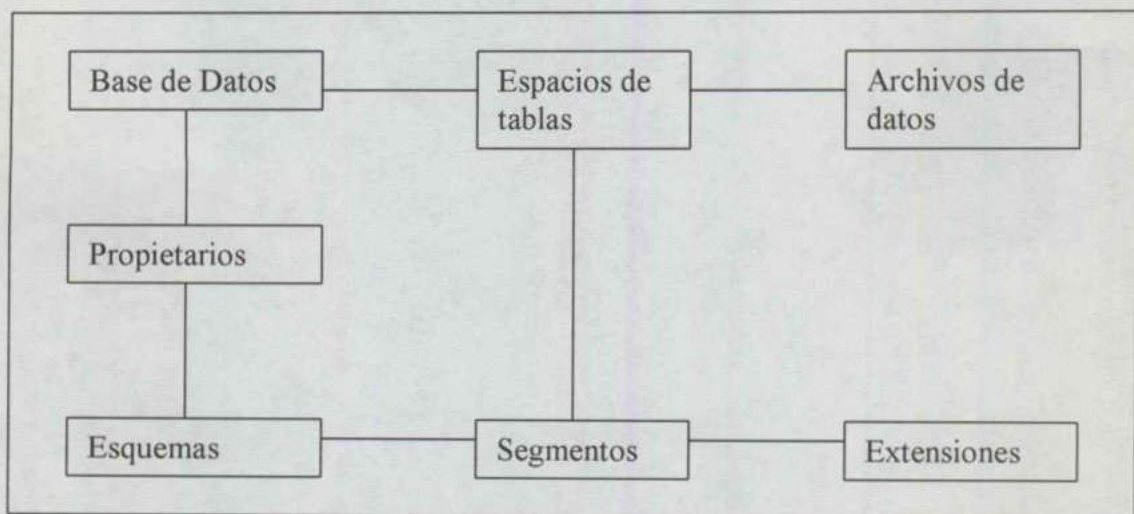
Cuando se crea un espacio de tablas, también se crean archivos de datos en los que guardar sus datos. Estos archivos asignan de forma inmediata el espacio especificado durante su creación, y nunca aumentan de tamaño. Existe, por tanto, una relación de uno a muchos entre espacios de tablas y archivos de datos.

Una base de datos puede tener varios usuarios, cada uno de los cuales dispone de un *esquema*. Los esquemas de usuario son un conjunto de objetos lógicos de base de datos tales como tablas e índices. Estos objetos hacen referencia a estructuras de datos físicas que están almacenadas en espacios de tablas. Los objetos del esquema de un usuario pueden almacenarse en varios espacios de tablas, y un único espacio de tablas puede contener objetos de varios esquemas.

Cuando se crea un objeto de base de datos (como una tabla o un índice), se asigna a un espacio de tablas por omisión para el usuario o a uno concreto mediante instrucciones específicas. Se crea un *segmento* en dicho espacio de tablas en el que almacenar los datos asociados con el objeto. El espacio asignado al segmento no se libera nunca hasta que el segmento se elimina, se comprime o se trunca.

Un segmento esta compuesto por secciones llamadas *extensiones*, que son conjuntos contiguos de bloques de Oracle. Cuando los datos nuevos ya no caben en las extensiones existentes, el segmento obtiene otra extensión. Este proceso de extensión continua hasta agotar el espacio libre disponible en los archivos de datos del espacio de tablas, o hasta llegar a un número máximo interno de extensiones por segmento. Si un segmento está formado por varias extensiones, nada garantiza que dichas extensiones sean contiguas.

Las interrelaciones lógicas entre los objetos de base de datos se muestran en la siguiente figura.



Como se aprecia en dicha figura, un espacio de tablas puede contener varios segmentos. Los tipos de segmentos disponibles en Oracle son:

- *TABLE
- *INDEX
- *ROLLBACK
- *TAMPORARY

Una de las principales funciones del DBA es la gestión del espacio que utiliza cada uno de ellos.

16.2 IMPLICACIONES DE LA CLAUSULA STORAGE

La cantidad de espacio que utiliza un segmento viene determinada por sus parámetros de almacenamiento. Estos parámetros los determina la base de datos en el momento de la creación del segmento; si no se incluyen parámetros de almacenamiento específicos en la orden `CREATE [TABLE|INDEX|CLUSTER|ROLLBACK SEGMENT]`, la base de datos utilizará los parámetros de almacenamiento por omisión del espacio de tablas en que va a almacenarse.

Los parámetros de almacenamiento especifican el tamaño `INITIAL` (Inicial) de la extensión, el tamaño de la extensión `NEXT` (Siguiete), el `PCTINCREASE` (Factor de crecimiento geométrico de cada segmento sucesivo), el `MAXEXTENTS` (Número máximo de extensiones) y el `MINEXTENTS` (Número mínimo de extensiones).

Los valores `INITIAL` y `MINEXTENTS` no pueden modificarse después de creado el segmento. Los valores por omisión de los parámetros de almacenamiento de cada espacio de tablas se encuentran en las vistas `DBA:TABLESPACES` y `USER_TABLESPACES`.

Cuando se crea un segmento, éste obtiene al menos una extensión (pueden definirse otros valores mediante `MINEXTENTS`). Esta extensión se utilizará para almacenar datos hasta que no quede espacio libre en ella (puede utilizarse la cláusula `PCTFREE` para reservar, dentro de cada bloque de cada extensión, un porcentaje de espacio que seguirá estando disponibles para las actualizaciones de las filas existentes).

Cuando se añaden datos adicionales al segmento, éste se extiende obteniendo una segunda extensión del tamaño especificado por el parámetro `NEXT`. No se garantiza que la segunda extensión esté contigua físicamente a la primera. Para

todos los segmentos excepto los segmentos de rollback (anulación), un segmento óptimo tendrá una extensión.

El parámetro PCTINCREASE está pensado para los usuarios que no conocen de antemano la cantidad de espacio que van a necesitar (también conocidos como <<desarrolladores de aplicaciones>>). Un valor distinto de cero en este parámetro puede ser peligroso (provoca que el tamaño de cada extensión sucesiva aumente geométricamente según el factor PCTINCREASE especificado).

17 SEGMENTOS

17.1 SEGMENTOS, EXTENSIONES Y BLOQUES

Los segmentos (segments) son la contrapartida física a los objetos lógicos de base de datos que almacenan los datos. En los segmentos de índice, se almacenan los datos asociados con los índices. Para una gestión eficaz de los segmentos es necesario que el DBA conozca los objetos que una aplicación va a utilizar, la forma en que se van a introducir los datos en esos objetos y las maneras en que se van a recuperar.

Como un segmento es una entidad física, debe estar asignado a un espacio de tablas de la base de datos (y por tanto encontrarse en uno de los archivos de datos de dicho espacio de tablas).

Un segmento consta de secciones llamadas extensiones (extents) (conjuntos contiguos de bloques de Oracle). Cuando las extensiones existentes en un segmento ya no pueden contener datos nuevos, el segmento consigue otra extensión.

Este proceso de extensión continúa hasta agotar el espacio libre disponible en los archivos de datos del espacio de tablas, o hasta alcanzar el número máximo de extensiones por segmento. Si un segmento consta de varias extensiones, no se garantiza que dichas extensiones sean contiguas.

17.2 SEGMENTOS DE TABLA

En los *segmentos de tabla*, llamados *segmentos de datos*, se almacenan las filas de datos asociadas con tablas o grupos. Cada segmento de datos contiene un bloque de cabecera que sirve como directorio del espacio del segmento.

