



ESCUELA SUPERIOR POLITECNICA DE CHIMBORAZO

FACULTAD DE INFORMÁTICA Y ELECTRÓNICA

ESCUELA DE INGENIERÍA DE SISTEMAS

**“DISEÑO E IMPLEMENTACIÓN DE LA ARQUITECTURA
DE DATOS BASADA EN COMPARATIVAS DE
RENDIMIENTO ENTRE SISTEMAS DE GESTIÓN DE
BASES DE DATOS.”**

TESIS DE GRADO

Previa obtención del título de:

INGENIERO EN SISTEMAS INFORMÁTICOS

Pilar Azucena Quizhpe Palaquibay

RIOBAMBA – ECUADOR

2009

Al Ing. Eduardo Villa, e Ing. Patricio Moreno, Director, y miembro de tesis de Grado, por su efectiva participación, apoyo y contribución en el desarrollo del presente trabajo investigativo.

Al Ing. Carlos Cevallos, Rector del Colegio "AMERICANO", y Gerente de la Empresa "**GuíaMundial.com**" por la motivación y apoyo brindado en cada una de las fases investigativas de este trabajo.

A Jesucristo mi amado Señor y Rey, quien ha sido mi fortaleza y mi refugio, en el trayecto de mi vida.

A mí adorado esposo Alexander Alcívar, quien se convirtió en la razón de mi vida, gracias por el amor que me diste... aunque estés en el cielo siempre te amaré...

A mis hijos, Alexander, David y Scarlett, que son la motivación para seguir viviendo y luchando cada día.

A mis padres Miguel y Azucena por haberme dado la vida, y ser el apoyo en estos momentos tan difíciles...

A mis hermanos Darwin y Marvin y mi suegra Corina... gracias por su apoyo incondicional.

Nombres

Fecha

Firma

Dr. Romeo Rodríguez

**Decano de la Facultad de Informática
y Electrónica.**

Ing. Iván Menes Camejo

**Director de la Escuela de Ingeniería de
Sistemas.**

Ing. Eduardo Villa

Director de Tesis.

Ing. Patricio Moreno

Miembro del Tribunal.

Lcdo. Carlos Rodríguez

Director Centro de Documentación.

Nota de la tesis:

“Yo, PILAR AZUCENA QUIZHPE PALAQUIBAY, soy responsable de las ideas y resultados expuestos en esta tesis; y el patrimonio intelectual de la tesis de grado pertenece a la ESCUELA SUPERIOR POLITÉCNICA DE CHIMBORAZO”.

Pilar Azucena Quizhpe Palaquibay

ÍNDICE GENERAL

PORTADA

AGRADECIMIENTO

DEDICATORIA

FIRMAS RESPONSABLES

ÍNDICE GENERAL

ÍNDICE DE FIGURAS

ÍNDICE DE TABLAS

ÍNDICE DE GRÁFICOS

INTRODUCCIÓN

CAPÍTULO I

1. GESTIÓN DE DATOS	15
1.1 Información y Sistemas	16
1.2 Sistemas de Información	17
1.3 Componentes de un Sistema de Información.....	17
1.4 Gestión de la información en los Sistemas de Información.....	17
1.5 Sistemas de Gestión de Bases de Datos.....	18
1.5.1 Origen de los Sistemas de bases de datos	18
1.5.2 Sistemas de Gestión de Bases de Datos (SGBD).....	21
1.5.3 Características de los SGBD	21
1.5.4 Estructura de los SGBD	22
1.5.5 Clasificación de los Sistemas de Gestión de Bases de Datos	25
1.5.6 Funciones de los sistemas de gestión de bases de datos.....	26

CAPÍTULO II

2. PRINCIPALES FUNCIONES Y CAPACIDADES DE LOS SGBD	29
2.1 Catálogo del Sistema	30
2.1.1 Estructura del Catálogo de Datos.....	30
2.2 Almacenamiento e Indexación de Datos	31
2.2.1 Organización de archivos y métodos de direccionamiento	32
2.2.2 Indexación.....	34
2.2.3 Particionamiento de Tablas e Índices.....	36
2.3 Procesamiento y Optimización de Consultas.....	37
2.3.1 Medidas del coste de la evaluación una consulta	37
2.4 Procesamiento de Transacciones y Control de Concurrencia.....	38
2.4.1 Transacción	38
2.4.2 Propiedades ACID.....	38
2.4.3 Estados de una Transacción	39
2.4.4 Ejecuciones Concurrentes	39
2.4.5 Importancia del control de concurrencia	40
2.4.6 Esquemas de control de concurrencia.....	41
2.5 Sistema de Recuperación.....	44
2.5.1 Transacciones Secuenciales y Recuperación	44
2.5.2 Transacciones Concurrentes y Recuperación	48
2.5.3 Fallo con pérdida de almacenamiento no volátil	48
2.5.4 Sistemas Remotos de Copias de Seguridad	48

CAPÍTULO III

3. FUNCIONAMIENTO DE LOS SISTEMAS DE GESTIÓN DE BASES DE DATOS ORACLE, SQL SERVER, MySQL.....	50
3.1 Introducción.....	50

3.2	Oracle	51
3.2.1	Arquitectura del Sistema	51
3.2.2	Catálogo del Sistema	54
3.2.3	Almacenamiento e Indexación.....	56
3.2.4	Particionamiento de Tablas	60
3.2.5	Variaciones y Extensiones SQL	61
3.2.6	Procesamiento y Optimización de Consultas	62
3.2.7	Transacciones y Concurrencia	64
3.2.8	Recuperación	68
3.3	MySQL	69
3.3.1	Arquitectura del Sistema	69
3.3.2	Catálogo del Sistema	72
3.3.3	Almacenamiento de Datos	73
3.3.4	Indexación de Datos	74
3.3.5	Particionamiento de Tablas	75
3.3.6	Variaciones y Extensiones SQL	75
3.3.7	Procesamiento y Optimización de Consultas	76
3.3.8	Transacciones y Concurrencia	78
3.3.9	Recuperación	83
3.4	SQL Server 2005	84
3.4.1	Arquitectura del Sistema	84
3.4.2	Catálogo del Sistema	85
3.4.3	Almacenamiento de Datos	86
3.4.4	Indexación de Datos	87
3.4.5	Particiones.....	87
3.4.6	Variaciones y Extensiones SQL	87
3.4.7	Procesamiento y Optimización de Consultas	90
3.4.8	Transacciones y Control de concurrencia	93
3.4.9	Recuperación	95
3.5	Características Primarias o Funcionalidad de los SGBDs	96
3.6	Comparativa de Características Primarias	104
3.7	Comparativa de Dimensionamiento	105
3.8	Comparativa de Costos de Licenciamiento	106
3.9	Comparativa de Soporte	108
3.9.1	Soporte Oracle	108
3.9.2	Soporte SQL Server	109
3.9.3	Soporte MySQL	109
3.10	Comparativa de Herramientas de Administración	111

CAPÍTULO IV

4.	ANÁLISIS DE RENDIMIENTO DE LOS SISTEMAS DE GESTIÓN DE BASES DE DATOS ORACLE, SQL SERVER Y MySQL.	112
4.1	Introducción	112
4.2	Descripción del Problema	113
4.3	Plataforma Hardware	113
4.4	Plataforma Software	113
4.5	Técnicas utilizadas en la comparación	114
4.6	Procedimiento previo a la ejecución de los test de rendimiento	114
4.7	Instalación de los Sistemas de Gestión de Bases de Datos	114
4.8	Prototipo de la Base de Datos	115
4.9	Criterios de Medición	117
4.10	Determinación de los escenarios de prueba	117
4.11	Definición de los grados de complejidad de las consultas	119
4.12	Determinación de las variables implícitas en el análisis	119
4.13	Diseño de los Test de Rendimiento	119
4.13.1	Test de Rendimiento No 01	120
4.13.2	Test de Rendimiento No 02	121
4.13.3	Test de Rendimiento No 03	122
4.13.4	Test de Rendimiento No 04	123
4.14	Ejecución de los SGBDS	124

4.14.1	SQL Server	124
4.14.2	MySQL	127
4.14.3	Oracle	128
4.15	Test de Rendimiento: Carga Masiva de Datos	130
4.15.1	Tablas de Lectura del Tiempo de los SGBD.....	130
4.15.2	Comparativa del tiempo de ejecución para la operación INSERT	131
4.15.3	Cálculo de la velocidad de inserción de los SGBD	132
4.16	Test de Rendimiento: Consultas de diversa complejidad	133
4.16.1	Tablas de Lectura de Tiempos de SQL Server	133
4.16.2	Comparativa de tiempos entre consultas de diversa complejidad SQL Server .	134
4.16.3	Resultados de la ejecución de Oracle.....	137
4.16.4	Resumen de Resultados para la operación consulta en ORACLE	138
4.16.5	Resultados de la ejecución de MySQL.....	141
4.16.6	Resumen de Resultados para la operación consulta en MYSQL.....	142
4.17	Comparativa entre los SGBD para consultas de diversa complejidad..	144
4.17.1	Resumen de Resultados para Consultas de Complejidad Simple.....	144
4.17.2	Resumen de Resultados para Consultas de Complejidad Media	145
4.17.3	Resumen de Resultados para Consultas de Complejidad Alta.....	146
4.18	Modelo de Puntuación	147
4.19	Resultados Generales de la medición.....	147
4.19.1	Comparativa de las Velocidades de Inserción entre los SGBD	147
4.19.2	Comparativa Velocidad de Recuperación de Datos	148
4.20	Comparativa General	151

CAPÍTULO V

5. DISEÑO DE LA ARQUITECTURA DE DATOS MEDIANTE EL ANALISIS DE REQUISITOS..... 152

5.1	Introducción.....	152
5.2	Especificación de requerimientos para la Base de Datos.....	153
5.2.1	Introducción	153
5.2.2	Sistema Actual	154
5.2.3	Sistema Propuesto.....	154
5.2.4	Diseño de la Arquitectura de Datos	154
5.2.5	Diseño Lógico.....	159
5.2.6	Implementación de la Base de Datos relacional	162
5.2.7	Tablas de la Base de Datos	166

CONCLUSIONES

RECOMENDACIONES

RESUMEN

SUMMARY

BIBLIOGRAFÍA

GLOSARIO

ÍNDICE DE FIGURAS

Figura I.01. Evolución de la Tecnología de Bases de Datos.....	20
Figura I.02. Estructura del sistema de base de datos	24
Figura II.01. Estructura de archivos en agrupaciones con cadena de punteros.....	33
Figura II.02. Estados de una Transacción.....	39
Figura II.03. Arquitectura de los sistemas remotos de copias de seguridad.	49
Figura III.01. Procesos de segundo plano Oracle	53
Figura III.02. Estructuras Físicas de Almacenamiento de ORACLE	54
Figura III.03. Estructuras Lógicas de Almacenamiento.....	56
Figura III.04. Arquitectura de MySQL	70
Figura III.05. Vista Lógica de la Arquitectura de MySQL.....	72
Figura III.06. Arquitectura de SQL Server.....	84
Figura III.07. Componentes del Motor Relacional y de Almacenamiento.....	84
Figura III.08. Página de datos de SQL Server.....	87
Figura III.09. Plan de Consulta	90
Figura IV.01. Interfaz de VmWare	115
Figura IV.02. Prototipo en SQL Server	116
Figura IV.03. Prototipo en MYSQL.....	117
Figura IV.04. Diseño de la operación en SQL Server.....	124
Figura IV.05. Acceso a SQL Server Profiler.....	124
Figura IV.06. Crear nueva traza.....	125
Figura IV.07. Propiedades de la Traza	125
Figura IV.08. Ejecución del Proceso.....	126
Figura IV.09. Duración del Proceso	126
Figura IV.10. Diseño de la Operación en MySQL	127
Figura IV.11. MySQL Command Line	127
Figura IV.12. MySQL Browser.....	128
Figura IV.13. iSQL Plus.....	128
Figura IV.14. Activación del reloj interno.	129
Figura IV.15. Tiempo de respuesta de Oracle	129
Figura V.01. Entidades	157
Figura V.02. Modelo Entidad - Relación.....	158
Figura V.03. Modelo relacional.....	161
Figura V.04. Tabla Continente	162
Figura V.05. Tabla País.....	162
Figura V.06. Tabla Ciudad.....	163
Figura V.07. Tabla Empresa	163
Figura V.08. Tabla Categoría	164
Figura V.09. Tabla Paquete	164
Figura V.10. Tabla detalle_paquete.....	165
Figura V.11. Tabla ciud_emp.....	165
Figura V.12. Tabla emp_cat	166
Figura V.13. Tablas de la Base de Datos	166

ÍNDICE DE TABLAS

Tabla I.I. Características de los métodos de acceso a datos	20
Tabla III.I. Estructuras Lógicas de Almacenamiento	57
Tabla III.II. Tipos de tablas de Oracle	58
Tabla III.III. Tipos de Índices de Oracle	59
Tabla III.IV. Particionamiento en Oracle	61
Tabla III.V. Bloqueos para sentencias DDL.....	66
Tabla III.VI. Reservas a nivel de tablas	66
Tabla III.VII. Características de los Motores de Almacenamiento	73
Tabla III.VIII. Modelos de Recuperación	95
Tabla III.IX. Experiencia.....	96
Tabla III.X. Frecuencias en la Experiencia.....	96
Tabla III.XI. Interfaces y conectores de programación.....	97
Tabla III.XII. Frecuencias Licenciamiento.....	97
Tabla III.XIII. Interfaces y conectores de programación	98
Tabla III.XIV. Frecuencias de Interfaces y Conectores de programación	98
Tabla III.XV. Sistemas Operativos Soportados.....	99
Tabla III.XVI. Frecuencias para Sistemas Operativos Soportados.....	99
Tabla III.XVII. Objetos Soportados	100
Tabla III.XVIII. Frecuencias de Objetos Soportados	100
Tabla III.XIX. Características Fundamentales	101
Tabla III.XX. Frecuencias en Características Fundamentales.....	101
Tabla III.XXI. Tipos de Índices	102
Tabla III.XXII. Frecuencia en Tipos de Índices Soportados	102
Tabla III.XXIII. Tablas y Vistas.....	103
Tabla III.XXIV. Frecuencia de Esquemas de Particionamiento.....	103
Tabla III.XXV. Cuadro Comparativo de Características Primarias.....	104
Tabla III.XXVI. Dimensionamiento y Capacidad de los SGBD.....	105
Tabla III.XXVII. Precios Licencias Oracle.....	106
Tabla III.XXVIII. Precios Licencias SQL Server.....	106
Tabla III.XXIX. Precios Licencias MySQL Server	107
Tabla III.XXX. Costos.....	107
Tabla III.XXXI. Herramientas de Administración	111
Tabla IV.I. Escenarios de Prueba	118
Tabla IV.II. Grados de Complejidad	119
Tabla IV.III. Variables e Indicadores	119
Tabla IV.IV. Test de Rendimiento No 01	120
Tabla IV.V. Test de Rendimiento No 02	121
Tabla IV.VI. Test de rendimiento No 03.....	122
Tabla IV.VII. Test de Rendimiento No 04	123
Tabla IV.VIII. Tiempo de ejecución de Oracle para carga masiva de datos	130
Tabla IV.IX. Tiempo de ejecución de SQL Server para carga masiva de datos.....	130
Tabla IV.X. Tiempo de ejecución de MySQL para carga masiva de datos	130
Tabla IV.XI. Tiempo de Ejecución para carga masiva de datos entre SGBD	131
Tabla IV.XII. Tiempo de respuesta de SQL Server para consultas simples.....	133
Tabla IV.XIII. Tiempo de respuesta de SQL Server para consultas medias	133
Tabla IV.XIV. Tiempo de respuesta de SQL Server para consultas altas.	133
Tabla IV.XV. Tiempo de ejecución de SQL Server para consultas de diversa complejidad ..	134
Tabla IV.XVI. Tiempo de respuesta de Oracle en consultas de complejidad simple.	137
Tabla IV.XVII. Tiempo de respuesta de Oracle en consultas de complejidad media.....	137
Tabla IV.XVIII. Tiempo de respuesta de Oracle en consultas de complejidad alta.....	137
Tabla IV.XIX. Tiempo de respuesta de Oracle en consultas de diversa complejidad.	138
Tabla IV.XX. Tiempo de Ejecución de MySQL para consultas de complejidad simple	141
Tabla IV.XXI. Tiempo de Ejecución de MySQL para consultas de complejidad media.....	141
Tabla IV.XXII. Tiempo de Ejecución de MySQL para consultas de complejidad alta.....	141
Tabla IV.XXIII. Tiempo de MySQL para consultas de diversa complejidad	142
Tabla IV.XXIV. Tiempo de Ejecución de los SGBD para consultas simples.....	144
Tabla IV.XXV. Tiempo de Ejecución de los SGBD para consultas complejidad media.....	145
Tabla IV.XXVI. Tiempo de Ejecución de los SGBD para consultas complejidad alta	146

Tabla IV.XXVII. Velocidad de Inserción de los SGBD	147
Tabla IV.XXVIII. Velocidad de recuperación - Volumen de información bajo a normal	148
Tabla IV.XXIX. Velocidad de recuperación - Volumen de información alto	149
Tabla IV.XXX. Comparativa General	151
Tabla V.I. Entidad Continente.....	159
Tabla V.II. Entidad País	159
Tabla V.III. Entidad Ciudad	159
Tabla V.IV. Entidad Empresa.....	160
Tabla V.V. Entidad Categoría.....	160
Tabla V.VI. Entidad Paquete.....	160

ÍNDICE DE GRÁFICOS

Grafica III.01. Puntuación Experiencia.....	96
Grafica III.02. Puntuación Licenciamiento	97
Grafica III.03. Puntuación Interfaces y Conectores de Programación	98
Grafica III.04. Puntuación Sistemas Operativos Soportados	99
Grafica III.05. Puntuación Objetos Soportados	100
Grafica III.06. Puntuación Características Fundamentales	101
Grafica III.07. Puntuación Tipos de Índices Soportados	102
Grafica III.08. Puntuación Esquemas de Particionamiento.....	103
Grafica III.09. Puntuación General Parte I	104
Grafica IV.01. Tiempo de respuesta de los SGBD en la operación INSERT	131
Grafica IV.02. Tiempo en consultas de diversa complejidad en SQL Server	134
Grafica IV.03. Tiempos en consultas de diversa complejidad en SQL Server	134
Grafica IV.04. Comparativa para consultas de diversa complejidad en Oracle.....	138
Grafica IV.05. Comparativa para consultas de diversa complejidad en Oracle.....	138
Grafica IV.06. Consultas de diversa complejidad en MySQL.....	142
Grafica IV.07. Consultas de complejidad simple en bases de datos pequeñas	144
Grafica IV.08. Consultas de complejidad simple en bases de datos grandes	144
Grafica IV.09. Consultas de complejidad media en bases de datos pequeñas.....	145
Grafica IV.10. Consultas de complejidad media en bases de datos grandes.....	145
Grafica IV.11. Consultas de complejidad alta en bases de datos pequeñas.....	146
Grafica IV.12. Consultas de complejidad alta en bases de datos grandes.	146
Grafica IV.13. Comparativa de Velocidad de Inserción	147
Grafica IV.14. Comparativa de Velocidad de Respuesta de Datos	148
Grafica IV.15. Comparativa de Velocidad de Respuesta de Datos	149

INTRODUCCIÓN

La información se ha posicionado como uno de los principales recursos que poseen las empresas actualmente. Los entes que se encargan de la toma de decisiones, han comenzado a comprender que la información no es sólo un subproducto de la conducción empresarial, sino que es uno de los factores más críticos para la determinación del éxito o fracaso de sus empresas. Si deseamos maximizar la utilidad que posee nuestra información, la empresa debe considerar su manejo de forma correcta y eficiente, tal y cómo se manejan los demás recursos existentes.

En este contexto, aparecen los sistemas de gestión de bases de datos que han pasado de ser simples repositorios de almacenamiento de datos, a verdaderas herramientas inteligentes que gestionan la información, haciéndola eficiente para aquellas personas que tienen determinadas expectativas y criterios para su interpretación, y convirtiéndose en parte fundamental de la estrategia de las empresas al ser la infraestructura de los sistemas de información empresarial.

En la actualidad, encontramos sofisticadas tecnologías de información dentro de los sistemas de gestión de bases de datos, por lo que resulta imprescindible contar con elementos de juicio a la hora de optar por una u otra solución. ¿Cuál se adecua mejor a las necesidades de la empresa? Para ello resulta indispensable observar el funcionamiento de los Sistemas de Gestión de Bases de Datos, además de evaluar el rendimiento de los SGBD, y así poder determinar las características que más convienen a la empresa.

La presente investigación tiene como objetivo efectuar un análisis comparativo que sirva como referencia para la selección de un determinado Sistema de Gestión de Bases de Datos, haciendo énfasis fundamentalmente en dos puntos de vista:

- Funcionamiento de los Sistemas de Gestión de Bases de Datos, en donde se estudian los servicios más relevantes y características de los sistemas es estudio, y ,
- El rendimiento de los sistemas en donde se determinan escenarios muy bien diferenciados, en donde cada sistema tiene su propio comportamiento.

La investigación comprende toda esta temática en cinco capítulos, que se encuentra estructurados de la siguiente manera:

Capítulo I: Comprende la primera parte teórica de la investigación, menciona conceptos de importancia sobre las gestión de datos y los sistemas que los coadyuvan, como los sistemas de información, sistemas de bases de datos, y sus características y funcionamiento en general.

Capítulo II: Comprende el estudio analítico del funcionamiento interno de los sistemas de gestión de bases de datos en general, enfatizando en los siguientes puntos.

- Catálogo del Sistema
- Almacenamiento e Indexación De Datos
- Procesamiento y Optimización De Consultas
- Procesamiento de Transacciones
- Control de Concurrencia
- Sistema de Recuperación
- Herramientas de Administración

Capítulo III: Comprende el estudio analítico del funcionamiento interno de cada uno de los sistemas de gestión de bases de datos seleccionados para nuestro estudio, estos son Oracle 10g, MySQL 5.1 y SQL Server 2005. Se enfatizará en las principales funciones y modo de trabajo analizados en el capítulo II.

Capítulo IV: Comprende la parte práctica de la investigación, para lo cual se crea un prototipo de evaluación, sobre el cual se diseñaran test de rendimiento, su correspondiente ejecución, y toma de datos, para posteriormente observar el comportamiento de los SGBD en gráficas comparativas, que nos permitan llegar a determinar las conclusiones y recomendaciones.

Capítulo V: Comprende el diseño de la arquitectura de datos, para la aplicación web de la empresa “**GuíaMundial.com**”, determinando previamente los requerimientos de la misma.

CAPÍTULO I

1. GESTIÓN DE DATOS

La gran demanda de información y la exigencia en la toma de decisiones inmediatas, han marcado un nuevo hito histórico en el desarrollo de sofisticadas tecnologías que van desde las redes de datos e Internet, hasta tener al conocimiento y la información en nuestros bolsillos (WAP). Estas tecnologías sirven como infraestructura especializada a los sistemas de información, mismos que explotan al máximo todas de sus capacidades.

Dentro del ámbito empresarial, los sistemas de información dependiendo de su tipo, facilitan la planificación, gestión y control de cada actividad desarrollada por los diferentes elementos de la empresa, mediante datos homogéneos obtenidos en tiempo real, variables estadísticas que permiten evaluar cuantitativamente los diferentes departamentos empresariales, de modo que se puedan aplicar criterios de eficacia, productividad y rentabilidad, junto al diseño de tiempos exigible a cada tarea desarrollada en la empresa, y así verificar la correcta actuación de acuerdo con los objetivos empresariales establecidos.

Otros sistemas ofrecen capacidades de suministro de valores y tasas, que permiten conocer y comparar los datos de la empresa con otros del mismo sector productivo, facilitando una mejor gestión de la decisión, pues los datos se poseen en el momento en que se producen o cuando todavía permite ciertos márgenes de maniobra. Todas estas perspectivas de

administración han crecido paralelamente a la evolución de la tecnología y se han logrado a través de ***los sistemas de bases de datos, los cuales se han convertido en el fundamento de los sistemas de información para la gestión corporativa.***

1.1 Información y Sistemas

“Según Toffler Alvin, el desarrollo y evolución de la humanidad ha pasado por tres eras, las cuales se han originado de las tres grandes transformaciones históricas de la generación de riqueza: la revolución agrícola que se basó en el poder de la tierra y el trabajo agrícola, la revolución industrial que se centró en las máquinas y las grandes industrias y la revolución de la información.

La información siempre ha existido pero posiblemente se encontraba dispersa en muchos sitios y en otras tantas cabezas por lo cual constituían datos crudos con valor pero que requerían prepararlos para una utilización productiva. La información es considerada como una representación de la realidad, como la expresión de conocimientos, la base para el funcionamiento de las organizaciones, el soporte para los procesos gerenciales, un recurso crítico de las organizaciones y un eslabón que une todos los componentes de la organización.

El proceso consiste en recolectar, organizar, analizar y contextualizar los datos para transformarlos en información, la cual mediante un proceso de análisis, evaluación y obtención de conclusiones útiles se convierte en conocimiento. De esta manera el conocimiento se transforma en el nuevo factor de poder económico y en el activo más importante de la estructura social. El conocimiento es importante pero puede ser volátil y su administración es compleja y necesaria por ello hay que garantizar su permanencia y una estructuración que permita su distribución y de esta forma convertirse en un activo real, lo que constituye el verdadero capital intelectual. ”

Los sistemas actuales cumplen con esas demandas, pues no solo almacenan datos, sino también los gestionan, haciendo que esta información pueda ser eficiente para aquellas personas que tienen determinadas expectativas y criterios para su interpretación.

1.2 Sistemas de Información

... Son un conjunto de elementos tecnológicos relacionados que cooperan para la consecución de un determinado fin empresarial, profesional o personal¹.

Un **sistema** de información (SI) es un conjunto organizado de elementos que interactúan entre sí, para procesar los datos y la información (incluyendo procesos manuales y automáticos) y distribuirla de la manera más adecuada posible en una determinada organización en función de sus objetivos.

Un sistema de información puede ser planteado como un proceso permanente de recolección, procesamiento, presentación, interpretación, uso, y aplicación de la información en la toma de decisiones, redefinición de objetivos, recursos y estructuras de la organización. Es así como las empresas son consideradas esencialmente como canales para el flujo de información, ideas e influencias, con el propósito de tomar decisiones y ejecutar acciones

1.3 Componentes de un Sistema de Información

Los elementos de un **sistema** de información son de 4 tipos:

- **Recursos Físicos:** hardware, software, documentos, archivadores, equipos de telecomunicaciones y de informática, etc.
- **Recursos Humanos:** usuarios y personal informático encargado de la alimentación de datos y utilización de los resultados que genera el sistema
- **Protocolos:** normas o métodos que rigen el uso y transmisión de los flujos de información.
- **Datos e Información**

1.4 Gestión de la información en los Sistemas de Información

El impacto significativo que ha tenido la información sobre la planificación y la toma de decisiones en las organizaciones, ha conducido a un reconocimiento siempre creciente de que la información es un recurso que tiene valor, y por lo tanto necesita estar organizada y

¹ **CABRERA SANCHEZ**, Gregorio. Sistemas de Gestores de Bases de Datos. Madrid. PARANINFO. 2001. 1 pp.

administrada. Aún cuando las empresas están acostumbradas a trabajar con activos intangibles tales como el dinero, las instalaciones y el personal, cuyo valor puede evaluarse con precisión, es muy difícil medir el valor de la información. Sin embargo es claro que si los directivos tienen información correcta, es más probable que pueda tomar decisiones pertinentes y certeras con un mayor impacto positivo en su negocio; y de manera contraria si su información es irrisoria, ellos deben trabajar con más incertidumbre y es menos probable que tomen decisiones eficaces. En este contexto, después de un largo proceso, aparecen los sistemas de bases de datos, cuya utilización ha sido trascendental para proporcionar información correcta y oportuna al sistema de información empresarial.

Un sistema de base de datos, adecuadamente diseñado, integra los datos comunes a varias unidades funcionales de la empresa y facilita su manipulación. Además de simplificar las operaciones cotidianas de los registros, recopilar la información en resúmenes estadísticos, realizar inferencias sobre las posibles tendencias del negocio y otras operaciones. En otras palabras el sistema de base de datos transforma datos puros en información (conocimiento).

1.5 Sistemas de Gestión de Bases de Datos

1.5.1 Origen de los Sistemas de bases de datos²

La aparición de las bases de datos, pretendió eliminar los problemas existentes con la gestión de datos mediante archivos, así como estandarizar un modo de gestión de datos más eficaz, universal e independiente de las máquinas y sistemas operativos sobre los que se implementare, generando un sistema de almacenamiento de datos, cuya interfaz de usuario sea fácil de modificar y sencilla de consultar.

Durante los cincuenta e inicios de los sesenta, el proceso de datos era mediante el tratamiento de archivos secuenciales. Las limitaciones de los sistemas orientados a archivos puramente secuenciales no los privaron de ser herramientas eficaces. Sin embargo, para ejecutar muchas tareas rutinarias en los negocios se necesitó el acceso directo a los datos es

² **HANSEN, Gary y HANSEN, James.** Diseño y Administración de Bases de Datos. 2da Edición. Madrid. Prentice-Hall. 2002. 569pp

decir la capacidad de tener acceso y procesar directamente un registro dado sin ordenar primero el archivo o leer los registros en secuencia. Durante los sesenta, debido a que el almacenamiento en disco utilizando el acceso directo llegó a estar ampliamente disponible, el procesamiento de archivos de acceso aleatorio llegó a ser factible y popular. Este método permitió el acceso directo a datos específicos en un archivo.

En la medida en que los sistemas computacionales de procesamiento de datos se hicieron más importantes, los negocios comenzaron a reconocer que la información era un recurso corporativo de valor considerable y que los datos necesarios para contestar numerosas preguntas del negocio estaban disponibles en sus archivos de procesamiento de datos. Como consecuencia comenzaron a presionar a los sistemas de información para la gestión en cuanto a la utilización de la potencia del computador para producir información a partir de los datos corporativos. Esto inicio la demanda de las bases de datos, los que garantizarían más efectivamente, el acceso a los datos y su manipulación. A mediados de los sesenta se introdujeron los primeros sistemas de bases de datos cuyo fundamento era una estructura jerárquica de los datos. Estos sistemas permitieron la recuperación de múltiples registros asociados con un registro único de otro archivo. Inmediatamente después se desarrollaron los sistemas de bases de datos en redes que soportaron interrelaciones entre registros de archivos diferentes mucho más complejas. Ambos modelos de bases de datos, el jerárquico y en red requirieron el uso de punteros físicos predefinidos para enlazar los registros relacionados.

En 1970, el artículo de E.F. Codd sobre el modelo de datos relacional revolucionó el pensamiento en la industria de las bases de datos. El enfoque de Codd proponía el acceso y la manipulación de los datos únicamente desde el punto de vista de sus características lógicas sin importar la estructura física sobre la que estuviesen mantenidas. Durante los años setenta y ochenta se desarrollaron numerosos sistemas de bases de datos relacionales, y aún en la actualidad estos dominan el mercado comercial. Un desarrollo adicional de gran importancia es la aparición de la plataforma cliente/servidor como la base para los cálculos y el acceso a las bases de datos en una organización. La potencia de la plataforma cliente/servidor descansa en el concepto de división de funciones. El cliente es el computador frontal que tiene interfaz directamente con el usuario, manipula la Interfaz Gráfica de

Usuario GUI y realiza los cálculos y otros programas de interés para el usuario final. El servidor es el que gestiona fundamentalmente el acceso a los datos atendiendo las diferentes peticiones del cliente, también realiza funciones de control y seguridad de acceso a los datos. Los sistemas de bases de datos, aún están evolucionando para permitir a los usuarios plantear y resolver problemas aún más complejos. A continuación se observa la cronología del desarrollo histórico de los métodos de acceso a los datos, que han llevado a la evolución de la tecnología de bases de datos:

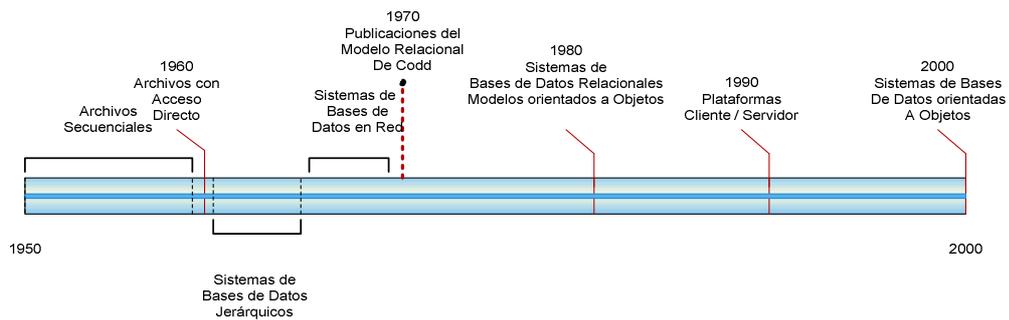


Figura 1.1. Evolución de la Tecnología de Bases de Datos
Fuente: Diseño y Administración de Bases de Datos

La tabla a continuación muestra una comparación entre algunas características de los diversos métodos de acceso a los datos.

Tabla 1.I. Características de los métodos de acceso a datos

Método de acceso a los datos	Características
Archivos Secuenciales	Todos los registros en un archivo deben de procesarse en secuencia
Archivos con Acceso Directo	Soporta el acceso directo a un registro específico. Es difícil el acceso a varios registros relacionados con un registro simple.
Bases de Datos Jerárquicas	Soportan el acceso a varios registros relacionados con un registro simple. Se restringen a las interrelaciones jerárquicas entre los datos. Dependientes de punteros físicos predefinidos.
Bases de Datos en Red	Soportan las interrelaciones jerárquicas y no jerárquicas en redes entre los datos. Dependientes de punteros físicos predefinidos
Bases de Datos Relacionales	Soportan todas las interrelaciones lógicas entre los datos.

Fuente: Diseño y Administración de Bases de Datos

1.5.2 Sistemas de Gestión de Bases de Datos (SGBD)

*“Un Sistema de Gestión de Bases de Datos –SGBD– (Data Base Management System DBMS) consiste en una colección de datos interrelacionados y un conjunto de programas para acceder a esos datos. El Objetivo primordial de un SGBD es proporcionar un entorno que sea a la vez conveniente y eficiente para ser utilizado al extraer y almacenar información de la base de datos”.*³

1.5.3 Características de los SGBD

Los sistemas de gestión de bases de datos se distinguen por características como⁴:

- Naturaleza auto descriptiva, pues poseen una definición completa de la base de datos, almacenada en el catálogo del sistema, que contiene información como la estructura de cada archivo, formato de almacenamiento de cada elemento de información y las restricciones que se aplican a los datos, evitando la incoherencia y duplicación de definiciones de datos en los programas. Los sistemas de bases de datos superan las limitaciones de los sistemas orientados a archivos, al tolerar una estructura de datos centralizada, e integrada eliminando los problemas de redundancia, de control e integridad de los datos, a diferencia de los sistemas de procesamiento de archivos tradicional, en donde la definición de los datos suele ser parte de los programas de aplicación mismos.
- Abstracción de Datos: Esta característica permite que el sistema oculte los detalles de cómo se almacenan y mantienen los datos, simplificando la interacción de los usuarios con el sistema.
- Manejo de múltiples vistas de los datos para los usuarios de la base de datos que pueden requerir una perspectiva o vista diferente de la mencionada base de datos.
- Compartimiento de datos y procesamiento de transacciones multiusuario, permitiendo que varios usuarios tengan acceso simultáneo a la base de datos, para lo cual el SGBD incluirá software de control de concurrencia para asegurar que

³ **SILBERSCHATZ, KORTH, SUDARSHAN.** Fundamentos de Bases de Datos. 4ta Edición. Madrid. Mc-Graw Hill. 2002. 787 pp.

⁴ **ELMARSI, Ramez y NAVATHE, Shankamt.** Sistemas de Bases de Datos. 2da Edición. Addison-Wesley Iberoamericana. 7-9p

cuando varios usuarios intenten actualizar los mismos datos, lo hagan de manera controlada para que el resultado de las actualizaciones sea correcto, y así garantizar la integridad y consistencia de los datos

1.5.4 Estructura de los SGBD

Los SGBD son sistemas de software muy complejos formados por módulos que se encargan de cada una de las responsabilidades del sistema. Los componentes funcionales de los sistemas de gestión de bases de datos se pueden dividir a grandes rasgos en gestor de almacenamiento y procesador de consultas⁵.

Gestor de Almacenamiento

Es el módulo responsable del almacenamiento, recuperación, y actualización de los datos en la base de datos, proporciona la interfaz entre los datos de bajo nivel en la base de datos y los programas de aplicación y consultas emitidas al sistema.

El gestor de memoria es importante porque las bases de datos requieren normalmente una gran cantidad de espacio de almacenamiento. Las bases de datos corporativas tienen un tamaño de entre cientos de gigabytes y, para las mayores bases de datos, terabytes de datos. Debido a que la memoria principal de los computadores no puede almacenar esta gran cantidad de información, esta se almacena en discos. Los datos se trasladan entre el disco de almacenamiento y la memoria principal cuando es necesario. Como la transferencia de datos a y desde el disco es lenta comparada con la velocidad de la unidad central de procesamiento, es fundamental que el sistema de base de datos structure los datos para minimizar la necesidad de movimiento de datos entre el disco y la memoria principal. Además es responsable de la interacción con el gestor de archivos.

Los componentes del gestor de almacenamiento son:

- **Gestor de autorización e integridad:** comprueba que se cumplen las restricciones de integridad y la autorización para que los usuarios puedan acceder a los datos.

⁵ **SILBERSCHATZ, KORTH, SUDARSHAN.** Fundamentos de Bases de Datos. 4ta Edición. Madrid. McGraw Hill. 2002. 787 pp.

- **Gestor de Transacciones:** asegura que la base de datos quede en un estado consistente (correcto), a pesar de los fallos del sistema y que las ejecuciones de transacciones concurrentes ocurran sin conflictos.
- **Gestor de Archivos:** Gestiona la reserva de espacio de almacenamiento de disco y las estructuras de datos usadas para representar la información almacenada en disco.
- **Gestor de memoria intermedia:** que es responsable de traer los datos del disco de almacenamiento a memoria principal y decidir qué datos tratar en memoria caché. El gestor de memoria intermedia es una parte crítica del sistema de bases de datos, ya que permite que la base de datos maneje tamaños de datos que son mucho mayores que el tamaño de la memoria principal.

El gestor de almacenamiento implementa varias estructuras de datos como parte de la implementación física del sistema:

- Archivos de datos, que almacenan la base de datos en sí.
- Diccionario de datos, que almacena metadatos acerca de la estructura de la base de datos, en particular, el esquema de la base de datos.
- Índices, que proporcionan acceso rápido a elementos de datos que tienen valores particulares.

Procesador de Consultas

El procesador de consultas simplifica y facilita el acceso a los datos mediante vistas de alto nivel que ayudan a que los usuarios del sistema no sean molestados innecesariamente con los detalles físicos de la implementación del sistema. Su objetivo es traducir las actualizaciones y las consultas escritas en un lenguaje no procedimental, en el nivel lógico, en una secuencia de operaciones en el nivel físico. Los componentes del procesador de consultas incluyen:

- Intérprete del LDD, que interpreta las instrucciones del LDD y registra las definiciones en el diccionario de datos.

- **Compilador del LMD**, que traduce las instrucciones del LMD en un lenguaje de consultas a un plan de evaluación que consiste en instrucciones de bajo nivel que entiende el motor de evaluación de consultas. Una consulta se puede traducir habitualmente en varios planes de ejecución alternativos que proporcionan el mismo resultado. El compilador del LMD también realiza optimización de consultas, es decir, elige el plan de evaluación de menor coste de entre todas las alternativas.
- **Motor de evaluación de consultas**, que ejecuta las instrucciones de bajo nivel generadas por el compilador del LMD.

En la figura se muestran los componentes del sistema de gestión de bases de datos y sus conexiones.

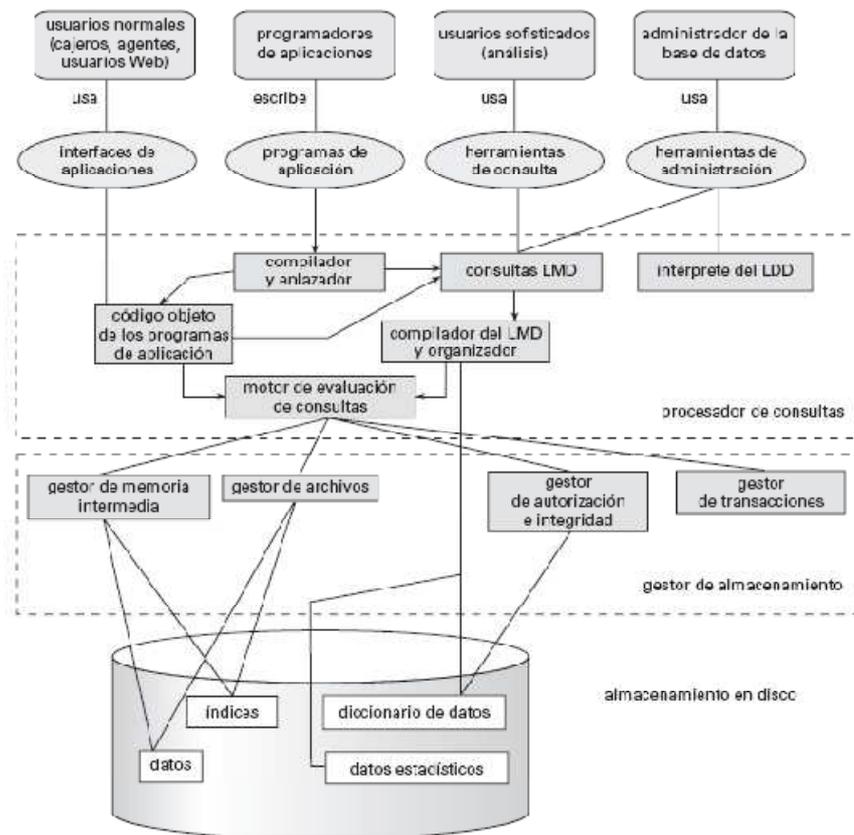


Figura 1.2. Estructura del sistema de base de datos
Fuente: Fundamentos de Bases de Datos

1.5.5 Clasificación de los Sistemas de Gestión de Bases de Datos

Existen varios criterios para clasificar a un Sistema de Gestión de Bases de Datos⁶.

El principal criterio que suele utilizarse para clasificar los SGBD es el modelo de datos en que se basa. Los modelos de datos empleados con mayor frecuencia en los SGBD comerciales actuales son el relacional, el de red y el jerárquico, algunos SGBD recientes se basan en modelos orientados a objetos o conceptuales. Entonces clasificaremos los SGBD como relacionales, de red, jerárquicos, orientados a objetos y otros.

Un segundo criterio para clasificar los SGBD es el número de usuarios a los que da servicio el sistema. Los sistemas mono usuario solo atiende a un usuario a la vez, y su principal uso se da en un computador personal. Los sistemas multiusuario, entre los que se cuentan la mayor parte de los SGBD, atienden varios usuarios al mismo tiempo.

Un tercer criterio es el número de sitios en los que está distribuida la base de datos. Casi todos los SGBD son *centralizados*: sus datos se almacenan en un solo computador. Los SGBD centralizados pueden atender a varios usuarios, pero el SGBD y la base de datos en sí residen por completo en una sola máquina.

En los SGBD *distribuidos* la base de datos real y el propio software del SGBD pueden estar distribuidos en varios sitios conectados por una red. Los SGBD *distribuidos homogéneos* utilizan el mismo SGBD en múltiples sitios. Una tendencia reciente consiste en crear software para tener acceso a varias bases de datos autónomas preexistentes almacenadas en SGBD *distribuidos heterogéneos*. Esto da lugar a los SGBD *federados* o *sistemas multibase de datos* en los que los SGBD participantes tienen cierto grado de autonomía local. Muchos SGBD distribuidos emplean una arquitectura cliente-servidor.

Un cuarto criterio es el coste del SGBD, y por último, los SGBD pueden ser de *propósito general* o de *propósito específico*. Cuando el rendimiento es fundamental, se puede diseñar y construir un SGBD de propósito especial para una aplicación específica, y este sistema no

⁶ **MARQUEZ, María.** Clasificación de los Sistemas de Gestión de Bases de Datos. [PDF] Apuntes de Ficheros y Bases de Datos. 2001. [Fecha de Consulta: Agosto del 2008] Disponible en: <http://www3.uji.es/~mmarques/f47/apun/apun.pdf>

sirve para otras aplicaciones. Muchos sistemas de reservas de líneas aéreas son SGBD de propósito especial y pertenecen a la categoría de *sistemas de procesamiento de transacciones en línea* (OLTP), que deben atender un gran número de transacciones concurrentes sin imponer excesivos retrasos.