Una tabla dimensionada correctamente tendrá una única extensión. Cuantas más extensiones tiene un segmento de datos, más trabajo cuesta recuperar los datos que contiene. En algunos casos, no es posible disponer de segmentos de datos con una única extensión; como las extensiones no se pueden distribuir por los archivos de datos, un segmento que sea mayor que el mayor archivo de datos disponibles tendrá que constar de varias extensiones. También se pueden utilizar las extensiones múltiples para distribuir un segmento por varios discos, aunque esta operación es mejor realizarla fuera de la base de datos.

Cuando un segmento de datos obtiene una extensión, la conserva hasta que el segmento se elimina o se trunca. El hecho de borrar filas de una tabla no afecta a la cantidad de espacio asignado a dicha tabla. El número de extensiones aumenta hasta que (1) se alcanza el valor MAXTENTS, (2) se alcanza el máximo número de extensiones permitidas para el sistema operativo o (3) el espacio de tablas se queda sin espacio.

El ajuste del parámetro PCTFREE permite minimizar la cantidad de espacio desperdiciado en un segmento de datos. Este parámetro especifica la cantidad de espacio que debe dejarse libre dentro de cada bloque de datos.

Este espacio libre puede utilizarse entonces cuando se introducen valores en columnas vacías, o cuando la actualización de otros valores de la fila fuerza a ésta a crecer. El valor adecuado para PCTFREE depende de la base de datos, ya que depende de la naturaleza de las actualizaciones que se lleven a cabo.

17.3 SEGMENTOS DE INDICE

Al igual que los segmentos de tabla, los segmentos de índice conservan el espacio que se les ha asignado hasta que se eliminan; no obstante, también pueden eliminarse de forma indirecta si se elimina su tabla o grupo.

Para minimizar la contienda, los índices deben almacenarse en un espacio de tablas separado de sus tablas asociadas.

Los índices están sujetos a los mismos problemas de espacio que las tablas. Sus segmentos tienen cláusulas de almacenamiento que especifican sus valores INITIAL, NEXT, MINEXTENTS, MAXEXTENTS y PCTINCREASE, y es tan probable que estén fragmentados como sus tablas.

Deben dimensionarse adecuadamente antes de crearse; de lo contrario, su fragmentación puede reducir drásticamente el rendimiento de la base de datos.

17.4 SEGMENTOS DE ROLLBACK (ANULACION)

Para mantener la consistencia de lectura entre varios usuarios de la base de datos y poder anular las transacciones, Oracle debe disponer de un mecanismo de reconstrucción de una <<imagen anterior>> de los datos para transacciones no confirmadas. Oracle utiliza segmentos de rollback dentro de la base de datos para llevar a cabo esta tarea. Los segmentos de rollback crecen hasta hacerse tan grandes como las transacciones que admiten. Mientras que las tablas óptimas tienen una extensión adaptada a sus requisitos de tamaño, los segmentos de rollback tendrán varias extensiones de igual tamaño que se sumen a su tamaño total óptimo (cuando se crea tienen un mínimo de dos extensiones). Cada extensión debe ser suficientemente grande para manejar dos datos de una única transacción. Si no es así, o si demasiados usuarios solicitan el mismo segmento de rollback, éste podrá extenderse.

Es recomendable disponer de un segundo espacio de tablas de segmentos de rollback si los usuarios realizan de forma periódica transacciones muy grandes (como grandes cargas de datos) y no se desea asignar ese espacio al espacio de tablas de los segmentos de rollback de forma permanente.

El espacio de tablas de los segmentos de rollback especializados puede añadirse y eliminarse a voluntad; los segmentos de rollback pueden especificarse mediante la siguiente orden:

La cláusula *OPTIMAL* (Optimo), que permite que los segmentos de rollback se compriman hasta un tamaño óptimo después de extenderse, es similar a la cláusula *PCTINCREASE*; sirve de ayuda en los sistemas que no se han implementado adecuadamente para la forma en que utilizan.

Oracle proporciona a los DBA mucha más flexibilidad en la gestión de los segmentos de rollback, por lo que éstos podrán mantenerse mucho más fácilmente.

17.5 SEGMENTOS TEMPORALES

Los segmentos temporales sirven para almacenar datos temporales durante operaciones de ordenación (como grandes consultas, creaciones de índices y uniones). Cada usuario dispone de un espacio en tablas temporal que se especifica al crear la cuenta por medio de *CREATE USER* o *ALTER USER*. Debe apuntar a algún lugar que no sea *SYSTEM* (valor por omisión).

Cuando se crea un segmento temporal, éste utiliza los parámetros de almacenamiento por omisión de ese espacio de tablas. Mientras existe, sus parámetros de almacenamiento no pueden modificarse cambiando los parámetros de almacenamiento por omisión del espacio de tablas.

Se extiende por sí mismo cuando es necesario y se elimina a sí mismo cuando la operación termina o se produce un error. Como el propio segmento temporal puede ser causa de errores (por superar el máximo número de extensiones o agotar el espacio libre del espacio de tablas), a la hora de dimensionar el espacio de tablas temporal es necesario tener en cuenta el tamaño de las grandes consultas y operaciones de ordenación.

El espacio de tablas temporal, que suele denominarse TEMP, está fragmentado por naturaleza. Los segmentos temporales se crean, extienden y eliminan constantemente. Es, por lo tanto, necesario maximizar la reutilización de las extensiones eliminadas. Para conseguirlo, debe elegirse un tamaño de extensión INITIAL y NEXT de 1/20 a 1/50 del tamaño del espacio de tablas. INITIAL y NEXT deben ser iguales para este espacio de tablas, y debe elegirse un PCTINCREASE de 0; el resultado serán segmentos compuestos de extensiones de idéntico tamaño. Cuando se eliminen estos segmentos, el siguiente segmento temporal que se forme podrá utilizar las extensiones eliminadas.

Oracle recombina automáticamente las extensiones libres contiguas, por lo que la reutilización de las extensiones no es tan preocupante como en las primeras versiones de Oracle.

17.6 ESPACIO LIBRE

Una extensión libre en un espacio de tablas es un conjunto de bloques libres contiguos en el espacio de tablas, cuando se elimina un segmento, se desasignan sus extensiones y se marcan como libres.

La metodología de Oracle afecta a la asignación de espacio dentro del espacio de tablas durante la siguiente solicitud de espacio.

En su búsqueda de una extensión libre suficientemente grande, la base de datos no mezcla extensiones contiguas libres a menos que no haya otra alternativa; de esta forma, tiende a utilizar la extensión libre más grande situada al final del espacio de tablas, mientras que las extensiones libres más pequeñas situadas al principio del espacio de tablas se utilizan menos, convirtiéndose en <<baches>> del espacio de tablas porque no tienen, por sí mismas, un tamaño adecuado para ser útiles.

Según progresa este patrón de utilización, la base de datos se aleja cada vez más de su asignación ideal del espacio.

En una base de datos ideal, todos los objetos se crean con su tamaño adecuado (en una extensión) y todo el espacio libre siempre está junto, formando un fondo común de recursos listo para ser utilizado.

17.7 ENCAJE FISICO

En las bases de datos Oracle es necesario planificar la asignación de archivos (mediante pesos de E/S) y verificar los pesos cuando el sistema entra en producción. Puede modificarse entonces la disposición de los archivos para equilibrar mejor los requisitos de E/S de los archivos.

El resultado será una base de datos que consiga sus objetivos de rendimiento sin sacrificar recuperabilidad, y sus objetivos de recuperabilidad sin sacrificar el rendimiento.

También es necesario dimensionar correctamente cada una de las facetas principales de la base de datos [tablas, índices, segmentos de rollback, (anulación) y segmentos temporales].

Para realizar un dimensionado correcto hace falta conocer la forma en que van a introducirse los datos, cómo van a almacenarse y los procesos que van a realizarse sobre ellos una vez introducidos.

Los costes de planificación de los parámetros de almacenamiento son mínimos comparados con los costes de manejar el sistema una vez puesto en producción.

El ajuste de postproducción debe ser un paso final y de escasa importancia en el proceso de planificación del diseño físico de la base de datos.

18 GESTION DEL PROCESO DE DESARROLLO

18.1 TECNOLOGIA

El hecho de seguir la metodología no basta si se hace aisladamente. Las entregas durante el proceso deben estar disponibles mientras se lleva a cabo el desarrollo. Como en la mayoría de los equipos de desarrollo hay varios desarrolladores (y a partir de ahora al menos un DBA), es necesario establecer un medio de comunicación. Los canales de comunicación servirán para mantener la consistencia en la planificación y ejecución.

Hacen falta cuatros soluciones tecnológicas para conseguir que la metodología funcione. En la actualidad hacen falta cuatro tecnologías independientes, ya que no se dispone de un paquete integrado de desarrollo de productos. Dichas tecnología son las herramientas CASE, los directorios compartidos, las bases de datos de gestión de proyectos y las bases de datos de discusión.

18.2 HERRAMIENTAS CASE

Una herramienta CASE (IcomputerAided Software Engineering; ingeniería asistida por ordenador) permite generar el diagrama de relaciones entre entidades y el diagrama de la base de datos. Oracle*CASE es una herramienta CASE multiusuario que puede crear el diagrama de relaciones entre entidades y dispone de un diccionario de datos integrado. Permite compartir entidades entre aplicaciones, y puede almacenar información sobre volúmenes de las tablas y tamaños de las filas.

Estas funciones ayudan a resolver algunas de las entregas que se han definido. Su capacidad multiusuario permite asegurar la coherencia entre distintos desarrolladores. También permite mantener o congelar distintas versiones de un modelo de datos.

Las órdenes SQL que crean los objetos de la base de datos para la aplicación deben generarse directamente desde una herramienta CASE, que también puede utilizarse para crear versiones genéricas de las aplicaciones basadas en los objetos definidos de la base de datos.

18.3 DIRECTORIOS COMPARTIDOS

No existe una herramienta específica en la que crear algunas de las entregas, como los requisitos de copia de seguridad. Estas entregas deben crearse en las herramientas más adecuadas de que se disponga en el emplazamiento. Los archivos resultantes deberán almacenarse en directorios compartidos del proyecto, de forma que todos los miembros del equipo implicado puedan acceder a ellos. Los formatos y convenios de denominación de estos archivos deben especificarse en las primeras fases del proceso de desarrollo.

18.4 BASES DE DATOS DE GESTION DE PROYECTOS

Es necesario disponer de una base de datos de gestión de proyectos para poder comunicar el estado de la aplicación y de sus entregas a personas ajenas al equipo de desarrollo. Esta base de datos le proporcionará a dichas personas una visión del proyecto y de sus objetivos actuales. De este forma se consigue que personas no implicadas directamente en el proyecto (como el personal de gestión de sistemas) puedan anticipar las necesidades futuras.

También permite analizar el impacto de los cambios en la programación o de los retardos en los objetivos del camino crítico. Este análisis puede provocar la modificación de los niveles de recursos asignados a las tareas del proyecto.

18.5 BASES DE DATOS DE DISCUSION

La mayor parte de la información de estas tres áreas (la herramienta CASE, los directorios de entregas compartidos y las bases de datos de gestión de proyectos) representan un consenso de opinión. Por ejemplo, varios miembros del equipo pueden tener distintas opiniones sobre la estrategia de copia de seguridad, lo mismo que el personal de gestión del sistema y el DBA. Para facilitar esta comunicación puede crearse un conjunto de bases de datos de discusión (generalmente mediante un producto de trabajo en grupo, en una red de área local). Pueden enviarse versiones provisionales a estas áreas antes de llevar la resolución final al directorio de entregas compartido.

18.6 GESTION DEL DESARROLLO DE PAQUETES

En la adecuada gestión de la implementación de los paquetes intervienen muchos de los temas que se han descrito para los procesos de desarrollo de aplicaciones en puntos anteriores. A continuación se proporciona una introducción a la forma de tratar los paquetes para que encajen lo mejor posible en el entorno de desarrollo.

18.7 GENERACION DE DIAGRAMAS

La mayoría de las herramientas CASE tienen la capacidad de realizar una operación de *ingeniería inversa* sobre los paquetes para generar un diagrama físico de la base de datos. Este proceso consiste en analizar las estructuras de las tablas y generar un diagrama físico de la base de datos que sea coherente con dichas estructuras.

Suele conseguirse mediante el análisis de los nombres de las columnas y de los índices para identificar columnas clave.

No obstante, no suele existir una correlación de uno a uno entre el diagrama físico de la base de datos y el diagrama de relaciones entre entidades. Los diagramas de relaciones entre entidades de los paquetes suele proporcionarlos el vendedor del paquete, y resultan muy útiles en la planificación de interfaces con la base de datos de paquetes.

18.8 REQUISITOS DE ESPACIO

La mayoría de los paquetes proporcionan una estimación precisa de su utilización de recursos de la base de datos durante la utilización en producción. No obstante, no suelen tener en cuenta sus requisitos de espacio durante cargas de datos y actualizaciones del software. Por este motivo, es conveniente crear un espacio de tablas de segmentos de rollback (anulación) especiales (RBS_2), que se utilice para manejar grandes cargas de datos.

También puede ser necesario disponer de un espacio de tablas de datos de reserva si el paquete crea copias de todas sus tablas durante las operaciones de actualización.

18.9 OBJETIVOS DE AJUSTE

Es necesario imponer objetivos de ajuste a los paquetes, lo mismo que ocurre con las aplicaciones personalizadas.

El establecimiento y seguimiento de estos valores de control ayudará a identificar las áreas del paquete que necesitan un ajuste.

18.10 REQUISITOS DE SEGURIDAD

Si los paquetes se han creado antes en otro sistema de base de datos, es muy probable que su conversión a Oracle no aproveche por entero las posibilidades funcionales de Oracle, entre las que se incluyen el bloqueo a nivel de fila, la utilización de objetos de secuencia y las capacidades de Oracle como funciones (roles) y disparadores (triggers) de base de datos. En el ajuste de uno de estos paquetes para que se adapte a las necesidades puede ser necesario modificar su código fuente.

Si el paquete asume que tiene autoridad de DBA completa, no deberá almacenarse en la misma base de datos. Esto es especialmente cierto en Oracle, ya que las cuentas con privilegios de DBA tienen acceso completo a todos los datos de todas las tablas de la base de datos.

La mayoría de los paquetes que necesitan autoridad de DBA lo hacen para poder añadir nuevos usuarios a la base de datos. Los DBA de Oracle deben determinar exactamente qué privilegios de nivel de sistema necesita realmente la cuenta del administrador del paquete (generalmente basta con CREATE SESSION y CREATE USER). Puede crearse entonces una función (role) especializada de nivel de sistema para proporcionar este conjunto limitado de privilegios de sistema al administrador del paquete.

Los paquetes desarrollados originalmente en bases de datos distintas de Oracle pueden utilizar la misma cuenta que otro paquete traducido a Oracle.

18.11 REQUISITOS DE LOS DATOS

Deben definirse claramente cualquier requisito de procesamiento que tengan los paquetes, en especial en el tema de la introducción de los datos. Estos aspectos suelen estar bien documentados en la documentación del paquete.

18.12 PLANES DE EJECUCION

Para generar planes de ejecución hace falta acceder a las sentencias SQL que se ejecutan sobre la base de datos. En Oracle, la forma de conseguirlo es mediante el código fuente; si no está disponible, no se dispondrá de planes de ejecución; en Oracle en el área de SQL compartido del SGA se conservan las sentencias SQL que se ejecutan sobre la base de datos.