1.5.6 Funciones de los sistemas de gestión de bases de datos

Según, E.F. CODD, el creador del modelo relacional, existen ciertos servicios primordiales que debe ofrecer todo SGBD⁷.

Catálogo del Sistema

El catálogo del sistema almacena las descripciones de los datos y es accesible por los usuarios, se lo denomina diccionario de datos y contiene información que describe los datos de la base de datos por ejemplo: tipo y tamaño de los datos, las relaciones entre los datos, restricciones de integridad sobre los datos, nombre de los usuarios autorizados a acceder a la base de datos, esquemas externos, conceptuales e internos, y correspondencia entre los esquemas, estadísticas tales como la frecuencia de las transacciones y el número de accesos realizados a los objetos de la base de datos.

Almacenamiento e Indexación De Datos

Un SGBD debe proporcionar a los usuarios la capacidad de almacenar datos en la base de datos, acceder a ellos y actualizarlos. Esta es la función fundamental de un SGBD y por supuesto, el SGBD debe ocultar al usuario la estructura física interna (la organización de los ficheros y las estructuras de almacenamiento).

Procesamiento y Optimización De Consultas

Este servicio hace referencia a la serie de actividades implicadas en la extracción de datos de una base de datos. Estas actividades incluyen la traducción de consultas expresadas en lenguajes de bases de datos de alto nivel en expresiones implementadas en el nivel físico del

⁷ **MARQUEZ, María.** Funciones de los Sistemas de Gestión de Bases de Datos. [PDF]. Apuntes de Ficheros y Bases de Datos. 2001. [Fecha de Consulta: Agosto del 2008] Disponible en: <http://www3.uji.es/~mmarques/f47/apun/apun.pdf>

sistema, así como transformaciones de optimización de consultas y la evaluación real de las mismas.

Procesamiento de Transacciones

Un SGBD debe proporcionar un mecanismo que garantice que todas las actualizaciones correspondientes a una determinada transacción se realicen, o que no se realice ninguna. Una *transacción* es un conjunto de acciones que cambian el contenido de la base de datos. Una transacción en el sistema informático de la empresa inmobiliaria sería dar de alta a un empleado o eliminar un inmueble. Una transacción un poco más complicada sería eliminar un empleado y reasignar sus inmuebles a otro empleado. En este caso hay que realizar varios cambios sobre la base de datos. Si la transacción falla durante su realización, por ejemplo porque falla el hardware, la base de datos quedará en un estado inconsistente. Algunos de los cambios se habrán hecho y otros no, por lo tanto, los cambios realizados deberán ser deshechos para devolver la base de datos a un estado consistente.

Control de Concurrencia

Un SGBD debe proporcionar un mecanismo que asegure que la base de datos se actualice correctamente cuando varios usuarios la están actualizando concurrentemente. Uno de los principales objetivos de los SGBD es el permitir que varios usuarios tengan acceso concurrente a los datos que comparten. El acceso concurrente es relativamente fácil de gestionar si todos los usuarios se dedican a leer datos, ya que no pueden interferir unos con otros. Sin embargo, cuando dos o más usuarios están accediendo a la base de datos y al menos uno de ellos está actualizando datos, pueden interferir de modo que se produzcan inconsistencias en la base de datos. El SGBD se debe encargar de que estas interferencias no se produzcan en el acceso simultáneo.

Sistema de Recuperación

Un SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos en caso de que ocurra algún suceso que la dañe. Como se ha comentado antes, cuando el sistema falla en medio de una transacción, la base de datos se debe devolver a un estado consistente. Este fallo puede ser a causa de un fallo en algún dispositivo hardware o un error del

software, que hagan que el SGBD aborte, o puede ser a causa de que el usuario detecte un error durante la transacción y la aborte antes de que finalice. En todos estos casos, el SGBD debe proporcionar un mecanismo capaz de recuperar la base de datos llevándola a un estado consistente.

Herramientas de Administración

Un SGBD debe proporcionar una serie de herramientas que permitan administrar la base de datos de modo efectivo. Algunas herramientas trabajan a nivel externo, por lo que habrán sido producidas por el administrador de la base de datos. Las herramientas que trabajan a nivel interno deben ser proporcionadas por el distribuidor del SGBD. Algunas de ellas son:

- Herramientas para importar y exportar datos.
- Herramientas para monitorizar el uso y el funcionamiento de la base de datos.
- Programas de análisis estadístico para examinar las prestaciones o las estadísticas de utilización.
- Herramientas para reorganización de índices.
- Herramientas para aprovechar el espacio dejado en el almacenamiento físico por los registros borrados y que consoliden el espacio liberado para re-utilizarlo cuando sea necesario.

CAPÍTULO II

2. PRINCIPALES FUNCIONES Y CAPACIDADES DE LOS SGBD

Los Sistemas de Gestión de Bases de Datos (SGBD), se han convertido en parte fundamental de la estrategia de las empresas. El valor de la información actualizada ha crecido tanto que las empresas que quieran incrementar o mantener su productividad deberán gestionar eficientemente todos los datos que manejan, y la mejor herramienta es un SGBD.

Dado que disponemos de varias opciones, resulta imprescindible contar con elementos de juicio a la hora de optar por una u otra solución, ¿Cuál se adecua mejor a nuestras necesidades? Para ello resulta indispensable observar el funcionamiento de los Sistemas de Gestión de Bases de Datos, para posteriormente contraponer estos conceptos con el modo de operación de los SGBD en cuestión, y así poder determinar las características que los hacen diferentes.

Además para evaluar la habilidad de un SGBD de satisfacer los requisitos de información, de una empresa, necesitamos considerar las funciones que proporciona y sus propiedades fundamentales.

2.1 Catálogo del Sistema

El catálogo del sistema constituye el núcleo de todo SGBD de aplicación general. El catálogo es una minibase de datos por sí misma, cuya función es almacenar los esquemas, o descripciones, de las bases de datos que el SGBD mantiene.

Un catálogo está acoplado íntimamente al software del SGBD; proporciona la información que contiene a los usuarios y al DBA, pero lo utilizan principalmente los diversos módulos de software del SGBD mismo, como son los compiladores de DDL y DML, el optimizador de consultas, el procesador de transacciones, los generadores de informes y el módulo encargado de hacer que se cumplan las restricciones.

La información almacenada en el catálogo de un SGBD relacional incluye descripciones de los nombres de las relaciones, nombres de los atributos, dominios (tipos de datos) de los atributos, claves primarias, atributos de clave secundaria, claves externas y otros tipos de restricciones, así como descripciones de nivel externo de las vistas y descripciones de nivel interno de las estructuras de almacenamiento e índices. También contiene información de seguridad y autorización, que especifica los permisos que tienen los usuarios para tener acceso a las relaciones y vistas de la base de datos, y quiénes son los creadores o propietario de cada relación⁸.

Uno de los beneficios más importantes del catálogo de sistema es que hace posible el desarrollo de herramientas de consulta de fácil actualización cuyo objetivo es permitir a los usuarios un acceso simple y transparente a la base de datos sin tener que aprender el lenguaje SQL⁹.

2.1.1 Estructura del Catálogo de Datos

El estándar SQL ANSI/ISO actual no especifica la estructura y los contenidos del catálogo de sistema. De hecho, el estándar no requiere la existencia de un catálogo de sistema en absoluto. Sin embargo, todos los productos SGBD basados en SQL más importantes

⁸ **ELMARSÍ, Ramez y NAVATHE, Shankamt.** Sistemas de Bases de Datos. 2da Edición. Addison-Wesley Iberoamericana. 484 -485pp

⁹ **DESCONOCIDO.** Catálogo del Sistema. [PDF]. Instituto Tecnológico Autónomo de México. 2002. [Fecha de Consulta: Agosto del 2008]. Disponible en: <http://cursos.itam.mx/akuri/2002/S12002/basesdedatos/CHAPTER6.PDF>

proporcionan un catálogo de sistema de una forma u otra. La estructura del catálogo y las tablas que contiene varían considerablemente de un producto SGBD a otro.

Debido a la creciente importancia de las herramientas de base de datos de propósito general que deben acceder al catálogo de sistema, el estándar SQL2 propuesto incluye una especificación de un catálogo de sistema mínimo. La especificación está basada generalmente en las tablas de sistema DB2, pero elimina los contenidos específicos de DB2. Si este intento de normalizar el catálogo del sistema va a tener éxito o no sigue siendo una cuestión abierta.¹⁰

2.2 Almacenamiento e Indexación de Datos

El rendimiento general de un sistema de base de datos se determina en gran medida por las estructuras de datos físicas usadas y la eficiencia con la cual el sistema trabaja sobre las mismas. El rendimiento, es un factor de gran importancia en la satisfacción del usuario con el sistema de base de datos y está intrínsecamente ligado con el tiempo de respuesta aceptable.

La información y el formato no se afectan mucho por la organización física de la base de datos, pero el tiempo de respuesta sí. El tiempo de respuesta, es el tiempo transcurrido entre la inicialización de una operación, sobre la base de datos y la disponibilidad del resultado. Un tiempo de respuesta lento, es la queja más frecuente que expresan los usuarios de los sistemas de bases de datos. Un buen diseño físico de la base de datos, almacenaría datos, de forma que su recuperación, actualización y manipulación, sea en el menor tiempo posible.

La empresa que va a adquirir un SGBD puede no desear involucrarse con asuntos de la organización física de los datos. En su lugar puede medir la eficacia de tal organización, ejecutando aplicaciones¹¹.

¹⁰ **DESCONOCIDO**. Catálogo del Sistema. [PDF]. Instituto Tecnológico Autónomo de México. 2002. [Fecha de Consulta: Agosto del 2008]. Disponible en: <http://cursos.itam.mx/akuri/2002/s12002/basesdedatos/chapter6.pdf>

¹¹ **HANSEN, Gary y HANSEN, James**. Diseño y Administración de Bases de Datos. 2da Edición. Madrid. Prentice-Hall. 2002. 569pp

2.2.1 Organización de archivos y métodos de direccionamiento

A continuación se mencionan las formas básicas de organización física de archivos sobre los dispositivos de almacenamiento¹²:

- **Organización de archivos en montículo.** Normalmente sólo hay un archivo por cada relación. En esta organización se puede colocar cualquier registro en cualquier parte del archivo en que haya espacio suficiente. No hay ninguna ordenación de los registros.
- **Organización de archivos secuenciales.** En esta organización los registros se guardan en orden secuencial, basado en el valor de la clave de búsqueda de cada registro. Minimiza el número de accesos a los bloques en el procesamiento de los archivos secuenciales, guardando físicamente los registros de acuerdo con el orden indicado por la clave de búsqueda, o en un orden tan cercano a éste como sea posible. La correspondencia entre el orden de la clave de búsqueda y el orden físico puede perderse totalmente, en cuyo caso el procesamiento secuencial será significativamente menos eficiente. Llegado a este punto, se debe reorganizar el archivo de modo que vuelva a estar físicamente en orden secuencial. Estas reorganizaciones resultan costosas y deben realizarse en momentos en los que la carga del sistema sea baja. La frecuencia con la que se necesitan las reorganizaciones depende de la frecuencia de inserción de registros nuevos.
- **Organización de archivos secuenciales indexados:** La organización secuencial indexada de un archivo brinda facilidades para acceder a los registros de ambas formas, secuencial y directamente. Los registros se almacenan en la secuencia física usual por la clave primaria, además se almacena en disco el índice de la localización del registro. Esto permite el acceso secuencial a los registros para aquellas aplicaciones que realicen una gran cantidad de actualizaciones, y por otra parte el acceso directo de acuerdo con las solicitudes del usuario.

El inconveniente principal de la organización de un archivo secuencial indexado reside en que el rendimiento, tanto para buscar en el índice como para buscar

¹² **SILBERSCHATZ, KORTH, SUDARSHAN.** Fundamentos de Bases de Datos. 4ta Edición. Madrid. McGraw Hill. 2002. 268 p.

secuencialmente a través de los datos, se degrada según crece el archivo. Aunque esta degradación se puede remediar reorganizando el archivo, el rendimiento de tales reorganizaciones no es deseable.

- **Organización de archivos en agrupaciones:** Una organización de archivos en agrupaciones almacena registros relacionados de dos o más relaciones en cada bloque. Esta organización permite leer muchos de los registros que satisfacen la condición de reunión utilizando un solo proceso de lectura de bloques. Este uso de la agrupación ha mejorado el procesamiento de una reunión particular sin embargo retarda el procesamiento de otros tipos de consulta, por ejemplo, *select * from cliente* en donde se necesita más accesos a los bloques que en el esquema en el que se guardaba cada relación en un archivo diferente. Es más, hallar todos los registros de cliente no resultaría posible sin alguna estructura adicional. Para encontrar todas las tuplas de la relación cliente en la estructura de archivos en agrupaciones hay que vincular todos los registros de esa relación utilizando punteros, tal y como se muestra a continuación.

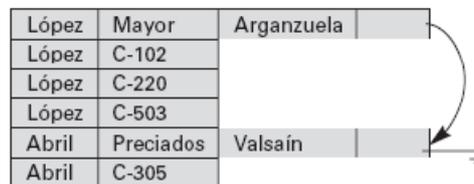


Figura 2.1. Estructura de archivos en agrupaciones con cadena de punteros.

Fuente: Fundamentos de Bases de Datos

La determinación del momento de utilizar la agrupación depende de los tipos de consulta que el diseñador de la base de datos considere más frecuentes. El uso cuidadoso de la agrupación puede producir ganancias de rendimiento significativas en el procesamiento de consultas.

- **Organización asociativa (hash) de archivos:** El uso de técnicas hashing como método de direccionamiento elimina la necesidad del mantenimiento y búsqueda de índices. La eliminación del índice evita la necesidad de hacer dos accesos al almacenamiento secundario para acceder a un registro: uno para leer el índice y otro para acceder al archivo. La idea básica consiste en la negociación del tiempo y el

esfuerzo que se emplea en el almacenamiento, mantenimiento, y búsqueda de un índice por el tiempo que necesita la Unidad Central de Procesamiento (CPU) para ejecutar un algoritmo hash, que genera la dirección del registro.

2.2.2 Indexación

Los índices de los sistemas de bases de datos sirven para agilizar la obtención de registros en respuesta a ciertas condiciones de búsqueda. Una estructura de índice permite el acceso rápido a los registros de un archivo, y está asociada a una clave de búsqueda concreta.

Para comprender el beneficio de un índice es necesario observar las ventajas y desventajas que proporcionan los diversos tipos de almacenamiento

Memoria Principal (RAM)

- Tiempo de acceso bajo
- Capacidad de almacenamiento bajo
- Es volátil

Memoria Secundaria (HDD, DVD,..)

- Tiempo de acceso **elevado** (*movimiento del cabezal*)
- Capacidad de almacenamiento **alto**
- **No es volátil**

La idea básica, es almacenar los datos en memoria secundaria, para acceder directamente a ellos a través de un índice que estará soportado por una estructura de datos, y no de forma secuencial ya que la mayor parte del tiempo se pierde en el movimiento del cabezal.

La creación de un índice hace que sea más eficiente localizar una fila concreta dentro de una tabla. De todas maneras esto añade algo de carga de trabajo adicional, porque la base de datos, debe mantener las filas de datos, y las entradas de índices correspondientes a la tabla, por aquella razón no debe hacerse uso indiscriminado de índices en una base de datos, porque la solución se convertiría en algo peor que el problema.

Existen varios tipos de índices, a continuación mencionaremos los más utilizados.

Vectores

Ventajas: Acceso a memoria, búsquedas binarias o dicotómicas.

Inconvenientes: Limitación de memoria, tamaño del vector (no son dinámicos), los procesos de inserción o borrado son costosos.

Listas Enlazadas

Ventajas: Acceso a memoria, no está limitada de tamaño, actualizar una lista es menos costoso que un vector

Inconvenientes: Limitación de la memoria, la búsqueda de los índices es secuencial.

ABB - BST (Arboles Binarios de Búsqueda - Binary Search Tree)

Ventajas: Acceso a memoria, el tamaño no está limitado, búsquedas binarias

Inconvenientes: Limitación de la memoria, **pueden degenerar en una lista** (sumando así los inconvenientes anteriores)

APE (Arboles Perfectamente Equilibrados)

Ventajas: Acceso a memoria, no tiene limitaciones de tamaño, búsquedas binarias, no degeneran en una lista.

Desventajas: Operaciones de reequilibrio muy costosas, mantenimiento de la información de equilibrio constantemente

Arboles AVL

Ventajas: acceso a memoria, el tamaño del árbol no está limitado, búsquedas dicotómicas, actualizar este árbol es menos costoso, no degeneran en una lista, rotaciones menos costosas que el reequilibrio de APE

Inconvenientes: Limitación de la memoria, se dan casos de desequilibrio haciendo necesarias las rotaciones, que son procesos costosos.

Una vez que se mencionan las ventajas e inconvenientes de todas estas estructuras de datos, se estudia los arboles B, indicando sus ventajas más importantes respecto a las estructuras anteriores.

Arboles B

Multicamino de búsqueda siempre esta ordenado

Todas las hojas están a la misma altura

Crece hacia arriba, no necesita rotaciones

2.2.3 Particionamiento de Tablas e Índices

El Particionamiento de tablas en índices se refiere a la capacidad de un SGBD para dividir una tabla o índice, en trozos mucho más pequeños. Esta capacidad ofrece ventajas en muchas áreas.

- La copia de seguridad y recuperación es más sencilla y rápida, puesto que se puede realizar sobre particiones individuales en lugar de sobre toda la tabla.
- Las operaciones de carga en un entorno de almacén de datos son menos intrusivas: se pueden agregar datos a una partición y entonces agregar la partición a una tabla, lo que es una operación instantánea. De igual forma, eliminar una partición con datos obsoletos desde una tabla es muy sencillo en un almacén de datos que mantenga una ventana de datos históricos.
- El rendimiento de la consulta se mejora sustancialmente, puesto que el optimizador puede reconocer que solamente se tiene que acceder a un sub-conjunto de las particiones de una tabla con el fin de resolver la consulta (poda de particiones). También el optimizador puede reconocer que en una reunión no es necesario intentar hacer corresponder todas las filas en una tabla con todas las filas en la otra, pero que las reuniones se necesitan realizar solamente entre pares coincidentes de divisiones (reunión por particiones).

2.3 Procesamiento y Optimización de Consultas

El procesamiento de consultas se refiere a la serie de actividades implicadas en la extracción de datos de una base de datos, así como transformaciones de optimización de consultas y la evaluación real de las mismas.¹³

Al optimizar el servidor de la base de datos, es también necesario mejorar el rendimiento de las consultas individuales, esto es incluso más importante que optimizar otros aspectos de la instalación del servidor, aspectos que afectan al rendimiento, como las configuraciones de hardware y software. Incluso si el servidor de la base de datos se ejecuta en el hardware más eficaz, su rendimiento puede verse afectado negativamente por un conjunto de consultas que se comporten incorrectamente, de hecho, una sola consulta incorrecta, a veces denominada "consulta descontrolada", puede provocar serios problemas de rendimiento a la base de datos.

De manera opuesta, el rendimiento general de la base de datos puede mejorarse en gran medida al optimizar el conjunto de las consultas más costosas o que se ejecutan con mayor frecuencia.

2.3.1 Medidas del coste de la evaluación una consulta

El coste de la evaluación de una consulta se puede expresar en términos de diferentes recursos, incluyendo accesos a disco, tiempo de UCP en ejecutar una consulta y, en sistemas de bases de datos distribuidos o paralelos, el coste de la comunicación.

El tiempo de respuesta para un plan de evaluación de una consulta (esto es, el tiempo de reloj que se necesita para ejecutar el plan), si se supone que no hay otra actividad ejecutándose en el sistema, podría tener en cuenta todos estos costes y utilizarlo como una buena medida del coste del plan.

Sin embargo, en grandes sistemas de bases de datos, los accesos a disco (que se miden como el número de transferencias de bloques de disco) son normalmente el coste más importante, ya que los accesos a disco son más lentos comparados con las operaciones en

¹³ **SILBERSCHATZ, KORTH, SUDARSHAN.** Fundamentos de Bases de Datos. 4ta Edición. Madrid. McGraw Hill. 2002. 319p.

memoria. Además, la velocidad de la UCP está aumentando mucho más rápidamente que las velocidades de los discos. Así, lo más probable es que el tiempo empleado en operaciones del disco siga influenciando el tiempo total de ejecución de una consulta. Por este motivo se considera el coste de los accesos a disco, una medida razonable del coste del plan de evaluación de una consulta. Se utilizará simplemente el término número de transferencias de bloques de disco como una medida del coste real. Para simplificar los cálculos del coste de los accesos a disco se asume que todas las transferencias de bloques tienen el mismo coste. Esta suposición ignora la varianza que surge de la latencia rotacional (tiempo de espera a que el dato deseado gire bajo la cabeza de lectura-escritura) y el tiempo de búsqueda (el tiempo que se emplea en mover la cabeza sobre el cilindro o pista deseada)¹⁴.

2.4 Procesamiento de Transacciones y Control de Concurrencia

2.4.1 Transacción

Una transacción es una colección de operaciones que forman una única unidad lógica de trabajo. Un sistema de base de datos debe asegurar que la ejecución de las transacciones se realice adecuadamente a pesar de la existencia de fallos: o se ejecuta la transacción completa o no se ejecuta en absoluto. Además debe gestionar la ejecución concurrente de las transacciones evitando introducir inconsistencias.

2.4.2 Propiedades ACID.

Para asegurar la integridad de los datos se necesita que el sistema de base de datos mantenga las siguientes propiedades de las transacciones:

A: Atomicidad

O se ejecutan todas las operaciones o ninguna.

C: Consistencia

La ejecución aislada de las transacciones conserva la consistencia.

¹⁴ **SILBERSCHATZ, KORTH, SUDARSHAN.** Fundamentos de Bases de Datos. 4ta Edición. Madrid. McGraw Hill. 2002. 320p.

I: (Isolation) Aislamiento

Las Transacciones concurrentes no se afectan entre sí.

D: Durabilidad

Si la transacción finaliza con éxito, los cambios permanecen incluso si hay fallos.

2.4.3 Estados de una Transacción

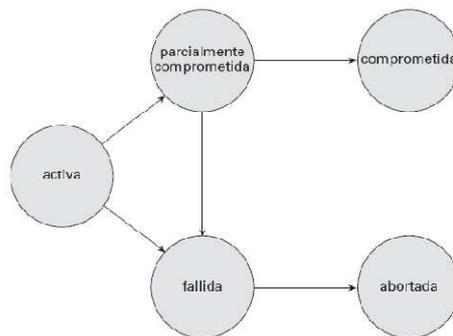


Figura 2.2. Estados de una Transacción

Fuente: Fundamentos de Bases de Datos

ACTIVA:	Estado Inicial
PARCIALMENTE COMPROMETIDA:	Tras Ejecutarse la Última instrucción.
FALLIDA:	Tras descubrir que no puede continuar.
COMPROMETIDA:	Tras Completarse con éxito.
ABORTADA:	Tras retroceder la T, y devolver la BD al estado previo al de la ejecución de T.

2.4.4 Ejecuciones Concurrentes

Los sistemas de procesamiento de transacciones permiten normalmente la ejecución de varias transacciones concurrentemente, esto provoca complicaciones en la consistencia de los mismos. Asegurar la consistencia a pesar de la ejecución concurrente de las

transacciones requiere un trabajo extra. Sin embargo, existen dos buenas razones para permitir la concurrencia.

- **Productividad y utilización de recursos mejoradas.** Esto se refiere al incremento de la cantidad del número de transacciones que se puede ejecutar en un tiempo dado. Análogamente, aumenta la utilización del procesador y del disco; en otras palabras, el procesador y el disco, están menos tiempo, desocupados o sin hacer ningún trabajo útil.
- **Tiempo de espera reducido.** Se reduce el tiempo medio de respuesta o sea el tiempo desde que una transacción comienza hasta que se completa.

El nivel de concurrencia, o número de transacciones activas, es probablemente uno de los parámetros más importante de un SGBD. Por lo tanto, los mecanismos de control de concurrencia deben encontrar el balance entre el mantenimiento de la consistencia de la base de datos y el mantenimiento de un alto nivel de concurrencia.

2.4.5 Importancia del control de concurrencia

La gestión de transacciones concurrentes en una base de datos brinda un eficiente desempeño del Sistema de Base de Datos, puesto que permite controlar la ejecución de transacciones que operan en paralelo, sin embargo esta puede causar el acceso a información compartida y, por lo tanto, interferir potencialmente unas con otras.¹⁵

El propósito del control de concurrencia, es asegurar que las transacciones se ejecuten atómicamente controlando la intercalación de transacciones concurrentes, dando la ilusión de que las transacciones se ejecutan secuencialmente, una después de la otra, sin ninguna intercalación.

El control de la concurrencia es muy importante para que el SGBD controle la integridad y la consistencia:

¹⁵ **VARGAS Magali, y CANO Yuri**, Concurrencia en las Bases de Datos. [ON-LINE]. Corporación universitaria Minuto de Dios. 2005. [Fecha de Consulta: Septiembre del 2008]. Disponible en: <http://www.monografias.com/trabajos24/concurrencia-base-datos/concurrencia-base-datos.shtml>

- Consistencia: si la mantenemos, estamos asegurando que los datos que vemos cuando hacemos una transacción, no cambian por otros usuarios hasta que no acabemos nuestra transacción.
- Integridad: gracias a esta propiedad aseguramos que los datos y estructuras reflejen los cambios en una secuencia correcta

Cuando se realizan varias transacciones a la vez, pueden producirse estados de inconsistencia en la base de datos. Por ejemplo, una transacción bancaria lee una cantidad de dinero de una cuenta y le resta 100 euros, y antes de actualizar ese dato en la base de datos, otra transacción lee ese dato. Si no se controlan esos pequeños detalles, los datos se actualizarían más tarde y la persona que lee de esa cuenta antes de que se confirme, verá un dato incorrecto. Por lo tanto, debemos controlar las transacciones para que no se produzcan estos errores. Además como ya se menciono anteriormente, maximizando la concurrencia, conseguimos maximizar la productividad y eficiencia de la base de datos.

2.4.6 Esquemas de control de concurrencia

Es necesario que el sistema controle la interacción entre las transacciones concurrentes, y dicho control se lleva a cabo mediante mecanismos llamados esquemas de control de concurrencia.

Se pueden usar varios esquemas de control de concurrencia para asegurar la secuencialidad, estos esquemas retrasan una operación o bien abortan la transacción que ha realizado la operación. Los más comunes son los protocolos de bloqueo, los esquemas de ordenación por marcas temporales, las técnicas de validación y los esquemas multiversión, niveles de aislamiento, etc.

Control por Bloqueos

Un bloqueo es una acción que impide el acceso a los datos bloqueados por parte de cualquier otra transacción, hasta que estos hayan sido desbloqueados. Así, un bloqueo asocia una variable a cada elemento de datos que describe respecto a las posibles operaciones que sobre él se pueden realizar.

Para ello tenemos el identificador del elemento bloqueado y el identificador de la transacción que lo bloquea; los bloqueos se obtienen solicitándolo al gestor de bloqueos, y garantiza derechos de exclusividad para la transacción.

Hay dos tipos de bloqueos:

- **Exclusivos:** un único bloqueo por recurso. Cierra el elemento a un acceso de cualquier tipo. Ninguna otra transacción puede leer o modificar los datos.
- **Compartidos:** muchos bloqueos por recurso. Cierra elementos a modificaciones, no a la lectura. Otras transacciones pueden leer el elemento, siempre y cuando no intenten modificarlo.

Los bloqueos no se pueden realizar de forma aleatoria ni arbitraria, ya que podrían producir inconsistencias. Para ello tenemos un protocolo de bloqueo, que indica cuando una transacción puede bloquear y desbloquear elementos logrando la consistencia en las transacciones concurrentes.

Esquemas Multiversión

Un esquema de control de concurrencia multiversión supone que cada operación escribir (Q) crea una nueva versión de Q. Cuando se solicita una operación leer (Q), el sistema selecciona una de las versiones de Q para ser leída. El esquema de control de concurrencia se encarga de que la selección de la versión que se va a leer se haga de una manera que garantice la serializabilidad. Esto se logra mediante el uso de horas de entrada. Una operación de lectura siempre se lleva a cabo con éxito, mientras que una operación de escritura puede ocasionar el retroceso de la transacción.

Niveles de Aislamiento

El estándar ANSI/ISO SQL 92 define una serie de niveles de aislamiento de una transacción con diferentes grados de impacto sobre el proceso de transacciones.

Este estándar pretende normalizar los sistemas gestores de bases de datos, con una serie de reglas o normas. SQL86 fue la primera de estas normas y se sustituyó por SQL89. Ésta, a su vez, fue sustituida por SQL92, hasta llegar a SQL2006, que es la norma actual.

Aunque el aislamiento entre las transacciones es generalmente deseable, el funcionamiento de muchas aplicaciones en este modo puede comprometer seriamente el rendimiento. Estos niveles de aislamiento se definen en relación a tres fenómenos que deben prevenirse cuando se produce concurrencia de transacciones.

Estos fenómenos son:

- Lectura errónea (DIRTY READS): estas se producen cuando una transacción lee datos que han sido escritos por otra transacción que aun no ha sido confirmada.
- Doble lectura (NON-REPETEABLE READS): una transacción lee datos que ya había leído, encontrando entre las dos lecturas diferencias (incoherentes) debido a que los datos han sido modificados o borrados por una transacción que ya ha sido confirmada.
- Lectura fantasma (PHANTOM READ): una transacción re-ejecuta una consulta encontrando que en el conjunto de filas resultantes ha sido ampliado por otra transacción que insertó nuevas filas y que ya ha realizado su COMMIT.

Como se menciona anteriormente, la norma SQL92 incluye cuatro niveles de aislamiento para las transacciones. Cada nivel de aislamiento especifica los tipos de acciones que no están permitidos realizar cuando se ejecutan transacciones concurrentes. Los niveles más altos incluyen las restricciones impuestas por los niveles más bajos:

- El nivel 0 (READ UNCOMMITTED) evita que otras transacciones cambien los datos que ya han sido modificados (mediante INSERT, DELETE, UPDATE, etc.) por una transacción no comprobada (es decir, que no se ha confirmado). Las otras transacciones se bloquean para que no modifiquen los datos hasta que la transacción se haya confirmado. No obstante, las otras transacciones todavía pueden leer los datos no confirmados, lo que da lugar a lecturas erróneas (DIRTY READS).
- El nivel 1 (READ COMMITTED) evita las lecturas erróneas. Estas lecturas tienen lugar cuando una transacción modifica una fila y luego una segunda transacción lee esa misma fila antes de que la primera transacción haya podido consignar el cambio. Si la primera transacción revierte el cambio, la información leída por la segunda transacción se convierte en inválida.

- El nivel 2 (REPEATABLE READ) evita las dobles lecturas (NON-REPEATABLE READS). Estas lecturas tienen lugar cuando una transacción lee una fila y luego una segunda transacción modifica dicha fila. Si la segunda transacción confirma el cambio, las lecturas siguientes realizadas por la primera transacción producen resultados diferentes a los de la primera lectura.
- El nivel 3 (Serializable) evita las lecturas fantasmas (PHANTOM READ). Estas lecturas tienen lugar cuando una transacción lee un conjunto de filas que cumplen una condición de búsqueda y luego una segunda transacción modifica los datos (mediante una instrucción INSERT, DELETE, UPDATE, etc.). Si la primera transacción repite la lectura con las mismas condiciones de búsqueda, el conjunto de filas resultante es distinto.

2.5 Sistema de Recuperación

Una computadora, al igual que cualquier otro dispositivo eléctrico o mecánico, está predispuesta a fallos. Éstos se producen por diferentes motivos como: fallos de disco, cortes de corriente, errores en el software, un incendio en la habitación de la computadora o incluso sabotaje. En cada uno de estos casos puede perderse información.

Por tanto, el sistema de bases de datos debe realizar con anticipación acciones que garanticen que las propiedades de atomicidad y durabilidad de las transacciones, se preservan a pesar de tales fallos.

Una parte integral de un sistema de bases de datos es un esquema de recuperación, el cual es responsable de la restauración de la base de datos al estado consistente previo al fallo. El esquema de recuperación también debe proporcionar alta disponibilidad; esto es, debe minimizar el tiempo durante el que la base de datos no se puede usar después de un fallo.

2.5.1 Transacciones Secuenciales y Recuperación

Para conseguir el objetivo de la atomicidad se debe efectuar primero la operación de salida de la información que describe las modificaciones en el almacenamiento estable sin modificar

todavía la base de datos. Asumiendo que las transacciones se ejecutan secuencialmente, hay dos formas de ejecutar tales salidas esto es¹⁶:

Recuperación Basada en el registro histórico

La estructura más ampliamente utilizada para guardar las modificaciones de una base de datos es el registro histórico. El registro histórico es una secuencia de registros de todas las actividades de actualización de la base de datos. Un registro de actualización del registro histórico describe una única escritura en la base de datos y tiene los siguientes campos:

- El identificador de la transacción es un identificador único de la transacción que realiza la operación escribir.
- El identificador del elemento de datos es un identificador único del elemento de datos que se escribe. Normalmente suele coincidir con la ubicación del elemento de datos en el disco.
- El valor anterior es el valor que tenía el elemento de datos antes de la escritura.
- El valor nuevo es el valor que tendrá el elemento de datos después de la escritura.

Existen otros registros del registro histórico especiales para registrar sucesos significativos durante el procesamiento de una transacción, tales como el comienzo de una transacción y el éxito o aborto de la misma.

Cuando una transacción realiza una escritura es fundamental que se cree el registro del registro histórico correspondiente a esa escritura antes de modificar la base de datos. Una vez que el registro del registro histórico existe, se puede realizar la salida de la modificación a la base de datos si se desea. Además, es posible deshacer una modificación que ya haya salido a la base de datos. Se deshazá utilizando el campo valor-anterior de los registros del registro histórico. Para que los registros del registro histórico sean útiles para recuperarse frente a errores del disco o del sistema, el registro histórico debe residir en almacenamiento estable.

¹⁶ **SILBERSCHATZ, KORTH, SUDARSHAN.** Fundamentos de Bases de Datos. 4ta Edición. Madrid. McGraw Hill. 2002. 413-441 pp.

Para garantizar la atomicidad frente a fallos se presentan dos técnicas de utilización del registro histórico:

- Técnica de modificación diferida: Almacena las modificaciones de la BD en el registro histórico, pero retarda la ejecución de todas las operaciones escribir de una transacción hasta que la transacción se compromete parcialmente.
- Técnica de modificación inmediata: Permite realizar la salida de las modificaciones de la base de datos a la propia base de datos mientras que la transacción está todavía en estado activo. Las modificaciones de datos escritas por transacciones activas se denominan modificaciones no comprometidas. En caso de una caída o de un fallo en la transacción, el sistema debe utilizar el campo para el valor anterior de los registros del registro histórico, para restaurar los elementos de datos modificados a los valores que tuvieran antes de comenzar la transacción.

Cuando ocurre un fallo en el sistema se debe consultar el registro histórico para determinar las transacciones que deben rehacerse y las que deben deshacerse. En principio es necesario recorrer completamente el registro histórico para hallar esta información. En este enfoque hay dos inconvenientes principales:

- El proceso de búsqueda consume tiempo.
- La mayoría de las transacciones que deben rehacerse de acuerdo con el algoritmo ya tienen escritas sus actualizaciones en la base de datos. Aunque el hecho de volver a ejecutar estas transacciones no produzca resultados erróneos, sí repercutirá en un aumento del tiempo de ejecución del proceso de recuperación.

Para reducir este tipo de sobrecarga se introducen los puntos de revisión. Durante la ejecución, a más de que el sistema actualiza el registro histórico utilizando una de las dos técnicas ya comentadas, el sistema realiza periódicamente puntos de revisión, en los cuales tiene lugar la siguiente secuencia de acciones:

- Escritura en almacenamiento estable de todos los registros del registro histórico que residan en ese momento en memoria principal.

- Escritura en disco de todos los bloques de memoria intermedia que se hayan modificado.
- Escritura en almacenamiento estable de un registro del registro histórico <revisión>. La presencia de un registro <revisión> en el registro histórico permite que el sistema pueda hacer más eficiente su procedimiento de recuperación.

Cuando se produce un fallo, el esquema de recuperación examina el registro histórico para determinar la última transacción T_i que comenzó su ejecución antes de que tuviera lugar el último punto de revisión.

Una vez que ha sido identificada la transacción T_i sólo es necesario aplicar las operaciones rehacer y deshacer a la transacción T_i y a las transacciones T_j que comenzaron su ejecución después que T_i . Sea T este conjunto de transacciones. Puede ignorarse el resto del registro histórico (la parte del principio) y puede borrarse cuando se desee.

Paginación en la Sombra

En la paginación en la sombra, durante la vida de una transacción se mantienen dos tablas de páginas: la tabla actual de páginas y la tabla de páginas sombra. Ambas tablas son idénticas cuando la transacción comienza. La tabla de páginas sombra y las tablas a las que apunta no sufren ningún cambio mientras dura la transacción. Cuando la transacción se compromete parcialmente se desecha la tabla de páginas sombra y la tabla actual se convierte en la nueva tabla de páginas. Si la transacción aborta, simplemente se desecha la tabla actual de páginas.

La técnica de la paginación en la sombra no puede aplicarse si se permite que varias transacciones se ejecuten concurrentemente, pero el esquema basado en registro histórico sí. No se permite que ninguna transacción pueda modificar un elemento de datos que ya se ha modificado por una transacción incompleta. Para garantizar esta condición puede utilizarse el bloqueo estricto de dos fases.

2.5.2 Transacciones Concurrentes y Recuperación

Se han desarrollado técnicas avanzadas de recuperación para soportar técnicas de bloqueo de alta concurrencia, como las utilizadas para el control de concurrencia con árboles B+. Estas técnicas se basan en el registro deshacer lógico y siguen el principio de repetir la historia. En la recuperación de un fallo del sistema se realiza una fase rehacer utilizando el registro histórico seguida de una fase deshacer sobre el registro histórico para retroceder las transacciones incompletas.

2.5.3 Fallo con pérdida de almacenamiento no volátil

A pesar de que es raro encontrarse con un fallo en el que se pierda información de almacenamiento no volátil, es necesario prepararse para afrontar este tipo de fallos. La idea básica es volcar periódicamente (una vez al día) el contenido entero de la base de datos en almacenamiento estable. Por ejemplo, puede volcarse la base de datos en una o más cintas magnéticas. Se utilizará el volcado más reciente para que la base de datos recupere un estado consistente cuando ocurra un fallo que conduzca a la pérdida de algunos bloques físicos de la base de datos.

Una vez que se complete esta operación, el sistema utilizará el registro histórico para llevar al sistema de base de datos al último estado consistente en el que estuvo antes de producirse el fallo. Para la recuperación por pérdida de almacenamiento no volátil se restituye la base de datos en el disco utilizando el último volcado realizado. Entonces se consulta el registro histórico y se rehacen todas las transacciones que se hubieran comprometido desde que se efectuó ese último volcado.

2.5.4 Sistemas Remotos de Copias de Seguridad

Los sistemas tradicionales de procesamiento de transacciones son sistemas centralizados o sistemas cliente-servidor. Esos sistemas son vulnerables frente a desastres ambientales como el fuego, las inundaciones o los terremotos. Hay una necesidad creciente de sistemas de procesamiento de transacciones que ofrezcan una disponibilidad elevada y que puedan funcionar pese a los desastres ambientales. Estos sistemas deben proporcionar una

disponibilidad elevada, es decir, el tiempo en que el sistema no es utilizable debe ser extremadamente pequeño.

Se puede obtener una disponibilidad elevada realizando el procesamiento de transacciones en un solo sitio, denominado sitio principal, pero tener un sitio remoto copia de seguridad, en el que se repliquen todos los datos del sitio principal.

El sitio remoto debe mantenerse sincronizado con el sitio principal, ya que las actualizaciones se realizan en el sitio principal. La sincronización se obtiene enviando todos los registros del registro histórico desde el sitio principal al sitio remoto copia de seguridad. El sitio remoto copia de seguridad debe hallarse físicamente separado del principal —por ejemplo, se puede ubicar en otra provincia— para que una catástrofe en el sitio principal no afecte al sitio remoto copia de seguridad.

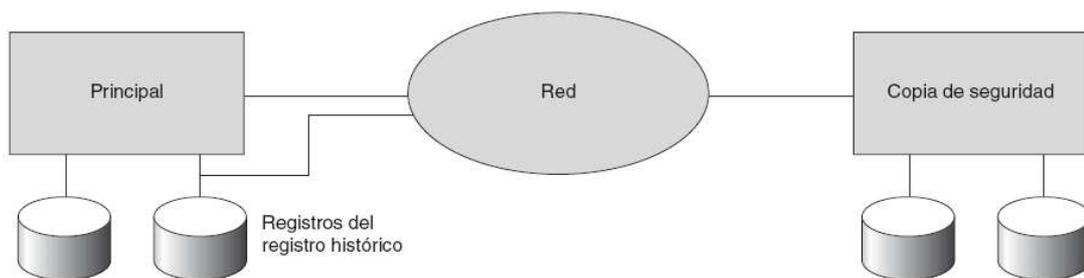


Figura 2.3. Arquitectura de los sistemas remotos de copias de seguridad.

Fuente: Fundamentos de Bases de Datos.

La disponibilidad aumenta mucho en comparación con los sistemas con un solo sitio, dado que el sistema puede recuperarse aunque se pierdan todos los datos del sitio principal. El rendimiento de los sistemas remotos para copias de seguridad es mejor que el de los sistemas distribuidos con compromiso de dos fases.

Una forma alternativa de conseguir alta disponibilidad es usar una base de datos distribuida con los datos replicados en más de un sitio. Son necesarias transacciones para actualizar todas las réplicas de cualquier elemento de datos que actualicen.

CAPÍTULO III

3. FUNCIONAMIENTO DE LOS SISTEMAS DE GESTIÓN DE BASES DE DATOS ORACLE, SQL SERVER, MySQL

3.1 Introducción

El mercado de Sistemas de Gestión de Bases de Datos, es excesivamente grande, sin embargo por ser los más utilizados, en el ámbito empresarial se escoge SQL Server, ORACLE y MySQL como objeto de este estudio.

Cada sistema tiene características, ventajas e inconvenientes, la elección de uno u otro sistema para gestionar nuestra base de datos vendrá definida por las necesidades empresariales. Por esa razón es muy importante conocer su modo de funcionamiento, características, y rendimiento, para posteriormente decidir cual SGBD se acopla mejor a los requerimientos de la empresa. Este capítulo se enfoca en el estudio de cada uno de los Sistemas de Gestión de Bases de Datos desde la perspectiva de su funcionamiento y características. El funcionamiento se lo analiza a partir de los siguientes aspectos: arquitectura del Sistema, catálogo del sistema, almacenamiento e indexación, variaciones y extensiones SQL, procesamiento y optimización de Consultas, control de concurrencia y recuperación, herramientas de administración.

Mientras que las características se definen partir de los siguientes parámetros.

- **Dimensionamiento:** El SGBD deberá garantizar que es capaz de manejar el volumen de datos requerido. Para ello, deberá comprobarse que es adecuado en cada uno de los siguientes puntos: Número total de bases de datos que se van a crear, número total de tablas por base de datos, número máximo de filas por tabla, longitud máxima de fila, número máximo de índices por tabla, número máximo de campos por índice.
- **Sistemas Operativos:** Se especificará los Sistemas Operativos sobre los que debe funcionar el SGBD.
- **Soporte Técnico:** Tipos de Servicio y programas de ayuda al cliente.

3.2 Oracle

Oracle es un sistema de gestión de base de datos relacional (o RDBMS por el acrónimo en inglés de Relational Data Base Management System), fabricado por Oracle Corporation, considerado como uno de los sistemas de bases de datos más completos, destacando su soporte de transacciones, estabilidad, escalabilidad, soporte multiplataforma. Oracle corre en computadoras personales (PC), microcomputadoras, mainframes y computadoras con procesamiento paralelo masivo. Soporta unos 17 idiomas, corre automáticamente en más de 80 arquitecturas de hardware y software distinto sin tener la necesidad de cambiar una sola línea de código. Esto es porque más el 80% de los códigos internos de Oracle son iguales a los establecidos en todas las plataformas de sistemas operativos.

3.2.1 Arquitectura del Sistema

Un servidor Oracle consta de dos elementos: Una instancia de la base de datos que reside en la memoria del servidor y está compuesta por un gran bloque de memoria asignado dentro de un área denominada SGA (System Global Area), junto con una serie de procesos de segundo plano que interactúan entre el SGA y los archivos de bases de datos, residentes en el disco.