El proceso de hacer concordar las sentencias SQL con partes específicas cuyas funciones y rendimiento sean fundamentales para el éxito de la aplicación, y al trabajar con el equipo de soporte del paquete para resolver los temas de rendimiento.

18.13 PROCEDIMIENTOS DE LA PRUEBA DE ACEPTACION

Los procedimientos de la prueba de aceptación de un paquete suelen crearse después de instalada la aplicación. No obstante, a los paquetes se les deberían imponer los mismos requisitos funcionales que deben cumplir las aplicaciones personalizadas. Por tanto, los procedimientos de la prueba de aceptación deberían desarrollarse antes de seleccionar el paquete; pueden generarse a partir de los criterios de selección del paquete.

Esta forma de comprobación permite comprobar las funciones que se necesitan, en lugar de las que los desarrolladores del paquete consideraron necesarias.

Hay que asegurarse de especificar cuáles son las opciones en case de que el paquete no supere su prueba de aceptación por motivos funcionales o de rendimiento. No obstante, pasarse por alto los factores fundamentales para el éxito de la aplicación sólo porque se trate de una aplicación comprada.

18.14 ENTORNO GESTIONADO

El resultado de la implementación de estos tres elementos clave (procesos culturales, procesos de gestión y tecnología) será un entorno de desarrollo con el control de calidad incorporado, que permite que las mejoras se realicen en el proceso de desarrollo. Las aplicaciones de producción se beneficiaran de ello en forma de una mejora del rendimiento, una mejor integración con otras aplicaciones de la empresa y un mantenimiento más sencillo.

19 PRODUCTOS ADICIONALES Y NUEVAS APLICACIONES DE ORACLE

19.1 TECNOLOGÍA APLICADA, CLAVE DE INNOVACIÓN EMPRESARIAL

Las Aplicaciones Oracle basadas en Internet Computing Architecture (ICA) es el primer conjunto de aplicaciones de gestión empresarial que ofrecen totalmente los beneficios de la tecnología Internet. Por primera vez, las empresas y corporaciones pueden implementar aplicaciones empresariales que adopten todos los beneficios del software cliente-servidor, pero sin incorporar el coste y la complejidad que conlleva la instalación de dicho software en cada uno de los puestos de trabajo del usuario.

19.2 ORACLE DESIGNER

Oracle es un juego de herramientas productivo basado en modelos, para el diseño y generación de lógica completa de lado servidor y de aplicaciones para el despliegue en Web o en cliente/servidor.

Use el Oracle Designer cuando desee modelar o generar aplicaciones sin codificación:

- Modelación de aplicaciones nuevas.
- Aplicaciones existentes de ingeniería de retroceso
- Generación de Formas y Reportes de Oracle Developer, o páginas dinámicas HTML usando los generadores Oracle Designer.

19.3 ORACLEWARE

Producto integrado de Oracle Corporation y Novell, Inc. que ofrece cómputo en redes para las empresas.

Oracle es un nuevo modelo diseñado y desarrollado por Oracle Novell para unir los sistemas de redes y productos de manejo de información en una sola solución d cómputo integrada, fácil de comprar, instalar, administrar y soportar.

Oracleware está integrado está integrado por el sistema operativo de Novell Netware o Unixware en la base de datos de servidor cooperativo Oracle. Oracle Office y Oracle Glue.

19.3.1 Integrado con Netware 3.12, Oracle, Oracle Office y Oracle Glue

Oracle Office es un sistema de mensajería cliente-servidor y de programación de eventos por su parte, Oracle Glue provee conexiones entre información basada en el servidor Oracle y un amplio rango de herramientas “front – end”.

Un programa de instalación integrado permite a todos los componentes del servidor de estos productos ser instalados rápidamente desde la consola Netware.

Oracleware proporciona en un solo paquete soluciones de redes aprobadas para expandir y fortalecer las redes existentes de una corporación o para adecuar en tamaño, aplicaciones de misión crítica dentro del ambiente de redes cliente-servidor.

19.4 ORACLE MOBILE AGENTS

Crear una red inalámbrica ahora que las barreras de la transmisión inalámbrica de datos se están volviendo menos imponentes, los fabricantes de software están ayudando a los programadores a extraer todo el poder de las redes inalámbricas. Una de las primeras muestras de ese esfuerzo es Oracle, cuyo juego de herramientas Oracle Mobile Agents resuelve el problema de intercalar información entre clientes inalámbricos distantes y los servidores corporativos de bases de datos.

Además el juego de herramientas puede manejar la comunicación con cualquier programa y no sólo con los productos de bases de datos de Oracle.

Oracle Mobile Agents está estructurado alrededor de un modelo Cliente-Servidor que interpone un agente de software entre el cliente y el servidor. El agente se ejecuta en la máquina del servidor y es responsable de manejar solicitudes de clientes externos, toma las solicitudes y obtiene respuestas del servidor. Teóricamente, el programador agente puede incorporar suficiente inteligencia en el agente para reducir en forma significativa la cantidad de tráfico de red entre el servidor y el cliente.

19.5 MANUFACTURING

Oracle Manufacturing proporciona la ventaja competitiva necesaria para aumentar la competitividad y aumentar la rentabilidad de la empresa en el mercado globalmente competitivo de hoy en día. Oracle Manufacturing tiene un solo objetivo centrado en el servicio: la entrega puntual, de menor coste y máxima calidad. Desde esta perspectiva, el principal requisito es garantizar rápidamente fechas de entrega fiables para cada producto configurado específicamente.

Con las funciones de Oracle Web Configurator, Oracle Supply Chain Planning y Oracle Global Capable-To-Promise, puede, con toda confianza, ofrecer compromisos de entrega exactos a sus clientes para satisfacer sus necesidades, al tiempo que aumenta la velocidad con que se llega al mercado.

19.6 SUPPLY CHAIN

La versión 11 de Aplicaciones Oracle ofrece gran cantidad de nuevas funciones de gestión de la cadena de suministro para la empresa global. Con sus nuevos módulos de producto e importantes mejoras de las capacidades existentes, utilizar Oracle Supply Chain Management es una forma de obtener las herramientas necesarias para competir en la economía global en red. En los módulos de gestión de la cadena de suministro de Oracle se incluyen la planificación integrada de varias instalaciones y capacidades de ejecución que agilizan la gestión de suministro y demanda. Los canales de ventas hacen pedidos con ubicaciones de distribución, lo que, a su vez, permite satisfacer la demanda mediante distintas opciones de suministro.

El sistema automatiza todavía más las funciones de suministro internas y de contratación externa gracias al soporte completo de transacciones EDI.

19.7 FINANCIALS

Oracle Financials transforma el departamento financiero en una fuerza estratégica de la gestión corporativa, en constante cambio en la actualidad. Específicamente, Oracle Financials permite cumplir las expectativas en cuatro áreas críticas

- Incrementando la eficacia operativa mediante la implementación de las mejores prácticas del mundo
- Activando una eficaz gestión financiera global para obtener una expansión internacional sin límites en nuevos mercados

- Aumentando el soporte de decisiones de la empresa con funciones complejas de análisis, generación de grandes volúmenes de informes y consulta flexible
- Satisfaciendo las cambiantes demandas de la organización para lograr una mejora continua rápida

19.8 PROJECTS

Oracle Projects soporta todo el ciclo de vida útil de los proyectos, desde el principio hasta la liquidación, en toda la empresa. Desde el plan inicial del proyecto hasta la auditoría final, Oracle Projects satisface las necesidades de directores generales, directores de proyectos, directores financieros y distintos colaboradores de proyectos.