Como se observa la arquitectura de Oracle en general, se compone de tres componentes básicos¹⁷:

Estructuras de memoria de Oracle

Oracle utiliza la memoria física del servidor para almacenar información que la instancia Oracle necesita. (Código ejecutable, información de sesión, procesos asociados a la base de datos, información compartida entre procesos, etc.). Contiene instrucciones SQL del diccionario de datos y del usuario, junto con la información en caché que en algún momento terminará por escribirse de forma permanente en el disco, como por ejemplo bloques de datos correspondientes a segmentos de la base de datos e información acerca de las transacciones completadas en la base de datos.

El área de datos asignada para una instancia Oracle se denomina *área global del sistema (SGA)*. Los ejecutables de Oracle residen en el *área de código software*, Además hay otra área denominada *área global del programa* que es privada para cada proceso de servidor y de segundo plano, asignándose un PGA a cada proceso.

- **Área global del sistema (SGA):** Es un grupo de estructuras de memoria de una instalación ORACLE que son compartidas por todos los usuarios de la instancia de base de datos. Cuando se arranca una instancia ORACLE, se asigna memoria al SGA basándose en los valores especificados en el archivo de parámetros de inicialización o incluidos dentro del propio código del software Oracle
- **Área global del programa (PGA):** Es un área de memoria asignada a un proceso y de carácter privado para el mismo. La configuración del PGA depende de la configuración de conexión de la base de datos Oracle, pueden ser una configuración de servidor compartido, o servidor dedicado.
- **Área de código software:** Las áreas de código software almacenan los archivos ejecutables Oracle que están ejecutándose como parte de la instancia ORACLE. Estas áreas de código son estáticas por su propia naturaleza y solo cambian cuando se instala una nueva versión del software.

¹⁷ **LONEY, Kevin y BRILA, Bob.** Oracle 10g: Manual del Administrador. Madrid. McGraw-Hill. 2004. 40p

registro relativa tanto a sucesos rutinarios como a condiciones de error que se presentan en la base de datos.

A continuación la figura ilustra la relación entre las estructuras físicas y las estructuras lógicas que serán comentadas en la sección almacenamiento e indexación.

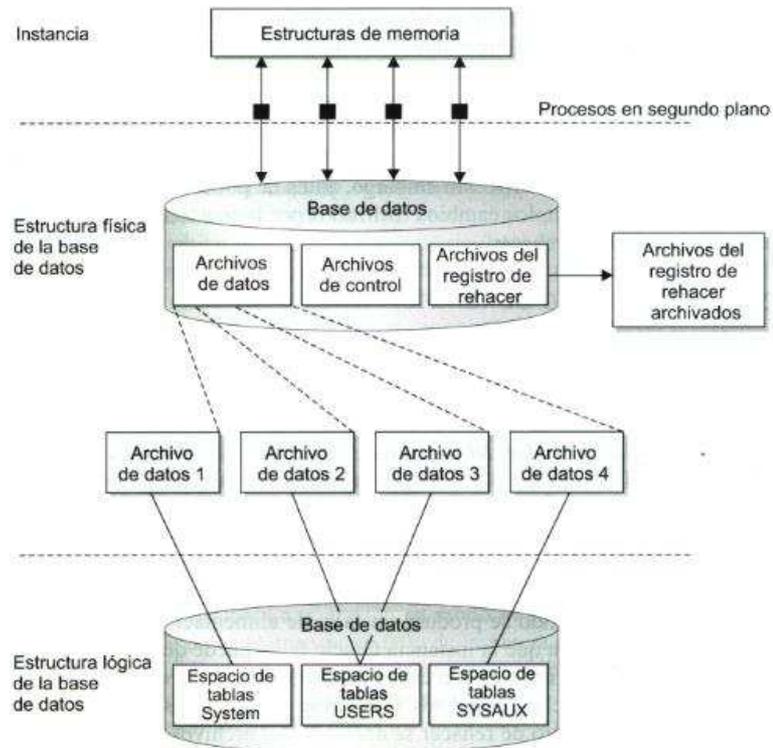


Figura 3.2. Estructuras Físicas de Almacenamiento de ORACLE
Fuente: Oracle Database 10g: Manual del Administrador

3.2.2 Catálogo del Sistema¹⁸

El catálogo de Oracle es un conjunto de tablas y vistas que contienen la definición de la estructura lógica y física de la base de datos, incluyendo los objetos existentes, la situación de los archivos de datos, la configuración de los usuarios, etc.

El catálogo de datos refleja con exactitud el estado de la base de datos en todo momento, actualizándose automáticamente en respuesta a acciones específicas como por ejemplo la modificación de la estructura de la base de datos.

¹⁸ **MANUEL, José.** El Catálogo de Oracle. [On-Line] Iniciación a Oracle. 2001. [Fecha de Consulta: Octubre del 2008] Disponible en: http://www.wikilearning.com/curso_gratis/iniciacion_a_oracle/3861

El catálogo de datos incluye el catálogo público que es accesible por todos los usuarios, y normalmente proporciona información sobre los objetos creados en la base de datos; y el catálogo del sistema que es accesible sólo por usuarios DBA y contiene información de los objetos en la base de datos, así como información específica de la base de datos en sí (versión, parámetros, procesos ejecutándose).

El usuario SYS es propietario de las tablas del catálogo de datos, en las que se almacena información sobre el resto de las estructuras de la base de datos. El usuario SYSTEM posee las vistas que permiten acceder a las tablas del catálogo de datos, para que las utilicen los usuarios restantes. Los usuarios pueden tener acceso a los objetos que poseen o a aquellos a los que se les ha concedido acceso.

Debido a que el catálogo de Oracle está continuamente actualizado, este maneja vistas dinámicas, cuyos datos no se almacenan en disco, sino que son tablas sobre datos contenidos en la memoria del servidor, por lo que almacenan datos actualizados en tiempo real como por ejemplo: información sobre los objetos que están en el caché, total de lecturas y escrituras físicas sobre un datafile de la base de datos del SGA, información acerca de los segmentos de rollback.

Sin embargo hay otros datos que no pueden actualizarse en tiempo real porque penalizarían mucho el rendimiento general de la base de datos, por lo que es necesario actualizar el catálogo de este tipo de datos ejecutando una sentencia especial que se encarga de volcar la información recopilada al catálogo:

```
ANALYZE [TABLE|INDEX] nombre [COMPUTE|ESTIMATE|DELETE] STATISTICS;
```

El catálogo público contiene ciertas tablas encargadas de almacenar información adicional sobre tablas, vistas y columnas. La información que se suele almacenar es información de análisis, valores posibles para las columnas y en general todo aquello que se haya concluido durante la etapa de análisis. Una vez que esta información está en la base de datos, se puede escribir procedimientos o scripts SQL que muestren la información para sacar informes de documentación de base de datos.

3.2.3 Almacenamiento e Indexación

Oracle almacena los datos lógicamente en forma de espacios de tablas y físicamente en forma de archivos. Oracle mantiene una relación entre su sistema de almacenamiento y el espacio de tablas SYSTEM¹⁹.

Estructuras Lógicas de Almacenamiento

Los archivos de datos de una base de datos de Oracle, se agrupan en uno o más espacios de tablas. Dentro de cada espacio de tablas, las estructuras lógicas son segmentos que a su vez se subdividen en extensiones y bloques. A continuación la figura ilustra la relación existente entre las estructuras lógicas de almacenamiento de una base de datos.

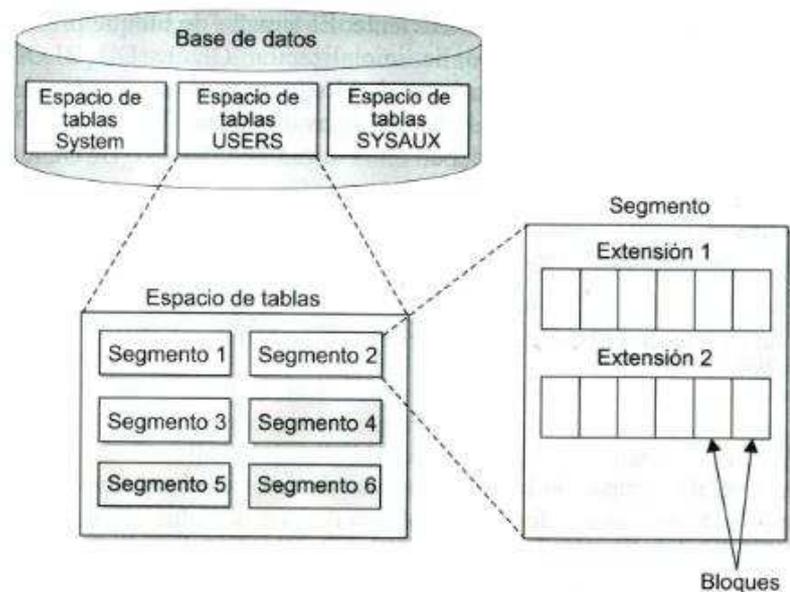


Figura 3.3. Estructuras Lógicas de Almacenamiento
Fuente: Oracle Database 10g: Manual del Administrador

Esta subdivisión lógica del almacenamiento permite a Oracle controlar de forma más eficiente la utilización del espacio de disco, pues permite la configuración de los parámetros de almacenamiento admitiendo un control detallado de cómo se asigna y gestiona el espacio, tales como: El tamaño de una extensión nueva que se va a asignar para proporcionar espacio a las filas que se insertan en una tabla, el porcentaje de utilización de espacio con el cual un bloque de la base de datos se considera lleno y con el cual no se introducirán más filas en ese bloque.

¹⁹ **LONEY, Kevin y BRILA, Bob.** Oracle 10g: Manual del Administrador. Madrid. McGraw-Hill. 2004. 5p

A continuación se resume en una tabla descriptiva las estructuras lógicas de almacenamiento utilizadas por Oracle.

Tabla 3.I. Estructuras Lógicas de Almacenamiento

Estructuras Lógicas de Almacenamiento	
Estructuras Lógicas	Descripción
Espacios de tablas	Es el nombre que tiene un conjunto de propiedades de almacenamiento que se aplican a los objetos que se crean en la base de datos bajo el espacio de tabla indicado. Las propiedades que se asocian a un espacio de tabla son: localización de los ficheros de datos, especificación de máximas cuotas de consumo de disco, control de la disponibilidad de los datos (en línea o fuera de línea), backup de los datos. Para una instalación 10g, se crea un mínimo de dos espacios de tablas: el espacio de tablas SYSTEM, y el espacio de tablas SYSAUX.
Segmentos	El espacio de tablas se divide en unidades lógicas, cada una de las cuales contiene datos para una estructura de datos específica, como por ejemplo una tabla o un índice
Extensión	Debajo del nivel de segmentos se asigna espacio a un nivel de granularidad, denominado extensión. Cada extensión consiste en un conjunto de bloques contiguos de la base de datos. Un bloque de la base de datos es el nivel más bajo de granularidad en el cual Oracle ejecuta E/S a disco
Bloques	Similar a los bloques de un S.O. Un bloque de base de la base de datos no tiene que tener el mismo tamaño que un bloque de un sistema operativo, pero debería ser un múltiplo con el fin de facilitar una E/S de disco eficiente

Fuente: Oracle 10g: Manual del Administrador

Realizado Por: Pilar A. Quizhpe P.

Estructuras Lógicas de la Base de Datos

A continuación se trata sobre los aspectos más importantes de las principales estructuras lógicas de la base de datos²⁰:

- **Tablas:** es la unidad básica de almacenamiento en una base de datos Oracle. Una tabla estándar en Oracle está organizada en montículo esto es, la ubicación de almacenamiento de una fila en una tabla no está basada en los valores contenidos en la fila y se fija cuando la fila se inserta. Otra forma de organización es la agrupación en donde las filas de tablas diferentes se almacenan juntas en el mismo bloque según algunas columnas comunes.

Una organización alternativa es una agrupación asociativa. Aquí, Oracle calcula la localización de una fila aplicando una función asociativa al valor para la columna de

²⁰ **LONEY, Kevin y BRILA, Bob.** Oracle 10g: Manual del Administrador. Madrid. McGraw-Hill. 2004. 9 - 30 pp

agrupación. La función asociativa asigna la fila a un bloque específico en la agrupación asociativa. Puesto que no es necesario el recorrido del índice para acceder a una fila según su valor de columna de agrupación, esta organización puede ahorrar cantidades significativas de E/S a disco. Oracle 10g mantiene una clasificación de las tablas según el tipo de organización de archivos que usan. A continuación se menciona las diversas tablas manejadas por Oracle:

Tabla 3.II. Tipos de tablas de Oracle

Tablas Manejadas por Oracle	
Tipo de tabla	Descripción
Tablas relacionales	Mantienen una organización en cúmulo o en montículo
Tablas temporales	Son temporales en el sentido de los datos almacenados en la tabla, más no desde el punto de vista de la definición de la propia tabla. Existen dos tipos de datos temporales: datos temporales mientras dura la transacción, y datos temporales mientras dura la sesión
Tablas con organización de índice	Se usa cuando la tabla no posee muchas columnas y el acceso a la misma se produce fundamentalmente a través de una única columna.
Tablas de Objetos	Las tablas de objetos tienen filas que son ellas mismas objetos, o instancias de una serie de definiciones de tipos
Tablas externas	Permiten al usuario acceder a un origen de datos. Los metadatos de la tabla se almacenan en el diccionario de datos Oracle, pero los contenidos de las tablas están almacenados externamente. No permite la creación de índices, y no se pueden realizar operaciones de inserción, actualización ni borrado en las tablas externas
Tablas en clúster	Almacena registros relacionados de dos o más relaciones en cada bloque, esto reduce la cantidad de operaciones de E/S para realizar una consulta, y la cantidad de espacio necesario para almacenar las columnas que las tablas tienen en común. Las tablas en clúster pueden ser: clústeres hash y clústeres hash ordenados
Tablas particionadas	Las tablas pueden particionarse o incluso subparticionarse en piezas más pequeñas. La partición es transparente, es decir no es necesario incluir referencias explícitas a ninguna partición concreta en ninguna transacción SQL del usuario final

Fuente: Oracle 10g: Manual del Administrador

Realizado Por: Pilar A. Quizhpe P.

- **Índices:** El manejo de los índices en ORACLE se realiza de forma inteligente, donde el programador sólo crea los índices sin tener que especificar, explícitamente, cuál es el índice que va a usar. Oracle 10g soporta varios tipos de índice cada uno de los cuales, es adecuado para un tipo concreto de tabla, método de acceso o entorno de aplicación:

Tabla 3.III. Tipos de Índices de Oracle

Tipos de Índices	
Tipo de Índice	Descripción
Índice Unívoco	Se implementa a través de un árbol binario B y se lo usa para imponer la restricción de clave principal en una tabla
Índice No Unívoco	Ayuda a acelerar el acceso a una tabla, sin necesidad de imponer la unicidad de los valores
Índice de Clave Inversa	Es un tipo de índice especial que se usa en entornos OLTP, invierte todos los bytes del valor de la clave de cada columna de índice
Índices de Mapas de Bits.	Los índices de mapas de bits utilizan una representación de mapa de bits para entradas de índice que pueden llevar a un ahorro sustancial de espacio (y, por consiguiente, ahorro de E/S a disco), cuando la columna indexada tiene un número moderado de valores distintos. Los índices de mapas de bits en Oracle utilizan la misma clase de estructura de árbol B para almacenar las entradas que un índice normal.
Índices basados en funciones	Además de crear índices sobre una o varias columnas de una tabla. Oracle permite crear índices sobre expresiones que involucran una o más columnas
Índices de reunión	Un índice de reunión es un índice donde las columnas clave no están en la tabla que se referencia mediante los identificadores de filas en el índice. Oracle soporta los índices de reunión mapa de bits principalmente para su uso con esquemas en estrella
Índices de dominio	Oracle permite que las tablas sean indexadas por estructuras de índices que no sean propias de Oracle. Esta característica de extensibilidad del servidor Oracle permite a los fabricantes de software desarrollar los llamados cartuchos con funcionalidad para dominios de aplicación específicos, tales como texto, datos espaciales e imágenes, con la funcionalidad de indexado más allá de la proporcionada por los tipos de índice Oracle estándar

Fuente: Oracle 10g: Manual del Administrador

Realizado Por: Pilar A. Quizhpe P.

- **Restricciones:** Es una regla o conjunto de reglas que pueden definirse sobre una o más columnas de una tabla para ayudar a imponer las reglas de negocio.

- **Vistas:** Las vistas permiten a los usuarios acceder a una presentación personalizada de los datos de una tabla, o incluso de una combinación entre múltiples tablas. Oracle brinda soporte para **Vistas materializadas.**

La característica de la vista materializada permite almacenar el resultado de una consulta SQL y utilizarlo en un procesamiento posterior. Además, Oracle mantiene el resultado materializado, actualizándolo cuando se actualizan las tablas a las que se hicieron referencia en la consulta. Las vistas materializadas se utilizan en el almacén de datos para acelerar el procesamiento de la consulta, pero esta tecnología también se utiliza para la réplica en entornos distribuidos y móviles

- **Usuarios y Esquemas:** Un usuario es una cuenta de acceso a la base de datos. Si el usuario crea y posee objetos en la base de datos, dichos objetos son parte de un esquema que tiene el mismo nombre que el usuario de la base de datos. El usuario siempre tiene privilegios completos y un total control sobre los objetos incluidos en su esquema. Un esquema puede poseer cualquier tipo de objeto: tablas, índices, secuencias, vistas, etc. Los usuarios pueden autenticarse utilizando tres métodos: autenticación de base de datos, autenticación del sistema operativo, y autenticación de red.
- **Perfiles:** Es un conjunto nominado de límites sobre los recursos a todos los usuarios de la base de datos. Como ejemplo de recursos tenemos: tiempo de procesador, sesiones concurrentes, lecturas lógicas, y tiempo de conexión.
- **Secuencias:** Las secuencias generan una serie única de números. Son útiles en la generación única de claves y pueden ser cíclicas o crecer hasta un valor máximo

3.2.4 Particionamiento de Tablas

Oracle soporta varias clases de división horizontal de tablas e índices y esta característica tiene una función principal en la capacidad de Oracle de soportar bases de datos muy grandes. Cada fila en una tabla dividida está asociada con una partición específica.

Esta asociación está basada en la columna o columnas de la división que son parte de la definición de una tabla dividida. Hay varias formas para hacer corresponder valores de

columna a divisiones, dando lugar a varios tipos de divisiones, cada una con distintas características: **divisiones por rangos, asociativas, compuestas y por listas**²¹.

Tabla 3.IV. Particionamiento en Oracle

Características del Particionamiento de Oracle	
Características	Descripción
Particiones por rangos	Los criterios de división son rangos de valores. Este tipo de división está especialmente indicado para columnas de fechas, en cuyo caso todas las filas en el mismo rango de fechas, digamos un día o un mes, pertenecen a la misma partición
Partición asociativa	Usa una función asociativa para hacer corresponder filas con divisiones según los valores en las columnas de la división
Partición Compuesta	La tabla se divide por rangos, pero cada partición tiene subparticiones mediante el uso de división asociativa
Partición por Listas	En la división por listas los valores asociados con una partición particular están en una lista. Este tipo de división es útil si los datos en la columna de división tienen un conjunto relativamente pequeño de valores discretos

Fuente: Oracle 10g: Manual del Administrador

Realizado Por: Pilar A. Quizhpe P.

Independientemente del tipo de esquema de particionamiento que se use cada miembro de una tabla particionada debe tener los mismos atributos lógicos, como por ejemplo nombres de columnas, tipos de datos, restricciones, etc. Sin embargo los atributos físicos de cada partición pueden ser diferentes, dependiendo de su tamaño y de la ubicación en disco. El aspecto clave es que la tabla particionada debe ser lógicamente coherente desde el punto de vista de la aplicación y del usuario.²²

3.2.5 Variaciones y Extensiones SQL

Oracle soporta todas las características principales de SQL: 1999, con algunas pequeñas excepciones tales como distintos tipos de datos. Oracle soporta un gran número de otras constructoras del lenguaje, algunas de las cuales casan con SQL: 1999, mientras que otras son específicas de Oracle en sintaxis o funcionalidad. Por ejemplo Oracle soporta las opera-

²¹ **SILBERSCHATZ, KORTH, SUDARSHAN.** Fundamentos de Bases de Datos. 4ta Edición. Madrid. McGraw Hill. 2002. 618 pp

²² **LONEY, Kevin y BRILA, Bob.** Oracle 10g: Manual del Administrador. Madrid. McGraw-Hill. 2004. 9 - 30 pp

ciones OLAP, incluyendo clasificación, agregación de traslado, cubos y abstracción. Oracle tiene dos lenguajes procedimentales principales, PL/SQL y Java. PL/SQL es una extensión de SQL. El interés de PL/SQL es mezclar la potencia de las instrucciones SQL con la flexibilidad de un lenguaje procedimental en un mismo programa.

Oracle proporciona varios tipos de disparadores, procedimientos almacenados, vistas materializadas, funciones y varias opciones para el momento y forma en que se invocan. Estos se pueden escribir en PL/SQL o Java o como llamadas a C. En general, Oracle ofrece soporte SQL para la gestión de variables, estructuras de control, uso de cursores, gestión de errores, objetos soportados nativamente (funciones, procedimientos, triggers), y gestión de transacciones.

3.2.6 Procesamiento y Optimización de Consultas

Oracle soporta una gran variedad de técnicas de procesamiento en su motor de procesamiento de consultas que permiten el acceso a los datos²³:

- **Exploración de tabla completa.** El procesador de la consulta explora toda la tabla y obtiene información sobre los bloques que forman la tabla del mapa de extensión y explorando esos bloques.
- **Exploración de índices.** El procesador crea una clave de comienzo y/o finalización a partir de las condiciones en la consulta y la utiliza para explorar una parte relevante del índice. Si hay columnas que se tienen que recuperar, que no son parte del índice, la exploración del índice sería seguida por un acceso a la tabla mediante el índice del identificador de fila. Si no hay disponible ninguna clave de inicio o parada la exploración sería una exploración de índice completa.
- **Exploración rápida completa de índices.** El procesador explora las extensiones de la misma forma que la extensión de tabla en una exploración de tabla completa. Si el índice contiene todas las columnas que se necesitan en el índice y no hay buenas claves de inicio y parada que puedan reducir significativamente esa porción del índice que se exploraría en una exploración de índices normal, este método puede ser la

²³ **SILBERSCHATZ, KORTH, SUDARSHAN.** Fundamentos de Bases de Datos. 4ta Edición. Madrid. McGraw Hill. 2002. 619-620 pp

forma más rápida de acceder a los datos. Esto es porque la exploración rápida completa aprovecha de forma completa la E/S de disco de varios bloques. Sin embargo, a diferencia de una exploración completa normal, que recorre los bloques hoja del índice en orden, una exploración rápida completa no garantiza que la salida preserve el orden del índice.

- **Reunión de índices.** Si una consulta necesita solamente un pequeño subconjunto de columnas de una tabla ancha, pero ningún índice contiene todas estas columnas, el procesador puede utilizar una reunión de índices para generar la información relevante sin acceder a la tabla, reuniendo varios índices que contienen en conjunto las columnas necesarias. Ejecuta las reuniones como reunión por asociación sobre los identificadores de filas desde los distintos índices.
- **Acceso a agrupaciones y agrupaciones asociadas.** El procesador accede a los datos utilizando la clave de agrupación.

Optimización de Sentencias SQL

Toda consulta SELECT se ejecuta dentro del servidor en varios pasos. Para la misma consulta, pueden existir distintos caminos para conseguir los mismos resultados, por lo que el servidor es el responsable de decidir qué camino seguir para conseguir el mejor tiempo de respuesta. La parte de la base de datos que se encarga de estas decisiones se llama Optimizador. El camino seguido por el servidor para la ejecución de una consulta se denomina "Plan de Ejecución" (EXPLAIN PLAN).

Oracle tiene dos tipos de optimizadores:

- **Optimizador por reglas (RULE):** Se basa en ciertas reglas para realizar las consultas. No realiza estimación de costes, se calcula un plan de ejecución en base a los posibles caminos de acceso, o sea se escoge el plan más rápido de ejecutar según un ranking de caminos de acceso. No tiene en cuenta el tamaño y distribución de los datos. Es sensible al orden de escritura de las condiciones en la sentencia. Oracle lo considera obsoleto y anuncia su desaparición en próximas versiones.

- **Optimizador por costes (CHOOSE):** El optimizador calcula todos los planes de ejecución posibles y se queda con el de menor coste, o uno lo suficiente bueno. Para estimar el coste se tiene en cuenta datos estadísticos del sistema, y de las tablas e índices que intervienen. El coste estimado de un plan de ejecución tiene en cuenta, para cada camino y operación a ejecutar, las operaciones de E/S, el tiempo de CPU, y la memoria principal disponible. Para averiguar el estado actual de la base de datos, se basa en los datos del catálogo público, por lo que es recomendable que esté lo más actualizado posible (a través de la sentencia ANALYZE), ya que de no ser así, se pueden tomar decisiones a partir de datos desfasados.

Oracle recomienda que todas las consultas se hagan por costes, aunque hay ciertos casos en los que una consulta no se resuelve (o tarda mucho) por costes y por reglas es inmediata. Hay dos modos de forzar a Oracle para que utilice un optimizador y hacer que una consulta se ejecute por reglas o por costes:

La primera es modificando la sesión activa para que todas las consultas sean optimizadas de una manera: `ALTER SESSION SET OPTIMIZER_GOAL = [RULE|CHOOSE];` con esto todas las consultas se ejecutarán utilizando el optimizador indicado. Otra manera es forzando a Oracle a que utilice un optimizador en una consulta concreta. Esto se hace a través de los "HINTS" o sugerencias. Un HINT es un comentario dentro de una consulta `SELECT` que informa a Oracle del modo en que tiene que trazar el plan de ejecución. Los HINTS, deben ir junto detrás de la palabra `SELECT`. Hay que tener en cuenta que si no es posible efectuar lo que se indica con el HINT, Oracle lo ignorará, los HINTS fuerzan el uso del Optimizador por costes (a excepción de rule), y que no afectan a subconsultas en la misma sentencia SQL. El objetivo de la optimización es la puesta a punto de las sentencias SQL, desde el punto de vista del tiempo de ejecución y consumo de recursos.

3.2.7 Transacciones y Concurrencia

Para solucionar los problemas de concurrencia, ORACLE maneja distintos tipos de bloqueo, un modelo de consistencia multiversión y varios niveles de aislamiento.

Modelo de consistencia multiversión

ORACLE es un sistema multiversión que utiliza protocolos basados en marcas de tiempo, por lo que el control de la concurrencia es de varias versiones sobre un objeto de datos. La forma de actuar es la siguiente: si se quiere realizar una transacción sobre un objeto, compara las marcas de tiempo de las distintas versiones del objeto, quedándose con una versión para mantener la serialización del plan de objetos que se está ejecutando. Para el caso particular de ORACLE, el modelo multiversión garantiza la consistencia en lectura.

Este modelo presenta las siguientes características para el control de la concurrencia:

- Usa un sistema basado en reservas, de manera que el planificador realiza las peticiones de reservas. El comportamiento por defecto, garantiza la consistencia de lectura, es decir, una transacción sólo ve los datos que están confirmados al comienzo de su ejecución. Cualquier cambio, que se haya graduado o no después de comenzar a ejecutarse la transacción, no estará disponible para esta transacción. Para evitar que las lecturas esperen a que otros usuarios escriban, se usa un mecanismo similar a la multiversión. Este mecanismo se basa en SCN (System Change Number) que actúa como una marca de tiempo y en los registros ROLLBACK, que lo que hacen es almacenar valores antiguos de los datos, de manera que si se quiere leer de una tabla, y algún registro ha sido modificado cuando la transacción ya estaba en curso, el sistema recupera el valor antiguo del registro consultando los segmentos de ROLLBACK. Cuando una consulta comienza, se le asigna un número (SCN), los datos leídos por una consulta tienen un SCN más pequeño. Los bloques con SCN más alto(es decir, más reciente) son reconstruidos con el segmento ROLLBACK.
- Los escritores tampoco esperan a los lectores, aunque si tienen que esperar a usuarios que escriben en los mismos datos que ellos. Cuando se lee y modifica al mismo tiempo, ORACLE crea vistas consistentes en lectura. Si se modifica antes de que hagamos un COMMIT, guardamos los valores antiguos de los datos en los Segmentos de ROLLBACK. Con la información del Segmento de ROLLBACK y la que hay actualizada en el Área Global del Sistema, creamos una vista consistente.

Cuando hacemos el COMMIT (confirmamos la transacción), se modifican los datos actualizándolos para todas las vistas posteriores.

- Soporta granularidad múltiple con dos niveles: de tabla y de tupla. Si reservamos una tupla, previamente hemos reservado la tabla. Para poder borrar o modificar una tabla, tenemos que tener una reserva exclusiva sobre dicha tabla.

Control de concurrencia por bloqueos

Bloqueos son los mecanismos que utiliza Oracle para evitar que dos transacciones accedan al mismo recurso. Oracle automáticamente obtiene los bloqueos necesarios cuando ejecuta alguna sentencia en SQL y utiliza el nivel menos restrictivo guiándose por las siguientes reglas:

- Operaciones de lectura no esperan a las de escritura sobre los mismos datos.
- Operaciones de escritura no esperan a las de lectura sobre los mismos datos.
- Operaciones de escritura solamente esperan a otras operaciones de escritura que intentan modificar la misma tupla.

ORACLE adquiere automáticamente los bloqueos a nivel de tuplas (TX), y se mantienen hasta que la transacción haga un COMMIT o ROLLBACK. A continuación mostramos los tipos de bloqueos que se realizan para las diferentes sentencias DDL:

Tabla 3.V. Bloqueos para sentencias DDL

Tipos de Bloqueos para sentencias DDL		
Sentencias SQL	Bloqueo tupla (TX)	Bloqueo Tabla (TM)
SELECT... FROM TABLE		
INSERT INTO TABLE	X	RX
UPDATE TABLE	X	RX
DELETE FROM TABLE	X	RX

Realizado Por: Pilar A. Quizhpe P.

Los tipos de reservas a nivel de tablas son los siguientes:

Tabla 3.VI. Reservas a nivel de tablas

Reservas a Nivel de Tablas						
Sentencias SQL	Bloqueo de tabla	RS	RX	S	SRX	X
SELECT... FROM TABLE	RX	S	S	S	S	S
INSERT INTO TABLE	RX	S	S	N	N	N
UPDATE TABLE	RX	S*	S*	N	N	N
DELETE FROM TABLE	RX	S*	S*	N	N	N

Realizado Por: Pilar A. Quizhpe P.

S: SI

N: NO

S*: habrá este tipo de bloqueo si no existen bloqueos de otras transacciones que pueden ser conflictivas.

A continuación vamos a definir los diferentes tipos de bloqueos:

- ROW SHARE MODE): Permite a otras transacciones, seleccionar, insertar, modificar o borrar tuplas en la misma tabla. NO permite a otras transacciones reservar la tabla en modo exclusivo (EXCLUSIVE MODE).
- RX (ROW EXCLUSIVE MODE): Permite a otras transacciones, seleccionar, insertar, modificar o borrar tuplas en la misma tabla. NO permite a otras transacciones reservar en(S, SRX, X).
- S (SHARE MODE): Permite a otras transacciones seleccionar tuplas. NO permite reservar en modo (SRX, X, RS).
- SRX (SHARE ROW EXCLUSIVE MODE): Sólo lo puede adquirir una transacción cada vez. No permite a otras transacciones reservar en modo (S, SRX, RX, X).
- X (EXCLUSIVE MODE): Sólo lo puede adquirir una transacción. Prohíbe al resto de las transacciones realizar ningún tipo de sentencia SQL y ningún tipo de reserva a nivel de tabla.

Niveles de Aislamiento de Transacciones

Los niveles de aislamiento que ofrece ORACLE son:

- READ COMMITTED: Es el nivel de aislamiento por defecto. Cada consulta de una transacción sólo ve los datos que fueron confirmados antes de que comenzara la transacción, es decir, si una sentencia DML requiere modificar filas que están bloqueadas por otra transacción, la sentencia esperará hasta que se libere el bloqueo. De esta manera se consigue Consistencia a Nivel de Sentencia.
- SERIALIZABLE: Solamente dispone de los datos cuyas transacciones fueron confirmadas y aquellos cambios efectuados por ella misma. Así conseguimos Consistencia a Nivel de Transacción.

- READ ONLY. Transacciones de sólo lectura, ven los datos que han sido confirmados antes de empezar, y no permiten modificaciones. Este último no pertenece a ninguna de las normas establecidas en SQL 92. Útil para informes que consultan muchas tablas que están continuamente siendo actualizadas.

3.2.8 Recuperación

Oracle permite determinar qué estructuras de datos están intactas y cuales necesitan recuperación, a través de estructuras básicas de recuperación como los archivos de control, registros históricos rehacer, registros históricos rehacer archivados y segmentos de retroceso.

El archivo de control contiene varios metadatos que son necesarios para operar en la base de datos, incluyendo la información sobre las copias de seguridad. Oracle registra cualquier modificación transaccional de una memoria intermedia de la base de datos en el registro histórico rehacer, que consiste en dos o más archivos. Registra la modificación como parte de la operación que la causa y sin considerar si la transacción finalmente se produce. Registra los cambios de los índices y segmentos de retroceso así como los cambios a la tabla de datos. Cuando se llenan los registros históricos rehacer se archivan mediante uno o varios procesos en segundo plano (si la base de datos se ejecuta en modo archive log). El segmento de retroceso contiene información sobre versiones anteriores de los datos (esto es, información para deshacer). Además de esta función en el modelo de consistencia de Oracle, la información se utiliza para restaurar la versión anterior de los datos cuando se deshace una transacción que ha modificado los datos.

Durante la recuperación de una copia de seguridad, Oracle ejecuta dos pasos para alcanzar un estado consistente de la base de datos como existía antes del fallo. En primer lugar Oracle rehace las transacciones aplicando los archivos históricos rehacer (archivados) a la copia de seguridad. Esta acción lleva a la base de datos a un estado que existía en la fecha del fallo, pero no necesariamente un estado consistente, puesto que los registros históricos deshacer incluyen datos no comprometidos. En segundo lugar, Oracle deshace las transacciones no comprometidas mediante el uso del segmento de retroceso. La base de datos está ahora en un estado consistente. La recuperación en una base de datos que ha

sido objeto de una actividad transaccional grande debido a la última copia de seguridad puede ser costosa en tiempo. Oracle soporta recuperación en paralelo en la cual se utilizan varios procesos para aplicar información de rehacer simultáneamente. Oracle proporciona una herramienta GUI, el gestor de recuperación, que automatiza la mayor parte de las tareas asociadas con copias de seguridad y recuperación.

3.3 MySQL

MySQL es un sistema de gestión de base de datos relacional, multihilo y multiusuario; incluye todos los elementos necesarios para instalar el programa, preparar diferentes niveles de acceso de usuario, administrar el sistema y proteger los datos, puede desarrollar sus propias aplicaciones de bases de datos en la mayor parte de lenguajes de programación utilizados en la actualidad y ejecutarlos en casi todos los sistemas operativos.

MySQL es una base de datos robusta que puede ser comparada con una base de datos comercial, compite con sistemas RDBMS propietarios como Oracle, SQL Server y DB2, disponiendo de procesamiento de transacciones a través del motor de almacenamiento InnoDB compatible con ACID y, que dispone de procedimientos almacenados, triggers, y vistas.

MySQL es lo suficientemente flexible para trabajar en entornos con gran demanda, tales como aplicaciones web; al mismo tiempo, puede impulsar aplicaciones empotradas, almacenes de datos, indexación de contenidos, sistemas de mensajería, sistemas redundantes de alta disponibilidad, procesamiento de transacciones en línea (OLTP), y mucho más.

3.3.1 Arquitectura del Sistema

La arquitectura de MySQL es muy diferente a la de otros servidores de bases de datos. La característica más inusual e importante de MySQL, es su arquitectura de motores de almacenamiento, cuyo diseño separa el procesamiento de consultas y otras tareas del servidor, del almacenamiento de datos y su recuperación, incluso a partir de la versión 5.1 se puede cargar motores de almacenamiento como plugins en tiempo de ejecución.

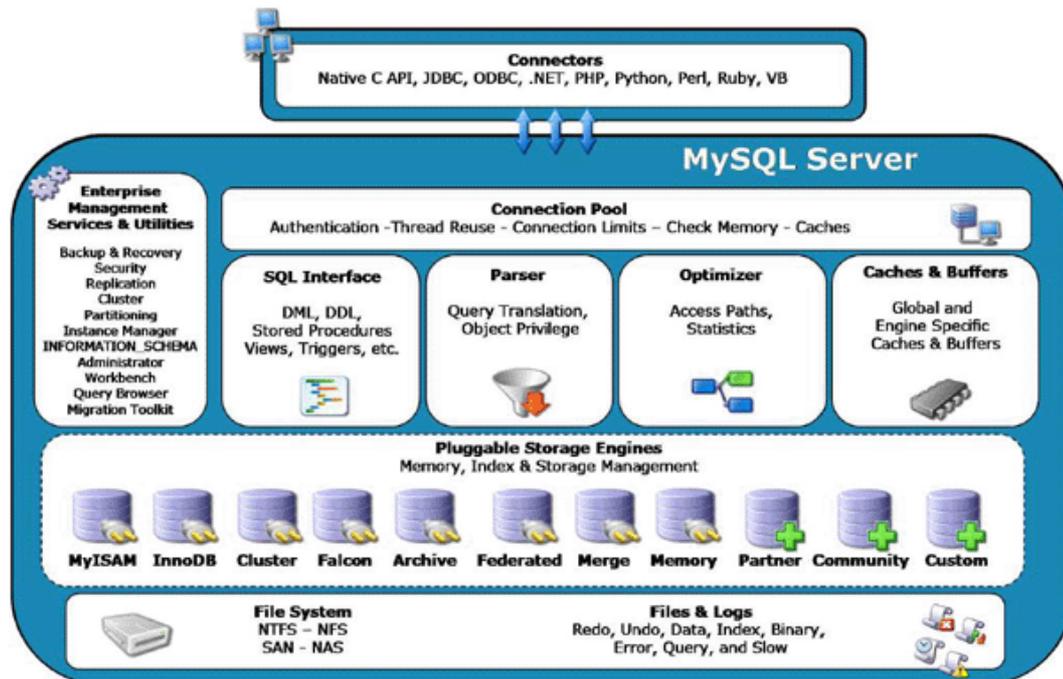


Figura 3.4. Arquitectura de MySQL
Fuente: www.mysql.com

Sobre la capa del motor de almacenamiento, existe un pool de conectores disponibles a través de los cuales se acceden a los módulos del servidor que analizan consultas, optimizan rutas de acceso, etc. El pool de conexión, proporciona autenticación, gestiona las amenazas, las conexiones, la memoria y las cachés.

Los motores de almacenamiento son los componentes del servidor de bases de datos que llevan a cabo acciones en los datos subyacentes que se mantienen en el nivel del servidor físico.

Los motores de almacenamiento evitan la carga de trabajo no productivo natural de los sistemas propietarios, que normalmente deben convivir con la carga de trabajo no productivo transaccional de sus arquitecturas de tablas, incluso cuando no es necesario. Por ejemplo, un centro en el que principalmente se consultan datos puede evitar todo el trabajo no productivo transaccional utilizando el motor de almacenamiento MyISAM, que es el predeterminado para MySQL. El motor MyISAM ofrece capacidad de consulta e inserción de alta velocidad, no es transaccional, proporciona bloqueo a nivel de tabla y trabaja con índices, es un buen motor para los almacenes de datos tradicionales.

La base de datos MySQL incluye un grupo de motores de almacenamiento internos. Los motores de almacenamiento externos están configurados para optimizar el rendimiento de productos y situaciones específicos, los suministran tanto desarrolladores de software independientes como la comunidad MySQL.

Al contar con varios motores de almacenamiento, tiene la posibilidad de aprovechar varias bases de datos. Por ejemplo: algunos motores de almacenamiento, como los utilizados para el archivado, son no transaccionales por naturaleza, pero permiten insertar y leer datos de forma muy eficiente. Otros motores de almacenamiento están configurados para ser eficientes en las operaciones transaccionales, mientras que un tercer grupo proporciona alta disponibilidad mediante técnicas de Clustering.

Si se migra a MySQL desde otra base de datos, no se necesita cambiar la forma en que las aplicaciones interactúan con el servidor de bases de datos: el código SQL es el mismo. Pero dentro del servidor MySQL, puede elegir los motores de almacenamiento que se adapten mejor a las necesidades de su aplicación.

Por ejemplo, si necesita una base de datos transaccional, puede utilizar una misma sentencia para transformar su base de datos no transaccional en un sistema transaccional. Si después decide que necesita un almacén de datos, un solo comando puede permitirle utilizar una potente base de datos orientada a columnas configurada para recuperar datos. Si su actividad principal consiste en capturar y almacenar datos, el motor de almacenamiento Archive está diseñado para gestionar eficazmente grandes volúmenes de inserciones y comprimir datos.

Arquitectura Lógica de MySQL

La capa más alta contiene los servicios que la mayoría de redes basadas en servidores o herramientas cliente/servidor necesitan: manejo de conexiones, autenticación, seguridad, etc. La segunda capa se compone por muchos de los cerebros de MySQL, incluido el código para el analizador sintáctico, análisis y optimización de consultas, gestión de cache y buffer, además de todas las funciones incorporadas (datos, reloj, matemáticas, encriptación).

Cualquier funcionalidad proporcionada al otro lado de los motores de almacenamiento vive en este nivel por ejemplo procedimientos almacenados, triggers, vistas, etc.

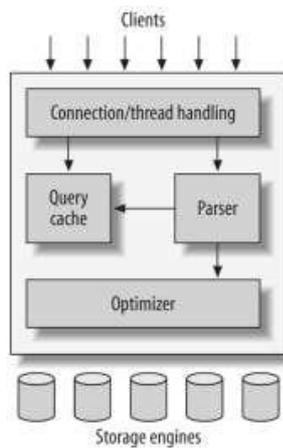


Figura 3.5. Vista Lógica de la Arquitectura de MySQL

La tercera capa es la responsable de almacenar y recuperar todos los datos guardados en MySQL, mediante los motores de almacenamiento, cada motor de almacenamiento tiene sus propios beneficios y desventajas. El servidor se comunica con los motores de almacenamiento a través del storage engine API.

Esta interface oculta las diferencias de los motores de almacenamiento y los hace transparentes principalmente a la capa de consulta. El API también contiene funciones en el nivel más bajo que ejecutan operaciones como por ejemplo “empezar una transacción” o “extraer filas que tienen su llave primaria”.

3.3.2 Catálogo del Sistema

El catálogo de datos denominado `INFORMATION_SCHEMA` proporciona acceso a los metadatos de la base de datos, tales como el nombre de la base de datos o tabla, el tipo de datos de una columna, o permisos de acceso. Almacena información acerca de todas las otras bases de datos que mantiene el servidor MySQL y está compuesto por varias vistas de sólo lectura, por lo que no existe ningún fichero asociado con ellas.

Cada usuario MySQL tiene derecho a acceder a estas tablas, pero sólo a los registros que se corresponden a los objetos a los que tiene permiso de acceso²⁴.

3.3.3 Almacenamiento de Datos

El almacenamiento e indexación de los datos es controlado por la capa de motores de almacenamiento. El motor de almacenamiento (storage-engine) es quien almacenará, manejará y recuperará información de una tabla en particular. MySQL soporta varios motores de almacenamiento que tratan con distintos tipos de tabla. Sin embargo MyISAM e InnoDB son los dos motores de almacenamiento más comunes en MySQL. Comparando MyISAM vs InnoDB, ninguno se destaca como la solución para la mayoría de los casos. Cada uno tiene sus pros y sus contras, por lo tanto al momento de decidir que motor de almacenamiento a utilizar dependerá mucho del escenario donde se aplique. Los motores de almacenamiento de MySQL incluyen algunos que tratan con tablas transaccionales y otros que no lo hacen.

A continuación tenemos un extracto de las características más importantes soportadas por los motores de almacenamiento tomadas del manual de referencia de MySQL.

Tabla 3.VII. Características de los Motores de Almacenamiento

Características de los Motores de Almacenamiento						
Característica	MyISAM	BDB	Memory	InnoDB	Archive	NDB
Soporte de Transacciones	No	Si	No	Si	No	Si
Granularidad de Bloqueo	Tabla	Página	Tabla	Registro	Registro	Registro
Control de Concurrencia	No	No	No	Si	Si	Si
Limites de Almacenamiento	Ilimitado	Ilimitado	Limitado	64TB	Ilimitado	Limitado
Indexación	B TREE Full Text	B TREE Hash	B TREE	B TREE Hash Clustered	No soporta	B TREE Hash
Costo de Almacenamiento	Bajo	Bajo	-	Alto	Muy Bajo	Alto
Costo de Memoria	Bajo	Bajo	Medio	Alto	Bajo	Alto
Velocidad de Inserción	Alto	Alto	Alto	Bajo	Muy Alto	Alto
Compresión de Datos	Si	No	No	No	Si	No
Encriptación de Datos	Si	Si	Si	Si	Si	Si

Fuente: Análisis comparativo de bases de datos de código abierto vs código cerrado.

²⁴ **MySQL AB y Sun Microsystems.** INFORMATION SCHEMA. [ON-LINE]. MySQL 5.0 Reference Manual. [Consultado: Octubre 2008]. Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/information-schema-tables.html>

MYSQL ofrece varios tipos de almacenamiento propios de MYSQL, diseñados de acuerdo a las necesidades de las aplicaciones. Por ejemplo²⁵:

- MYISAM, es el default de los tipos de almacenamiento, y es uno de los más usados en aplicaciones web, datawarehousing y otros tipos de aplicación
- INNODB, usado para aplicaciones transaccionales, soporta algunas características incluida ACID, boqueo a nivel de fila, los usuarios que escriben información no bloquean a otros usuarios que leen información y viceversa, soporta claves foráneas.
- BDB, es otra alternativa de manejo transaccional como INNODB que soporta: COMMIT, ROLLBACK y otras características transaccionales.
- MEMORY, almacena todos los datos en RAM para proporcionar un acceso rápido en ambientes que requieren una vista rápida de la información.
- MERGE, usado en ambientes de datawarehousing, pues permite realizar un grupo lógico de series de tablas MYISAM idénticas y referenciadas estas como un único objeto.
- ARCHIVE, provee una solución perfecta de almacenamiento y recuperación de grandes tablas históricas, archivos o información de auditoría.
- FEDERATED, ofrece la habilidad de realizar una especie de link entre los datos de servidores MYSQL creando una base de datos lógica de varios servidores ubicados físicamente en distintos lugares. Muy bueno para ambientes distribuidos con un manejo inteligente de los datos.
- CLUSTER/NDB, para ambientes con alto nivel de rendimiento que requieren un alto nivel de disponibilidad.

3.3.4 Indexación de Datos

Cada motor de almacenamiento tiene su propio comportamiento en materia de índices y cada una de ellos lo procesa de manera diferente. No todos los tipos de índices están disponibles para los distintos motores de almacenamiento. Es importante tener claro cómo se va a utilizar los índices que se van a necesitar ante de seleccionar el tipo motor de

²⁵ **GILFILLAN, Ian.** *La Biblia de MySQL*. Barcelona. Anaya Multimedia.

almacenamiento. En ocasiones, lo que parece ser el motor de almacenamiento perfecto se convierte en la peor elección porque o se puede utilizar un determinado tipo de índice en ella o no.

Cuando indexamos una columna en particular, MySQL crea otra estructura de datos (un índice) que usa para almacenar información extra acerca de los valores en la columna indexada. Los valores indexados son llamados frecuentemente claves. Aunque esta es la manera simple de explicar los índices, realmente es un poco más complejo, ya que MySQL almacena todas las claves del índice en una estructura de datos de árbol B-TREE. Esta estructura de datos de árbol le permite a MySQL encontrar claves muy rápidamente.

3.3.5 Particionamiento de Tablas

Entre las principales mejoras destaca la partición en tablas e índices, ya que soporta cinco formas de particionamiento horizontal de datos, lo que permite acelerar los tiempos de respuesta a las peticiones examinando sólo la parte de los datos necesarios.

3.3.6 Variaciones y Extensiones SQL

MySQL se basa en la potencia del lenguaje SQL para la gestión de datos, sin embargo en la actualidad MySQL introduce extensiones espaciales para permitir la generación, almacenamiento, y análisis de elementos geográficos, estas características están solo disponibles para tablas MyISAM.

- **Tipos de datos espaciales de MySQL** que corresponden a las clases OpenGis.
- **Procedimientos Almacenados:** Los procedimientos almacenados y funciones son nuevas funcionalidades a partir de la versión de MySQL 5.0. Un procedimiento almacenado es un conjunto de comandos SQL que pueden almacenarse en el servidor. Una vez que se hace, los clientes no necesitan relanzar los comandos individuales pero pueden en su lugar referirse al procedimiento almacenado.
- **Disparadores (Triggers):** A partir de MySQL 5.0.2 se incorporó el soporte básico para disparadores (triggers). Un disparador es un objeto con nombre dentro de una base de datos el cual se asocia con una tabla y se activa cuando ocurre en ésta un evento en particular. Por ejemplo, las siguientes sentencias crean una tabla y un

disparador para sentencias `INSERT` dentro de la tabla. El disparador suma los valores insertados en una de las columnas de la tabla:

- **Vistas:** Las vistas (incluyendo vistas actualizables) fueron introducidas en la versión 5.0 del servidor de base de datos MySQL. MySQL no soporta vistas materializables sin embargo, pueden llegar a ser emuladas con software específico para esta tarea.

3.3.7 Procesamiento y Optimización de Consultas

MySQL descompone gramaticalmente las consultas para crear una estructura interna denominada árbol sintáctico, entonces aplica una variedad de optimizaciones. Esto puede incluir reescribir la consulta determinando el orden en la cual esta leerá las tablas, escogiendo cual índice usar, etc. Además, se puede pasar hints al optimizador a través de palabras claves en la consulta, lo que afecta la forma de realizar el proceso. También se puede preguntar al servidor varios aspectos de la optimización, este le deja saber qué decisión ha tomado y da un punto de referencia para arreglar consultas, esquemas, y parámetros de tal manera que todo se ejecute tan eficiente como sea posible.

El motor de almacenamiento afecta, como el servidor optimiza la consulta, ya que el optimizador pregunta al motor de almacenamiento acerca de sus capacidades, el costo de ciertas operaciones y las estadísticas sobre la tabla de datos. Por ejemplo, algunos motores de almacenamiento soportan tipos de índices que pueden ser provechosos para ciertas consultas.

Optimización de consultas

MySQL permite acelerar las consultas haciendo uso de los índices de una manera inteligente. Los índices son usados para encontrar rápidamente los registros que tengan un determinado valor en alguna de sus columnas.

Cuando MySQL trata de responder una consulta, examina una variedad de estadísticas acerca de nuestros datos y decide como buscar los datos que deseamos de la manera más rápida. Sin embargo, cuando en una tabla no existen índices en los cuales pueda auxiliarse MySQL para resolver una consulta se tendrán que leer todos los registros de la tabla de manera secuencial. Esto es comúnmente llamado un "escaneo completo de una tabla", y es

muchas veces algo que se debe evitar. En particular, debemos evitar los escaneos completos de tablas por las siguientes razones:

- **Sobrecarga de CPU.** El proceso de checar cada uno de los registros en una tabla es insignificante cuando se tienen pocos datos, pero puede convertirse en un problema a medida que va aumentando la cantidad de registros en nuestra tabla. Existe una relación proporcional entre el número de registros que tiene una tabla y la cantidad de tiempo que le toma a MySQL revisarla completamente.
- **Concurrencia.** Mientras MySQL está leyendo los datos de una tabla, éste la bloquea, de tal manera que nadie más puede escribir en ella, aunque si pueden leerla. Cuando MySQL está actualizando o eliminando filas de una tabla, éste la bloquea, y por lo tanto nadie puede leerla.
- **Sobrecarga de disco.** En una tabla muy grande, un escaneo completo consume una gran cantidad de entrada/salida en el disco. Esto puede alentar significativamente nuestro servidor de bases de datos, especialmente si tenemos un disco IDE algo antiguo.

En resumen, lo mejor es tratar de que los escaneos completos de tablas sean mínimos especialmente si nuestra aplicación necesita escalabilidad en tamaño, número de usuarios, o ambos.

Cuando MySQL encuentre que hay un índice en una columna, lo usará en vez de hacer un escaneo completo de la tabla. Esto reduce de manera importante los tiempos de CPU y las operaciones de entrada/salida en disco, a su vez que se mejora la concurrencia porque MySQL bloqueará la tabla únicamente para obtener las filas que necesite (en base a lo que encontró en el índice). Cuando se tiene grandes cantidades de datos en nuestras tablas, la mejora en la obtención de los datos puede ser muy significativa.

Como es de observarse las tablas con pocos índices devolverán los resultados muy rápido. Pero la inclusión de demasiados índices, aunque no suele ser normal, también ocasiona degradación de la base de datos. Los índices ocupan espacio de disco y, como están ordenados, cada vez que se realice una operación de inserción o de actualización, es necesario volver a organizar el índice para incluir los cambios, lo que da como resultado una

carga de trabajo adicional significativa. La eficiencia en el uso de los índices depende también de la configuración de MySQL.

3.3.8 Transacciones y Concurrencia

Como se sabe, quien dota a MySQL de capacidad transaccional, es el motor de almacenamiento InnoDB, por tal razón se analizará el modelo de transacciones que este motor ofrece. En el modelo de transacciones de InnoDB, la meta es combinar las mejores propiedades de una base de datos multiversión con el tradicional bloqueo de dos fases.

InnoDB bloquea a nivel de fila y ejecuta consultas por defecto como lecturas consistentes (consistent reads) no bloqueadas, al estilo de Oracle.

Modos de Bloqueo de la tabla InnoDB

InnoDB implementa un bloqueo a nivel de fila estándar, donde hay dos tipos de bloqueos:

- **Bloqueo Compartido (Shared) (s)** le permite a una transacción leer una fila. Si la transacción A sostiene un **bloqueo compartido (s)** sobre una tupla t , entonces una solicitud de otra transacción B para un bloqueo x sobre t no puede ser atendido inmediatamente. Una solicitud de otra transacción B para un bloqueo s sobre t puede ser atendido inmediatamente. En consecuencia, tanto A como B sostendrán un bloqueo s sobre t .
- **Bloqueo Exclusivo (Exclusive) (x)** le permite a una transacción actualizar o eliminar una fila. Si una transacción A sostiene un **bloqueo exclusivo (x)** sobre una tupla t , entonces una solicitud de otra transacción B para establecer un bloqueo de cualquier tipo sobre t no puede ser atendida inmediatamente. En lugar de eso, la transacción B debe esperar a que la transacción A libere el bloqueo en la tupla t .

Adicionalmente, InnoDB soporta:

- **Bloqueo de Granularidad Múltiple** (multiple granularity locking), el cual permite que existan simultáneamente bloqueos en registros y bloqueos en tablas enteras.

Para hacer práctico el nivel de bloqueo de granularidad múltiple, se emplean tipos adicionales de bloqueo, llamados bloqueos de intención (intention locks).

- **Bloqueos de intención:** son bloqueos de tabla en `InnoDB`. La idea detrás de los mismos es que una transacción indique qué tipo de bloqueo (compartido o exclusivo) requerirá más tarde sobre una fila de esa tabla.

En `InnoDB` se utilizan dos tipos de bloqueos de intención (asumiendo que la transacción T ha solicitado un bloqueo del tipo indicado en la tabla R):

- Intención compartida (Intention Shared) (`IS`): La transacción T trata de establecer bloqueos S en tuplas individuales de la tabla T .
- Intención exclusiva (Intention exclusive) (`IX`): La transacción T trata de establecer bloqueos X en las tuplas.

Luego, el protocolo de bloqueo de intención es el siguiente:

- Antes de que de una determinada transacción logre un bloqueo S en una determinada fila, primero debe conseguir establecer un bloqueo `IS` o superior en la tabla que contiene a la fila.
- Antes de que de una determinada transacción logre un bloqueo X en una determinada fila, primero debe conseguir establecer un bloqueo `IX` en la tabla que contiene a la fila.

Por lo tanto, los bloqueos de intención solamente bloquean solicitudes sobre tablas completas. El propósito principal de `IX`, e `IS` es mostrar que alguien está bloqueando una fila, o va a bloquear una fila en la tabla.

Niveles de Aislamiento

A partir de MySQL 5.0, **InnoDB** ofrece los cuatro niveles de aislamiento de transacciones descritos por el estándar SQL. Cada nivel de aislamiento especifica los tipos de acciones que no están permitidos realizar en cuanto a este proceso.

- **READ UNCOMMITTED**

Las sentencias **SELECT** son ejecutadas sin realizar bloqueos, pero podría usarse una versión anterior de un registro. Por lo tanto, las lecturas no son consistentes al usar este nivel de aislamiento. En otro caso, este nivel de aislamiento funciona igual que **READ COMMITTED**.

- **READ COMMITTED**

Similar en parte al mismo nivel de aislamiento de Oracle. En este nivel de aislamiento, todas las sentencias **SELECT...FOR UPDATE** y **SELECT...LOCK IN SHARE MODE** bloquean solamente los registros de índice, no los espacios vacíos que los preceden, por lo tanto se permite la libre inserción de nuevos registros junto a los bloqueados.

Las sentencias **UPDATE** and **DELETE** que empleen un índice único con una condición de búsqueda única bloquean solamente el registro de índice hallado, no el espacio que lo precede. En las sentencias **UPDATE** y **DELETE** que actúan sobre rangos de registros, **InnoDB** debe bloquear los espacios vacíos y bloquear las inserciones de otros usuarios en los espacios vacíos que hay dentro del rango. Esto es necesario debido a que las "filas fantasma" deben ser bloqueadas para que funcionen la replicación y recuperación en MySQL.

Las lecturas consistentes se comportan como en Oracle: Cada lectura consistente, incluso dentro de la misma transacción, establece y lee su propia captura tomada de la base de datos.

- **REPEATABLE READ**

Este es el nivel de aislamiento predeterminado de **InnoDB**. Las sentencias **SELECT... FOR UPDATE**, **SELECT...LOCK IN SHARE MODE**, **UPDATE**, y **DELETE** que utilicen un índice único con una condición de búsqueda única, bloquean solamente el registro de índice hallado, no el espacio vacío que lo precede. Con otras condiciones de búsqueda, estas operaciones emplean bloqueo de clave siguiente (next-key), bloqueando el rango de índice cubierto por la operación incluyendo los espacios vacíos, y bloqueando las nuevas inserciones por parte de otros usuarios.

En lecturas consistentes (consistent reads), hay una importante diferencia con respecto al nivel de aislamiento anterior: En este nivel, todas las lecturas consistentes dentro de la misma transacción leen de la captura de la base de datos tomada por la primera lectura. Esta práctica significa que si se emiten varias sentencias **SELECT** dentro de la misma transacción, éstas serán consistentes unas con otras.

- **SERIALIZABLE**

Este nivel es similar a **REPEATABLE READ**, pero todas las sentencias **SELECT** son convertidas implícitamente a **SELECT... LOCK IN SHARE MODE**.

Esquemas Multiversión

InnoDB utiliza su característica de multiversión para presentar a una consulta una captura de la base de datos en un momento determinado. La consulta ve los cambios realizados exactamente por aquellas transacciones confirmadas antes de ese momento, y no los cambios hechos con posterioridad o por transacciones no confirmadas. La excepción a esto es que la consulta ve los cambios efectuados por la transacción a donde pertenece.

Si se está ejecutando con el nivel de aislamiento predeterminado **REPEATABLE READ**, entonces todas las lecturas consistentes dentro de la misma transacción leen la captura creada por la primera lectura en esa transacción. Se puede refrescar esta captura confirmando la transacción actual y emitiendo nuevas consultas. Lectura consistente es el modo por defecto en el cual **InnoDB** procesa las sentencias **SELECT** en los niveles de

aislamiento `READ COMMITTED` y `REPEATABLE READ`. Una lectura consistente no establece ningún bloqueo en las tablas a las que accede, y, por lo tanto, otros usuarios están libres para modificar las tablas sobre las que se está haciendo la lectura consistente.

Bloqueos en Modo Compartido

En ciertas circunstancias, no es conveniente una lectura consistente. Por ejemplo, se podría desear agregar una fila en la tabla hijo, y estar seguro de que dicha fila tiene una fila padre en la tabla padre. El siguiente ejemplo muestra cómo implementar integridad referencial en el código de la aplicación. Suponiendo que se utiliza un esquema multiversión para leer la tabla padre y efectivamente puede verse el registro padre para la fila hijo que se agregará, no puede agregarse en forma segura la fila hijo dentro de la tabla hijo, porque puede haber ocurrido que entretanto otro usuario haya borrado el registro padre de la tabla padre, sin que se tenga conocimiento de ello. La solución es llevar a cabo el `SELECT` en un modo con bloqueo, utilizando `LOCK IN SHARE MODE`:

```
SELECT * FROM parent WHERE NAME = 'Jones' LOCK IN SHARE MODE;
```

Realizar una lectura en modo compartido (`share mode`) significa que se leen los últimos datos disponibles, y se establece un bloqueo en modo compartido en los registros que se leen.

Un bloqueo en modo compartido evita que otros actualicen o eliminen la fila que se ha leído. Además, si los datos más actualizados pertenecen a una transacción todavía no confirmada de otra conexión, se espera hasta que la transacción se confirme. Luego de ver que la mencionada consulta devuelve el registro padre 'Jones', se puede agregar con seguridad el registro hijo en la tabla hijo y confirmar la transacción.

Detección de interbloqueos y cancelación de transacciones

`InnoDB` detecta automáticamente un deadlock de transacciones y cancela una o más transacciones para evitarlo. `InnoDB` intenta escoger para cancelar: transacciones pequeñas, el tamaño de la transacción es determinado por el número de filas insertadas, actualizadas, o eliminadas. `InnoDB` se mantiene al tanto de los bloqueos de tablas siempre y cuando esté configurado para este trabajo, y la capa `MySQL` por debajo sepa acerca de bloqueos a nivel

de fila. En otro caso, InnoDB no puede detectar deadlocks cuando está involucrado un bloqueo de tabla establecido por una sentencia `LOCK TABLES` o por otro motor de almacenamiento que no sea InnoDB. Cuando InnoDB lleva a cabo una cancelación completa de una transacción, todos los bloqueos de la transacción son liberados. Sin embargo, si solamente se cancela como resultado de un error una sentencia SQL individual, algunos de los bloqueos impuestos por la sentencia SQL podrían mantenerse. Esto se debe a que InnoDB guarda los bloqueos de fila en un formato en el que no puede luego saber qué sentencia SQL originó cada bloqueo.

3.3.9 Recuperación

MYSQL recupera una caída del servidor, a través del motor InnoDB, el cual verifica automáticamente los registros (logs) y ejecuta una recuperación de la base de datos del tipo roll-forward, es decir, hasta el momento anterior a la falla. InnoDB revierte automáticamente las transacciones no grabadas que existían al momento de la caída. Si la base de datos se corrompe o falla el disco, habrá que efectuar la recuperación desde una copia de respaldo. En el caso de corrupción, primero habría que encontrar una copia de respaldo realizada antes de la corrupción. Luego de restaurar la copia de respaldo base, debe realizarse la recuperación a partir de los ficheros de registro binario.

En algunos casos de corrupción de base de datos es suficiente con volcar, eliminar, y volver a crear una o unas pocas tablas corruptas. Se puede emplear la sentencia SQL `CHECK TABLE` para verificar si una tabla está corrupta, aunque `CHECK TABLE`, naturalmente, no puede detectar cada posible clase de corrupción. Se puede emplear `innodb_tablespace_monitor` para verificar la integridad de la gestión de espacio de ficheros dentro de los ficheros de espacio de tablas.

En algunos casos, una aparente corrupción de página de base de datos se debe en realidad a que el sistema operativo está corrompiendo su propio cache de ficheros, y los datos en el disco podrían estar en buenas condiciones. Lo mejor es, antes que nada, intentar el reinicio del ordenador. Ello puede eliminar errores que dan la sensación de tener corrupción en la página de base de datos.

3.4 SQL Server 2005

SQL Server 2005 es un sistema gestor de datos que analiza y administra datos haciendo posible avances en tres ámbitos clave: gestión de información empresarial, productividad de los desarrolladores e inteligencia empresarial. Microsoft SQL Server 2005 es una plataforma de base de datos para transacciones en línea (OLTP) a gran escala, bodegas de datos (datawarehousing), y aplicaciones de comercio electrónico; a su vez es un plataforma de inteligencia de negocios con integración de datos, análisis, y soluciones de reporte.

3.4.1 Arquitectura del Sistema

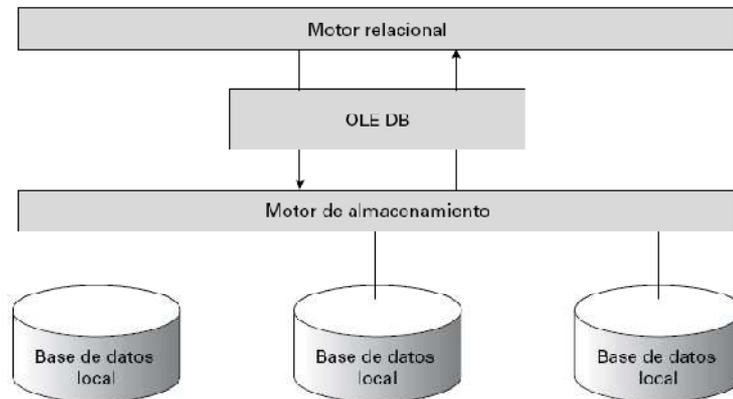


Figura 3.6. Arquitectura de SQL Server
Fuente: Fundamentos de Bases de Datos

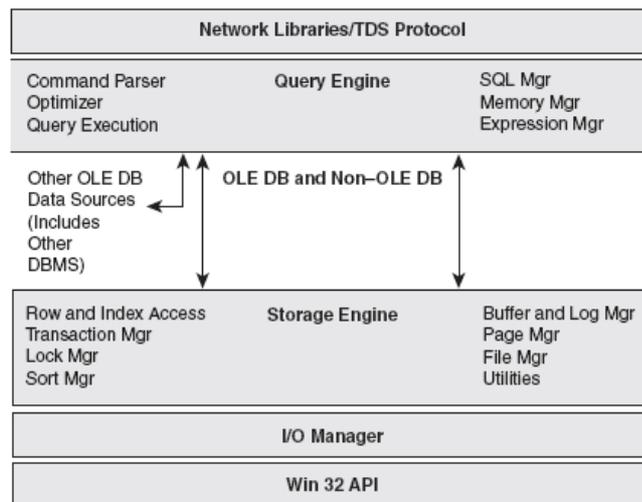


Figura 3.7. Componentes del Motor Relacional y de Almacenamiento
Fuente: Fundamentos de Bases de Datos

El servidor de la base de datos de SQL Server tiene dos partes principales: El motor relacional y el motor de almacenamiento.

- El motor relacional (MR): compila las instrucciones en un plan de ejecución optimizado.
- El motor de almacenamiento (MA): transfiere los datos desde las memorias intermedias de los datos hasta los operadores de ejecución de la consulta, la cual se ejecuta en el motor relacional

La arquitectura de SQL Server claramente separa los componentes del motor relacional y de almacenamiento con el servidor y estos componentes utilizan interfaces OLE-DB para comunicarse entre sí. Esta comunicación entre el motor relacional y el motor de almacenamiento mediante la interfaz OLE-DB permite al motor relacional procesar las consultas con cualquier origen de datos que exponga tales interfaces. Dichos orígenes de datos pueden ser otros sistemas SQL Server (consultas distribuidas) y otros proveedores de datos OLE-DB relacionales y no relacionales (consultas heterogéneas).

3.4.2 Catálogo del Sistema

El catálogo del sistema proporciona información para las bases de datos de SQL Server. El núcleo de los catálogos del sistema de SQL Server es un conjunto de vistas que muestran metadatos que describen los objetos de una instancia de SQL Server. Los metadatos son datos que describen los atributos de los objetos de un sistema. Las aplicaciones basadas en SQL Server pueden tener acceso a la información de los catálogos del sistema utilizando lo siguiente:

- **Vistas del Catálogo:** proporcionan acceso a los metadatos almacenados en cada base de datos del servidor.
- **Vistas del Esquema de Información:** proporcionan una vista interna e independiente de las tablas del sistema de los metadatos de SQL Server. Las vistas de esquema de información permiten que las aplicaciones funcionen correctamente aunque se hayan realizado cambios significativos en las tablas del sistema

subyacentes. Cada vista de esquema de información contiene metadatos para todos los objetos de datos almacenados en esa base de datos en concreto.

- **Vistas de Compatibilidad:** Muchas de las tablas del sistema de versiones anteriores de SQL Server se implementan ahora como un conjunto de vistas. Se conocen como vistas de compatibilidad y sólo se proporcionan por compatibilidad con versiones anteriores. Muestran los mismos metadatos que están disponibles en SQL Server 2000. No obstante, no muestran ninguno de los metadatos relacionados con las características incluidas en SQL Server 2005 y versiones posteriores.
- **Funciones del Catálogo ODBC:** El controlador ODBC de SQL Server Native Client permite que una aplicación determine la estructura de base de datos a través de llamadas a funciones de catálogo de ODBC. Las funciones de catálogo devuelven información en conjuntos de resultados y se implementan utilizando procedimientos almacenados de catálogo para consultar las tablas del sistema en el catálogo.

3.4.3 Almacenamiento de Datos

La unidad fundamental del almacenamiento de datos en SQL Server es la página. El espacio en disco asignado a un archivo de datos (.mdf o .ndf) de una base de datos se divide lógicamente en páginas numeradas de forma contigua de 0 a n . Las operaciones de E/S de disco se realizan en el nivel de página. Es decir, SQL Server lee o escribe páginas de datos enteras. Las extensiones son una colección de ocho páginas físicamente contiguas; se utilizan para administrar las páginas de forma eficaz. Todas las páginas se almacenan en extensiones.

Las filas de datos se colocan en las páginas una a continuación de otra, empezando inmediatamente después del encabezado. Al final de la página, comienza una tabla de desplazamiento de fila y cada una de esas tablas contiene una entrada para cada fila de la página. Cada entrada registra la distancia del primer byte de la fila desde el inicio de la página. Las entradas de la tabla de desplazamiento de fila están en orden inverso a la secuencia de las filas de la página.

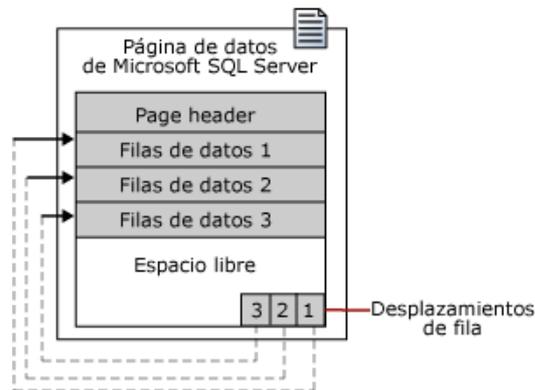


Figura 3.8. Página de datos de SQL Server

3.4.4 Indexación de Datos

La forma más común de indexación son los índices no agrupados, que también se conocen como índices secundarios, estos índices son construidos sobre la estructura de árboles B+ sobre una o más columnas de la tabla base. SQL Server permite al usuario construir índices secundarios sobre columnas calculadas que son columnas cuyo valor es una expresión, normalmente basada en el valor de otras columnas en esa fila.

3.4.5 Particiones

En SQL Server 2005, las páginas de tablas e índices están incluidas en una o varias particiones. De forma predeterminada, una tabla o un índice sólo incluyen una partición que contiene todas las páginas de tablas o índices, cuando una tabla o un índice utilizan varias particiones, se crean particiones horizontales de los datos para que se asignen los grupos de filas a particiones individuales, en función de la columna especificada. Las particiones se pueden colocar en uno o varios grupos de archivos de la base de datos y la tabla o el índice se tratarán como una sola entidad lógica cuando se realicen consultas o actualizaciones en los datos. SQL Server soporta división en particiones por rango.

3.4.6 Variaciones y Extensiones SQL

Transact-SQL es el lenguaje de bases de datos soportado por SQL Server. Transact-SQL es un lenguaje completo de programación de bases de datos que incluye instrucciones de

definición y manipulación de datos, instrucciones iterativas y condicionales, variables, procedimientos y funciones. Transact-SQL cumple el nivel de entrada de la norma SQL-92 pero también soporta varias características desde los niveles intermedios y superiores. Transact-SQL también soporta extensiones a la norma SQL-92.

Funciones definidas por el usuario

SQL Server soporta las funciones que devuelven un tipo escalar o una tabla. Las funciones escalares se pueden utilizar en cualquier expresión escalar dentro de una instrucción `LMD` o `LDD` de `SQL`. Las funciones que devuelven tablas se pueden utilizar en cualquier lugar donde se permita una tabla en una instrucción `SELECT`. Las funciones que devuelven una tabla cuyo cuerpo contiene una única instrucción `SQL SELECT` se tratan como una vista (entre líneas expandida) en la consulta que hace referencia a la función. Puesto que las funciones que devuelven una tabla permiten la introducción de argumentos, las funciones entre líneas que devuelven tablas se pueden considerar vistas parametrizadas.

Vistas indexadas

Además de las vistas tradicionales como se definieron en la norma ANSI de SQL, SQL Server soporta las vistas indexadas (materializadas). Las vistas indexadas pueden mejorar sustancialmente el rendimiento de las consultas complejas de ayuda a la toma de decisiones que recuperan un gran número de filas y agregan grandes cantidades de información en sumas recuentos y medias. SQL Server soporta la creación de índices agrupados en una vista y subsecuentemente cualquier número de índices no agrupados. Una vez que se indexa una vista, el optimizador puede utilizar sus índices en consultas que hacen referencia a la vista o sus tablas base. Las consultas existentes se pueden beneficiar de la eficiencia mejorada de la recuperación de los datos directamente de la vista indexada sin tener que ser reescrita para hacer referencia a la vista. Las instrucciones de actualización de las tablas base se propagan automáticamente a los índices de la vista.

Vistas divididas

Las vistas divididas se utilizan para dividir los datos entre varias tablas, bases de datos o ejemplares de SQL Server con el fin de distribuir la carga de trabajo. Si los datos se dividen

entre varios servidores, se pueden proporcionar los mismos beneficios en el rendimiento que un agrupamiento de servidores y se puede utilizar para soportar las necesidades de procesamiento de los mayores sitios Web o centros de datos corporativos. Una tabla base se puede dividir en varias tablas miembro, cada una de las cuales tiene un subconjunto de filas de la tabla original. Cada servidor debe tener definición de la tabla dividida.

Vistas actualizables

Generalmente las vistas puede ser el objetivo de las instrucciones update, delete o insert si la modificación de los datos se aplica a solamente una de las tablas base de la vista. Las actualizaciones de las vistas divididas se pueden propagar a varias tablas base. Por ejemplo, la siguiente instrucción update incrementará los precios para el editor «0736» en un 10 por ciento.

```
update vistatítulos set precio = precio * 1.10 where id_editorial = '0736'
```

Para las modificaciones de los datos que afectan a más de una tabla base, la vista se puede actualizar si hay un desencadenador `INSTEAD` definido para la operación. Los desencadenadores insert para las operaciones insert, update o delete se pueden definir en una vista para especificar las actualizaciones que se deben ejecutar en las tablas base para implementar las modificaciones correspondientes en la vista.

Desencadenadores (disparadores)

Los desencadenadores son procedimientos Transact-SQL que se ejecutan automáticamente cuando se envía una instrucción update, insert o delete a una tabla base o vista. Hay dos clases generales de desencadenadores que difieren en el tiempo con respecto a la instrucción de desencadenamiento, bajo la que se realiza la acción.

Los desencadenadores `AFTER` se ejecutan después de la instrucción de desencadenamiento y se aplican posteriores restricciones declarativas. Los desencadenadores `INSTEAD` se ejecutan en lugar de la acción de desencadenamiento. Los desencadenadores `INSTEAD` son similares a los desencadenadores `BEFORE`, pero realmente reemplazan la acción de desencadenamiento. En SQL Server los desencadenadores `AFTER` se pueden definir solamente sobre tablas base mientras que los desencadenadores `INSTEAD` se pueden definir sobre tablas base o vistas.

Los desencadenadores INSTEAD permiten que se pueda actualizar prácticamente cualquier vista.

3.4.7 Procesamiento y Optimización de Consultas²⁶

En el siguiente diagrama se ilustran las entradas y salidas del optimizador de consultas durante la optimización de una única instrucción SELECT:



Figura 3.9. Plan de Consulta

El optimizador de consultas de SQL Server es un optimizador basado en el costo. Cada plan de ejecución posible tiene asociado un costo en términos de la cantidad de recursos del equipo que se utilizan. El optimizador de consultas debe analizar los planes posibles y elegir el de menor costo estimado. La estimación se basa en las estadísticas de distribución de datos que están disponibles para el optimizador cuando evalúa cada tabla implicada en la consulta.

Frecuentemente se asume que el costo estimado de ejecución es un buen indicador de cuánto tiempo tardará la consulta en ejecutarse y que esta estimación les permite distinguir los planes buenos de los que no lo son. Puesto que se trata de una estimación, podría ser equivocado, en ocasiones los planes con costos estimados más altos pueden ser mucho más efectivos en términos de CPU, E/S y del tiempo de ejecución, a pesar de que sus estimaciones sean más altas.

Algunas instrucciones SELECT complejas tienen miles de planes de ejecución posibles, por lo que el optimizador de consultas no evalúa todas las combinaciones posibles, sino que utiliza algoritmos complejos para encontrar un plan de ejecución "lo suficientemente bueno", que tenga un costo razonablemente cercano al mínimo posible. Esto es porque el encontrar un

²⁶ **Microsoft TechNet.** Optimización el Rendimiento de Consultas en SQL Server. [ON-LINE]. SQL Server. [Consultado en: Noviembre 2008]. Disponible en: <http://technet.microsoft.com/es-es/magazine/2007.11.sqlquery.aspx>

plan perfecto no siempre es posible; e incluso cuando es posible, el costo de la evaluación de todas las posibilidades para encontrar que el plan perfecto podría fácilmente tener más peso que cualquier ganancia de rendimiento.

El optimizador de consultas de SQL Server no elige sólo el plan de ejecución con el costo de recursos mínimo, sino que elige el plan que devuelve resultados al usuario con un *costo razonable de recursos* y devuelve *resultados lo más rápido posible*.

Por ejemplo, el procesamiento de una consulta en paralelo suele utilizar más recursos que el procesamiento en serie, pero completa la consulta más rápidamente. El optimizador de SQL Server utilizará un plan de ejecución en paralelo para devolver resultados si esto no afecta negativamente a la carga del servidor.

El optimizador de consultas es importante porque permite que el servidor de la base de datos se ajuste dinámicamente a las condiciones cambiantes de la base de datos, sin necesitar la entrada de un programador o de un administrador de base de datos. Esto permite a los programadores centrarse en la descripción del resultado final de la consulta, pues se tiene la seguridad de que el optimizador de consultas creará un plan de ejecución eficaz para el estado de la base de datos cada vez que se ejecuta la instrucción.

Almacenar en caché y volver a utilizar un plan de ejecución

SQL Server 2005 tiene un búfer de memoria que se utiliza para almacenar planes de ejecución y búferes de datos. El porcentaje del búfer de memoria que se asigna a los planes de ejecución o a los búferes de datos varía dinámicamente según el estado del sistema. La parte del búfer de memoria que se utiliza para almacenar los planes de ejecución se denomina caché de procedimientos. Cuando se ejecuta una instrucción en SQL Server 2005, el motor relacional mira primero en la caché de procedimientos para ver si existe un plan de ejecución para la misma instrucción SQL.

SQL Server 2005 vuelve a utilizar cualquier plan existente que encuentre, lo que evita el trabajo que supone volver a compilar la instrucción SQL. Si no existe ningún plan de ejecución, SQL Server 2005 genera uno nuevo para la consulta. SQL Server 2005 tiene un algoritmo eficiente para encontrar cualquier plan de ejecución existente de una instrucción

SQL dada. En la mayor parte de los sistemas, los recursos mínimos que utiliza este recorrido son menos que los recursos que se ahorran al poder utilizar de nuevo los planes existentes en lugar de compilar cada instrucción SQL. Los algoritmos que hacen coincidir las instrucciones SQL nuevas con los planes de ejecución existentes no utilizados de la caché requieren que todas las referencias a objetos estén completas. Por ejemplo, la primera de estas instrucciones `SELECT` no coincide con un plan existente, pero la segunda sí:

Planes paralelos

Un plan paralelo es un tipo especial de plan de ejecución. Si se ejecuta una consulta en un servidor con más de una CPU y la consulta puede optar por la paralelización, puede que se elija un plan paralelo, generalmente, el optimizador de consultas sólo considerará un plan paralelo para una consulta que tenga un costo que supere cierto umbral configurable.

SQL Server 2005 proporciona consultas en paralelo para optimizar la ejecución de consultas y las operaciones de índice en los equipos que tienen más de un microprocesador (CPU). Debido a que SQL Server puede realizar una operación de índice o consulta en paralelo mediante varios subprocesos del sistema operativo, la operación se puede completar de forma rápida y eficaz, sin embargo, por la sobrecarga generada al administrar múltiples subprocesos paralelos de ejecución, los planes paralelos son más costosos de ejecutar y eso se refleja en su costo estimado.

Gracias al uso de la eficacia de procesamiento de varias CPU, los planes paralelos tienden a generar los resultados más rápido que los planes estándar. Dependiendo de su situación específica, incluidas variables como los recursos disponibles y la carga simultánea de otras consultas, esta situación puede ser deseable para la instalación.

El número real de subprocesos que utiliza una consulta en paralelo se determina en la inicialización de la ejecución del plan de consulta y viene determinado por la complejidad del plan y el grado de paralelismo. El grado de paralelismo determina el número máximo de CPU que se están utilizando; no significa el número de subprocesos que se están utilizando.

El optimizador de consultas de SQL Server no utiliza un plan de ejecución en paralelo para una consulta si se cumple alguna de las siguientes condiciones:

- El costo de la ejecución en serie de la consulta no es lo suficientemente alto como para considerar un plan de ejecución en paralelo alternativo.
- El plan de ejecución en serie se considera más rápido que cualquier otro plan de ejecución en paralelo de una consulta determinada.
- La consulta contiene operadores escalares o relacionales que no se pueden ejecutar en paralelo. Es posible que algunos operadores hagan que una sección del plan de consultas se ejecute en modo de serie, o que todo el plan se ejecute en modo de serie.

3.4.8 Transacciones y Control de concurrencia

Controles de Concurrencia

En las versiones de SQL Server anteriores a la 2005 se usaba el control pesimista y basado en bloqueos. En la versión 2005 aparece el versionado de filas. El esquema de control de concurrencia pesimista asume que hay suficientes modificaciones de la información en el sistema como para que sea probable que dada una operación de lectura, ésta se verá afectada por una operación de escritura ejecutada concurrentemente, es decir el sistema es pesimista sobre la posibilidad de la aparición de conflictos.

El comportamiento por defecto cuando se usa este sistema es usar bloqueos. El control optimista de la concurrencia asume que hay pocas operaciones de modificación de los datos haciendo improbable que un proceso modifique información mientras otro esté leyendo; el comportamiento por defecto en este caso sería usar versionado de filas para ver el estado de la información antes de que se produjera la modificación.

El proceso sigue de la siguiente manera: cuando se actualiza un registro de una tabla, al nuevo registro se le asigna el número de la transacción que realiza la actualización, la versión anterior del registro se almacena en una base de datos temporal, y el nuevo registro contiene un puntero a su versión anterior, que a su vez puede enlazar a su propia versión anterior, proporcionando así una lista enlazada. Las versiones de registros sólo deben almacenarse mientras haya operaciones que puedan requerirlos.

Niveles de Aislamiento de Transacciones

Para controlar como afectan los bloqueos a las transacciones, podemos modificar el nivel de aislamiento de las transacciones a través de la instrucción **SET TRANSACTION ISOLATION LEVEL**. Con esta instrucción controlamos como interpreta la transacción los bloqueos existentes y como genera nuevos bloqueos con sus operaciones de lectura/escritura.

Su sintaxis general es la siguiente: **SET TRANSACTION ISOLATION LEVEL** <opción>.

Donde <opción> puede tomar estos valores:

- **READ COMMITTED:** La transacción no puede leer datos modificados por otras transacciones. Permite que otras transacciones puedan modificar los datos que se han leído. Esta opción es la predeterminada para SQL Server 2005.
- **READ UNCOMMITTED:** La transacción es capaz de leer los datos modificados por otras transacciones pero que aún no han sido confirmadas (pendientes de COMMIT).
- **REPEATABLE READ:** La transacción no puede leer datos modificados por otras transacciones y otras transacciones no pueden modificar los datos que se han leído.
- **SERIALIZABLE:** Las instrucciones no pueden leer datos que hayan sido modificados, pero aún no confirmados, por otras transacciones y ninguna otra transacción puede modificar los datos leídos por la transacción actual ni insertar filas nuevas con valores de clave que pudieran estar incluidos en el intervalo de claves hasta que la transacción actual finalice.

En este escenario, cualquiera de las opciones posibles tiene sus inconvenientes, por lo que la gestión de los bloqueos siempre ha sido un tema difícil de solventar en versiones anteriores de SQL Server. En SQL Server 2005 se ha incluido un nuevo nivel de aislamiento, el nivel **SNAPSHOT** que asegura la coherencia de los datos para toda la transacción:

- **SNAPSHOT:** Activa el versionado de fila. Las instrucciones que se ejecuten en la transacción no verán las modificaciones de datos efectuadas por otras transacciones, en su lugar reciben una "copia coherente" de cómo estaban los datos al comienzo de la transacción. De este modo actúan otros gestores de bases de datos muy populares, como por ejemplo, ORACLE.

El nivel de aislamiento **SNAPSHOT** no está disponible de forma predeterminada. La opción Snapshot usa el versionado de filas para proporcionar consistencia de lectura a nivel de transacciones. Las operaciones de lectura no piden bloqueos. Cuando leen filas modificadas por otra transacción, recuperan la versión de la fila que existía antes de que la transacción comenzara. Se activa con la opción `ALLOW_SNAPSHOT_ISOLATION`, que por defecto está desactivada.

3.4.9 Recuperación

SQL Server maneja diferentes modelos de recuperación que controlan el mantenimiento del registro de transacciones.

Tabla 3.VIII. Modelos de Recuperación

Modelo de recuperación	Descripción	Riesgo de pérdida de trabajo	¿Recuperación hasta un momento dado?
Simple	Sin copias de seguridad de registros. Recupera automáticamente el espacio de registro para mantener al mínimo los requisitos de espacio, eliminando, en esencia, la necesidad de administrar el espacio del registro de transacciones.	Los cambios realizados después de la copia de seguridad más reciente no están protegidos. En caso de desastre, es necesario volver a realizar dichos cambios.	Sólo se puede recuperar hasta el final de una copia de seguridad.
Completa	Requiere copias de seguridad de registros. No se pierde trabajo si un archivo de datos se pierde o resulta dañado. Se puede recuperar hasta cualquier momento, por ejemplo, antes del error de aplicación o usuario.	Normalmente ninguno. Si el final del registro resulta dañado, se deben repetir los cambios realizados desde la última copia de seguridad de registros.	Se puede recuperar hasta determinado momento, siempre que las copias de seguridad se hayan completado hasta ese momento.
Por medio de registros de operaciones masivas	Requiere copias de seguridad de registros. Complemento del modelo de recuperación completa que permite operaciones de copia masiva de alto rendimiento. Reduce el uso del espacio de registro mediante el registro masivo de la mayoría de las operaciones masivas.	Si el registro resulta dañado o se han realizado operaciones masivas desde la última copia de seguridad de registros, se pueden repetir los cambios desde esa última copia de seguridad. En caso contrario, no se pierde el trabajo.	Se puede recuperar hasta el final de cualquier copia de seguridad. No admite recuperaciones a un momento dado.

Fuente: www.microsoft.com

3.5 Características Primarias o Funcionalidad de los SGBDs

Se realiza un extracto del soporte de las características primarias de los Sistemas de Gestión de Bases de Datos, basado en la recopilación de información general en la WEB. Se procede a categorizar las características primarias de los SGBDs, cada categoría tiene un número de ítems que será valorado de la siguiente manera. 0 pto: para el SGBD que mantenga soporte completo para la característica, y 1 pto: para el SGBD que no cumpla con la característica. A cada SGBD se le asignará un peso sobre 10, de acuerdo al número de ítems con los que cumpla en cada categoría

Experiencia: A medida que una empresa acumula experiencia en la producción de un producto o servicio, más aprende a producir de forma eficiente. Por tal razón se ha considerado la experiencia de la empresa como un medidor de la eficiencia de los SGBDs.

Tabla 3.IX. Experiencia

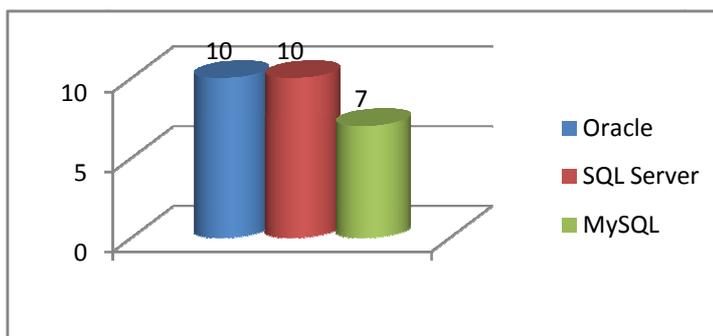
Experiencia			
	Oracle	SQL Server	MySQL
Creador	Oracle	Microsoft	MySQL AB
Fecha Primera Versión	1977	1989	Noviembre 1996
Experiencia	20 Años	20 Años	14 Años

Realizado Por: Pilar A. Quizhpe P.