19.9 HUMAN RESOURCES

Oracle HRMS ofrece la capacidad necesaria para aumentar al máximo el potencial de la fuerza de trabajo mediante reclutamiento, plantilla, formación, compensación, beneficios y planificación eficaces.

Cada módulo forma parte de una amplia serie que proporciona un cuadro unificado del capital en HR dentro de la empresa. Oracle ofrece una interfaz gráfica de usuario elaborada y de fácil manejo que facilita el acceso a información de HR. Oracle HRMS soporta todas las funciones que lleva a cabo el departamento de HR, habilitando una mejor gestión de procedimientos, lo que ahorra tiempo y reduce los costes.

Habiendo analizado y evaluado las mejores prácticas de HR, ofrecemos un sistema de gran funcionalidad, aunque adaptable, que permite que las organizaciones implementen políticas de HR únicas. Más importante aún, al tratar la gestión de HR como un conjunto de procesos integrados y entrelazados, el

departamento de HR aporta valor añadido a las empresas garantizando soporte de las estrategias de HR y objetivos corporativos más amplios.

19.20 FRONT OFFICE

Oracle Front Office permite a la empresa una gran capacidad de respuesta ante los clientes, lo cual posibilita que las compañías comprendan mejor sus relaciones con los clientes, así como su valor y rentabilidad. Las compañías pueden concentrarse ahora en aumentar sus ingresos, reducir los costes de ventas y servicios, generar ventas adicionales en base a servicio de valor añadido y soporte personalizado al cliente. Las Aplicaciones Front Office de Oracle están formadas por módulos integrados dirigidos a la fuerza de ventas, atención al cliente, servicio, y automatización de centros de llamadas.

Este enfoque integrado es aplicable a toda la empresa u organización, bien sea directamente desde las oficinas corporativas, o indirectamente a través de los canales de socios colaboradores.

19.21 DESARROLLO DE APLICACIONES MULTICAPA

JDeveloper está integrado con el Oracle Application Server (OAS) 4.0, lo que permite a los desarrolladores construir aplicaciones multicapa. JDeveloper tiene varios wizards para facilitar la creación de aplicaciones distribuidas. Estas aplicaciones pueden ser ejecutadas en el servidor de aplicaciones de Oracle (OAS), ya sea utilizando el cartucho Java/Web (JWeb) o el cartucho Java/CORBA Objects (JCO).

- *Oracle Application Server Java/Web Cartridge.* Este ambiente permite a los desarrolladores ejecutar aplicaciones multicapa a través de clientes HTML ultra delgados. El *HTML-Java Wizard* de JDeveloper ayuda a los desarrolladores a agregar contenido dinámico a las páginas HTML. Otro

wizard, el *PL/SQL-Java Wizard* proporciona acceso Java a los procedimientos almacenados PL/SQL existentes.

- *Oracle Application Sever JCO Cartridge*. Este ambiente de ejecución esta dirigido a desarrolladores de aplicaciones multicapa basados en componentes. El *Deployment Wizard* de JDeveloper permite al desarrollador empaquetar objetos Java, de manera que puedan ser ejecutados en el Oracle Application Server como Objetos Java/CORBA. Con JDeveloper, los desarrolladores pueden también escribir applets y aplicaciones y reutilizar objetos Java/CORBA.

19.22 APLICACIONES DE ORACLE

Oracle Contabilidad y finanzas

Oracle Administración de la cadena de suministros

Oracle Fabricación

Oracle Fabricación

Oracle Proyectos

Oracle Recursos Humanos

Oracle Administración de Mercados

20 COMPARATIVO CON OTRAS BASES DE DATOS

20.1 EL ANALISIS

Como cualquier manejador de bases de bases de datos este tiene tanto sus fuerzas como sus debilidades y dependiendo de lo que se quiera trabajar en ella se va a hacer un análisis minucioso para ver cuál manejador será el más apropiado.

Cuando se habla de bases de datos, por lo común se recurre a la tipología que considera tres divisiones que pueden ser útiles para el análisis.

- ✓ Bases de datos para grandes empresas.

Son las que tienen por objeto abarcar gran cantidad de información, además de poder atender a muchos usuarios concurrentemente. Por lo general requieren de un servidor que debe mantener y quizá hasta procesar la información.

- ✓ Bases de datos para desarrollo de programas

Están orientadas al programador y le permiten desarrollar aplicaciones con objetivos concretos como facturación, nómina, inventario, análisis financiero o cualquier tema que pudiera surgir. Además las aplicaciones pueden distribuirse entre los usuarios.

- ✓ Bases de datos para usuario final.

Este tipo de bases permiten que el usuario elabore sus propios reportes, etiquetas, listados o cartas personalizadas. Son fáciles de usar y tienen mucha ayuda en

línea. Sin embargo, no son muy eficientes para manejar grandes cantidades de información y suelen ser básicamente para un solo usuario.

20.2 ALPHA FIVE

Alpha Software Corp desarrolló un manejador de base de datos Alpha Five que ofrece soluciones rápidas sin programación al crear sistemas de bases de datos. Sin embargo para los usuarios más avanzados ofrece un lenguaje de programación.

El enfoque para desarrollar aplicaciones en Alpha Five es crear tarjetas en las que el usuario define botones, variables y gráficas.

Las características de los elementos de una tarjeta se determinan al seleccionar sus elementos, así como las acciones a realizar y las condiciones que se deben cumplir. En forma opcional se puede definir código adicional en su lenguaje de programación Xbasic.

Xbasic es similar a Basis con Xbase. Cuenta con más de 300 instrucciones entre las que se encuentran: Import, Append y Update entre otras. Todas las operaciones que realiza el usuario son guardadas en código del mismo tipo.

Una vez que el usuario ha realizado una tarea, se pueden editar y modificar las instrucciones ejecutadas, mismas que al finalizar es posible guardar y modificar en un programa. Esto permite crear aplicaciones escribiendo un mínimo de líneas.

Alpha Five soporta 40 índices por archivo, 1024 campos por registro, 255 caracteres por campo, 20000 caracteres por registro y 24 archivos relacionados.

Cuenta con soporte para las redes Novell Netware, Windows para trabajo en grupo y redes compatibles con Netbios.

Requiere una computadora con procesador 386 DX o mejor. Microsoft Windows 3.1 o superior, se recomienda 8 MB de RAM y un monitor VGA o de mejor calidad para su óptimo desempeño.

20.3 CA-VISUAL OBJECTS

Clipper para Windows es su nombre informal y es un producto que en cierta medida es el sucesor del veterano Clipper. Quizá tardó demasiado tiempo en aparecer pues a penas vio la luz a finales de 1994 y con muchos problemas técnicos.

La versión llamada pre-release inundó por seis meses el mercado con la promesa de que algún día saldría por fin libre de errores. Dicha promesa tal parece que nunca se cumplirá pues, al poco tiempo de haber salido como producto final, apareció una colección de parches para esa versión.

Es probable, que con el tiempo, la situación cambie y el producto alcance un cierto grado de madurez pero por ahora, sigue siendo inestable.

Las ideas básicas del producto son visionarias: un ambiente de programación orientado a objetos con un depósito de bibliotecas prefabricadas que facilitan y aceleran el tiempo de desarrollo de un programa, es un compilador real lo cual hace que los programas ejecutables sean pequeños y puedan ser distribuidos sin costo a los clientes o usuarios del sistema, herramientas visuales de diseño de pantallas y reportes, soporte ODBC, y muchas características más que lo harían el líder en productos de desarrollo de bases de datos dentro de Windows.

La realidad es que el compilador es muy lento y requiere de un gran equipo para funcionar correctamente, las aplicaciones generadas son inestables y causan frecuentes fallos de protección general dentro de Windows, el enlace ODBC no es tan transparente como debiera ser, y muchos otros defectos adicionales que lo hacen, prácticamente un programa inutilizable.