Tabla 3.X. Frecuencias en la Experiencia

Experiencia		
	Frecuencia	Puntaje
Oracle	20	10
SQL Server	20	10
MySQL	14	7

Realizado Por: Pilar A. Quizhpe P.



Grafica 3.1. Puntuación Experiencia

Realizado Por: Pilar A. Quizhpe P.

En este caso tanto Oracle como SQL Server reciben el más alto puntaje, al tener la mayor experiencia en el mercado de desarrollo de los Sistemas de Gestión de Bases de Datos, con 20 años de experiencia.

Licenciamiento

Tabla 3.XI. Interfaces y conectores de programación

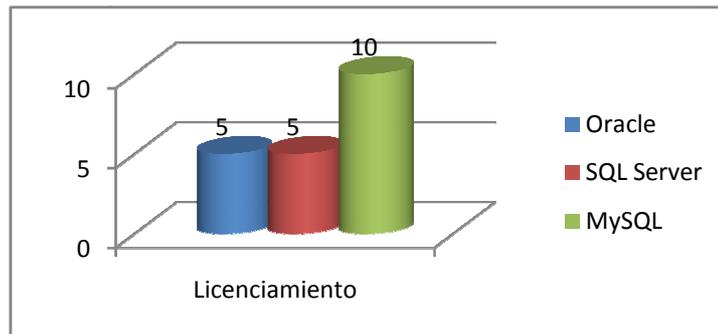
Interfaces y Conectores de Programación			
	Oracle	SQL Server	MySQL
Licenciamiento Comercial	Soporte Completo	Soporte Completo	Soporte Completo
Licenciamiento Libre			Soporte Completo

Realizado Por: Pilar A. Quizhpe P.

Tabla 3.XII. Frecuencias Licenciamiento

Licenciamiento		
	Frecuencia	Puntaje
Oracle	1	5
SQL Server	1	5
MySQL	2	10

Realizado Por: Pilar A. Quizhpe P.



Grafica 3.2. Puntuación Licenciamiento

Realizado Por: Pilar A. Quizhpe P.

MySQL obtiene la mejor puntuación al tener gran ventaja, pues provee licenciamiento a medida, es decir puede distribuirse bajo los términos de la Licencia Publica General GNU, para aplicaciones no comerciales, y también bajo licenciamiento comercial, disponible para aquellos que quieran distribuir aplicaciones no GPL.

Interfaces y conectores de programación: son el conjunto de funciones y procedimientos (o métodos si se refiere a programación orientada a objetos) que ofrece cierta biblioteca para ser utilizado por otro software como una capa de abstracción.

Tabla 3.XIII. Interfaces y conectores de programación

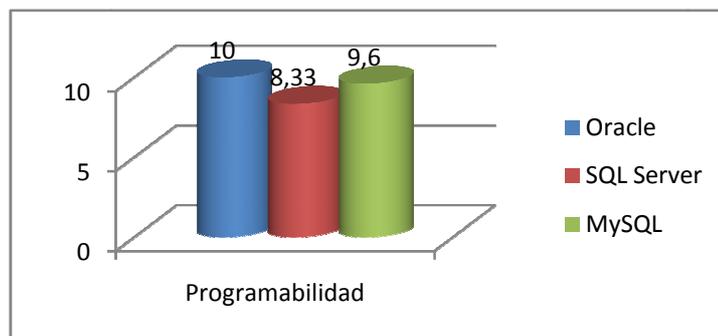
Interfaces y Conectores de Programación			
	Oracle	SQL Server	MySQL
C	Soporte Completo	Soporte Completo	Soporte Completo
C++	Soporte Completo	Soporte Completo	Soporte Completo
TCL	Soporte Completo		
Delphi	Soporte Completo	Soporte Completo	Soporte Completo
Perl	Soporte Completo	Soporte Completo	Soporte Completo
Python	Soporte Completo	Soporte Completo	Soporte Completo
PHP	Soporte Completo	Soporte Completo	Soporte Completo
Java	Soporte Completo	Soporte Completo	Soporte Completo
Ruby	Soporte Completo		Soporte Completo
.NET	Soporte Completo	Soporte Completo	Soporte Completo
ODBC	Soporte Completo	Soporte Completo	Soporte Completo
JDBC	Soporte Completo	Soporte Completo	Soporte Completo

Realizado Por: Pilar A. Quizhpe P.

Tabla 3.XIV. Frecuencias de Interfaces y Conectores de programación

Programabilidad		
	Frecuencia	Puntaje
Oracle	12	10
SQL Server	10	8.33
MySQL	11	9.16

Realizado Por: Pilar A. Quizhpe P.



Grafica 3.3. Puntuación Interfaces y Conectores de Programación

Realizado Por: Pilar A. Quizhpe P.

Oracle obtiene el mayor puntaje puesto que provee mayor capacidad de Programabilidad, al soportar todas las interfaces y conectores de programación mencionados.

Soporte de Sistemas operativos: Un buen parámetro de medición es la portabilidad del SGBD, mientras más sistemas operativos soporte mayor es la portabilidad, de la BD.

Tabla 3.XV. Sistemas Operativos Soportados

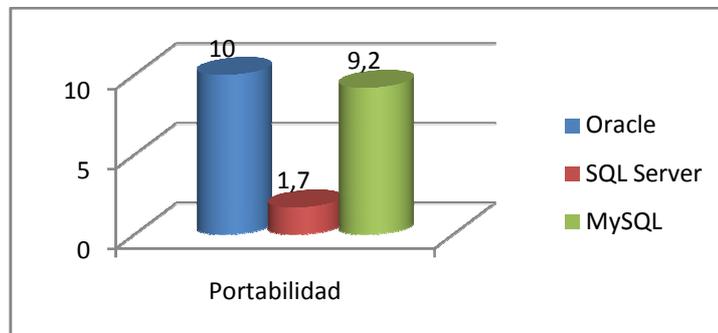
Sistemas Operativos Soportados			
	Oracle	SQL Server	MySQL
Windows	Soporte Completo	Soporte Completo	Soporte Completo
Linux	Soporte Completo		Soporte Completo
Mac OS	Soporte Completo		Soporte Completo
FreeBSD	Soporte Completo		
IBM AIX	Soporte Completo		Soporte Completo
Solaris	Soporte Completo		Soporte Completo
HP - UX	Soporte Completo		Soporte Completo
QNX	Soporte Completo		Soporte Completo
SCO Unix	Soporte Completo		Soporte Completo
Novell Netware	Soporte Completo		Soporte Completo
SGI IRIX	Soporte Completo		Soporte Completo
Soporte 64 bits	Soporte Completo	Soporte Completo	Soporte Completo

Realizado Por: Pilar A. Quizhpe P.

Tabla 3.XVI. Frecuencias para Sistemas Operativos Soportados

Portabilidad		
	Frecuencia	Puntaje
Oracle	12	10
SQL Server	2	1.7
MySQL	11	9.2

Realizado Por: Pilar A. Quizhpe P.



Grafica 3.4. Puntuación Sistemas Operativos Soportados

Realizado Por: Pilar A. Quizhpe P.

Tanto Oracle como MySQL obtienen los mejores puntajes en cuanto a soporte de sistemas operativos, lo que los definiría como SGBDs con gran capacidad de portabilidad. SQL Server, es todo lo contrario al solamente poder ejecutarse en su propia plataforma Windows.

Objetos Soportados: En general cada SGBD, maneja objetos de las bases de datos, en común, pero no todos los SGBD proveen soporte para los objetos mencionados en la tabla, lo que los pone en desventaja en relación a los SGBDs que si los soportan. Existen una gran cantidad de objetos soportados en un SGBD, pero a continuación se mencionan los más importantes.

Tabla 3.XVII. Objetos Soportados

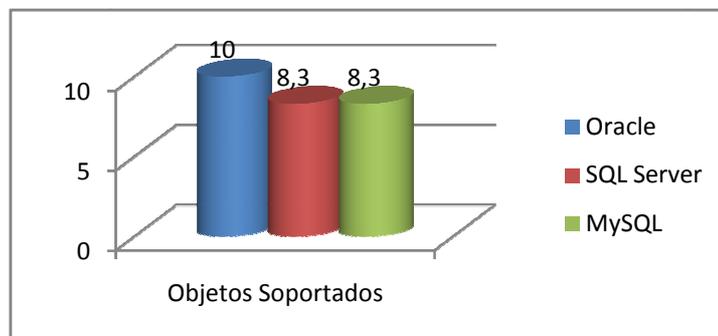
Objetos Soportados			
	Oracle	SQL Server	MySQL
Dominio	Soporte Completo		
Cursor	Soporte Completo	Soporte Completo	Soporte Completo
Trigger	Soporte Completo	Soporte Completo	Soporte Completo
Función	Soporte Completo	Soporte Completo	Soporte Completo
Procedimiento	Soporte Completo	Soporte Completo	Soporte Completo
Rutina Externa	Soporte Completo	Soporte Completo	Soporte Completo

Realizado Por: Pilar A. Quizhpe P.

Tabla 3.XVIII. Frecuencias de Objetos Soportados

Soporte Completo		
	Frecuencia	Puntaje
Oracle	6	10
SQL Server	5	8.3
MySQL	5	8.3

Realizado Por: Pilar A. Quizhpe P.



Grafica 3.5. Puntuación Objetos Soportados

Realizado Por: Pilar A. Quizhpe P.

Oracle obtiene el mejor puntaje al proveer soporte para todos los objetos mencionados, sin embargo la diferencia es mínima en relación a SQL Server y MySQL que tan solo no proveen soporte para objetos de tipo dominio.

Características Fundamentales: Son las características que definen un SGBD, por lo que el carecimiento de cualquiera de ellas, básicamente inhibiría el verdadero objetivo de estos sistemas.

Tabla 3.XIX. Características Fundamentales

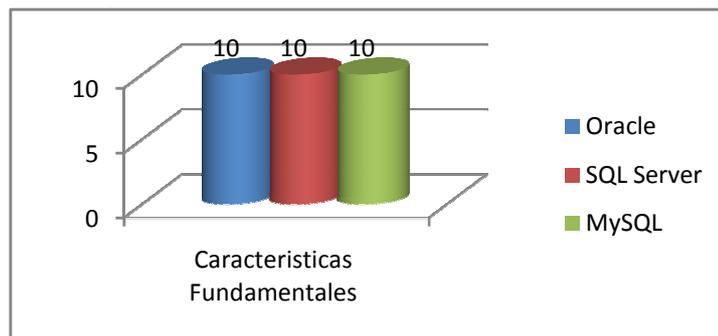
Características Fundamentales			
	Oracle	SQL Server	MySQL
ACID	Soporte Completo	Soporte Completo	Soporte Completo
Integridad Referencial	Soporte Completo	Soporte Completo	Soporte Completo
Transacciones	Soporte Completo	Soporte Completo	Soporte Completo
Unicode	Soporte Completo	Soporte Completo	Soporte Completo

Realizado Por: Pilar A. Quizhpe P.

Tabla 3.XX. Frecuencias en Características Fundamentales

Soporte Completo		
	Frecuencia	Puntaje
Oracle	4	10
SQL Server	4	10
MySQL	4	10

Realizado Por: Pilar A. Quizhpe P.



Grafica 3.6. Puntuación Características Fundamentales

Realizado Por: Pilar A. Quizhpe P.

Oracle, SQL Server y MySQL obtienen el mismo puntaje al cumplir con las características básicas que un Sistema de Gestión de Bases de Datos en General debe soportar.

Tipos de índices: Los tipos de índices son importantes, porque permiten agilizar el proceso de búsqueda de los, datos, al haber más tipos de índices se pueden aprovechar las características inherentes a cada uno de estos.

Tabla 3.XXI. Tipos de Índices

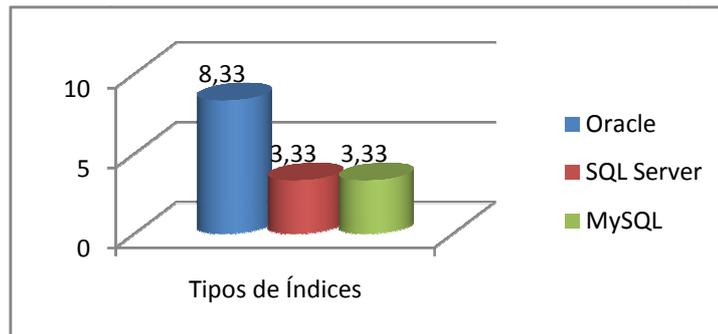
Índices			
	Oracle	SQL Server	MySQL
Árbol R-/R+	Soporte Completo	Soporte Completo	Soporte Completo
Hash	Soporte Completo		Soporte Completo
Expresión	Soporte Completo	Soporte Completo	
Parcial			
Reversa	Soporte Completo		
Mapa de Bits	Soporte Completo		

Realizado Por: Pilar A. Quizhpe P.

Tabla 3.XXII. Frecuencia en Tipos de Índices Soportados

Soporte Completo		
	Frecuencia	Puntaje
Oracle	5	8.33
SQL Server	2	3.33
MySQL	2	3.33

Realizado Por: Pilar A. Quizhpe P.



Grafica 3.7. Puntuación Tipos de Índices Soportados

Realizado Por: Pilar A. Quizhpe P.

Una vez más Oracle recibe el más alto puntaje al contar con una mayor cantidad de tipos de índices, dando más opciones del programador de encontrar mejores técnicas en el procesamiento y optimización de consultas.

Particionamiento: Esta característica es muy importante para SGBDS en ambientes distribuidos, haciendo posible particionar las tablas de acuerdo al requerimiento o necesidad del cliente.

Tabla 3.XXIII. Tablas y Vistas

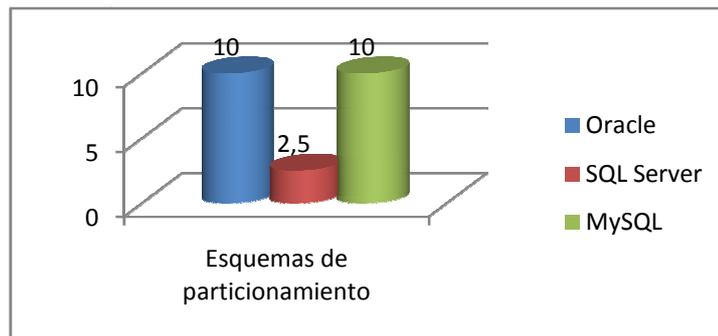
Particionamiento			
	Oracle	SQL Server	MySQL
Rango	Soporte Completo	Soporte Completo	Soporte Completo
Hash	Soporte Completo		Soporte Completo
Compuesto	Soporte Completo		Soporte Completo
Lista	Soporte Completo		Soporte Completo

Realizado Por: Pilar A. Quizhpe P.

Tabla 3.XXIV. Frecuencia de Esquemas de Particionamiento

Soporte Completo		
	Frecuencia	Puntaje
Oracle	4	10
SQL Server	1	2.5
MySQL	4	10

Realizado Por: Pilar A. Quizhpe P.



Grafica 3.8. Puntuación Esquemas de Particionamiento

Realizado Por: Pilar A. Quizhpe P.

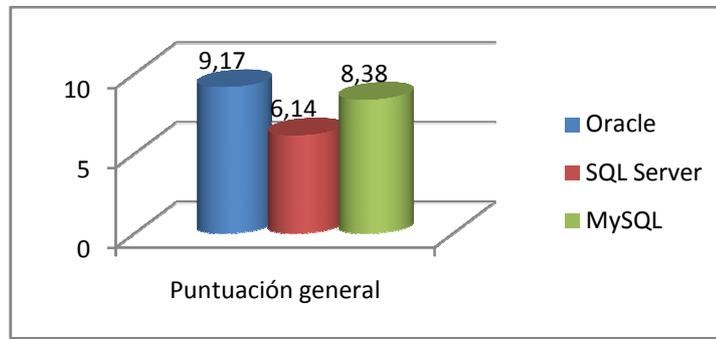
Oracle al igual que MySQL obtienen el más alto puntaje, porque proporcionan mayor cantidad de técnicas de particionamiento, mientras que SQL Server se queda relegado por no proveer mayor soporte de técnicas de particionamiento.

3.6 Comparativa de Características Primarias

Tabla 3.XXV. Cuadro Comparativo de Características Primarias

Comparativa General			
	Oracle	SQL Server	MySQL
Experiencia	10.00	10.00	7.00
Licenciamiento	5.00	5.00	10.00
Interfaces y conectores de programación	10.00	8.30	9.20
Sistemas Operativos Soportados	10.00	1.70	9.20
Objetos Soportados	10.00	8.30	8.30
Características Fundamentales	10.00	10.00	10.00
Tipo de Índices	8.33	3.33	3.33
Métodos de Particionamiento	10.00	2.50	10.00
Promedio General	9.17	6.14	8.38

Realizado Por: Pilar A. Quizhpe P.



Grafica 3.9. Puntuación General Parte I

Realizado Por: Pilar A. Quizhpe P.

Se identifica a Oracle como el mejor en cuanto a características al soportar toda la funcionalidad que se espera de un servidor "serio", permitiendo implementar diseños "activos", con triggers, procedimientos almacenados, e integridad referencial declarativa bastante potente, además de de diversas técnicas de partición para la mejora de la eficiencia de replicación e incluso la administración de bases de datos distribuidas.

Además, Oracle puede ejecutarse en todas las plataformas, desde una PC hasta un supercomputador, así como puede ejecutarse en multitud de sistemas operativos, incluso existe una versión personal para Windows 9x, lo cual es un punto a favor para los desarrolladores que se llevan trabajo a casa. MySQL actualmente es una sólida base de datos, con grandes capacidades para crecer, elevadas prestaciones, soporte transaccional y las ventajas de estar disponible bajo el modelo de fuente abierta, MySQL es una de las más populares para crear todo tipo de servicios Web. Con la versión 5 de MySQL se cuenta con un excelente conjunto de funcionalidad ajustándose al estándar SQL de forma más concreta.

También resulta interesante el soporte de particiones de las tablas, procedimientos y funciones almacenadas así de triggers, aunque esto último de forma limitada. En cuanto a SQL Server, podemos decir que es un sistema que ofrece características similares a sus contendientes, con una interfaz muy amigable e intuitiva, sin embargo su peor defecto es su falta de portabilidad.

3.7 Comparativa de Dimensionamiento

Tabla 3.XXVI. Dimensionamiento y Capacidad de los SGBD

Dimensionamiento y Capacidad de los SGBD						
Versión	Oracle	Pto	SQL Server	Pto	MySQL	Pto
Límites de la Base de Datos						
Tamaño de la BD:	Ilimitado	10	524.272	5	Ilimitado	10
Número de Tablas:	Ilimitado	10	Limitado por el número de Objetos en la BD	5	Ilimitado	10
Archivos por Base de Datos:	Usualmente 1022	5	32767	8	Ilimitado	10
Límites de la Tabla						
Número de Filas	Ilimitado	10	Ilimitado	10	Limitado por el máximo de la tabla	8
Número de Columnas	1000	5	4096	8	1000	5
Índices por Tabla	Ilimitado	10	16	5	64	5
Límites de los Índices						
Tamaño Máximo	40% del tamaño del bloque	8	900 bytes	5	1000 bytes	6
Numero de Campos	32	8	16	5	16	5
Puntuación Total		8.25		6.37		7.38

Realizado Por: Pilar A. Quizhpe P.

Se valora sus características, de la siguiente manera:

Ilimitado: 10, **Limitado:** 5 – 8 Dependiendo de la cantidad, **Depende de algún factor:** 9.

La característica de dimensionamiento, es muy importante, porque indica la capacidad que proporciona el sistema de gestión de bases de datos, al garantizar que es capaz de manejar el volumen de datos requerido y comprobando que es adecuado en cada uno de los puntos evaluados.

3.8 Comparativa de Costos de Licenciamiento

Tabla 3.XXVII. Precios Licencias Oracle

ORACLE				
	Named User Plus	Software Update License & Support	Processor License	Software Update License & Support
Oracle Database				
Standard Edition One	149	32.78	4,995	1,098.90
Standard Edition	300	66.00	15,000	3,300.00
Enterprise Edition	800	176.00	40,000	8,800.00
Personal Edition	400	88.00	-	-
Lite Mobile Server	-	-	20,000	4,400.00
Lite Client	50	11.00	-	-
Enterprise Edition Options				
Real Application Clusters	400	88.00	20,000	4,400.00
Active DataGuard	100	22.00	5,000	1,100.00
Partitioning	200	44.00	10,000	2,200.00
Real Application testing	200	44.00	10,000	2,200.00
Advanced Compression	200	44.00	10,000	2,200.00
Total Recall	100	22.00	5,000	1,100.00
Advanced Security	200	44.00	10,000	2,200.00
Label Security	200	44.00	10,000	2,200.00
Database Vault	400	88.00	20,000	4,400.00
OLAP	400	88.00	20,000	4,400.00
Data Mining	400	88.00	20,000	4,400.00
Warehouse Builder Enterprise ETL	200	44.00	10,000	2,200.00
Warehouse Builder Data Qualyte	300	66.00	15,000	3,300.00
Content Database Suite	1000	220.00	50,000	11,000.00
Records Database	100	220.00	50,000	11,000.00
Spatial	200	44.00	10,000	2,200.00
Database Enterprise Management				
Diagnostics Pack	60	13.20	3,000	660.00
Tuning Pack	60	13.20	3,000	660.00
Change Management Pack	60	13.20	3,000	660.00
Configuration Management Pack	60	13.20	3,000	660.00
Provisioning Pack for Database	60	13.20	3,000	660.00
Data Masking Pack	200	44.00	10,000	220.00
Datawarehousing Products				
Express Server	800	176.00	40,000	8,800.00
Express Analyzer	800	176.00	-	-
Express Objects	5,000	1,100.00	-	-

Fuente: Oracle Global Price List, 1 Enero 2008

Tabla 3.XXVIII. Precios Licencias SQL Server

SQL Server				
	Licencias de Usuario	Server License	Processor License	Software Update License & Support
SQL Server Database				
Standard Edition	10	2,799.00	5,999.00	
WorkGroup Edition	5	739.00	3,899.00	
Enterprise Edition	25	13,499.00	24,499.00	

Fuente: <http://www.infoworld.com/>

Tabla 3.XXIX. Precios Licencias MySQL Server

MySQL				
	MySQL Enterprise Basic	MySQL Enterprise Silver	MySQL Enterprise Gold	MySQL Enterprise Platinum
MySQL Server				
Precios	599.00	1,999.00	2,999.00	4,999.00
Software				
MySQL Enterprise Server	Profesional	Profesional	Advanced	Advanced
MySQL Enterprise Monitor				
Enterprise Dashboard		Incluido	Incluido	Incluido
Notifications and Alerts		Incluido	Incluido	Incluido
Custom Advisor		Incluido	Incluido	Incluido
Upgrade Advisor		Incluido	Incluido	Incluido
Administration Advisor		Incluido	Incluido	Incluido
Security Advisor		Incluido	Incluido	Incluido
Query Analyzer			Incluido	Incluido
Replication Monitor			Incluido	Incluido
Replication Advisor			Incluido	Incluido
Memory Usage Advisor			Incluido	Incluido
Schema Advisor				Incluido
Performance Advisor				Incluido

Fuente: www.mysql.com

A continuación se procede a valorar a los SGBDs, en función de su precio y el tipo de licencias.

Tabla 3.XXX. Costos

Tipo de Licenciamiento y Costos			
	Oracle	SQL Server	MySQL
Costos	Extremadamente Altos	Altos	Cómodos
Calificación	6	8	10

Realizado Por: Pilar A. Quizhpe P.

Como se observa MySQL tiene gran ventaja, en precios para su servidor, pues a más de distribuirse bajo los términos de la Licencia Publica General GNU, para aplicaciones no comerciales, tiene precios extremadamente convenientes, en relación a los otros dos SGBD competidores. En cuanto a Oracle, se observa que cada funcionalidad añadida, tiene exorbitantes precios de licencias, lo que lo hace inalcanzable para pequeñas y medianas empresas, en donde MySQL tiene más acogida, por sus precios, razón que ha hecho de Oracle el motor de base de datos relacional más usado por compañías multinacionales en general, pues por sus costos de licenciamiento, solamente son accesibles, por dichas empresas. Además se debe pagar precios extremadamente altos, por soporte técnico, mientras que en SQL Server, y MySQL, el soporte está incluido dentro de la compra del Software de base de datos.

3.9 Comparativa de Soporte

3.9.1 Soporte Oracle

EL servicio de soporte de Oracle ofrece los siguientes programas:

Software Update Licence & Support: Proporciona a los clientes, el derecho a versiones actualizadas del producto y apoyo técnico 24 x 7, este servicio está disponible 5 años desde la fecha de versión del producto. La actualización del producto incluye versiones actualizadas del software, mantenimiento de versiones y parches. Además los clientes tienen acceso directo de expertos Oracle para cuestiones específicas del producto como instalación y operación del software de Oracle. El soporte Web es proporcionado por OracleMetaLink, cuyas características incluyen notificaciones proactivas, librerías y foros técnicos, información del ciclo de vida del producto, una base de datos de los bugs del producto, la capacidad de registrar las solicitudes de asistencia técnica.

Advanced Customer Support: Es un nivel de soporte avanzado para los clientes de Oracle, entregando soluciones de soporte adaptado y flexible, construido para satisfacer los requerimientos específicos del negocio del cliente. Los clientes de este tipo de soporte tienen la flexibilidad para adquirir servicios o combinaciones de servicios estándar con ofertas específicas para proporcionar una solución completa.

Paquetes de Soporte para incidentes en el Servidor: Este tipo de soporte proporciona asistencia técnica basada en la web y está limitado a los siguientes productos:

Oracle Application Server Support Package (1,000 USD x 10 incidentes x 1 servidor):
Internet Application Server Enterprise Edition, Internet Application Server Standard Edition,
Internet Application Server Java Edition

Oracle Database Server Support Package: (2,000 USD x 10 incidentes x 1 servidor): Oracle
Database Enterprise Edition, Oracle Database Standard Edition, Standard Edition One,
Partitioning, Real Application Clusters

Extended Support: Comprando el soporte extendido, los clientes con el programa "Software Update Licence & Support", pueden respaldar sus productos por 3 años adicionales, pasado los 5 años iniciales desde la fecha de disponibilidad general del producto.

3.9.2 Soporte SQL Server

Microsoft ofrece un mínimo de cinco años de Soporte Principal y cinco años de Soporte Extendido para productos empresariales y de desarrollo.

Soporte Principal: El Soporte técnico principal es la primera fase del ciclo de vida de soporte. A nivel de Service Pack del que se ofrece soporte, el soporte técnico principal incluye: Soporte técnico por incidencias (soporte sin cargo por incidencias, soporte con cargo por incidencias, soporte con tarifa por horas, soporte por reclamación de garantía), Soporte técnico a actualizaciones de seguridad, capacidad de solicitar correcciones que no son de seguridad

Soporte Extendido: Durante la fase de Soporte Extendido, las opciones de soporte de pago continúan estando disponibles y los parches de seguridad se siguen ofreciendo sin coste adicional. Sin embargo, los clientes que quieran soporte de parches no relativos a seguridad necesitan firmar un acuerdo de Soporte Extendido de Hotfix. El beneficio para los clientes de Soporte Extendido de Hotfix de Software Assurance incluye la tasa anual del Acuerdo de Soporte Extendido de Hotfix en los productos cubiertos con Software Assurance y estos clientes no necesitan firmar dicho acuerdo en los 90 días de transición de Soporte Principal del Producto hacia el Soporte Extendido. Un Hotfix es un paquete único compuesto por uno o más módulos de código utilizados para dirigir un cambio en un producto. Los Hotfix están diseñados para arreglar una situación específica de un cliente y no pueden ser distribuidos fuera de la organización del cliente sin consentimiento legal por escrito otorgado por Microsoft.

3.9.3 Soporte MySQL

MYSQL Network provee soporte y servicios directos de los desarrolladores de MYSQL asegurando una respuesta confiable, segura y rápida. Las personas interesadas en la herramienta tienen acceso a listas de correo, foros, documentación en línea, información

técnica de temas específicos. MySQL brinda diferentes tipos de asistencias técnicas tipos para los clientes con aplicaciones de base de datos crítica:

MySQL Network Basic: Provee consultores independientes para pequeños negocios con servicios que mejoran la productividad. Tiene acceso a software certificado y optimizado, mantenimiento, UPDATES, ingreso a la base de conocimiento de MYSQL. Tiene un máximo de dos incidencias, acceso web y un tiempo máximo de respuesta de 2 días. Este tipo de soporte cuesta \$595.00 por servidor por año.

MySQL Network Silver: Provee los mismos servicios que el básico en lo que se refiere a software, pero en la parte de resolución de problemas, puede tener un número de incidentes ilimitado, acceso web, acceso telefónico 8x5 y un límite de 4 horas en el tiempo de respuesta. El costo es 1995.00 por servidor y por año.

MySQL Network Gold: tiene las mismas particularidades del básico y plata, pero en la resolución de problemas tiene un número de incidentes ilimitado, acceso web, acceso telefónico 24x7, máximo tiempo de repuesta a incidencias 2 horas, además, en caso de emergencias, tiene un tiempo máximo de respuesta de 30 minutos, también dispone de resolución de problemas en forma remota. El costo es de 2995.00 por servidor por año.

MySQL Network Platinum: Además de las particularidades de software de las anteriores, tiene como opción el desarrollo de proyectos especiales, tiene un número de incidentes ilimitado, acceso web, acceso a llamadas telefónicas 24x7, máximo tiempo de respuesta en incidencias 30 min, en caso de emergencia tiene también 30 minutos de tiempo base. Para soporte de consultaría tiene resolución de problemas remotamente, revisión de código, revisión de performance, revisión de código para manejar las diferentes API'S que soporta MySQL y funciones definidas por el usuario. El costo es de 4995.00 por servidor por año.

Cada una de las empresas provee paquetes de soporte técnico, para cada uno de sus productos, la diferencia radica en los precios y las ofertas, sin embargo se puede calificar

como aceptable el soporte brindado por cada una de las empresas desarrolladoras del producto, por lo q cada producto se lo califica con 10.

3.10 Comparativa de Herramientas de Administración

Las Herramientas de Administración, constituyen un punto importantísimo en la selección de un SGBD, pues facilitan la gestión del administrador de la base de Datos, en la manipulación de los componentes del catálogo de datos.

Tabla 3.XXXI. Herramientas de Administración

Herramientas de Administración			
	Oracle	SQL Server	MySQL
Administración	Enterprise Manager	Management Studio	Control Center
Importación y Exportación	Soporte Completo	Soporte Completo	Soporte Completo
Auto Tuning	Soporte Completo	Soporte Completo	
Análisis de Consultas	Soporte Completo	Soporte Completo	Soporte Completo
Modelado			Soporte Completo
Desarrollo	Soporte Completo	Soporte Completo	Soporte Completo
Monitor de Rendimiento	Soporte Completo	Soporte Completo	Soporte Completo
Alta Disponibilidad	Soporte Completo	Soporte Completo	Soporte Completo
Total de Herramientas	6	6	6
Calificación sobre 10	8.5	8.5	8.5

Realizado Por: Pilar A. Quizhpe P.

CAPÍTULO IV

4. ANÁLISIS DE RENDIMIENTO DE LOS SISTEMAS DE GESTIÓN DE BASES DE DATOS ORACLE, SQL SERVER Y MySQL.

4.1 Introducción

El mercado de sistemas de gestión de bases de datos ofrece numerosas alternativas en cuanto a sistemas servidores para grandes, medianas, y pequeñas empresas. Sin embargo al tomar una decisión sobre qué sistema elegir, es necesario tener en cuenta cuál es el más optimo cual ofrece mayores garantías, en cuanto a rendimiento a los proyectos de las compañías a las que deben ajustarse.

Estas razones son las que han motivado a seleccionar los sistemas que serán sometidos a prueba, estos son: Oracle 10g, SQL Server 2005 y MySQL 5.1, sistemas gestores de bases de datos más utilizados a nivel empresarial y que en el caso de Oracle y MySQL se encuentran disponibles para la mayoría de plataformas disponibles.

4.2 Descripción del Problema

Un sistema de gestión de base de datos, se debe desempeñar a un cierto nivel para que sea válido a los usuarios; tiempos inaceptables de respuesta a las solicitudes del usuario llevará a la búsqueda de otros medios para satisfacer sus necesidades de información.

En este contexto, el problema se enfoca en determinar el desempeño (tiempo de respuesta) del SGBD sometido a situaciones de carga extrema, criterio de medición que permitirá determinar el rendimiento de los SGBD desde perspectivas como almacenamiento y recuperación de datos (velocidad de ejecución de consultas de complejidad simple, complejidad media, y complejidad alta, así como la velocidad de inserción masiva de datos), etc.

4.3 Plataforma Hardware

La computadora usada en las pruebas de rendimiento se corresponde con las siguientes características:

- **Tipo:** CLON
- **Procesador:** Intel Core 2 Duo
- **Velocidad:** 3 GHz
- **RAM:** 1 Gb
- **Disco Duro:** 250 Gb

4.4 Plataforma Software

- **Virtualizador:** VmWare Workstation
- **Sistema Operativo:** MS Windows XP PRO Services Pack 3
- **Gestor de Base de Datos:** MS SQL Server Enterprise 2005
- **Gestor de Base de Datos:** Oracle 10g Enterprise Edition
- **Gestor de Base de Datos:** MySQL Server 5.1 Enterprise Edition
- **Herramientas de Gestión:** Oracle Express Manager
SQL Manager Studio 2005
MySQL GUI Tools

4.5 Técnicas utilizadas en la comparación

La comparativa se basa en la técnica de análisis de sensibilidad (¿what if?) que se convierte en el primer paso para reconocer la incertidumbre, es decir examina cómo el cambio en una variable afecta un resultado.

Esto es de mucha utilidad porque permite identificar las variables más críticas o construir escenarios posibles que permitirán analizar el comportamiento de un resultado bajo diferentes supuestos. En otras palabras, el análisis de sensibilidad permite medir el cambio en un resultado, dado un cambio en un conjunto de variables, tanto en términos relativos, como en términos absolutos²⁷.

4.6 Procedimiento previo a la ejecución de los test de rendimiento

El procedimiento previo a la ejecución de los test de rendimiento, se resume en los siguientes pasos:

- Instalación y configuración del Gestor de Base de Datos.
- Creación del prototipo de Base de Datos, sobre el que se aplicará las pruebas.
- Implementación del prototipo de base de datos en cada SGBD.
- Determinación de los escenarios de prueba.
- Determinación de variables implícitas en el análisis de sensibilidad.
- Diseño de los test de Rendimiento.
- Ejecución y resultados del análisis de sensibilidad.

4.7 Instalación de los Sistemas de Gestión de Bases de Datos

Se utiliza la herramienta VM Workstation Machine, para implementar sistemas virtuales, en donde se instalan los sistemas de gestión de bases de datos Oracle 10g y MySQL 5. El sistema SQL Server es instalado en la máquina real. Cada sistema tiene asignado 1 GB de memoria RAM.

²⁷ **VÉLEZ, Ignacio.** *Sensitivity Analysis*. [PDF]. Documento electrónico adquirido en Internet. 2003. [Fecha de Consulta: 20 de Marzo del 2009] Disponible en SSRN: <http://ssrn.com/abstract=986887>

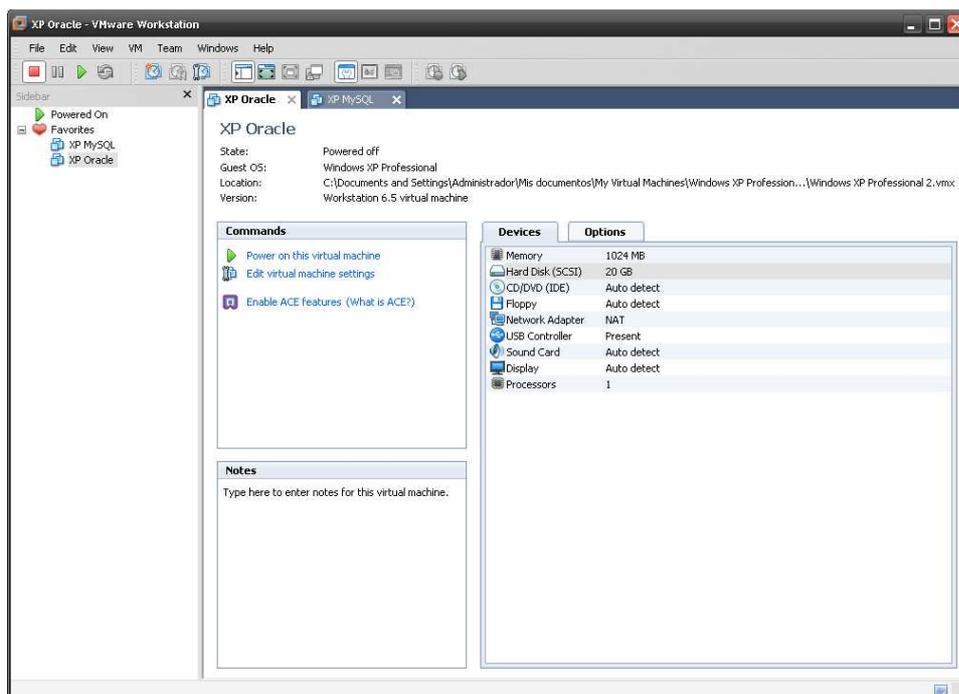


Figura 4.1. Interfaz de VmWare

VmWare es un virtualizador que permite ejecutar (simular) varios ordenadores (sistemas operativos) dentro de un mismo hardware de manera simultánea, permitiendo así el mayor aprovechamiento de recursos.

No obstante, y al ser una capa intermedia entre el sistema físico y el sistema operativo que funciona en el hardware emulado, la velocidad de ejecución de este último es menor, pero en la mayoría de los casos es suficiente para usarse en entornos de producción²⁸

El proceso subsiguiente de instalación no es detalle de este trabajo investigativo, por ser básica en el estudio de formación pregrado.

4.8 Prototipo de la Base de Datos

La base de datos usada para la prueba de rendimiento, es un "modelo de bases de datos para estructuras planas" y se compone de 6 tablas referenciadas entre sí²⁹. El modelo ha sido escogido como prototipo de evaluación para este proyecto, sobre el cual se aplican las

²⁸ **Desconocido.** VmWare. [ON-LINE] [Fecha de Consulta: Enero 2009]. Disponible en <http://es.wikipedia.org/wiki/VMware>

²⁹ **Saravia Roger.** Modelo de Bases de Datos para Estructuras Planas. Maestría en Ingeniería del Software. La Paz - Bolivia. Universidad Mayor de San Andrés. 2007. 6-8p

pruebas que permitan medir el rendimiento de cada sistema gestor de base de datos sometiéndolo a cargas de trabajo combinado con operaciones numéricas.

Las tablas están relacionadas de acuerdo al siguiente detalle:

ELEMENTO (codele, *tipoe*, *codprop*, *fix*, *fiy*, *fiz*, *codnudini*, *codnudfin*)

PROPIEDADES (codprop, *elasticidad*, *inercia*, *area*)

CARGELEM (codce, *tipo*, *inicio*, *longitud*, *fx*, *fy*, *mz*, *codele*)

NUDO (codnud, *tponu*, *x*, *y*, *codbor*, *dx*, *dy*, *gz*, *rx*, *ry*, *rz*)

CARGNUD (codcn, *px*, *py*, *pz*, *codnud*)

BORDE (codbor, *borx*, *bory*, *borz*)

A continuación se observa el prototipo implementado en el servidor SQL server y MySQL.

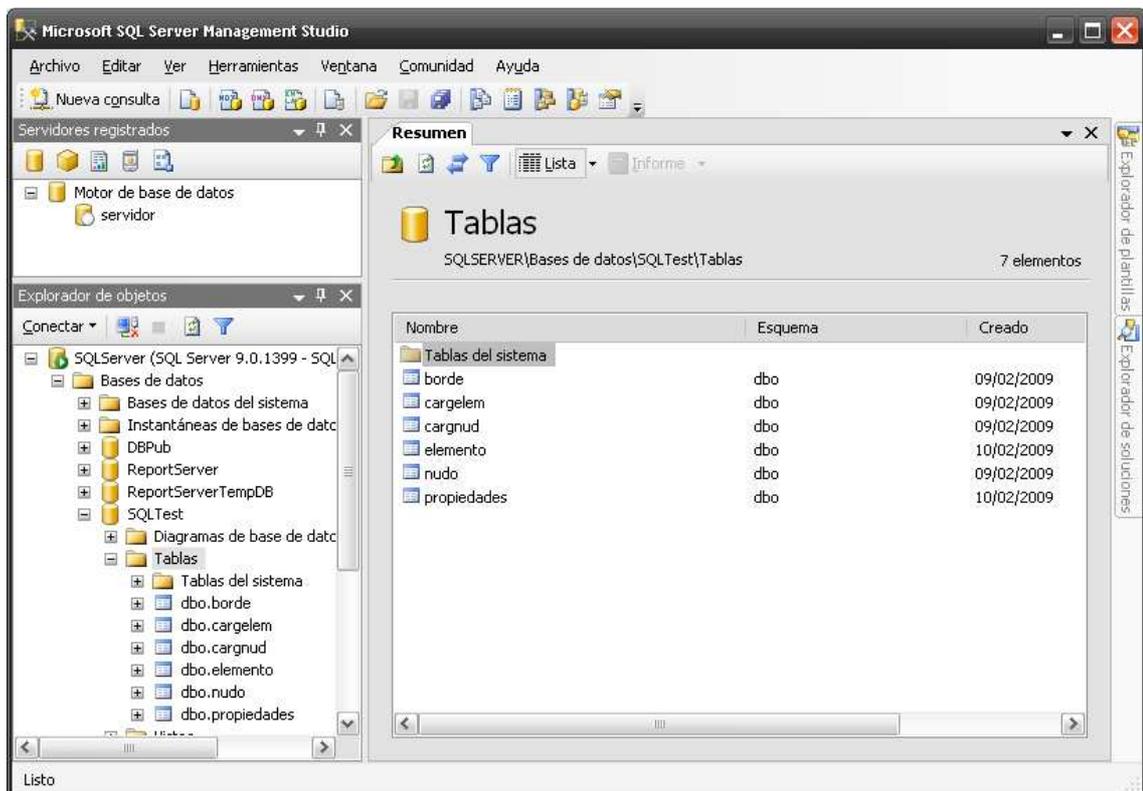


Figura 4.2. Prototipo en SQL Server

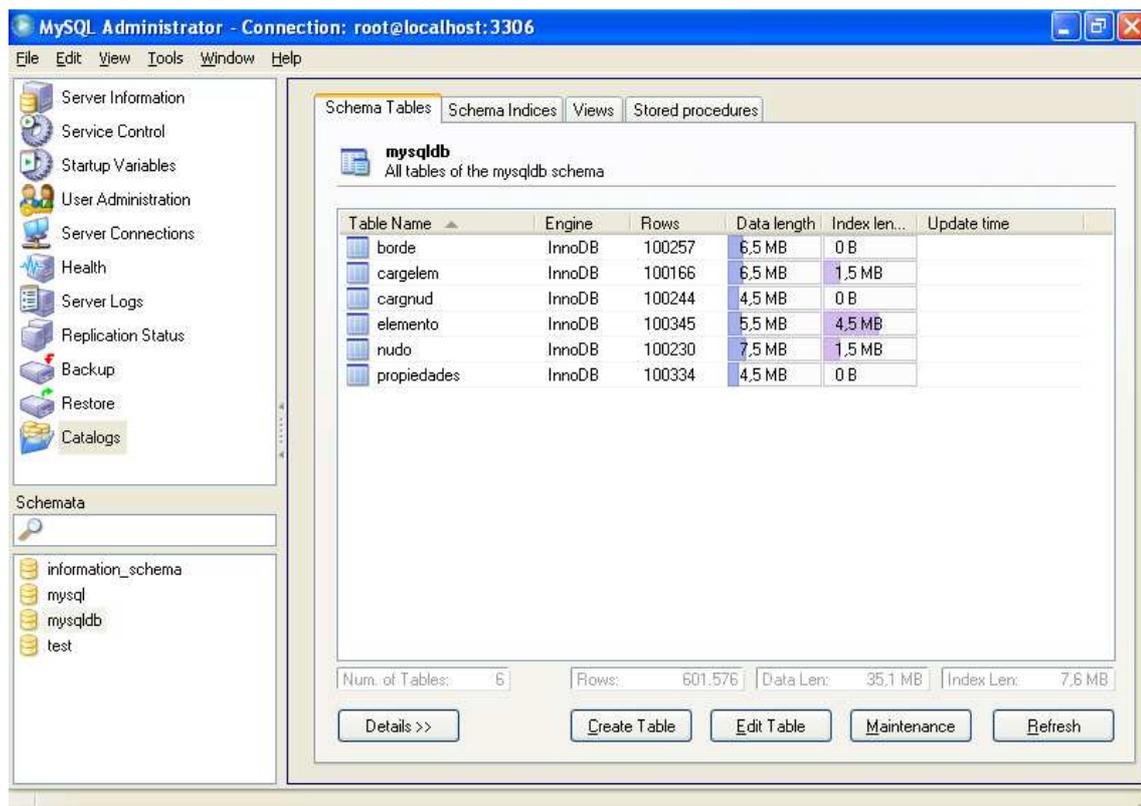


Figura 4.3. Prototipo en MYSQL

4.9 Criterios de Medición

Un SGBD es en primer lugar un administrador de información, por ello es importante poder determinar con qué efectividad administra sus recursos, es decir cuál es el rendimiento que presenta el sistema de gestión de bases de datos, para tal o cual operación.