20.4 DBASE 5.0

Este conocido acrónimo de Data Base o Base de Datos (DBase), originalmente se basó en un sistema similar desarrollado para el gobierno de Estados Unidos y que fue adaptado para las computadoras personales en el sistema operativo DOS. El consumidor estadounidense lo encontró sencillo y rápido, pero a la vez con capacidades que sólo se podían obtener en computadoras y programas de mayor costo, de ahí su gran aceptación.

En su versión IV, Ashton Tate tuvo serios problemas con la calidad del producto, debido a desacuerdos con sus empleados. Todo esto llevó a que la firma aceptara sus problemas financieros. Entre tanto, Borland International, que se rumoraba desarrollaba un compilador de Dbase, decidió comprar la compañía. Poco después se liberó una nueva versión de Dbase que resolvía muchos problemas de su antecesora. Sin embargo, Borland tenía ya un producto de base de datos (Paradox), hacía pensar que estaba lista la sepultura para Dbase. No obstante Borland ha continuado con Dbase, llevándolo a Windows y mejorándolo para DOS.

Se espera que pronto se libere la versión "Voyager" que permitirá: seguridad a nivel de cada tabla, más asistentes para la creación de reportes, tablas y formas, nuevas capacidades en el Open Database Connectivity Support, mejoras para el soporte de gráficas y para el control de formas de páginas múltiples, expansión de las capacidades de programación orientada por objetos y al fin un compilador, el cual es opcional.

20.5 DB/2 PARA OS/2

En sus inicios, IBM era una empresa que enfocó sus esfuerzos a satisfacer la demanda de grandes empresas con voluminosos equipos, cobrando también respetables sumas de dinero por sus servicios. Sin embargo, esta tendencia ha

cambiado a través de los años: primero fabricando equipo más pequeño como IBM/PC y luego con programas para esas computadoras que resuelven los problemas que hace 10 años solucionaba un sistema 36.

DB/2 es el resultado de ese esfuerzo de miniaturización en materia de programas. Es una base de datos relacional de 32 bits, orientada a empresas medianas y grandes que requieren compartir información numerosa y mantener buena velocidad de respuesta.

Su lenguaje estándar es SQL y aunque puede operar con cualquier otra base de datos, prefiere las IBM por lo que puede representar una gran ventaja para quienes cuentan con un sistema AS/400 o similar.

La integración con los productos de escritorio se puede hacer a través del ODBC de Windows o a través de *named pipes*. Aunque no incluye utilerías para conectarse a la red, sí lo están los clientes, para DOS y Windows que permiten la administración remota de la base de datos bajo el protocolo IPX/SPX, NETBIOS, TCP/IP, además de las IBM y Novell.

La principal herramienta de búsqueda se denomina Query Manager y es sencilla de usar pues corre en ambiente gráfico y permite seleccionar los campos sólo con el clic del ratón. Recuerda en gran medida al manejo de los queries de Access de Microsoft.

DB/2 es una excelente base de datos con buenas posibilidades dentro de la industria. Es evidente que los usuarios de IBM de línea dura se inclinarán por esta opción y no sufrirán ninguna decepción, pues cumple con todo lo que se puede esperar de un producto de este tipo.

Carece, sin embargo de los manuales necesarios para los primeros pasos de la instalación y el manejo administrativo del sistema.

Los requerimientos para DB/2 son: PC compatibles, con 8 MB de RAM y / MB de disco duro, sistema operativo OS/2 2.1 o superior y que esté habilitado para funcionar con cualquier protocolo que opere dentro de OS/2.

20.6 DELPHI 95

Es una herramienta de desarrollo visual que trabaja bajo una versión mejorada de Borland Pascal. El producto está enfocado a plataformas cliente-servidor y desarrollo de aplicaciones para bases de datos, aunque puede emplearse para escribir casi cualquier tipo de software.

Delphi 95 va más allá de lo hecho antes en Turbo Pascal. El sistema se enfoca al diseño de las pantallas, mediante el drag & drop (tómese y tírese), por lo que poner botones, controles, ventanas o imágenes significa tomar la opción del menú y colocarla en la ventana en donde se está creando la forma y escribir el código para que los componentes creados correspondan a los eventos.

Entre las características más relevantes de Delphi 95 se encuentran: ser un lenguaje compilado y que la arquitectura de los componentes es completamente orientado a objetos.

Los componentes modelados bajo el paradigma de los objetos, puede ser heredados y modificados por los desarrolladores sin necesidad de conocer cómo funcionaban los objetos originales.

Delphi también genera código ejecutable directamente en windows, soporta todos los servicios de OLE, DLL y DDE que windows provee y permite que el código

fuelle de Turbo pascal 7.0 pueda utilizarse en las aplicaciones de Delphi casi sin modificaciones.

Sus requerimientos son alrededor de 50 MB en disco duro, y al menos 4 MB de RAM. El sistema viene en disco compacto, pero puede también adquirirse en discos flexibles. Contiene alrededor de 14 manuales y herramientas adicionales para agilizar las labores comunes, como la de crear reportes y gráficas.

20.7 MICROSOFT SQL SERVER FOR WINDOWS NT

Este producto está orientado a empresas grandes o medianas con base de datos departamentales que manejan mucha información. Los requisitos para correrlo son, en comparación con productos similares, pequeños.

El servidor SQL (Standard query Language) puede operar con cualquier base de datos de este tipo, pues se comunica a través de protocolos estándar como TCP/IP, IPX/SPX y NETBEUI. Estas características le permiten tener gran integración con productos de escritorio como Access, Visual Basic, Fox para Windows, Approach y cualquier otro que soporte el mecanismo ODBC. También se puede comunicar con otros servidores de SQL como los de Oracle, Informix y Sybase.

La instalación es sencilla y no requiere que el usuario sea experimentado. Al terminar la instalación del producto debe reinicializarse Windows NT.

El producto es, entre los SQL evaluados, es más fácil para instalar, con menor número de problemas en esa etapa y el más fácil de configurar después.

Las herramientas de administración son muy sencillas de utilizar y en general se puede decir que llegan a ser intuitivas.

No se limitan al servidor pues gran parte de ellas pueden ser ejecutadas desde cualquier computadora que corra Windows 3.1 sin mayor problema.

Los programas administrativos de Microsoft SQL son: SQL Administrator, SQL Object Manager, isql e isql/w. Con ellos se pueden crear los dispositivos y las bases de datos, además de respaldar y diagnosticar los problemas del sistema, entre otras labores comunes.

La ventaja que representan estos programas es que todos pueden ser ejecutados tanto en el servidor como en cualquier estación de trabajo Windows que los tenga.

Sus requerimientos son: una computadora 486 DX a 33 MHz para el servidor con 16 MB de RAM, espacio en disco duro de 35 MB, sin considerar el reservado para la información que se manejará y el sistema operativo Windows NT (que ocupa alrededor de 70 MB más).

20.8 ORACLE CDE2

Oracle CDE2 (Cooperative Development Enviroment) es un ambiente visual integrado para hacer desarrollos abiertos en plataformas cliente/servidor. Consta fundamentalmente de tres partes Oracle Forms 4.5, Oracle Graphics 2.5 y Oracle Reports 2.5.

Estos productos maximizan la productividad gracias a sus componentes reusables, lo que permite la creación de aplicaciones escalables con una interfaz gráfica elegante y funcional a través de todas las plataformas más importantes cliente-servidor. Las herramientas de CDE2 hacen posible la creación de aplicaciones partiendo de la base de datos definidas previamente.

Debido a la manera como se interactua con el sistema, es posible generar programas prácticamente sin tener que escribir una sola línea de código, aunque es

claro que esto sólo tiene sentido cuando se trata de aplicaciones muy sencillas y sin mayores complicaciones.