El rendimiento en el ámbito informático se define como un *criterio de medición o cuantificación de la velocidad/resultado con que se realiza una tarea o proceso*³⁰.

Cuando se diseña una base de datos, es necesario asegurarse de que realiza todas las operaciones importantes de forma rápida y correcta, manteniendo el máximo rendimiento incluso con las limitaciones que puedan existir.

4.10 Determinación de los escenarios de prueba.

La importancia del rendimiento radica en los tiempos de respuesta razonables en los procesos críticos del servidor, por esta circunstancia, es necesario observar el

³⁰**ALEGSA.** Definición de Rendimiento. [ON-LINE]. Diccionario Informático. <http://www.alegsa.com.ar/dic/rendimiento.php>

comportamiento del SGBD en materia de rendimiento desde diferentes perspectivas como el tipo y complejidad de las operaciones, así como con el volumen de información que maneja la base de datos.

Este análisis permitirá determinar los alcances del sistema, de tal manera que el analista decida dónde concentrar esfuerzos para maximizar el desempeño del servidor, ya sea mejorando las características del hardware, configurar los elementos del servicio de base de datos para que su uso sea óptimo en velocidad y en uso de memoria, o bien rediseñando las estrategias de programación sobre la base de datos.

A continuación se observan los escenarios, sintetizados lógicamente en la siguiente tabla:

Tabla 4.I. Escenarios de Prueba

Volumen de Información		Operaciones			
		Inserción	Selección		
			Complejidad		
Cantidad de Filas		Simple	Media	Alta	
Baja a Normal	1000				
	5000				
	10000				
	50000				
Alto	100000				
	500000				

Realizado Por: Pilar A. Quizhpe P.

Se pretende analizar el efecto que produce la variación del número de filas insertadas y la variación del número de filas recuperadas en el tiempo de respuesta de los SGBD en estudio. Para esto se definieron 6 escenarios con los siguientes valores: 1000, 5000, 10000, 50000, 100000, 500000 filas.

Además teniendo en cuenta las prestaciones más importantes que deben ser cumplidas por los sistemas de gestión de bases de datos y que permitan evaluar el criterio de medición de rendimiento, mencionado en la sección 4.8, se escogen las operaciones más críticas de los SGBD, que son: inserción, y selección de datos, mismas que serán evaluadas a partir de diversos grados de complejidad, para de esta manera observar el comportamiento de los SGBD en estudio.

4.11 Definición de los grados de complejidad de las consultas

Se han definido tres tipos de consultas, con diferentes grados de complejidad.

Tabla 4.II. Grados de Complejidad

Grados de Complejidad				
Complejidad	Atributos	Joins	Ordenamientos	Dirección
Simple	2	1	1	1 Descendente
Media	4	2	2	2 Descendentes
Alta	10	5	3	3 Descendentes

Realizado Por: Pilar A. Quizhpe P.

4.12 Determinación de las variables implícitas en el análisis

Se determinan las variables e indicadores de acuerdo a los escenarios establecidos, en la sección 4.10.

Tabla 4.III. Variables e Indicadores

Variables e Indicadores		
Variables	Indicadores	Observaciones
Volumen de Información	Cantidad de Filas	Variación de las filas entre [1000,500000]
Costes de Almacenamiento	Tiempo de Inserción	Tiempo que se demora en insertar las cantidad de filas especificado.
Costes de acceso a Datos	Tiempo de respuesta	Tiempo que se demora en seleccionar la cantidad de filas especificado
Rendimiento	Velocidad de Inserción	La velocidad con que se realiza el proceso
	Velocidad de Respuesta	

Realizado Por: Pilar A. Quizhpe P.

4.13 Diseño de los Test de Rendimiento

Los test de rendimiento, están diseñados para medir los tiempos de respuesta en los escenarios determinados con anterioridad, a partir de los cuales podemos determinar la velocidad de ejecución de los SGBD tanto para la inserción como para la selección de datos con diversos grados de complejidad.

4.13.1 Test de Rendimiento No 01

Tabla 4.IV. Test de Rendimiento No 01

Test No 01			
Operación	INSERCIÓN		
Objetivo	Obtener el tiempo que consume cada SGBD para completar el proceso de inserción de X filas en cada una de las tablas de la base de datos, para posteriormente calcular la velocidad de inserción del SGBD.		
Proceso	Consiste en ejecutar procedimientos almacenados pasando como parámetro de entrada el número de filas que deberán ser insertadas en cada una de las tablas de la base de datos prototipo. Para esta tarea, será activado el tiempo de reloj de cada uno de los SGBD, de tal manera que pueda obtenerse el tiempo que demora en el proceso.		
Variables	Costes de Almacenamiento Rendimiento Volumen de Información	Indicadores	Tiempo de Inserción Velocidad de Ejecución Cantidad de Filas
Descripción	Se tomará 5 lecturas por cada proceso.		
Herramientas	SQL Server Profiler MySQL Command Line Client iSQL PLUS		
Procedimiento			
<ol style="list-style-type: none"> 1. Crear los procedimientos necesarios para insertar la información en cada una de las tablas de la Base de Datos 2. Ejecutar Monitor de Rendimiento 3. Crear una traza 4. Ejecutar los procedimientos almacenados 5. Parar la ejecución de la traza 6. Obtener el tiempo consumido en la operación. 			
Código			
<pre> CREATE PROCEDURE SP_INSERT(IN NUMROWS INT) BEGIN DECLARE I INT; SET I=0; WHILE (I<=NUMROWS) DO INSERT INTO BORDE VALUES(I+10,'F','L','F'); SET I=I+1; END WHILE; SET I=0; WHILE (I<=NUMROWS) DO INSERT INTO NUDO VALUES(I+10,'A',I*2,I*3,11,I*1.5,I*2.5,I*3.5,I*1.8,I*2.8,I*3.8); SET I=I+1; END WHILE; SET I=0; WHILE (I<=NUMROWS) DO INSERT INTO CARGNUD VALUES(I+2,I*1.2,I*1.3,I*1.4,I+10); SET I=I+1; END WHILE; SET I=0; WHILE (I<=NUMROWS) DO INSERT INTO PROPIEDADES VALUES(I, I*2,I*3,I*3.5); SET I=I+1; END WHILE; SET I=0; WHILE (I<=NUMROWS) DO INSERT INTO ELEMENTO VALUES(I+10,'V',I,I*2.1,I*3.1,I*4.1,I+10,I+10); SET I=I+1; END WHILE; SET I=0; WHILE (I<=NUMROWS) DO INSERT INTO CARGELEM VALUES(I+10,'P',I*1.1,I*1.2,I*2.1,I*3.1,I*4.1,I+10); SET I=I+1; END WHILE; END; </pre>			

Realizado Por: Pilar A. Quizhpe P.

4.13.2 Test de Rendimiento No 02

Tabla 4.V. Test de Rendimiento No 02

Test No 02	
Operación	Selección
Objetivo	Obtener el tiempo de respuesta que consume el SGBD para completar el proceso de selección de complejidad simple en la base de datos con X cantidad de información, para posteriormente calcular la velocidad de consulta del SGBD.
Proceso	<p>Consiste en construir una consulta de complejidad simple, la misma que será evaluada sobre una base de datos, con X número de filas en cada una de las tablas.</p> <p>Para esta tarea, será activado el tiempo de reloj de cada uno de los SGBD, de tal manera que pueda obtenerse el tiempo que demora en el proceso.</p>
Descripción	Se tomará 5 lecturas por cada proceso.
Procedimiento	
<ol style="list-style-type: none"> 1. Crear la consulta de complejidad simple. 2. Implementar la consulta de complejidad simple en cada SGBD, con su correspondiente sintaxis. 3. Ejecutar la consulta y obtener el tiempo de respuesta del SGBD 4. Tabular los datos obtenidos 5. Variar el volumen de información, en cada SGBD. 1. Repetir el mismo proceso anterior 	
Código	
<pre>SELECT elemento.tipoe, cargelem.mz FROM elemento JOIN cargelem ON elemento.codele = cargelem.codele;</pre>	

Realizado Por: Pilar A. Quizhpe P.

4.13.3 Test de Rendimiento No 03

Tabla 4.VI. Test de rendimiento No 03

Test No 03	
Operación	Selección
Objetivo	Obtener el tiempo de respuesta que consume el SGBD para completar el proceso de selección de complejidad media en la base de datos con X cantidad de información, para posteriormente calcular la velocidad de consulta del SGBD.
Proceso	<p>Consiste en construir una consulta de complejidad media, la misma que será evaluada sobre una base de datos, con X número de filas en cada una de las tablas.</p> <p>Para esta tarea, será activado el tiempo de reloj de cada uno de los SGBD, de tal manera que pueda obtenerse el tiempo que demora en el proceso.</p>
Descripción	Se tomará 5 lecturas por cada proceso.
Procedimiento	
<ol style="list-style-type: none"> 1. Crear la consulta de complejidad media. 2. Implementar la consulta de complejidad media en cada SGBD, con su correspondiente sintaxis. 3. Ejecutar la consulta y obtener el tiempo de respuesta del SGBD 4. Tabular los datos obtenidos 5. Variar el volumen de información, en cada SGBD. 6. Repetir el mismo proceso anterior 	
Código	
<pre> SELECT NUDO.DX, NUDO.RZ, ELEMENTO.FIY, CARGELEM.MZ FROM CARGELEM JOIN ELEMENTO ON CARGELEM.CODELE = ELEMENTO.CODELE JOIN NUDO ON ELEMENTO.CODNUDINI = NUDO.CODNUD AND ELEMENTO.CODNUDFIN = NUDO.CODNUD ORDER BY NUDO.RY DESC,ELEMENTO.FIX DESC </pre>	

Realizado Por: Pilar A. Quizhpe P.

4.13.4 Test de Rendimiento No 04

Tabla 4.VII. Test de Rendimiento No 04

Operación	Selección
Objetivo	Obtener el tiempo de respuesta que consume el SGBD para completar el proceso de selección de complejidad media en la base de datos con X cantidad de información, para posteriormente calcular la velocidad de consulta del SGBD.
Proceso	<p>Consiste en construir una consulta de complejidad media, la misma que será evaluada sobre una base de datos, con X número de filas en cada una de las tablas.</p> <p>Para esta tarea, será activado el tiempo de reloj de cada uno de los SGBD, de tal manera que pueda obtenerse el tiempo que demora en el proceso.</p>
Descripción	Se tomará 5 lecturas por cada proceso.
Procedimiento	
<ol style="list-style-type: none"> 1. Crear la consulta de alta complejidad. 2. Implementar la consulta de alta complejidad en cada SGBD, con su correspondiente sintaxis. 3. Ejecutar la consulta y obtener el tiempo de respuesta del SGBD 4. Tabular los datos obtenidos 5. Variar el volumen de información, en cada SGBD. 6. Repetir el mismo proceso anterior 	
Código	
<pre> SELECT CARGNUD.PX, CARGNUD.PY, BORDE.BORX, BORDE.BORY, ELEMENTO.TIPOE, CARGELEM.TIPO, CARGELEM.INICIO, PROPIEDADES.CODPROP, PROPIEDADES.ELASTICIDAD, NUDO.TIPONU FROM CARGNUD JOIN NUDO ON CARGNUD.CODNUD = NUDO.CODNUD JOIN BORDE ON NUDO.CODBOR = BORDE.CODBOR JOIN ELEMENTO ON NUDO.CODNUD = ELEMENTO.CODNUDINI AND NUDO.CODNUD = ELEMENTO.CODNUDFIN JOIN PROPIEDADES ON ELEMENTO.CODPROP = PROPIEDADES.CODPROP JOIN CARGELEM ON ELEMENTO.CODELE = CARGELEM.CODELE ORDER BY NUDO.RX DESC, ELEMENTO.FIX DESC, CARGELEM.MZ DESC </pre>	

Realizado Por: Pilar A. Quizhpe P.

Se crea una traza, y a continuación se procede a establecer las propiedades de la misma.

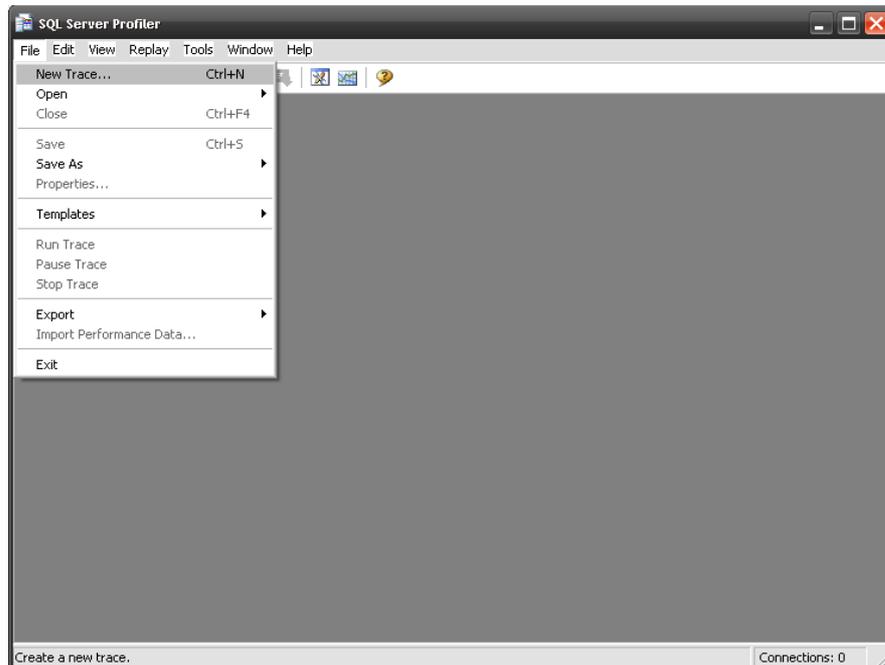


Figura 4.6. Crear nueva traza

Al presionar el botón RUN, se ejecuta la traza, la cual queda activada, rastreando todos los procesos de SQL Server.

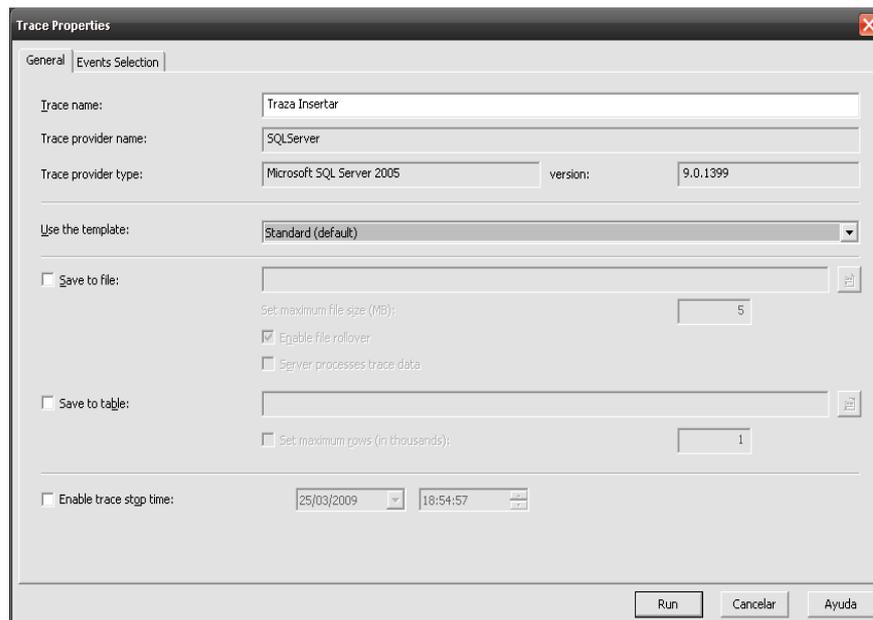


Figura 4.7. Propiedades de la Traza

Se procede a ejecutar la operación, misma que será monitoreada por SQL Server Profiler

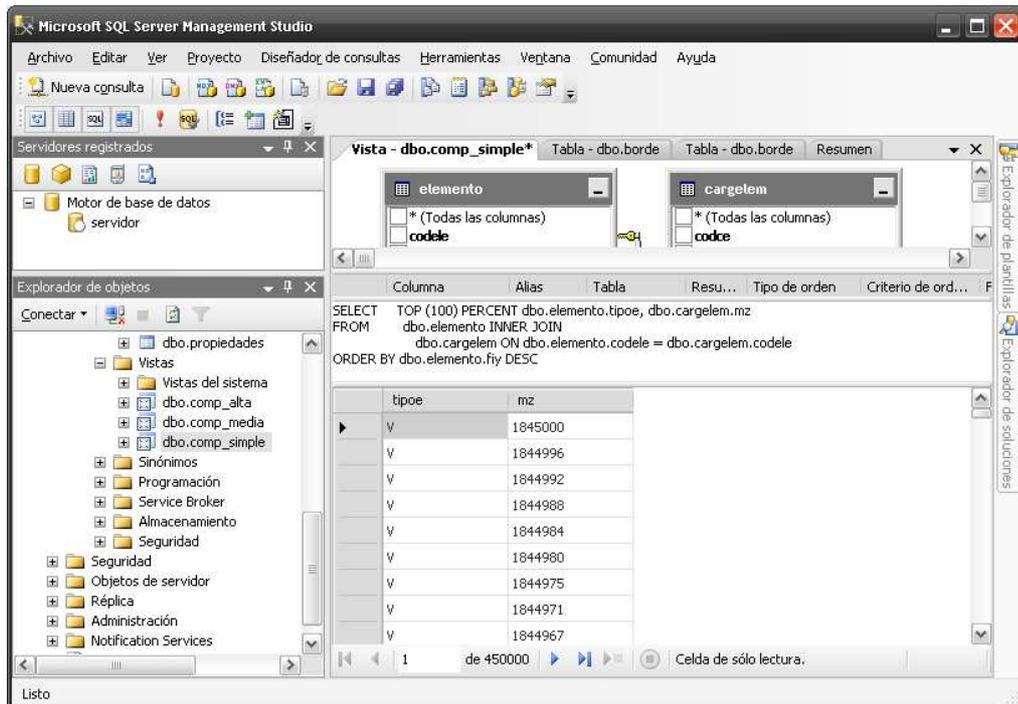


Figura 4.8. Ejecución del Proceso

Se observa los resultados del proceso en la Herramienta SQL Server Profiler.

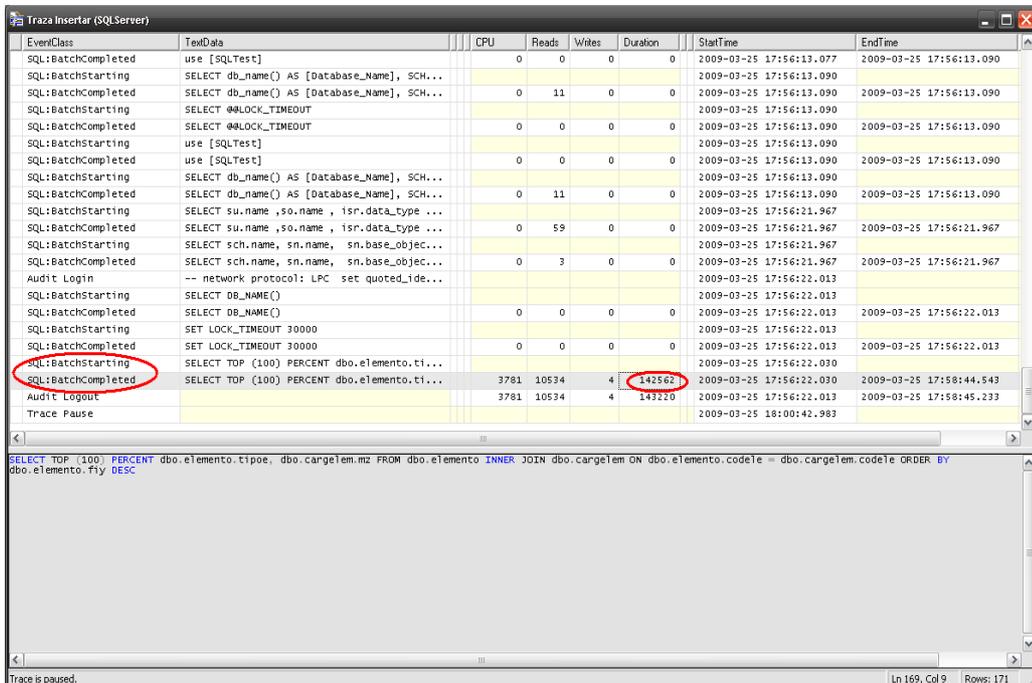


Figura 4.9. Duración del Proceso

4.14.2 MySQL

Se procede a diseñar las operaciones utilizando la herramienta MySQL Administrator.

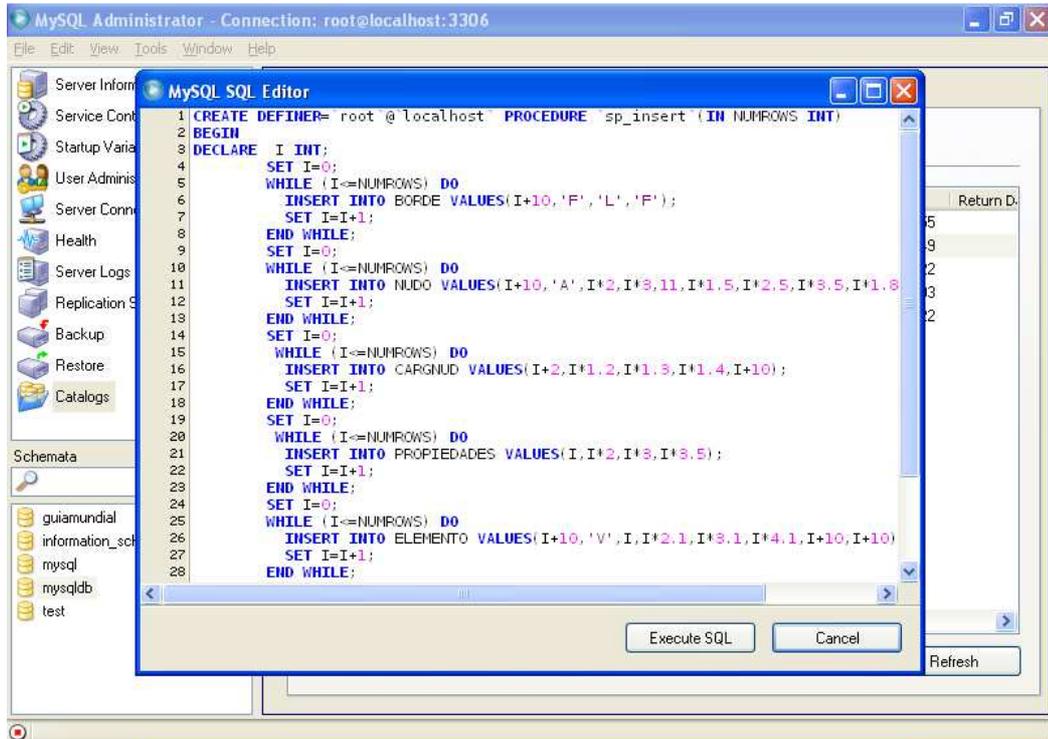


Figura 4.10. Diseño de la Operación en MySQL

Para la toma de datos de los procesos de inserción se usa la herramienta MySQL Command Line, la misma que provee el tiempo de ejecución que toma el SGBD para realizar el proceso.

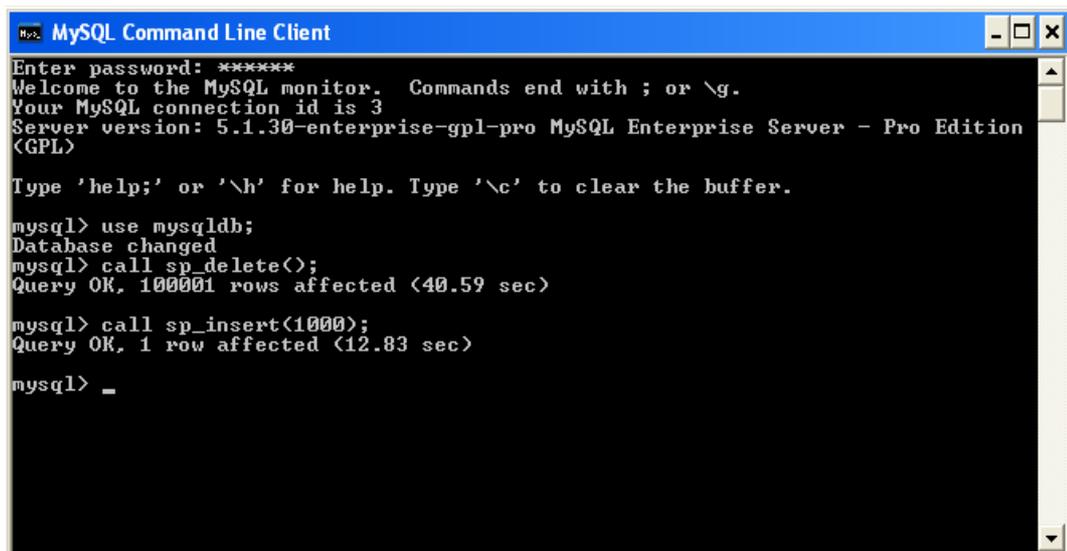


Figura 4.11. MySQL Command Line

Para la toma de datos de los procesos de consulta, se usa la herramienta MySQL Browser, que permite obtener los tiempos de consulta de MySQL

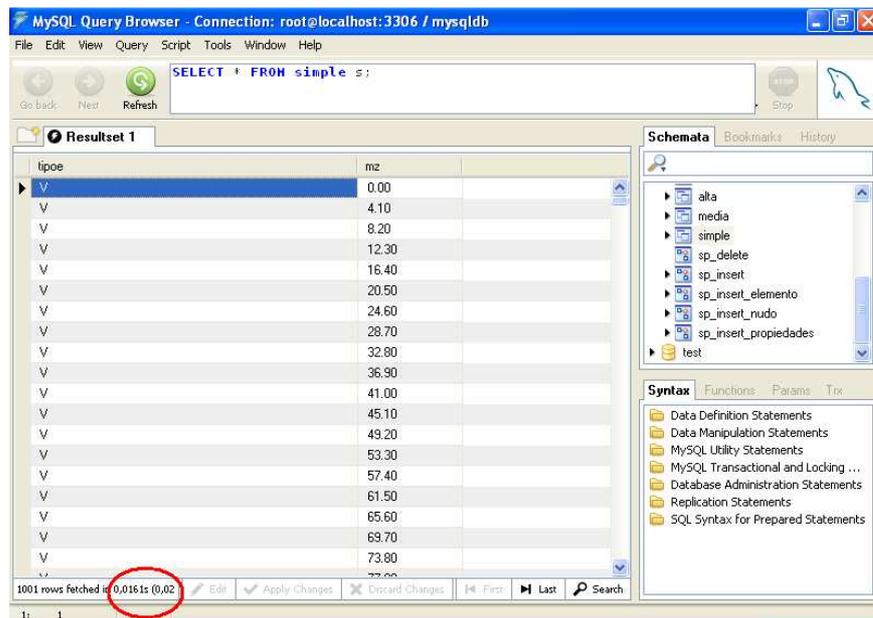


Figura 4.12. MySQL Browser

4.14.3 Oracle

En Oracle se utiliza la herramienta iSQL Plus, para diseñar los procesos que deberán ser evaluados.

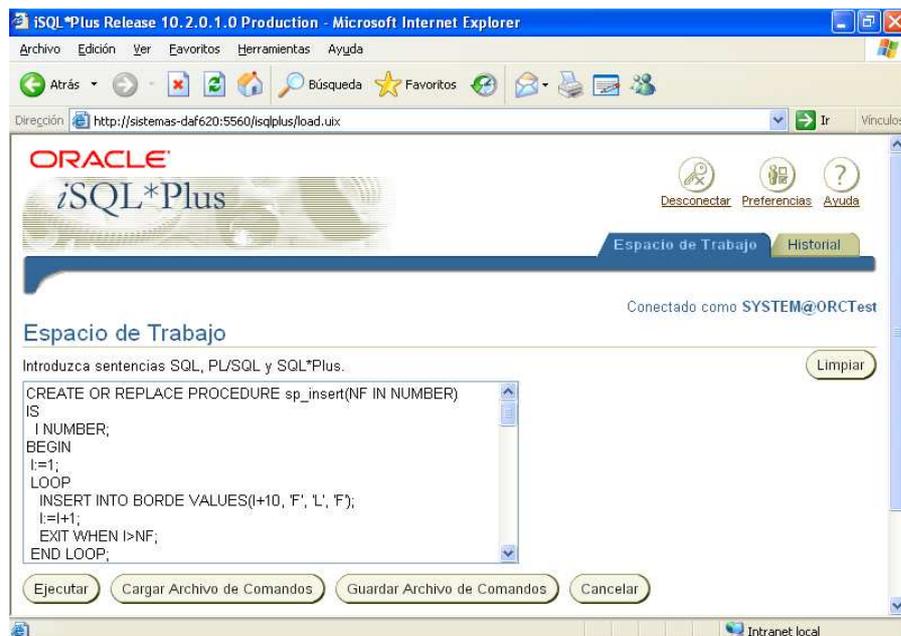


Figura 4.13. iSQL Plus

Es necesario activar el reloj interno de Oracle con el siguiente comando.

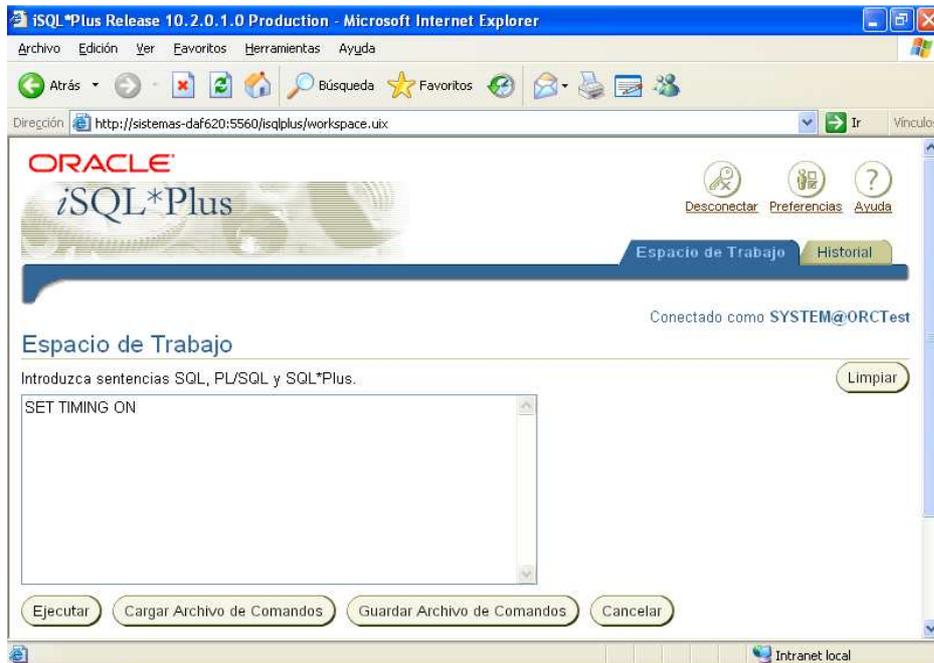


Figura 4.14. Activación del reloj interno.

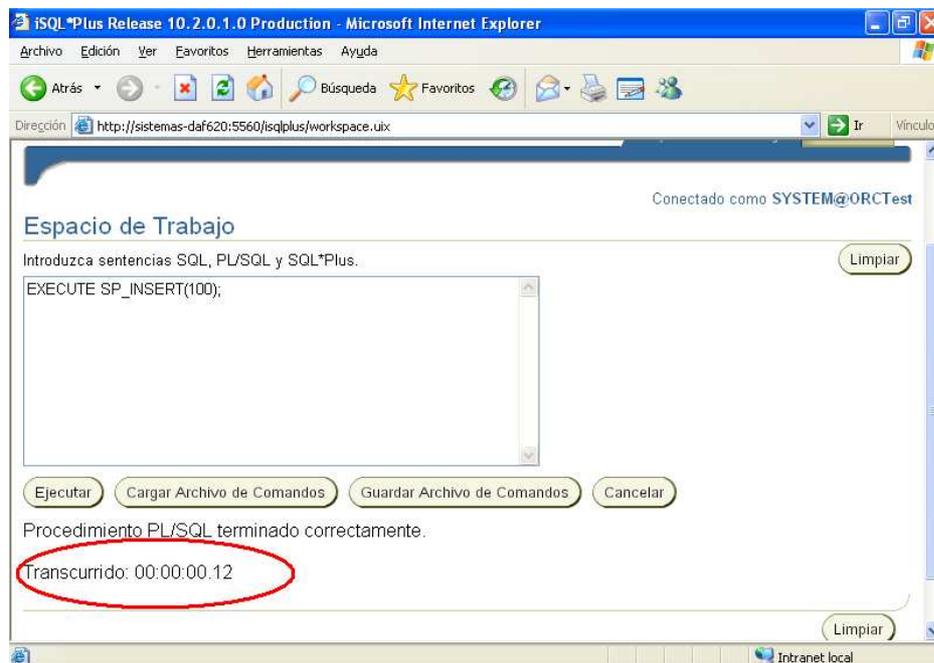


Figura 4.15. Tiempo de respuesta de Oracle

4.15 Test de Rendimiento: Carga Masiva de Datos

Se procede a tabular los resultados de la ejecución de los scripts preparados para la inserción masiva de filas, en cada uno de los Sistemas de Gestión de Bases de Datos obteniendo los siguientes datos.

4.15.1 Tablas de Lectura del Tiempo de los SGBD

Resultados de la ejecución de Oracle

Tabla 4.VIII. Tiempo de ejecución de Oracle para carga masiva de datos

Oracle (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	1980	2510	2200	2620	2700	2402
5.000	7820	9180	10750	9450	10180	9476
10.000	17090	19250	18540	20120	19460	18892
50.000	77750	76940	77980	76450	77820	77388
100.000	155090	153250	156450	157210	155130	155426
500.000	877640	875480	876990	877650	879840	877520

Realizado Por: Pilar A. Quizhpe P.

Resultado de la ejecución de SQL Server

Tabla 4.IX. Tiempo de ejecución de SQL Server para carga masiva de datos

SQL Server (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	1275	1288	1261	1148	1278	1250
5.000	7416	6896	6650	6646	6570	6837
10.000	12543	12476	12212	12529	12308	12414
50.000	57526	57972	57434	58103	57393	57686
100.000	115640	114699	113564	114488	113675	114413
500.000	562350	548616	550719	563489	571109	559257

Realizado Por: Pilar A. Quizhpe P.

Resultados de la ejecución de MySQL Server

Tabla 4.X. Tiempo de ejecución de MySQL para carga masiva de datos

MySQL Server (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	13690	9170	9300	9200	8980	10068
5.000	42250	44300	44480	44410	44030	43894
10.000	85090	88170	87150	89230	87450	87418
50.000	426030	425780	465280	415460	478650	442240
100.000	871550	872120	871400	871200	870980	871450
500.000	4321090	4320980	4320970	4321890	4321540	4321294

Realizado Por: Pilar A. Quizhpe P.

4.15.2 Comparativa del tiempo de ejecución para la operación INSERT

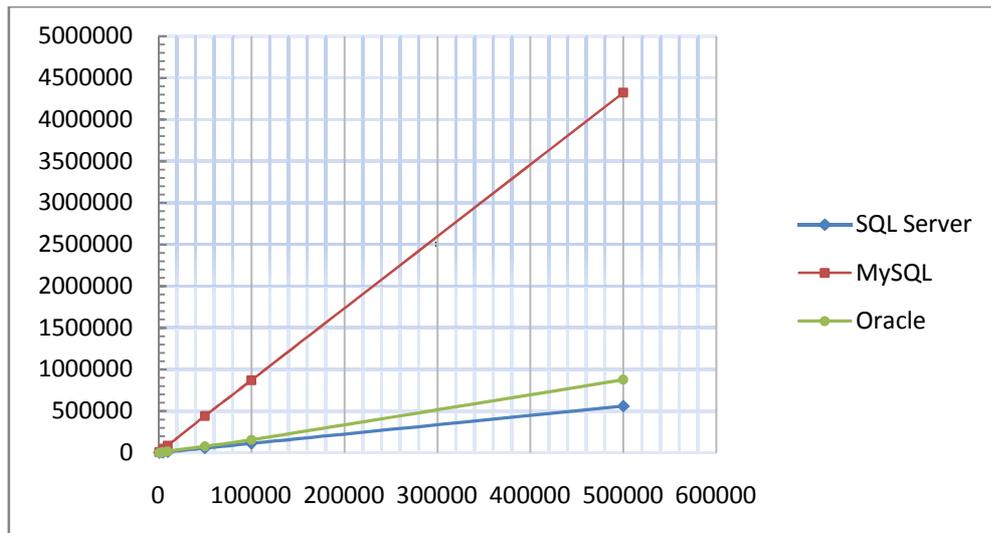
Tablas Resumen de Lectura de Tiempos

Tabla 4.XI. Tiempo de Ejecución para carga masiva de datos entre SGBD

Operación INSERT (milisegundos)			
Filas	Oracle	SQL Server	MySQL
1.000	2402	1288	10068
5.000	9476	6896	43894
10.000	18892	12476	87418
50.000	77388	57972	442240
100.000	155426	114699	871450
500.000	877520	548616	4321294

Realizado Por: Pilar A. Quizhpe P.

Gráfica Comparativa



Grafica 4.1. Tiempo de respuesta de los SGBD en la operación INSERT

Realizado Por: Pilar A. Quizhpe P.

Se observa que las variaciones de la función (tiempo de respuesta) son proporcionales a los de la variable independiente (filas), por lo que matemáticamente se analizará la grafica como función lineal, de donde podemos calcular la velocidad de inserción de los datos utilizando la pendiente de la función.

4.15.3 Cálculo de la velocidad de inserción de los SGBD

Velocidad de Inserción de Oracle

Tomamos $P_1(1000, 2402)$ y $P_2(500000, 877520)$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{877520 - 2402}{500000 - 1000} = \frac{875118}{499000}$$

$$\text{Tiempo de ejecución por fila } m = 1.75 \frac{ms}{filas}$$

$$\text{Velocidad de Inserción} = \frac{1 \text{ filas}}{1,75 \text{ ms}} = 0.57 \frac{filas}{ms}$$

Velocidad de Inserción de SQL Server:

Tomamos $P_1(1000, 1288)$ y $P_2(500000, 548616)$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{548616 - 1288}{500000 - 1000} = \frac{547328}{499000}$$

$$\text{Tiempo de ejecución por fila } m = 1.09 \frac{ms}{filas}$$

$$\text{Velocidad de Inserción} = \frac{1 \text{ filas}}{1,09 \text{ ms}} = 0.92 \frac{filas}{ms}$$

Velocidad de Inserción de MySQL Server:

Tomamos $P_1(1000, 10068)$ y $P_2(500000, 4321294)$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{4321294 - 10068}{500000 - 1000} = \frac{4311226}{499000}$$

$$\text{Tiempo de ejecución por fila } m = 8.64 \frac{ms}{filas}$$

$$\text{Velocidad de Inserción} = \frac{1 \text{ filas}}{8.64 \text{ ms}} = 0.12 \frac{filas}{ms}$$

4.16 Test de Rendimiento: Consultas de diversa complejidad

4.16.1 Tablas de Lectura de Tiempos de SQL Server

A continuación se observan, los resultados obtenidos para la operación SELECT.

Tabla 4.XII. Tiempo de respuesta de SQL Server para consultas simples

SQL Server - Complejidad Simple (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	2	3	4	3	3	3
5.000	685	678	665	669	595	658
10.000	1458	1547	1365	1464	1341	1435
50.000	6630	5732	6367	6170	5966	6173
100.000	13553	13166	13812	13103	13612	13449
500.000	176758	167370	189607	183569	160335	175528

Realizado Por: Pilar A. Quizhpe P.

Tabla 4.XIII. Tiempo de respuesta de SQL Server para consultas medias

SQL Server - Complejidad Media (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	152	161	151	153	159	155
5.000	1109	1096	1103	1032	1079	1084
10.000	1804	1239	1842	1804	1811	1700
50.000	6743	5793	5919	6348	6088	6178
100.000	14138	13385	13339	14358	13224	13689
500.000	168747	239497	166607	163543	161098	179898

Realizado Por: Pilar A. Quizhpe P.

Tabla 4.XIV. Tiempo de respuesta de SQL Server para consultas altas.

SQL Server - Complejidad Alta (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	381	314	310	300	325	326
5.000	2102	1622	1760	2002	2098	1916
10.000	2513	2273	2467	2654	2455	2472
50.000	7491	8531	7596	7333	7700	7730
100.000	19159	20019	20597	17048	17616	18887
450.000	179921	232233	147513	175924	212248	189568
500.000	El servidor no fue capaz de cumplir con el requerimiento					

Realizado Por: Pilar A. Quizhpe P.

Nota: SQL Server no pudo cumplir con la consulta de alta complejidad en las 500.000 filas, por lo que se tomo la muestra para un número de 450.000 filas en donde se observa un incremento drástico en el tiempo de respuesta para las consultas de alta complejidad.

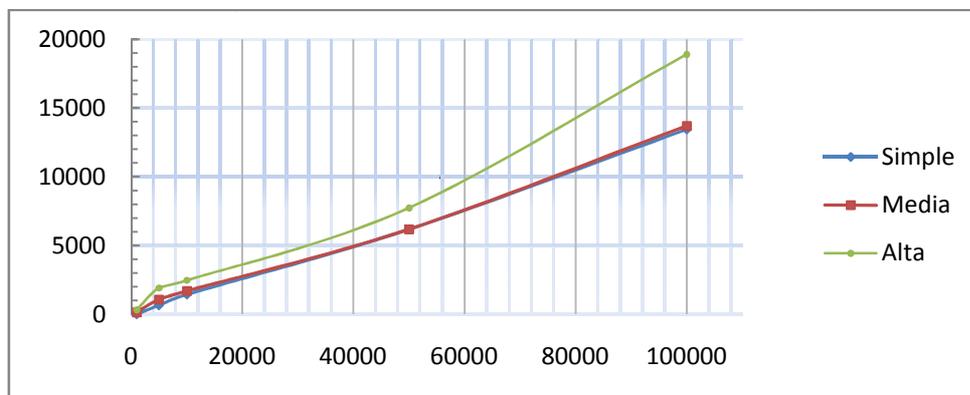
4.16.2 Comparativa de tiempos entre consultas de diversa complejidad SQL Server

Tabla 4.XV. Tiempo de ejecución de SQL Server para consultas de diversa complejidad

SQL Server - Consultas de diversa complejidad (milisegundos)			
Filas	Simple	Media	Alta
1.000	3	155	326
5.000	658	1084	1916
10.000	1435	1700	2472
50.000	6173	6178	7730
100.000	13449	13689	18887
500.000	175528	179898	-

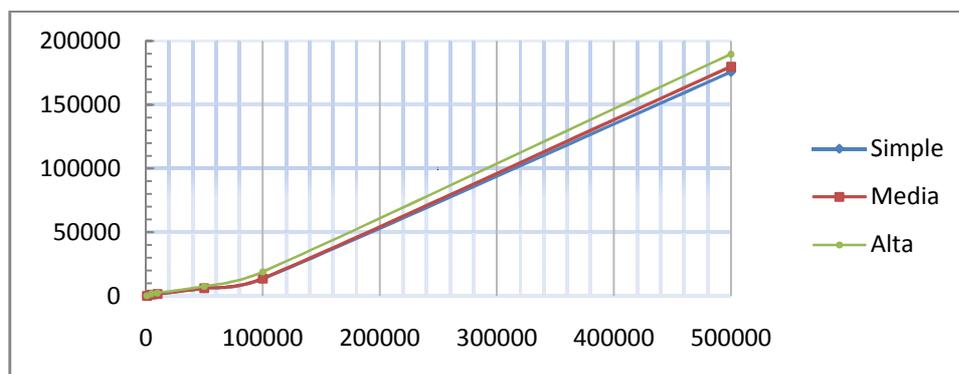
Realizado Por: Pilar A. Quizhpe P.

Gráfica Comparativa



Gráfica 4.2. Tiempo en consultas de diversa complejidad en SQL Server

Realizado Por: Pilar A. Quizhpe P.



Gráfica 4.3. Tiempos en consultas de diversa complejidad en SQL Server

Realizado Por: Pilar A. Quizhpe P.

Debido al incremento drástico del tiempo de respuesta para el intervalo de filas [100.000; 500.000], se analiza por partes todo el intervalo [1000, 500.000].

Observamos que los tiempos de respuesta de SQL Server tienen una cierta regularidad en la forma de producirse en el intervalo [1000,100000], esto nos permite sacar la conclusión de que las variaciones de la función o sea el tiempo de respuesta son proporcionales o casi proporcionales a los de la variable independiente o sea el número de filas, por lo que matemáticamente se puede admitir que dicha función es lineal y usar para estimar los valores, la interpolación lineal, a partir de la fórmula: $y - y_1 = \frac{y_2 - y_1}{x_2 - x_1}(x - x_1)$, de donde podemos calcular la velocidad de recuperación de los datos con la pendiente de la recta: $m = \frac{y_2 - y_1}{x_2 - x_1}$

Cálculo de la Velocidad de Recuperación de Datos de SQL Server, para consultas de complejidad simple:

Tomamos $P_1(1000,3)$ y $P_2(100000,13449)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{13499 - 3}{100000 - 1000} = \frac{13496}{99000}$$

$$\text{Tiempo de ejecución por fila } m = 0.14 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{0.14 \text{ ms}} = 7.34 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de SQL Server, para consultas de complejidad media:

Tomamos $P_1(1000,155)$ y $P_2(100000,13689)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{13689 - 155}{100000 - 1000} = \frac{13534}{99000}$$

$$\text{Tiempo de ejecución por fila } m = 0.14 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{0.14 \text{ ms}} = 7.31 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de SQL Server, para consultas de alta complejidad:

Tomamos $P_1(1000,326)$ y $P_2(100000,18887)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{18887 - 326}{100000 - 1000} = \frac{18561}{99000}$$

$$\text{Tiempo de ejecución por fila } m = 0.19 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{0.19 \text{ ms}} = 5.33 \frac{filas}{ms}$$

Para el intervalo [100000,500000], como las variaciones del tiempo de respuesta son proporcionales a los del número de filas, se utilizará de igual manera la función lineal para obtener la velocidad de respuesta de SQL Server.

Cálculo de la Velocidad de Recuperación de Datos de SQL Server, para consultas de complejidad simple:

Tomamos $P_1(100000, 13449)$ y $P_2(500000, 175528)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{175528 - 13449}{500000 - 100000} = \frac{162079}{400000}$$

$$\text{Tiempo de ejecución por fila } m = 0.41 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{0.41 \text{ ms}} = 2.47 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de SQL Server, para consultas de complejidad media:

Tomamos $P_1(100000, 13689)$ y $P_2(500000, 179898)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{179898 - 13689}{500000 - 100000} = \frac{166209}{400000}$$

$$\text{Tiempo de ejecución por fila } m = 0.42 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{0.42 \text{ ms}} = 2.41 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de SQL Server, para consultas de alta complejidad:

Tomamos $P_1(1000, 326)$ y $P_2(450000, 189568)$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{189568 - 326}{500000 - 1000} = \frac{189242}{499000}$$

$$\text{Tiempo de ejecución por fila } m = 0.37 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{8.64 \text{ ms}} = 2.67 \frac{filas}{ms}$$

Se observa que el aumento del tiempo de respuesta es inversamente proporcional a la velocidad de la consulta, además la variación del volumen de información incide directamente en el rendimiento de los SGBDs.

4.16.3 Resultados de la ejecución de Oracle

Tablas de Lectura de Tiempos

Tabla 4.XVI. Tiempo de respuesta de Oracle en consultas de complejidad simple.

Oracle - Complejidad Simple (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	60	70	60	60	70	64
5.000	400	540	360	460	480	448
10.000	670	730	760	710	710	716
50.000	2710	2480	2480	2500	2450	2524
100.000	4370	4410	4370	4390	4000	4308
500.000	529790	529480	529870	529100	529480	529444

Realizado Por: Pilar A. Quizhpe P.

Tabla 4.XVII. Tiempo de respuesta de Oracle en consultas de complejidad media

Oracle - Complejidad Media (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	290	290	140	310	150	236
5.000	640	780	570	640	320	590
10.000	1010	1120	960	1010	1000	1020
50.000	3530	3510	3480	3030	3460	3402
100.000	6310	6340	6120	6400	6310	6296
500.000	359750	359890	359920	359800	359750	359862

Realizado Por: Pilar A. Quizhpe P.

Tabla 4.XVIII. Tiempo de respuesta de Oracle en consultas de complejidad alta.

Oracle - Complejidad Alta (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	500	310	320	320	340	358
5.000	1230	900	870	870	930	960
10.000	1650	1530	1420	1480	1530	1522
50.000	5260	5590	5490	5590	5480	5482
100.000	9250	9870	10120	9520	9120	9576
500.000	1578420	1578310	1578490	1578360	1579120	1578540

Realizado Por: Pilar A. Quizhpe P.

El reloj interno de ORACLE reportó los tiempos tabulados en las tres tablas anteriores, sin embargo el resultado de las consultas estaba disponible cierto tiempo después, debido al tiempo de carga de las páginas Web en el cual se despliega la versión ORACLE

4.16.4 Resumen de Resultados para la operación consulta en ORACLE

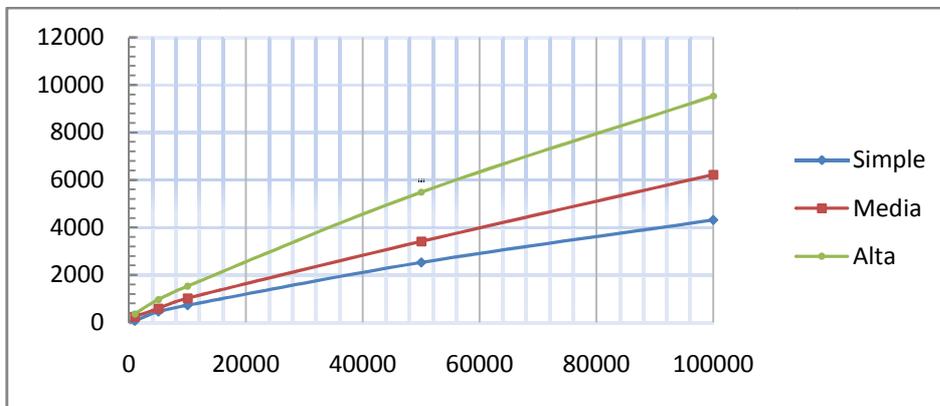
Tabla de Resultados

Tabla 4.XIX. Tiempo de respuesta de Oracle en consultas de diversa complejidad.

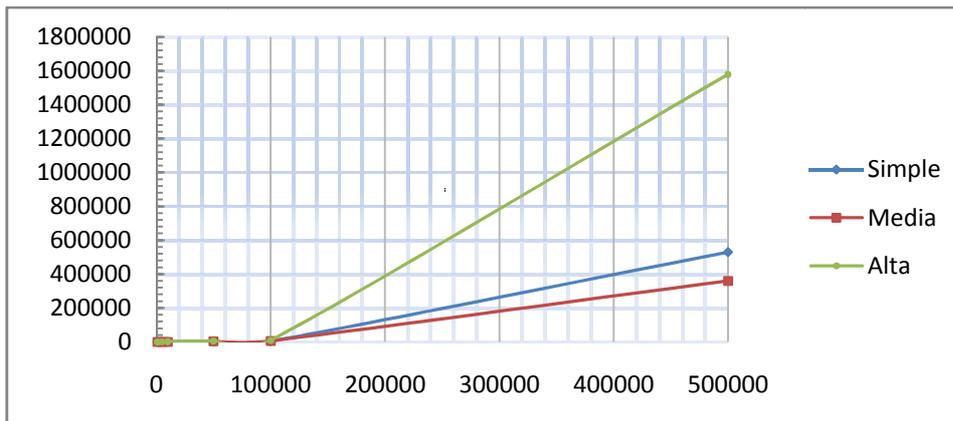
Oracle - Consultas de diversa complejidad (milisegundos)			
Filas	Simple	Media	Alta
1.000	64	236	358
5.000	448	590	960
10.000	716	1020	1522
50.000	2524	3402	5482
100.000	4308	6212	9576
500.000	529444	359862	1578540

Realizado Por: Pilar A. Quizhpe P.

Gráfica Comparativa



Grafica 4.4. Comparativa para consultas de diversa complejidad en Oracle
Realizado Por: Pilar A. Quizhpe P.



Grafica 4.5. Comparativa para consultas de diversa complejidad en Oracle
Realizado Por: Pilar A. Quizhpe P.

De la misma manera que SQL Server, Oracle sufre un incremento drástico en el tiempo de respuesta, desde las 100000 filas en adelante, por lo que en primer lugar se considera la gráfica en el intervalo [1000,100000], en donde se interpola las coordenadas con una función lineal, a través de la cual se determina la velocidad de respuesta de Oracle para ese intervalo específicamente.

Cálculo de la Velocidad de Recuperación de Datos de Oracle, para consultas de complejidad simple:

Tomamos $P_1(1000,64)$ y $P_2(100000,4308)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{4308 - 64}{100000 - 1000} = \frac{4244}{99000}$$

$$\text{Tiempo de ejecución por fila } m = 0.04 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{0.04 \text{ ms}} = 23.32 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de Oracle, para consultas de complejidad media:

Tomamos $P_1(1000,236)$ y $P_2(100000,6212)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{6212 - 236}{100000 - 1000} = \frac{5976}{99000}$$

$$\text{Tiempo de ejecución por fila } m = 0.06 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{0.06 \text{ ms}} = 16.57 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de Oracle, para consultas de alta complejidad:

Tomamos $P_1(1000,358)$ y $P_2(100000,9576)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{9576 - 358}{100000 - 1000} = \frac{9218}{99000}$$

$$\text{Tiempo de ejecución por fila } m = 0.09 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{0.09 \text{ ms}} = 10.74 \frac{filas}{ms}$$

Tomando en cuenta que en el intervalo [100000,500000], las variaciones del tiempo son proporcionales a la variación de las filas, se utiliza la función lineal para obtener la velocidad de respuesta de Oracle para las diversas consultas.

Cálculo de la Velocidad de Recuperación de Datos de Oracle, para consultas de complejidad simple:

Tomamos $P_1(100000, 4308)$ y $P_2(500000, 529444)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{529444 - 4308}{500000 - 100000} = \frac{588136}{400000}$$

$$\text{Tiempo de ejecución por fila } m = 1.47 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{1.47 \text{ ms}} = 0.68 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de Oracle, para consultas de complejidad media:

Tomamos $P_1(100000, 6212)$ y $P_2(500000, 359862)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{359862 - 6212}{500000 - 100000} = \frac{353650}{400000}$$

$$\text{Tiempo de ejecución por fila } m = 0.88 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{0.88 \text{ ms}} = 1.13 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de Oracle, para consultas de alta complejidad:

Tomamos $P_1(100000, 9576)$ y $P_2(500000, 1578540)$

$$m = \frac{y_2 - y_1}{x_2 - x_1} \quad m = \frac{1578540 - 9576}{500000 - 100000} = \frac{1568964}{400000}$$

$$\text{Tiempo de ejecución por fila } m = 3.92 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1 \text{ filas}}{3.92 \text{ ms}} = 0.25 \frac{filas}{ms}$$

4.16.5 Resultados de la ejecución de MySQL

Tablas de Lectura de Tiempos

A continuación se observan, los resultados obtenidos para la operación SELECT.

Tabla 4.XX. Tiempo de Ejecución de MySQL para consultas de complejidad simple

MySQL Server - Complejidad Simple (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	31	25	26	29	25	27
5.000	61	61	63	58	71	63
10.000	111	106	113	131	101	112
50.000	1792	1331	1935	1368	1289	1543
100.000	3383	3096	2813	3520	2700	3102
500.000	23095	21612	22026	21071	22079	21977

Realizado Por: Pilar A. Quizhpe P.

Tabla 4.XXI. Tiempo de Ejecución de MySQL para consultas de complejidad media

MySQL Server - Complejidad Media (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	30	27	33	31	30	30
5.000	74	86	73	78	76	77
10.000	166	170	201	206	173	183
50.000	1172	1241	1148	1164	1203	1186
100.000	1734	2076	1755	2413	1956	1987
500.000	11571	11534	11295	11574	11369	11469

Realizado Por: Pilar A. Quizhpe P.

Tabla 4.XXII. Tiempo de Ejecución de MySQL para consultas de complejidad alta

MySQL Server - Complejidad Alta (milisegundos)						
Filas	Lectura 1	Lectura 2	Lectura 3	Lectura 4	Lectura 5	Promedio
1.000	45	42	46	41	51	45
5.000	383	118	115	123	123	172
10.000	280	275	227	310	296	278
50.000	1238	1181	1737	1185	1240	1316
100.000	2976	2572	2605	3099	2585	2767
500.000	29865	16996	16012	14833	16380	18817

Realizado Por: Pilar A. Quizhpe P.

4.16.6 Resumen de Resultados para la operación consulta en MYSQL

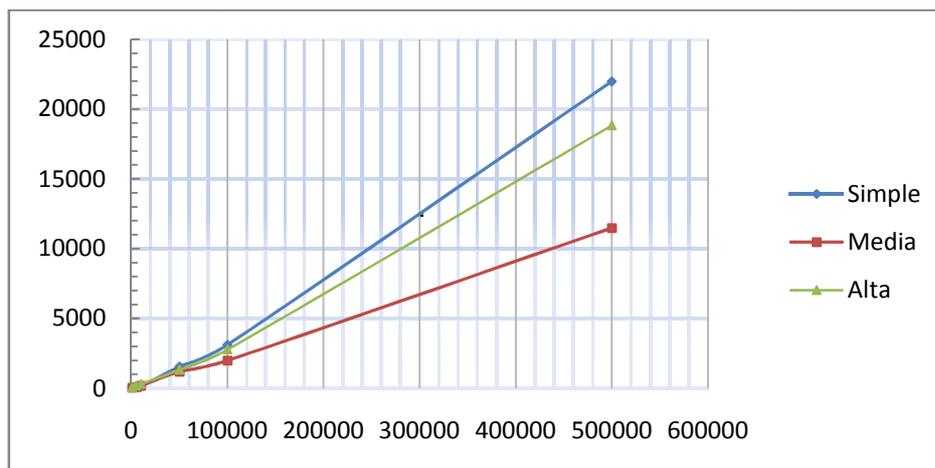
Tabla de Resultados

Tabla 4.XXIII. Tiempo de MySQL para consultas de diversa complejidad

MySQL Server - Consultas de diversa complejidad (milisegundos)			
Filas	Simple	Media	Alta
1.000	27	30	45
5.000	63	77	172
10.000	112	183	278
50.000	1543	1186	1316
100.000	3102	1987	2767
500.000	21977	11469	18817

Realizado Por: Pilar A. Quizhpe P.

Gráfica Comparativa



Grafica 4.6. Consultas de diversa complejidad en MySQL

Realizado Por: Pilar A. Quizhpe P.

Observamos que los tiempos de respuesta de MySQL tienen una cierta regularidad en la forma de producirse en todo el intervalo de pruebas, esto nos permite sacar la conclusión de que las variaciones de la función (tiempo) son proporcionales (o casi proporcionales) a los de la variable independiente (filas) en todo el intervalo, por lo que matemáticamente se procede a interpolar las coordenadas con una función lineal, a partir de la cual se calculan los valores requeridos.

Cálculo de la Velocidad de Recuperación de Datos de SQL Server, para consultas de complejidad simple:

Tomamos $P_1(1000, 27)$ y $P_2(500000, 21977)$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{21977 - 27}{500000 - 1000} = \frac{21950}{499000}$$

$$\text{Tiempo de ejecución por fila } m = 0.044 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1}{0.044} \frac{filas}{ms} = 22.73 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de SQL Server, para consultas de complejidad media:

Tomamos $P_1(1000, 30)$ y $P_2(500000, 11469)$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{11469 - 30}{500000 - 1000} = \frac{11439}{499000}$$

$$\text{Tiempo de ejecución por fila } m = 0.023 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1}{0.023} \frac{filas}{ms} = 43.62 \frac{filas}{ms}$$

Cálculo de la Velocidad de Recuperación de Datos de SQL Server, para consultas de alta complejidad:

Tomamos $P_1(1000, 45)$ y $P_2(450000, 18817)$

$$m = \frac{y_2 - y_1}{x_2 - x_1}$$

$$m = \frac{18817 - 45}{500000 - 1000} = \frac{18772}{499000}$$

$$\text{Tiempo de ejecución por fila } m = 0.038 \frac{ms}{filas}$$

$$\text{Velocidad de Consulta} = \frac{1}{0.038} \frac{filas}{ms} = 26.58 \frac{filas}{ms}$$

4.17 Comparativa entre los SGBD para consultas de diversa complejidad.

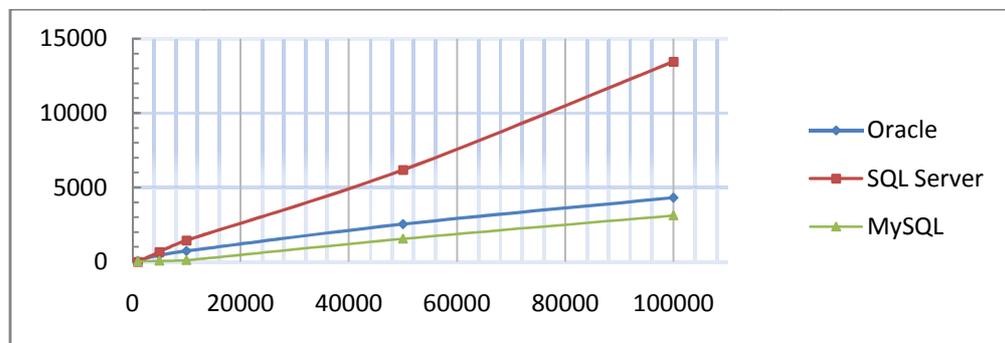
4.17.1 Resumen de Resultados para Consultas de Complejidad Simple

Tabla 4.XXIV. Tiempo de Ejecución de los SGBD para consultas simples.

Consultas de complejidad simple (milisegundos)			
Filas	Oracle	SQL Server	MySQL
1.000	64	3	27
5.000	448	658	63
10.000	716	1435	112
50.000	2524	6173	1543
100.000	4308	13449	3102
500.000	529444	175528	21977

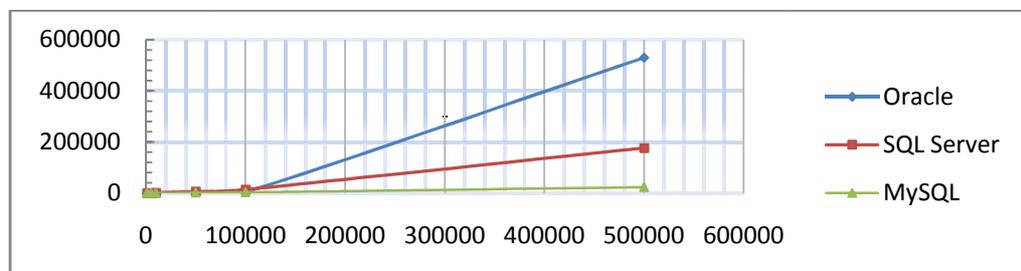
Realizado Por: Pilar A. Quizhpe P.

Gráfica Comparativa para Complejidad Simple



Gráfica 4.7. Consultas de complejidad simple en bases de datos pequeñas

Realizado Por: Pilar A. Quizhpe P.



Gráfica 4.8. Consultas de complejidad simple en bases de datos grandes

Realizado Por: Pilar A. Quizhpe P.

MySQL ofrece mejor rendimiento, para las consultas de complejidad simple, al tener el menor tiempo de respuesta en la ejecución de las pruebas. Oracle, mantiene un excelente desempeño después de MySQL, pero sufre una drástica caída del rendimiento en la extracción de 100000 filas en adelante, lo que se observa en el aumento del tiempo de respuesta. En primera instancia SQL Server tiene un rendimiento bajo, en relación a Oracle y MySQL, pero la caída del rendimiento de Oracle, hace que este se posicione como el segundo mejor en rendimiento para las consultas de complejidad simple.

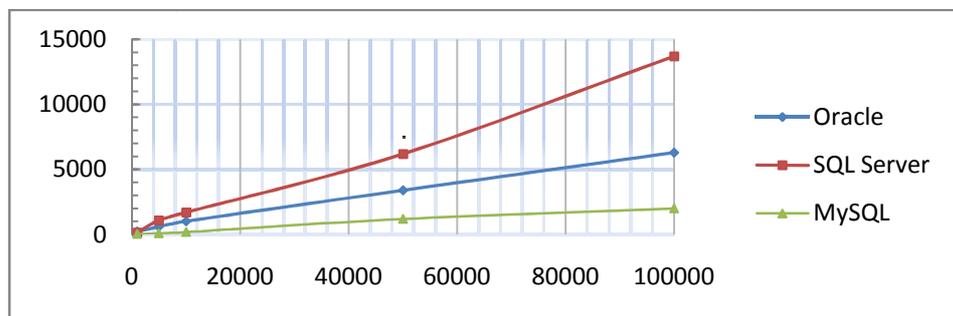
4.17.2 Resumen de Resultados para Consultas de Complejidad Media

Tabla 4.XXV. Tiempo de Ejecución de los SGBD para consultas complejidad media

Consultas de complejidad media (milisegundos)			
Filas	Oracle	SQL Server	MySQL
1.000	236	155	30
5.000	590	1084	77
10.000	1020	1700	183
50.000	3402	6178	1186
100.000	6296	13689	1987
500.000	359862	179898	11469

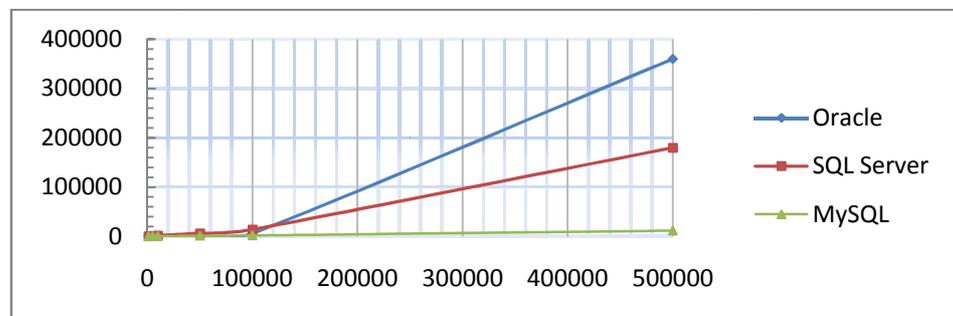
Realizado Por: Pilar A. Quizhpe P.

Gráfica Comparativa para Complejidad Media



Gráfica 4.9. Consultas de complejidad media en bases de datos pequeñas

Realizado Por: Pilar A. Quizhpe P.



Gráfica 4.10. Consultas de complejidad media en bases de datos grandes

Realizado Por: Pilar A. Quizhpe P.

Para las consultas de complejidad media se observa como nuevamente MySQL ofrece el mejor desempeño, seguido por Oracle en una primera etapa limitada hasta la extracción de las 100000 filas, en donde empieza el decrecimiento drástico de su desempeño. SQL server en primera instancia, se muestra como el SGBD de más bajo desempeño, en las consultas de complejidad media, sin embargo se reincorpora, como el segundo mejor al ofrecer un mejor desempeño que Oracle en la extracción de 100000 filas en adelante.

4.17.3 Resumen de Resultados para Consultas de Complejidad Alta

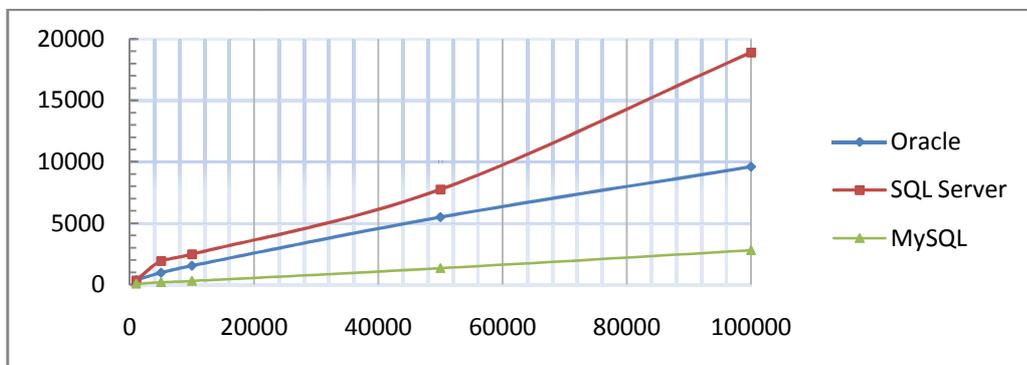
Tabla de Resumen de Tiempos

Tabla 4.XXVI. Tiempo de Ejecución de los SGBD para consultas complejidad alta

Consultas de complejidad alta (milisegundos)			
Filas	Oracle	SQL Server	MySQL
1.000	358	326	45
5.000	960	1916	172
10.000	1522	2472	278
50.000	5482	7730	1316
100.000	9576	18887	2767
500.000	1578540	-	18817

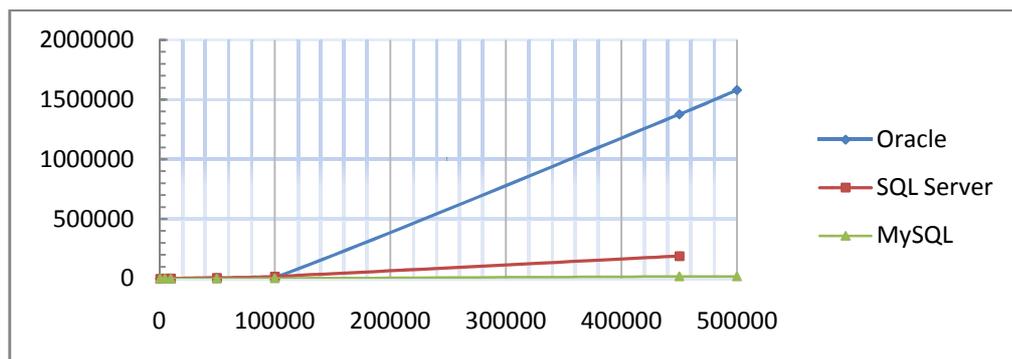
Realizado Por: Pilar A. Quizhpe P.

Gráfica Comparativa para Complejidad Alta



Grafica 4.11. Consultas de complejidad alta en bases de datos pequeñas.

Realizado Por: Pilar A. Quizhpe P.



Grafica 4.12. Consultas de complejidad alta en bases de datos grandes.

Realizado Por: Pilar A. Quizhpe P.

En cuanto a consultas de alta complejidad, MySQL ofrece el mejor desempeño, seguido por SQL Server y por último Oracle, el mismo que sufre una gran caída en su desempeño, a partir de la extracción de las 100.000 filas en adelante. SQL Server no fue capaz de cumplir con la prueba para las 500.000 filas.

4.18 Modelo de Puntuación

Se procede a calificar con una puntuación de 10/10, los promedios generales de cada escenario en los que se realiza la prueba, mediante una regla de proporcionalidad, comúnmente conocida como regla del tres. De esta manera el mejor promedio obtendrá 10, mientras que las demás calificaciones seguirán bajando en la misma proporción que bajen los promedios generales.

4.19 Resultados Generales de la medición

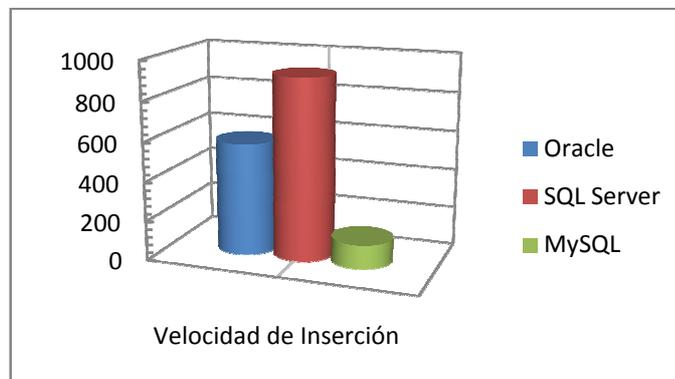
4.19.1 Comparativa de las Velocidades de Inserción entre los SGBD

Se analiza el desempeño de cada Sistema de Gestión de bases de Datos en la escritura intensiva de datos.

Tabla 4.XXVII. Velocidad de Inserción de los SGBD

Velocidad de Inserción			
	Oracle	SQL Server	MySQL
Velocidad	0.57	0.92	0.12
Calificación	6.1	10	1.3

Realizado Por: Pilar A. Quizhpe P.



Gráfica 4.13. Comparativa de Velocidad de Inserción

Realizado Por: Pilar A. Quizhpe P.

En donde se observa que SQL Server se posiciona como el mejor al ofrecer el menor tiempo de respuesta en la ejecución de la operación INSERT y con una velocidad de inserción de 0.92 filas insertadas por milisegundo.

SQL Server es alrededor de 1.61 veces más rápido que Oracle y 7.67 veces más rápido que MySQL; por otra parte Oracle es 4.75 veces más rápido que MySQL. Por lo tanto se puede recomendar a SQL Server apto para aplicaciones o entornos en

donde exista alta demanda de inserciones, pues es el que mejor trabajará seguido de Oracle. Definitivamente MySQL no cumple con el desempeño deseado, para aplicaciones en donde exista alta demanda de escritura de datos.

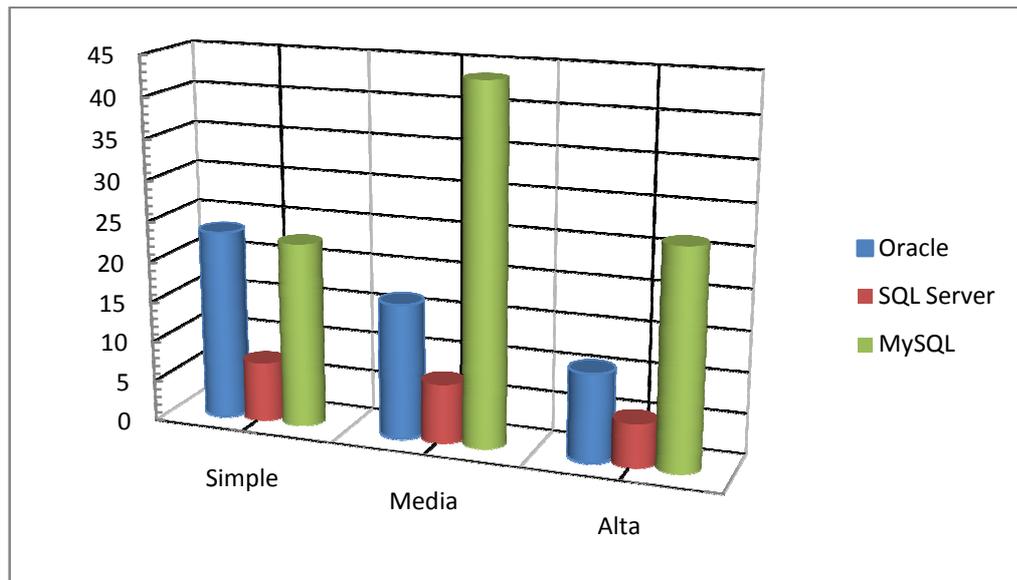
4.19.2 Comparativa Velocidad de Recuperación de Datos

Rendimiento en Bases de Datos con bajo a normal volumen de información.

Tabla 4.XXVIII. Velocidad de recuperación - Volumen de información bajo a normal

Velocidad de Recuperación de Datos			
Complejidad	Oracle	SQL Server	MySQL
Simple	23.32	7.34	22.73
Media	16.57	7.31	43.62
Alta	10.74	5.33	26.58
Promedio de Velocidad	16.87	6.66	30.98
Calificación	5.4	2.14	10

Realizado Por: Pilar A. Quizhpe P.



Grafica 4.14. Comparativa de Velocidad de Respuesta de Datos

Realizado Por: Pilar A. Quizhpe P.

Se analiza cada Sistema de Gestión de Bases de Datos, con volúmenes de información que se clasifican como bajo a normal (en este caso hasta 100000 filas por tabla). En donde se observa que MySQL es una base de datos muy rápida en la lectura, mostrando el mejor rendimiento entre los demás SGBD, para cualquier tipo de consultas.

En segundo lugar se encuentra Oracle ofreciendo una rápida respuesta en la extracción intensiva de datos tanto en consultas simples, medias y de alta

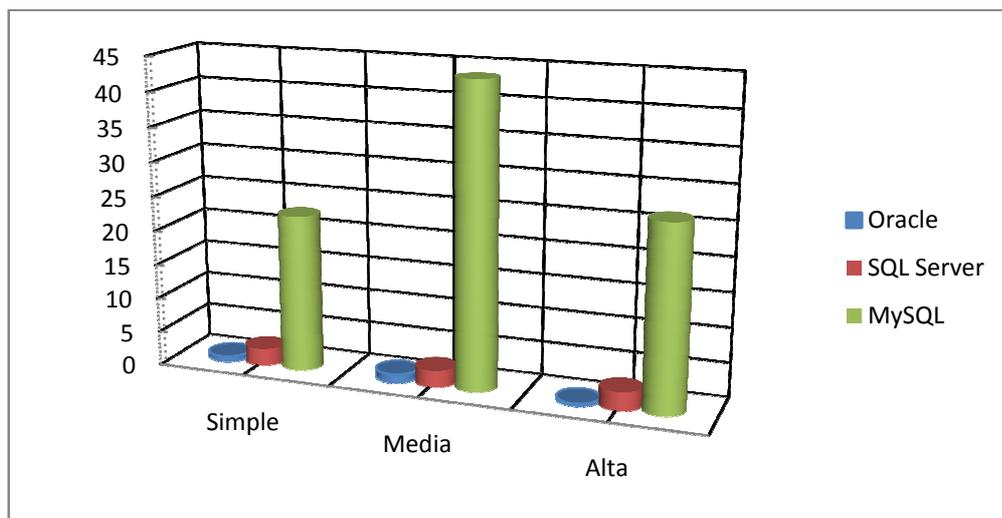
complejidad. En último lugar tenemos a SQL Server, que prácticamente no sería muy recomendable para aplicaciones o entornos en donde se requiera extracciones intensivas de datos, al no proveer la velocidad de respuesta necesaria que ameritan este tipo de aplicaciones. En las consultas de complejidad simple y con volúmenes de información de bajo a normal, MySQL es alrededor de 0.97 veces más rápido que Oracle y alrededor de 3.1 veces más rápido que SQL Server. Por otra parte Oracle es alrededor de 3.17 veces más rápido que SQL Server. Para consultas de complejidad media MySQL es 5.97 veces más rápido que SQL Server y alrededor de 2.61 veces, más rápido que Oracle. Mientras que Oracle es 3.17 veces más rápido que SQL Server. Por último en consultas de complejidad media MySQL es alrededor de 2.47 veces más rápido que Oracle y 4.99 veces más rápido que SQL Server, mientras que Oracle es alrededor de 2.02 veces más rápido que SQL Server.

Rendimiento en Bases de Datos con gran volumen de información.

Tabla 4.XXIX. Velocidad de recuperación - Volumen de información alto

Velocidad de Recuperación de Datos			
Complejidad	Oracle	SQL Server	MySQL
Simple	0.68	2.47	22.73
Media	1.13	2.41	43.62
Alta	0.25	2.67	26.58
Promedio de Velocidad	0.67	2.52	30.98
Calificación	0.2	0.8	10

Realizado Por: Pilar A. Quizhpe P.



Grafica 4.15. Comparativa de Velocidad de Respuesta de Datos

Realizado Por: Pilar A. Quizhpe P.

Un fenómeno extraño, es el comportamiento de Oracle, el mismo que al ejecutar las pruebas con gran volumen de información sufre un descenso drástico de su desempeño o sea de su velocidad de respuesta, misma que disminuye de 23.32 filas extraídas cada milisegundo a 0.68 filas extraídas cada milisegundo en las consultas de complejidad simple; de 16.57 a 1.13 filas por segundo para consultas de complejidad media, y de 10.74 a 0.25 filas por segundo. De igual manera SQL Server que de 7.34 filas extraídas cada milisegundo cae a 2.47 filas extraídas cada milisegundo. Sin embargo no es la misma caída en comparación al descenso sufrido por Oracle.

Es interesante como MySQL se mantiene con un excelente desempeño para bases de datos con gran volumen de información. Siendo 33.42 veces más rápido que Oracle, y alrededor de 9.20 veces más rápido que SQL Server, para consultas de complejidad simple. Para consultas de complejidad media 38.60 veces más rápido que Oracle y 18.1 veces más rápido que SQL Server. De igual manera, en consultas de alta complejidad, es 106.32 veces más rápido que Oracle y 9.95 veces más rápido que SQL Server. En definitiva MySQL es el mejor SGBD para realizar consultas de datos.

De manera general, en base al proceso de puntuación se determina que: MySQL posee una eficiencia 46% mejor que Oracle y 78.60% mejor que SQL Server, para consultas de diversa complejidad, en bases de datos con volumen de información normal. MySQL posee una eficiencia 98% mejor que Oracle y 92% mejor que SQL Server para consultas de diversa complejidad, en bases de datos con gran volumen de información. SQL Server posee una eficiencia 39% mejor que Oracle y 87% mejor que MySQL para operaciones de inserción.

Como se observa, el análisis permite al analista decidir dónde concentrar más esfuerzos tanto en el proceso de planeación como de configuración de los SGBD, por ejemplo Oracle responde muy bien para bases de datos con volumen de información bajo a normal, pero es necesario que exista un tuning exhaustivo del SGBD, que permita mejorar el performance o desempeño del sistema, en las bases de datos con volúmenes de información mayor.

4.20 Comparativa General

A continuación se determina el mejor SGBD basado en funcionalidad, soporte técnico, costos, herramientas de administración y rendimiento, de acuerdo a las calificaciones obtenidas en las siguientes categorías.

Tabla 4.XXX. Comparativa General

Comparativa General			
	Oracle	SQL Server	MySQL
Características Primarias	9.17	6.14	8.38
Dimensionamiento	8.25	6.37	7.38
Costos	6.00	8.00	10.00
Soporte Técnico	10.00	10.00	10.00
Herramientas de Administración	8.50	8.50	8.50
Rendimiento en Inserción	6.10	10.00	1.30
Rendimiento en Recuperación Bases de Datos Pequeñas	5.40	2.14	10.00
Rendimiento en Recuperación Bases de Datos Grandes	0.20	0.80	10.00
Promedio General	5.96	6.49	8.19

Realizado Por: Pilar A. Quizhpe P.

De acuerdo a los resultados MySQL se consolida como el Sistema de gestión de Bases de Datos, con buenas características y con un excelente rendimiento para aplicaciones intensivas en consulta de datos.

CAPÍTULO V

5. DISEÑO DE LA ARQUITECTURA DE DATOS MEDIANTE EL ANALISIS DE REQUISITOS.

5.1 Introducción

La arquitectura de datos identifica y define las mejores clases de datos que apoyan las funciones del negocio, definidas en el modelo de negocios. Es la primera de las arquitecturas a ser definidas, porque la calidad de los datos es el producto básico de la función de los sistemas de información.

La arquitectura de datos consiste de entidades de datos, cada una de las cuales tiene atributos y relaciones con otras entidades de datos. La arquitectura de datos, se crea en relación al modelo de datos, mismo que permite describir los elementos de la realidad que intervienen en un problema dado y la forma en que se relacionan esos elementos entre sí.

Una vez que se ha determinado el SGBD de mejor rendimiento para la aplicación de la empresa “**GuíaMundial.com**”, se procede a realizar el correspondiente análisis de requerimientos de la base de datos, a partir del cual se diseñará el modelo arquitectónico y la implementación de la base de datos en el Sistema Gestor de Base de Datos.

5.2 Especificación de requerimientos para la Base de Datos

5.2.1 Introducción

A. Propósito General

Internet se ha convertido en "un instrumento fundamental" en todo el mundo tanto en la búsqueda como en la reserva de servicios en diferentes países del mundo. Por ejemplo, según lo confirma un estudio encargado por NH, el 51% de usuarios utiliza la Red para obtener información sobre reservas en un hotel, y la cifra se eleva al 73% en el caso de los viajeros de empresas. El estudio recuerda que Internet "simplifica" las tareas de cara a "buscar las mejores ofertas, tomar decisiones basadas en información actualizada y reducir las posibilidades de que el servicio responda a las expectativas generadas".

Actualmente, las empresas publicitan sus servicios en la web, sabiendo, que su público objetivo ya no está limitado, sino que se ha expandido a nivel mundial. En este contexto la empresa "**GuíaMundial.com**", ofrece paquetes publicitarios en la web, para la promoción de las empresas que contraten sus servicios, y al mismo tiempo convirtiéndose en una guía de servicios para los internautas de la web.

B. Alcance

Se implementará una base de datos, a partir de su modelo arquitectónico determinado en base al análisis de los requerimientos del usuario de la aplicación.

La base de datos manejará todas las entidades, atributos y relaciones, que permitan la gestión de los perfiles de la aplicación. En este caso, el perfil para el usuario final, perfil para administración, y un perfil para la empresa contratante.

C. Objetivos de la base de Datos

- Eliminar la actual redundancia e inconsistencia de los datos, a través de la reestructuración del modelo de datos.
- Optimizar el desempeño del sistema en el acceso a los datos.

5.2.2 Sistema Actual

Actualmente, el sistema se encuentra implementado con un manejador de contenidos y páginas estáticas. El principal inconveniente del sistema actual, es que no existe un modelo de datos formal, lo que ha repercutido, en redundancia de la información y bajo desempeño de la aplicación en general.

5.2.3 Sistema Propuesto.

A. Visión General

El objetivo principal de la aplicación es ofrecer paquetes publicitarios, a través de los cuales se definirá la manera en que las empresas contratantes del servicio serán promocionadas en el sitio web. Cada empresa será ubicada en base a su sitio dentro del globo terráqueo, es decir, por continente, país, y ciudad, y por su categorización dentro de la base de datos.

En definitiva, en cuanto a operaciones el sistema propuesto varía al dejar de ser un sitio web de páginas estáticas a un sitio de páginas generadas dinámicamente, por lo que se hará un enfoque especial en el diseño de una arquitectura de datos formal, que permita gestionar y optimizar el almacenamiento de la información y por consiguiente mayor eficiencia en el desempeño de la aplicación en general.

5.2.4 Diseño de la Arquitectura de Datos

A. Diseño Conceptual

Descripción del Proyecto.

La empresa publicitaria “**GuíaMundial.com**”, desea obtener el diseño de la arquitectura de datos con el objeto de establecer una sólida plataforma para el posterior desarrollo de una aplicación web, a través del cual pueda ofertar paquetes publicitarios en la web, y convertirse en una guía de servicios ON - LINE.

B. Requisitos Funcionales

La base de datos deberá manejar información relativa a los módulos de administración que manejará la aplicación, en este caso se han determinado tres módulos o perfiles de usuario los cuales manejan la información de la siguiente manera:

Módulo de Administración

El modulo de administración manejará todo lo información relacionada a la administración de usuarios, teniendo usuarios administrativos que gestionan la aplicación web, y usuarios especiales que puedan ingresar y actualizar información correspondiente a sus empresas.

Además manejará todas las operaciones relativas a inserción, actualización, eliminación y selección de información correspondiente a: continentes, países, ciudades, empresas, categorías, paquetes, banners, pagos diarios, cumpliendo las siguientes restricciones.

1. Un continente tendrá varios países, y un país pertenecerá a un solo continente.
2. Cada continente se almacenará con un código que lo identifique, su nombre, una descripción que lo caracterice, y adicionalmente un campo de observaciones, para el administrador.
3. Un país tiene varias ciudades, y una ciudad le pertenece a un único país.
4. Cada país, será almacenado con su correspondiente código que lo identifique, su nombre, una descripción general del país, y por ultimo un campo de observaciones para el país.
5. Una ciudad tiene varias empresas y una empresa puede localizarse en varias ciudades.
6. Cada ciudad se almacenará con su correspondiente código que la identifique, su nombre, una descripción general del país, y por ultimo un campo de observaciones para la ciudad.