El verdadero poder de CDE2 se basa en el lenguaje PL/SQL, el cual permite explotar todo el poder de la base de datos Oracle y de la plataforma cliente. Por supuesto, este lenguaje se emplea una vez que ha sido definida la base de datos a usar, así como las formas y ventanas en donde el usuario final va interactuar con el sistema. De hecho, Oracle CDE2 trabaja de manera similar a los otros ambientes visuales de desarrollo (Delphi 95 y Power Builder). Se colocan botones u otros componentes mediante drag & drop (mover y soltar) y después se programan las acciones a realizar vía el lenguaje PL/SQL, único usado en Oracle.

Una de las virtudes de los desarrollos en CDE2 es la portabilidad que mantienen las aplicaciones. De hecho, se pueden construir aplicaciones empleando Windows, Apple Macintosh o Motif y usarlas en cualquiera de estas máquinas e incluso en terminales en modo texto.

Gracias al Oracle Translation Manager, es posible modificar las aplicaciones para diferentes lenguajes humanos. Esto significa que una aplicación escrita en español puede ser modificada vía la herramienta mencionada, por el equipo inglés de programación en muy poco tiempo.

CDE2 es un sistema completo y complejo. Contiene muchas herramientas y un sinfín de posibilidades. Por ejemplo, el Oracle Forms 4.5 tiene entre sus muchas características un navegador para los objetos (ve todos y cada uno de ellos, los crea y remueve), una paleta de propiedades de los objetos y componentes del sistema, editor gráfico de menús, orientación a objetos, (para reagrupar componentes, cambiar y añadir clases de propiedades y métodos) y la posibilidad de heredar o ignorar propiedades de los objetos padres, entre otros.

Los otros componentes, Oracle Graphics 2.5 y Oracle Reports 2.5 poseen demasiadas características relevantes. Oracle Graphics es posible insertar gráficas generadas por Oracle en las formas del sistema o en los reportes directamente, recibir datos tanto de Reports como del Forms y graficar en una multitud de formatos (barrar, pie, líneas, etc.).

Por su parte, el Oracle Reports tiene opciones para imprimir tablas en formatos diversos, un control completo sobre los datos de las bases de información y puede agrupar datos, filtrarlos, hacer cálculos en las formas. Además soporta extensiones para C, C++, COBOL, FORTRAN, Pascal, Ada, PL/1.

Los requerimientos del sistema son 16 MBRAM y un mínimo de 100 MB en disco duro para poder instalar todas las aplicaciones.

20.9 POWERBUILDER 4.0

Este producto de la PowerSoft es toda una herramienta visual para generar aplicaciones en Windows que funcionen con bases de datos, es decir, el sistema está orientado a ser un manejador de bases de datos para plataformas cliente-servidor, lo que permite al usuario crear poderosos front-ends.

Para crear una aplicación es necesario definir la base de datos con la que se planea trabajar y después hacerlo con las ventanas en donde correrá la aplicación. Los botones de comandos, iconos, cajas de diálogo se pueden crear empleando el drag & drop en las ventanas creadas. Las acciones que deberán cumplir los botones en ellas debe escribirse en PowerScript (el lenguaje de PowerBuilder).

Finalmente se debe convertir todo el desarrollo en un programa ejecutable que corre bajo Windows (que sirve como front-end de la aplicación).

La versión de PowerBuilder tiene soporte completo para 16 y 32 bits bajo windows en procesadores Intel, incluyendo windows NT, Win OS/2 y Windows 3.11.

Esta versión de PowerBuildes incorpora las bases de datos de Lotus Notes, aparte de las ya soportadas: Oracle, SQL Server, Watcom SQL, ODBC, Informix, SQLBase, DB Manager y DB/2 y XDB. Sus requerimientos son: PC 386 o superior, 4 MB de RAM mínimo (recomendable 8 MB) y al menos 20 MB en disco duro.

20.10 SUPERBASE 3.0

Superbase Inc., es una compañía que nació en 1994. Su producto es una herramienta para el desarrollo de bases de datos. Esto no limita el acceso de otras fuentes de información ya que cuentan con extensiones SQL para enlazarse con bases de datos de este tipo. Por ejemplo, Informix, Sybase, Oracle y DB2, entre otros.

Superbase cuenta con un lenguaje SBL (Super Base Language), que es compatible con la sintaxis utilizada en Visual Basic y permite el manejo de programación orientada a objetos (encapsulamiento, herencia y polimorfismo).

El editor de formas permite arrastrar y dejar (drag & drop) elementos en su pantalla y sus editores modificar las características de cada elemento simplificando la programación de formas electrónicas. El uso del reporteador en programa resulta sencillo y su acceso es rápido. Superbase puede ser una opción económica y sencilla para el usuario de bases de datos.

Una de sus principales características es que requiere 12 MB de espacio en disco y puede funcionar en computadoras con poca memoria RAM (2 MB). Se puede instalar en modo monousuario o en red, sin costo adicional.

20.11 VISUAL FOX 3.0

Fox, es un producto de bases d datos que utiliza el lenguaje xbase. Este es un viejo rival de dBASE. Actualmente en México, Fox cuenta con más de 15, 000 usuarios registrados. El ambiente para el desarrollo permite consultar toda la información de un proyecto desde una ventana “dividida” en carpetas donde se consultan las distintas partes de un proyecto: documentos, clases, código y otros elementos que se requieren integrar al sistema que se están desarrollando. Al seleccionar un elemento, éste se expande revelando sus características. Para crear aplicaciones y sus componentes se utiliza drag & drop, elementos en pantalla, con lo que se pueden modificar las características de cada uno de los componentes.

Para facilitar el desarrollo en Fox se pueden crear clases para la programación con objetos (encapsular, heredar y polimorfismo). Esto se realiza en forma visual con la ayuda de un asistente, puede editar el código y escribir las instrucciones.

Fox puede ser una solución para el desarrollo de cliente-servidor, gracias a que la versión estándar incluye el Kit de Conectividad.

Con esta herramienta puede funcionar como una interfaz de usuario, permitiendo crear aplicaciones en una PC y migrarlas a un verdadero ambiente cliente servidor, sin tener que modificar el programa.

Requiere por lo menos 8 MB de RAM física para mantener un rendimiento adecuado en el ambiente de desarrollo, mientras que las aplicaciones exigen un mínimo de 6 MB. Adicionalmente se recomienda el uso de Windows para trabajo en grupos, Windows 95 o el NT.

20.12 GUPTA

Los productos que ofrece este proveedor de software para desarrollo de aplicaciones para cliente-servidor tienen el objetivo de proporcionar un buen desempeño y escalabilidad desde computadoras portátiles hasta servidores LAN.

20.12.1 Los principales de GUPTA

Gupta SQL Windows. Ambiente de desarrollo de aplicaciones en lenguaje de cuarta generación

Gupta SQL Base. Servidor de base de datos relacionales para un solo usuario o para una red local.

Gupta Quest. Herramienta para la administración y monitoreo de bases de datos.

Gupta SQL Console. Herramienta para la administración y monitoreo de bases de datos.

Gupta SQL Network. Producto que permite el intercambio de datos entre cliente-servidor, minicomputadoras y mainframes.

Gupta le puede ayudar en proyectos de gran escala cliente-servidor en una arquitectura abierta con su línea de productos.

20.13 INFORMIX

El objetivo de esta compañía es proveer el desarrollo de aplicaciones en la administración de datos para sistemas abiertos. Ofrece: servidores de datos, productos para la conectividad y herramientas para el desarrollo de aplicaciones y para el acceso por el usuario final.