7. El sistema debe clasificar a las empresas por su categoría, además de su ubicación.
8. Cada empresa se almacenará con su correspondiente código que la identifique, su razón social, descripción de la empresa, observaciones del administrador para la empresa, teléfono, fax, código de inscripción, fecha de registro, entidad donde la empresa está registrada, representante legal de la empresa, y por ultimo un campo para almacenar la imagen del logo.
9. Una categoría servirá como criterio de clasificación de la empresa.
10. Una empresa puede ser clasificada en varias categorías y una categoría puede tener asociada a ella varias empresas.
11. Una categoría se almacenara con un código que la identifique, el nombre y la descripción de la categoría.
12. Una empresa establecida en una ciudad específica puede comprar varios paquetes, y un paquete puede ser adquirido por varias empresas.
13. Un paquete publicitario consta de un código, denominación del paquete, descripción del paquete, duración, y costo.

Módulo de Usuario

El modulo del usuario manejará operaciones relativas a búsqueda de la información de su interés como por ejemplo a: continentes, países, ciudades, empresas, categorías, cumpliendo las siguientes restricciones.

1. El usuario puede acceder a la información de los continentes, países, ciudades, empresas, categorías, paquetes publicitarios. Sin embargo no puede visualizar información, sobre la adquisición de paquetes.
2. Si al usuario le interesa contratar algún tipo de paquete publicitario, deberá darse de alta como usuario especial, para poder realizar la compra de un paquete.
3. Si el usuario especial decide comprar un paquete, deberá dar de alta la información concerniente a su empresa, indicando el tipo deseado de paquete, posteriormente, se le dará la opción de pagar a través de tarjeta de

crédito, por lo cual será necesario que ingrese los datos de su tarjeta de crédito.

4. Un usuario especial tendrá la capacidad de actualizar los datos correspondientes a su empresa.

C. Entidades, relaciones y diagramas E - R

Se procede a identificar y definir las mejores clases de datos o entidades, que apoyan las funciones del negocio, en base a los requerimientos funcionales, definidos anteriormente.

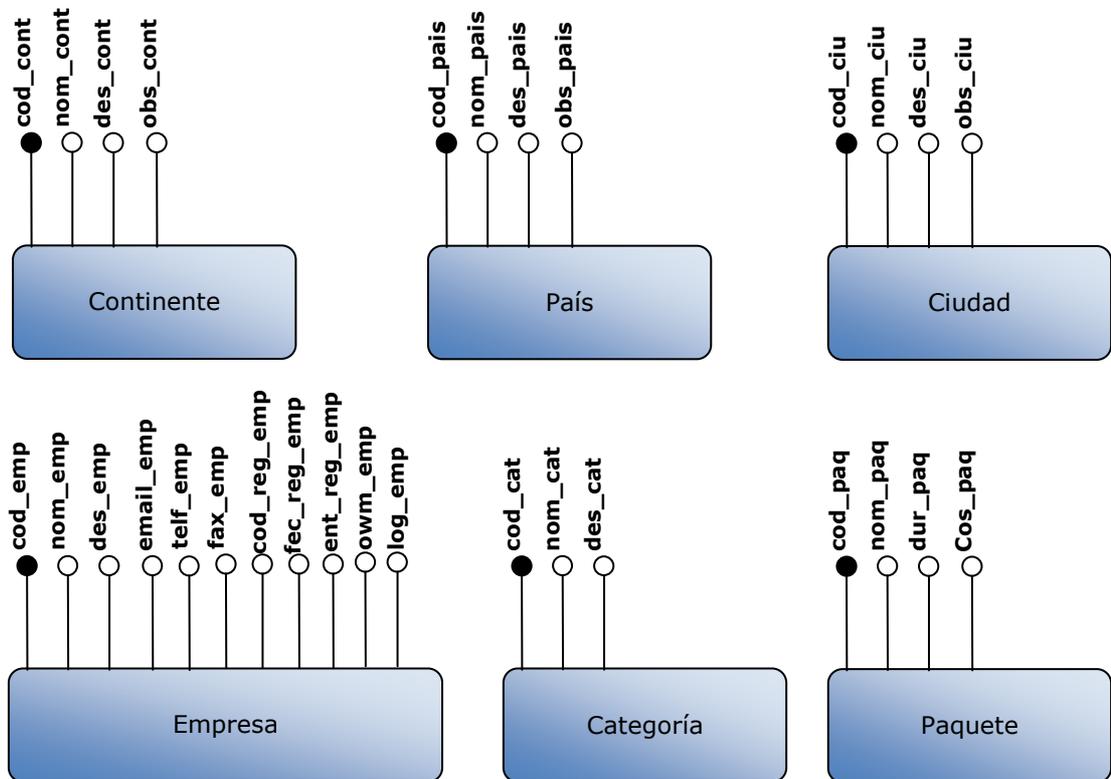


Figura 5.1. Entidades

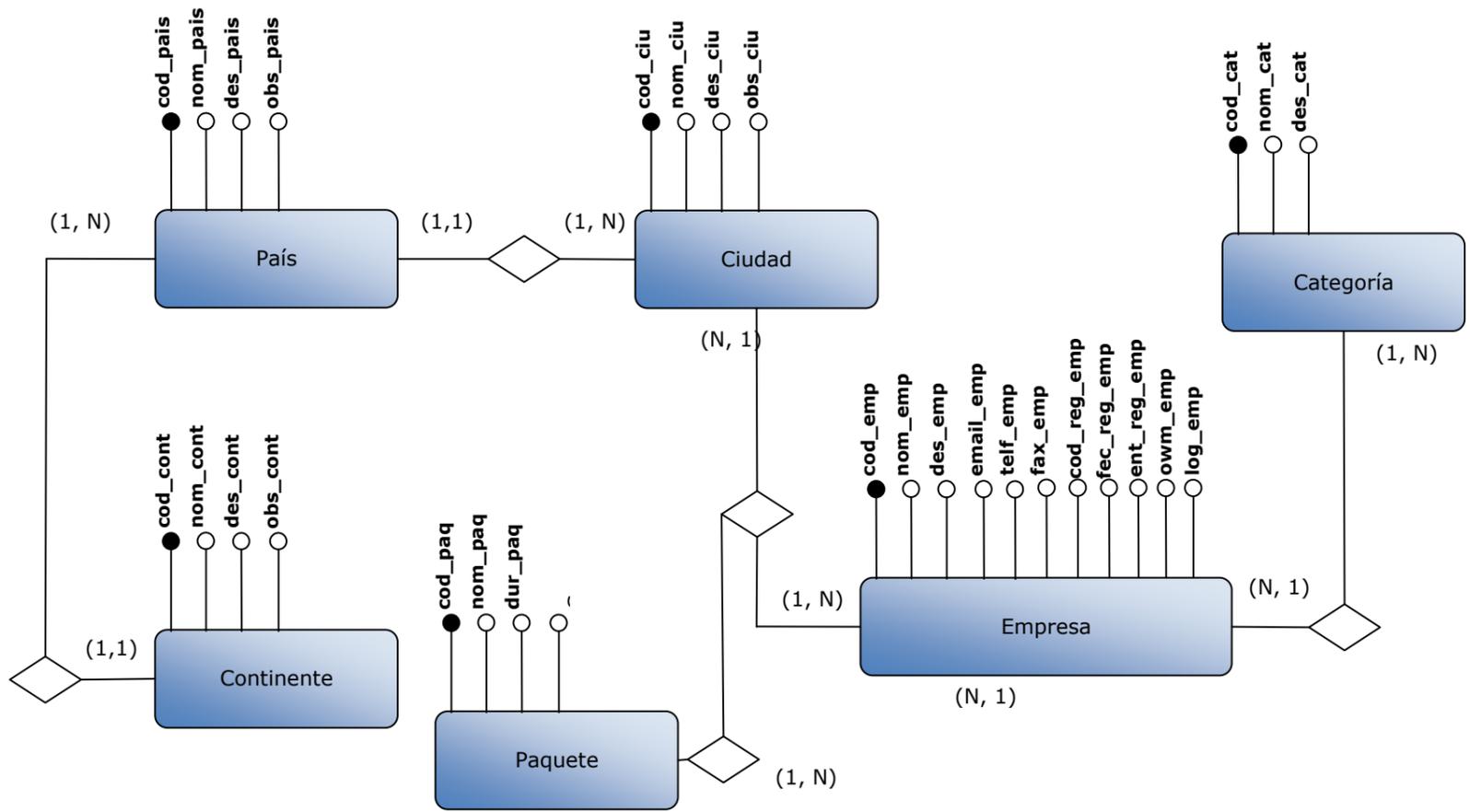


Figura 5.2. Modelo Entidad - Relación

5.2.5 Diseño Lógico

A. Entidades

Tabla 5.I. Entidad Continente

Gestión de Continentes		
Entidad:	Continente	
Atributos	Tipo de Dato	Descripción
COD_CONT	INTEGER	Código de identificación del continente dentro de la aplicación.
NOM_CONT	VARCHAR(10)	Nombre del Continente
DES_CONT	VARCHAR(200)	Descripción General para el Continente mencionado
OBS_CONT	VARCHAR(200)	Observaciones del administrador, respecto al continente

Tabla 5.II. Entidad País

Gestión de Países		
Entidad:	País	
Atributos	Tipo de Dato	Descripción
COD_PAIS	INTEGER	Código de identificación del país dentro de la aplicación.
NOM_PAIS	VARCHAR(10)	Nombre del país
DES_PAIS	VARCHAR(200)	Descripción General del país mencionado
OBS_PAIS	VARCHAR(200)	Observaciones del administrador, respecto al país.

Tabla 5.III. Entidad Ciudad

Gestión de Ciudades		
Entidad:	Ciudad	
Atributos	Tipo de Dato	Descripción
COD_CIU	INTEGER	Código de identificación de la ciudad dentro de la aplicación.
NOM_CIU	VARCHAR(10)	Nombre de la ciudad
DES_CIU	VARCHAR(200)	Descripción General de la ciudad mencionada
OBS_CIU	VARCHAR(200)	Observaciones del administrador, respecto a la ciudad

Tabla 5.IV. Entidad Empresa

Entidad:	Empresa	
Atributos	Tipo de Dato	Descripción
COD_EMP	INTEGER	Código de identificación de la empresa dentro de la aplicación.
NOM_EMP	VARCHAR(10)	Nombre de la empresa
DES_EMP	VARCHAR(200)	Descripción General de la empresa mencionada
EMAIL_EMP	VARCHAR(20)	Observaciones del administrador, respecto a la empresa
TELF_EMP	VARCHAR(10)	Teléfono de la Empresa
FAX_EMP	VARCHAR(10)	Número de Fax de la Empresa
COD_REG_EMP	VARCHAR(15)	Código de Inscripción del registro Mercantil
FEC_REG_EMP	DATE	Fecha de Registro
ENT_REG_EMP	VARCHAR(20)	Entidad en donde la Empresa está registrada.
OWN_EMP	VARCHAR(20)	Representante legal de la Empresa
LOG_EMP	BLOB	Imagen del Logo de la Empresa

Tabla 5.V. Entidad Categoría

Gestión de Categorías		
Entidad:	Categoría	
Atributos	Tipo de Dato	Descripción
COD_CAT	INTEGER	Código de identificación de la Categoría dentro de la aplicación.
NOM_CAT	VARCHAR(10)	Nombre de la Categoría, a través de la cual
DES_CAT	VARCHAR(200)	Descripción General de la Categoría mencionada

Tabla 5.VI. Entidad Paquete

Entidad:	Paquete	
Atributos	Tipo de Dato	Descripción
COD_PAQ	INTEGER	Código de identificación de la ciudad dentro de la aplicación.
NOM_PAQ	VARCHAR(10)	Denominación del paquete.
DET_PAQ	VARCHAR(200)	Detalle del paquete
DUR_PAQ	VARCHAR(10)	Especifica el tiempo de duración del paquete
COSTO_PAQ	NUMERIC	Precio de la tarifa establecida para el paquete

A. Modelo Lógico

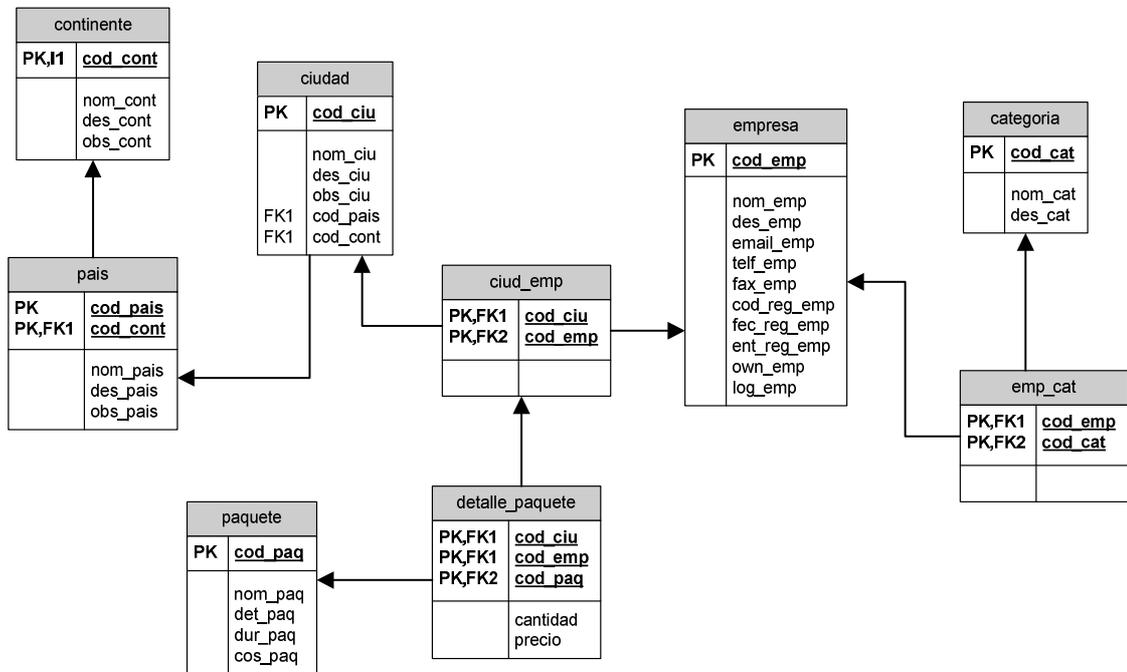


Figura 5.3. Modelo relacional

5.2.6 Implementación de la Base de Datos relacional

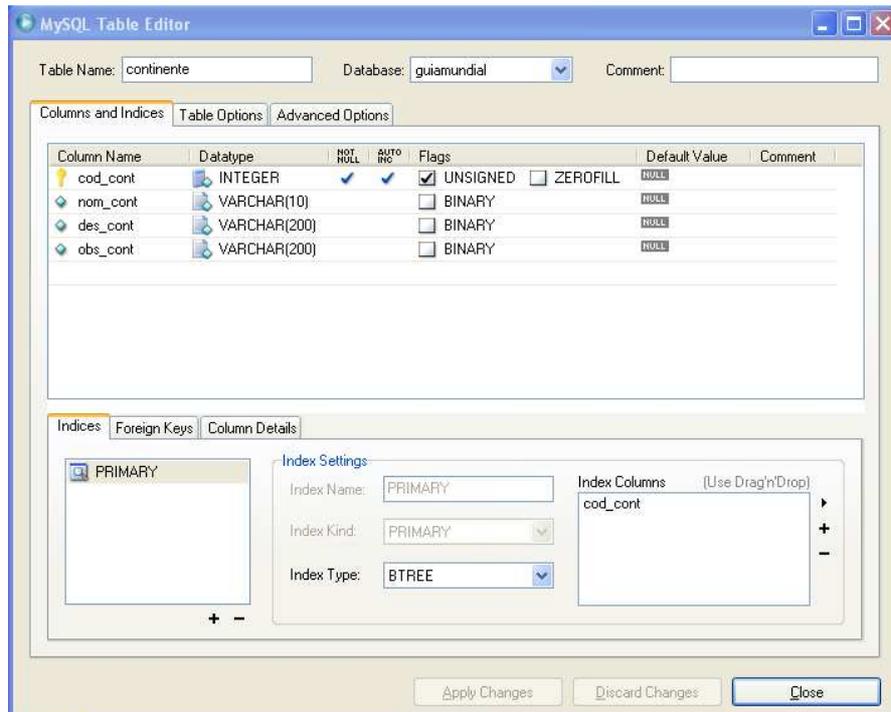


Figura 5.4. Tabla Continente

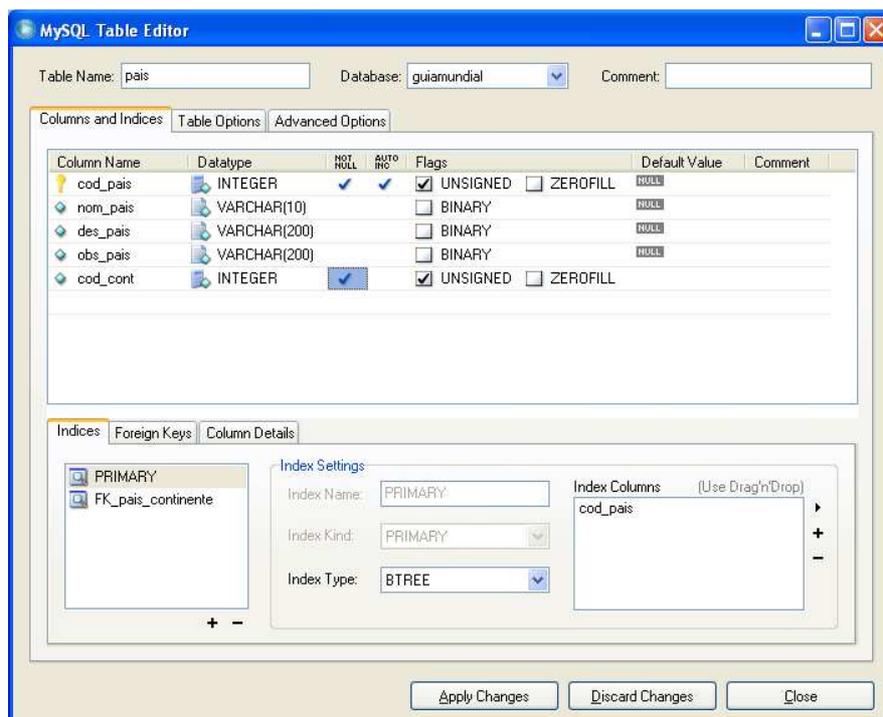


Figura 5.5. Tabla País

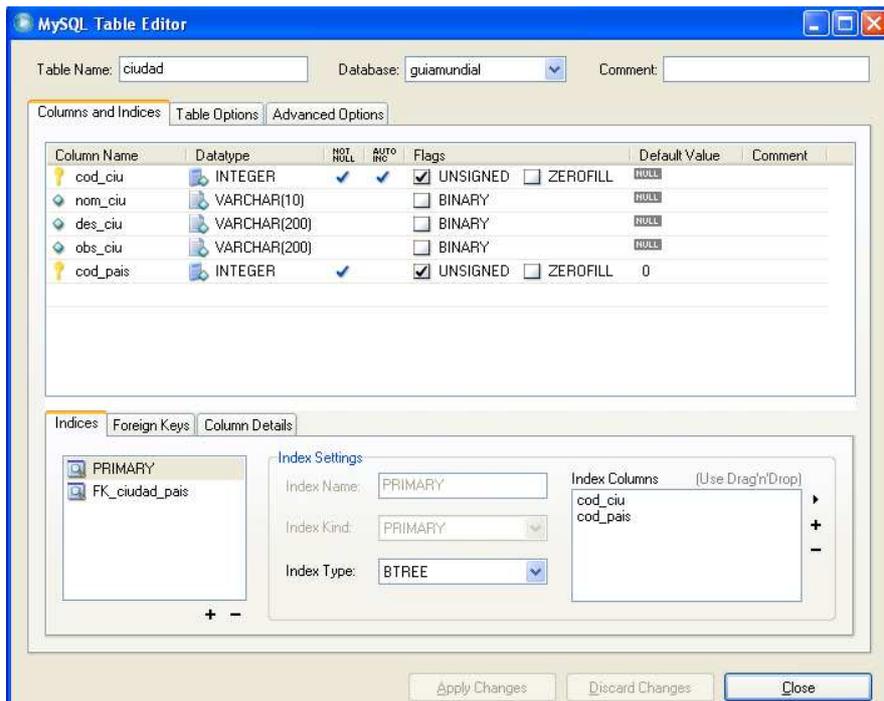


Figura 5.6. Tabla Ciudad

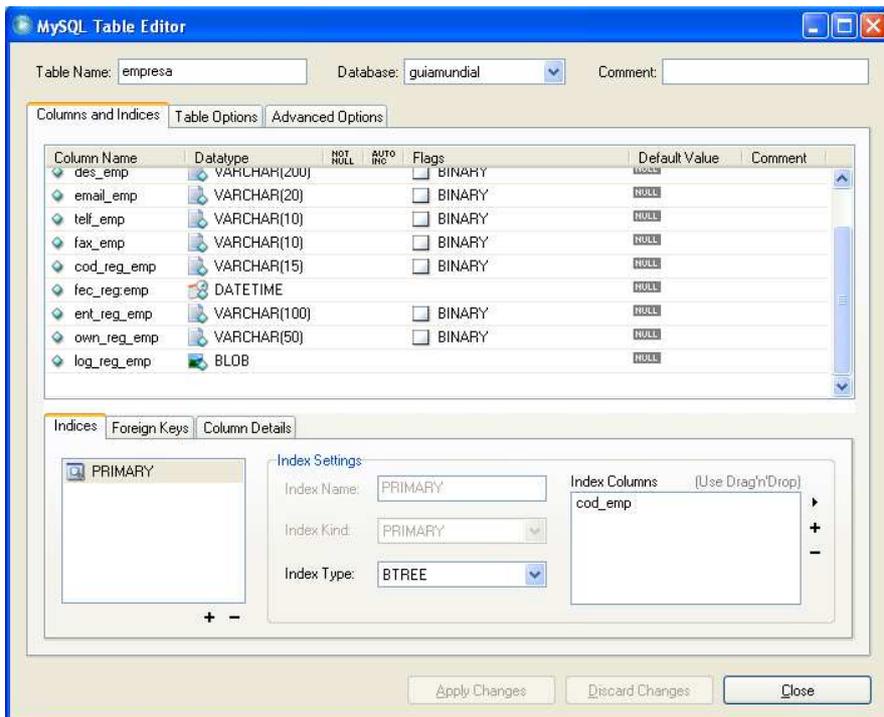


Figura 5.7. Tabla Empresa

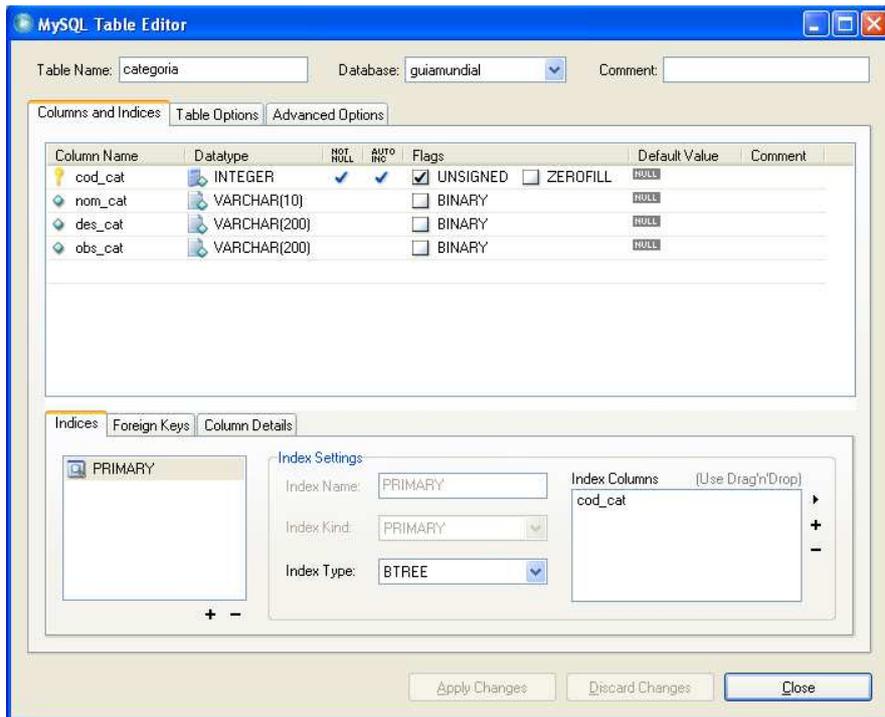


Figura 5.8. Tabla Categoría

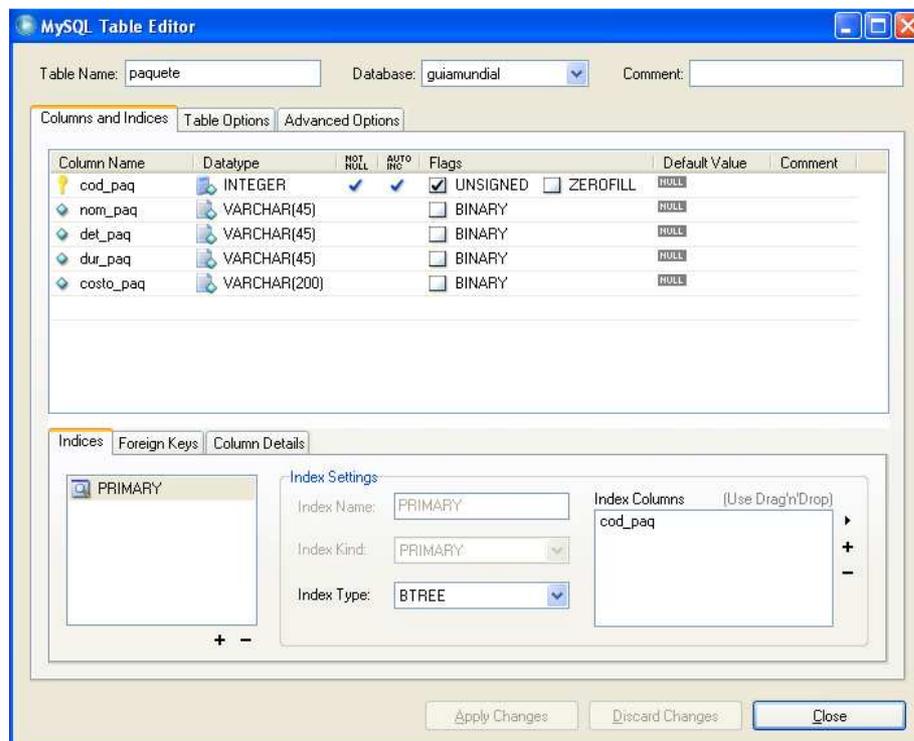


Figura 5.9. Tabla Paquete

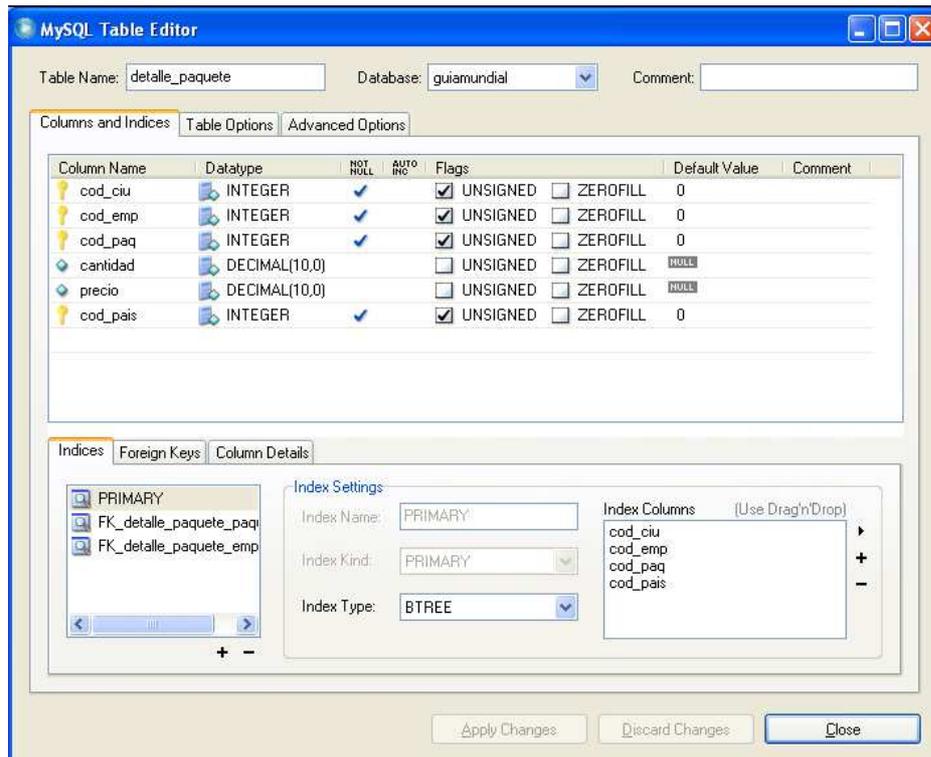


Figura 5.10. Tabla detalle_paquete

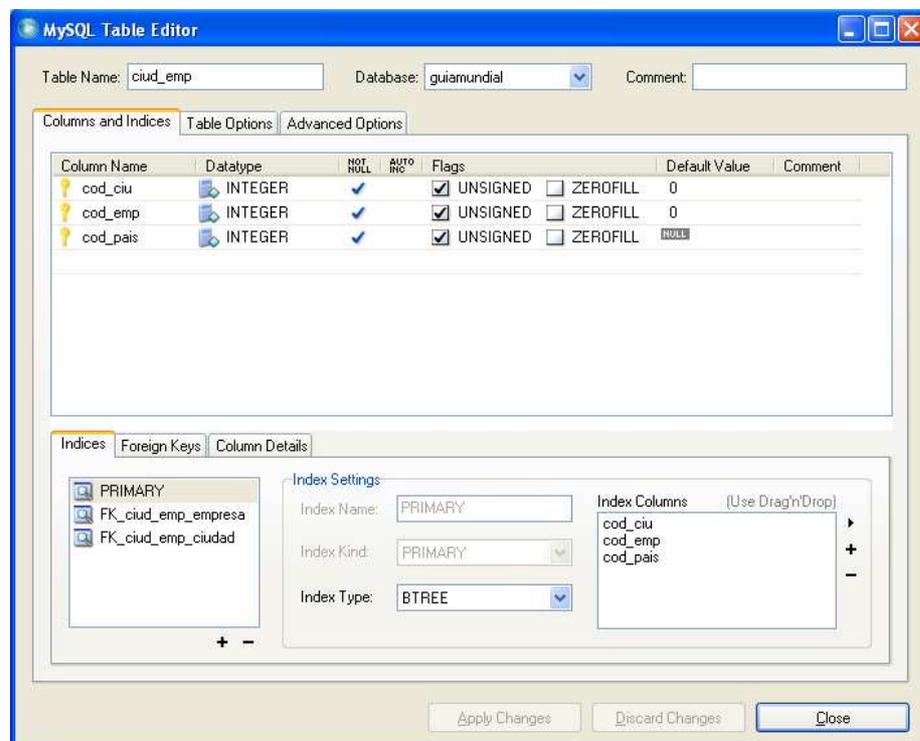


Figura 5.11. Tabla ciud_emp

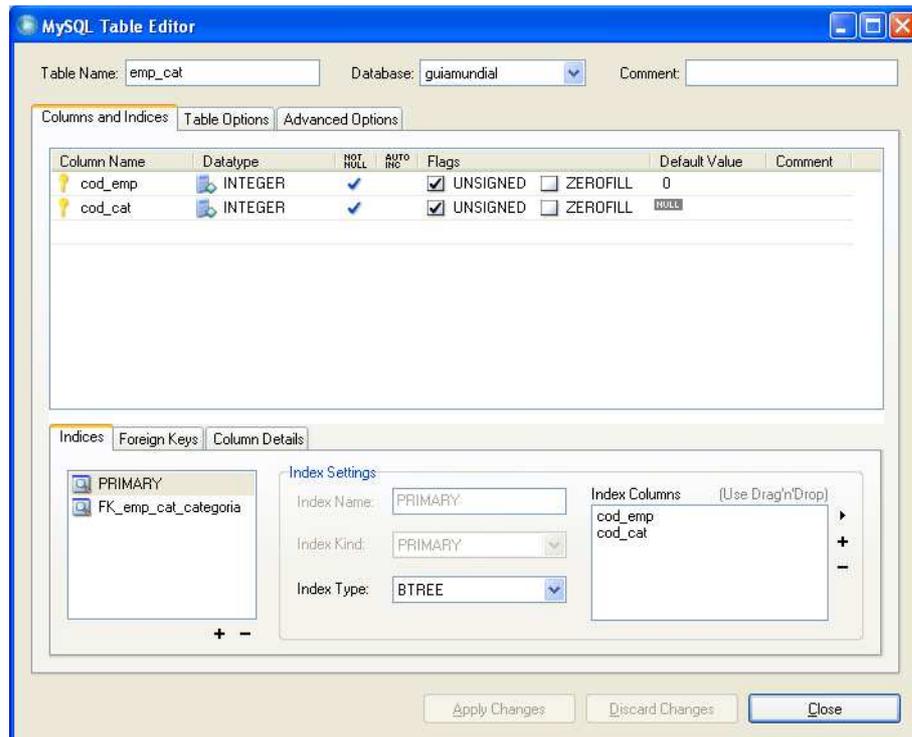


Figura 5.12. Tabla emp_cat

5.2.7 Tablas de la Base de Datos

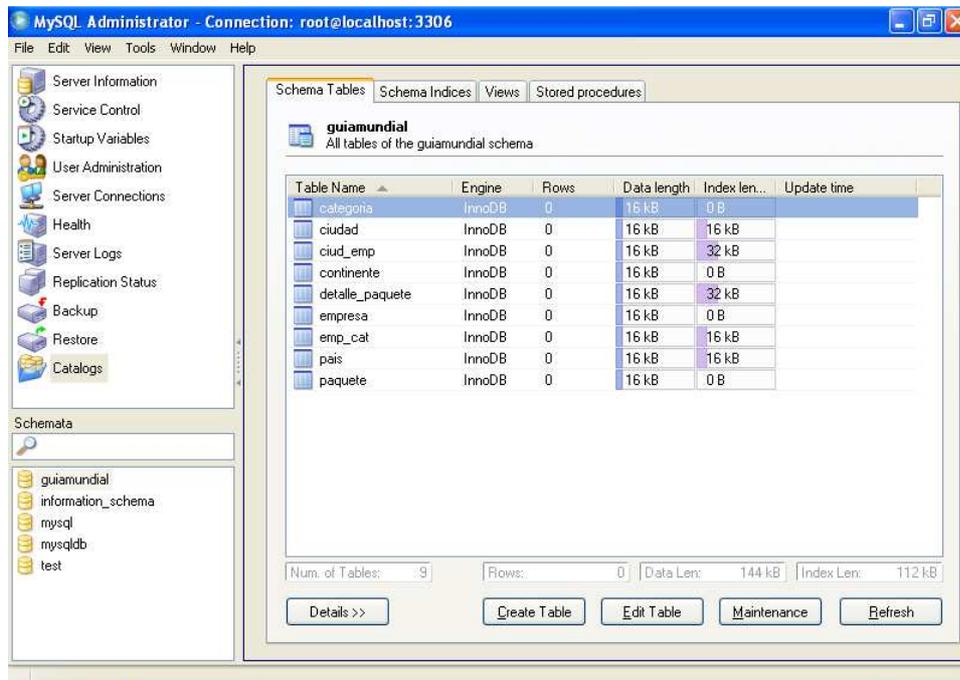


Figura 5.13. Tablas de la Base de Datos

CONCLUSIONES

- Después de un meticuloso análisis se determina que cada sistema tiene características, ventajas e inconvenientes, por tal razón la elección de uno u otro sistema para gestionar una base de datos vendrá definida por las necesidades de la empresa, por ejemplo en cuanto a dimensionamiento, costos, soporte y el aspecto más importante el rendimiento.
- La técnica de análisis usada permite decidir dónde concentrar más esfuerzos tanto en el proceso de planeación como de configuración de los SGBD, en este caso Oracle responde muy bien en operaciones de selección, al rendir $16.87 \frac{\text{filas}}{\text{ms}}$, en bases de datos con volumen de información bajo a normal, pero es necesario que exista un tuning exhaustivo del SGBD, que permita mejorar el performance o desempeño del sistema, para bases de datos con volúmenes de información mayor, en donde el rendimiento cae drásticamente a $0.2 \frac{\text{filas}}{\text{ms}}$.
- En primera instancia Oracle competía como el más fuerte de los participantes, no solo por la confiabilidad, escalabilidad y otros factores de gran peso, sino por la experiencia de la empresa en el mercado de los SGBD, sin embargo, sufre un gran descenso en su desempeño, cuando no cuenta con los recursos físicos deseables para su ejecución.
- SQL Server, su mayor defecto es su falta de portabilidad, lo que influye que su nivel de utilización quede relegado, al sector de empresas que han apostado por Microsoft, no obstante, estas empresas son una gran cantidad en el sector empresarial.
- SQL Server ofrece un gran rendimiento sobre sus competidores en la operación de inserción, con una velocidad de $0.92 \frac{\text{filas}}{\text{ms}}$ frente a $0.57 \frac{\text{filas}}{\text{ms}}$ de Oracle y $0.12 \frac{\text{filas}}{\text{ms}}$ de

MySQL, consolidándose como la plataforma más apta en aplicaciones intensivas de escritura de datos. Sin embargo al igual que Oracle sufre un gran descenso en su desempeño en la operación de selección con una velocidad promedio de $6.66 \frac{\text{filas}}{\text{ms}}$ para bases de datos pequeñas y una velocidad promedio de $2.52 \frac{\text{filas}}{\text{ms}}$ para bases de datos grandes.

- MySQL se consolida como el Sistema de Gestión de bases de Datos con mejor rendimiento para aplicaciones intensivas de consultas de datos, tanto en pequeñas, medianas y grandes bases de datos, además proporciona una funcionalidad completa con soporte para procedimientos almacenados, triggers, particionamiento utilizado en sistemas distribuidos, soporte Clustering, extensiones SQL (OpenGis), con la intuitividad y facilidad en su administración.
- SQL Server y Oracle se consolidan como los mejores en cuanto a velocidad de inserción de datos sin embargo el volumen de información manejada por los Sistemas de Gestión de Bases de Datos, incide directamente en el coste de almacenamiento de datos.
- Cuanto mayor es la base de datos, mayores son los requisitos de hardware. Por ejemplo, los requisitos de hardware de una base de datos que contenga datos que se actualicen con poca frecuencia serán inferiores a los requisitos de un almacén de datos de 1 terabyte que contenga datos de acceso frecuente de ventas, productos y clientes de una gran compañía. Además de los requisitos de almacenamiento en disco, se necesitará más memoria y procesadores más rápidos para que el almacén de datos pueda guardar en la memoria caché más datos y para que las consultas que hacen referencia a grandes cantidades de datos se puedan procesar con más rapidez.

RECOMENDACIONES

- Es recomendable utilizar para este tipo de proyectos técnicas de análisis de sensibilidad, porque permite calcular el valor posible de una variable entre un infinito número de valores posibles y aunque estos datos de entrada están sujetos a error nos dará un punto de referencia como en este caso, para decidir dónde concentrar más esfuerzos en el proceso de elección, planeamiento e incluso configuración de los SGBD.
- Para elegir un Sistema de gestión de base de Datos, es importante analizar cual se acomoda más a las necesidades de la empresa, es decir cuánto la empresa está dispuesta a invertir, además que tipo de aplicaciones va a soportar, y sobre todo con que volumen de información se va a trabajar, pues cada SGBD tiene sus pros y contras en cada uno de estos aspectos.
- Se debería ampliar este tipo de proyectos, evaluando a los Sistemas de Gestión de bases de Datos cada uno configurado, optimizado, y con los recursos ideales para su máximo desempeño.
- Se recomienda este estudio, como una guía base para comparativas mucho más profundas, en áreas específicas, como por ejemplo alta disponibilidad, concurrencia, seguridad, etc.
- Se ha comprobado que en este caso el software libre tiene las mismas características y mejor rendimiento que el software comercial, por lo que MySQL se recomienda como alternativa para aquellas personas o entidades que requieran la necesidad de Sistema de Gestión de Bases de Datos con buenas prestaciones y alto rendimiento, a un bajo costo o incluso gratis, de acuerdo al escenario de utilización.

- Se sugiere, como un próximo tema de tesis: el análisis y evaluación de las mejores tecnologías de desarrollo y la consecuente implementación de la aplicación mencionada.

RESUMEN

El proyecto investigativo se enfoca en la determinación del Sistema de Gestión de Bases de Datos (SGBD) de mayor rendimiento, a través del análisis comparativo entre los sistemas: Oracle 10g, SQL Server 2005 y MySQL 5.1, para posteriormente utilizarlo como plataforma de base de datos de la aplicación web de la empresa “**guíamundial.com**”, establecida en Panamá, con sede en Riobamba, provincia de Chimborazo. Cada SGBD, se analiza desde diferentes perspectivas, como el funcionamiento interno, características, y el rendimiento, criterio que se evalúa mediante la técnica de análisis de sensibilidad ¿what if?, determinando las variables que inciden directamente en el rendimiento de los SGBD; y el diseño de los test contruidos a partir de las variables determinadas. El proceso de evaluación consiste en tomar lecturas del tiempo de respuesta de los SGBD en cada operación ejecutada, utilizando herramientas propias del software, y posteriormente calcular la velocidad/resultado con que se realiza la tarea o proceso, indicador que determina el rendimiento del sistema de donde se concluyen los siguientes resultados: MySQL posee una eficiencia 46% mejor que Oracle y 78.60% mejor que SQL Server, para consultas de diversa complejidad, en bases de datos con volumen de información normal. MySQL posee una eficiencia 98% mejor que Oracle y 92% mejor que SQL Server para consultas de diversa complejidad, en bases de datos con gran volumen de información. SQL Server posee una eficiencia 39% mejor que Oracle y 87% mejor que MySQL para operaciones de inserción. Como se observa, el análisis ha permitido determinar los SGBDs que minimicen costes de almacenamiento, costes de acceso, pero no inconsistencias en los datos, por lo que se recomienda ampliar el campo de investigación de este análisis al área de integridad y seguridad. Finalmente, se recomienda usar MySQL, por sus características, costos y rendimiento, especialmente en entornos web en donde los requerimientos de información son más exigentes, mientras que SQL Server y Oracle pueden ser utilizados en entornos en donde exista alta demanda de escritura de datos.

SUMMARY

This research project focuses in the determination of the Database Manager System (DBMS) of major performance, through the comparative analysis among the Oracle 10g, SQL Server 2005 and MySQL 5.1 systems, for further use as a database platform for the web application of the **"guiamundial.com"** company, established in Panama, with branch in Riobamba, Chimborazo province. Each DBMS is analyzed from different perspectives such as the internal operation, characteristics, and performance, this criterion is analyzed by the sensibility analysis technique "what if", establishing the variables that directly influences in the DBMS performance; and the design of the performance test built based on the certain variables. The evaluation process consists on taking readings of execution time in each operation of the DBMS, using software tools characteristic of the DBMS, and later on calculating the velocity/result with which the task or process has been made, indicator that determines the performance from where the following results are concluded: MySQL has 46% efficiency greater than Oracle and 78, 6% greater than SQL server in diverse complexity "SELECT" operations with normal volume information databases. MySQL has 98% efficiency greater than Oracle and 92% greater than SQL Server in diverse complexity "SELECT" operations with high volume information databases. SQL has 39% efficiency greater than Oracle and 87% greater than MySQL in "INSERT" operations. As it is observed, the analysis allows establishing the DBMS that minimize the storage costs, access cost, but no data inconsistency, for this reason it is recommended to extend the research of this analysis to the integrity and security area. Finally, it is recommended to use MySQL due to its characteristics, costs and performance, especially in web environments where the requirements of information are more demanding, while SQL Server and Oracle can be used in environments where there is a high demand of data writing.

BIBLIOGRAFÍA

BIBLIOGRAFÍA GENERAL

BERTON, Rodolfo. *Métricas de Performance en Administración de Bases de Datos Distribuidas en Redes LAN y WAN.* UNLP, 2004. 160pp

BURBANO, Javier. *Análisis comparativo de bases de datos de código abierto vs código cerrado.* Universidad Católica de Quito, 2006. 111pp

CABRERA, Gregorio. *Sistemas Gestores de Bases de Datos.* Madrid. Paraninfo 2001. 205pp

CAMPS, CASILLAS. *Bases de Datos - Software Libre,* Cataluña, Eureka Media, 2005. 460pp

ELMASRI