20.13.1 Principales productos

OnLine Dynamic Server. Servidor de datos diseñado para obtener el mejor provecho de las arquitecturas de uno o varios procesadores en una computadora.

La versión 7 tiene paralelismo y su administración es en línea, además de contar con los productos de conectividad. La versión 8 tiene una replicación de datos a nivel de bases de datos tabla y podrá usar nuevas arquitecturas de procesadores.

Informix-OnLine. Este es un servidor de archivos para ambientes en donde existe mucha tráfico de información. También hay una versión para el uso de discos ópticos.

Informix-SE. Servidor de datos, fácil de usar, requiere poco mantenimiento y ofrece capacidades de cliente-servidor. No maneja mucha información y es para pocos usuarios.

C-ISAM. Librería de funciones en lenguaje C para crear y utilizar archivos secuenciales indexados.

STAR. Para la conectividad Informix ofrece este producto que permite transparentar las actividades en distintos servidores y en ubicaciones diferentes.

NET. Este producto crea conexiones cliente-servidor entre Informix y las herramientas de desarrollo en una computadora cliente.

NewEra. Una herramienta para el desarrollo permite crear aplicaciones cliente-servidor en un ambiente gráfico, orientado a objetos. Mientras que NewEra ViewPoint proporciona más herramientas para simplificar la creación de aplicaciones. Su versión profesional incluye la administración de estructuras.

Informix 4GL. Esta familia incluye las herramientas para el desarrollo de aplicaciones con un lenguaje de cuarta generación.

Informix-SQL. Representa una herramienta para el desarrollo que cuenta con editor de esquema, constructor de menús, de formas, editor SQL y un reporteador.

Por otra parte, Informix-ESQL para C, COBOL, Fortran y Ada permite incluir sentencias SQL en estos lenguajes.

20.14 PROGRESS

El ambiente de desarrollo de aplicaciones permite crear programas con interfaces gráficas. Está diseñado para desarrollar y utilizar sistemas en un ambiente para programación, un sistema de administración de bases de datos relacionales y la arquitectura Dataserver de Progress. Esta arquitectura permite construir sistemas, independientemente de la base de datos utilizada, que pueden ser usados por otros manejadores de bases de datos.

Progress cuenta con una división enfocada a los desarrolladores de Visual basic (crescent) para construir aplicaciones cliente-servidor para complementar o aumentar la capacidad de este lenguaje. Los productos que se ofrece para Visual Basic son:

Power Pack Pro. Conjunto de herramientas con controles para datos y la administración de código.

QuickPak. Herramientas para programadores novatos o expertos en el desarrollo de aplicaciones.

XREF. Herramienta para la administración de desarrollo de aplicaciones. Sirve para documentar y administrar aplicaciones. Está dirigido a los programadores de nivel medio y avanzado.

NetPack Professional. Colección de funciones y programas para redes que permite acceder los servicios avanzados de la red en los sistemas desarrollados.

PDQCom. Librería para comunicaciones seriales. Permite mejorar el control MSCom que se envía con Visual Basic Professional Toolkit, para el uso de servicios de comunicación.

Quickpak Scientific. Librería de subrutinas y funciones para el análisis numérico y estadístico. Util para resolver problemas analíticos en el desarrollo de aplicaciones científicas de ingeniería e investigación de mercados, entre otras técnicas.

20.15 EL CONCEPTO CLIENTE- SERVIDOR

Este modelo significa que para manejar una base de datos se requiere más de una máquina. En una computadora llamada servidor se almacena la información, mientras que en otro equipo denominado cliente se consulta ésta. Este modelo es muy útil cuando hay una base de datos centralizada en una organización, para muchos usuarios simultáneos.

Aunque hoy en día es tema de moda, el modelo cliente-servidor lleva muchos años de existir, sin embargo, no era una buena opción pues exigía poder de cómputo tanto en el equipo servidor como en el cliente. No obstante, con el abaratamiento del hardware, este modelo ha cobrado importancia en los últimos años.

Los beneficios que tiene este esquema, es que la computadora llamada servidor sólo se dedica a explorar y mantener la base de datos por lo que deberá tener muy bien optimizados los mecanismos de búsqueda y mantenimiento.

Por otro lado, la computadora llamada cliente se encargará de mostrar al usuario la información solicitada en forma gráfica y claramente expuesta, sin degradar la velocidad de las consultas de los demás usuarios.

21 GLOSARIO

Bases de Datos: Es un conjunto de datos almacenados.

Esquemas: El conjunto de objetos que posee una cuenta de usuario se denomina esquema (shema) del usuario.

Procedimiento: Un procedimiento (procedure) es un bloque de sentencias PL/SQL que se almacena en el diccionario de datos y al que pueden llamar las aplicaciones.

RDBMS: Sistema de gestión de bases de datos relacionales.

Tablespaces: Espacios de tablas es una división lógica de una base de datos.

22 CONCLUSIONES

Al término del presente trabajo podemos darnos cuenta de la fuerza con que cuenta ORACLE, se ven ventajas y desventajas, así como sus características. Vemos que es una herramienta muy potente, y la cual nos puede ayudar a la solución de problemas de cualquier índole, siempre y cuando se manejen bases de datos.

Las intensas presiones competitivas de hoy requieren respuestas rápidas y ORACLE permite el mejoramiento continuo de los procesos de negocios, sistematiza por completo la empresa con más de 30 módulos integrados de software cliente-servidor para la administración contable, administración de la cadena de suministros, fabricación, sistemas de proyectos, recursos humanos y la administración de mercados.

La solución para automatizar e integrar los sistemas administrativos de las empresas medianas, que requieren crecer ante las demandas del Siglo XXI; integra a una suite de aplicaciones financieras listas para trabajar en ellas de inmediato gracias a que se encuentran parametrizadas, preconfiguradas y cuentan con metodología de implementación. Fast Forward *la vía rápida*, fácil y económica de hacer eficiente y efectivo el control administrativo de su empresa, con la visión del año 2001.

ORACLE ofrece el desarrollo basado en modelos con generación de aplicaciones en ingeniería de reversa (de dibujos a códigos y viceversa).

Cuando se cuenta con ciclos de desarrollo más rápidos y comunicación mejorada entre usuarios y desarrolladores, ORACLE proporciona un ambiente sorprendentemente productivo que protege su inversión en desarrollo y asegura que la aplicación que se construye sea la correcta desde el principio. ORACLE le permite eliminar la complejidad de las aplicaciones de escritorio e incrementa el acceso hacia esas aplicaciones corporativas a costos más bajos.

ORACLE es una herramienta de desarrollo de aplicaciones para crear formas de bases de datos de alto rendimiento, para empresas, reportes y gráficas para el despliegue en Web o cliente/servidor; provee un ambiente visual declaratorio, provee una plataforma escalable, robusta, segura y extensible para soportar aplicaciones capa intermedia (middleware).

23 BIBLIOGRAFIA

- 📖 ORACLE 7 the COMPLETE REFERENCE
George Koch
Ed. Mc Graw Hill

- 📖 ORACLE 7.0 Administration & Management
Michael R. Ault
Ed. Willey – Qed

- 📖 ORACLE Manual del Administrador
Kelvin Loney
Ed. Mc Graw Hill

- 📖 SOLUCIONES AVANZADAS
Año 6 No. 62 p. 24

- 📖 PERSONAL COMPUTING MEXICO
Año 7 No. 86 p. 38-51

- 📖 BYTE MEXICO
Vol. 22 No. 9 septiembre p. 111-113

- 📖 BYTE MEXICO
Año 7 No. 72 p. 30

- 📖 BYTE MEXICO
Año 6 No. 88 p. 50-52

- 📖 BYTE MEXICO
Año 9 No. 93 p. 34



BYTE MEXICO

Año 9 No. 91 p. 52-54 1995

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA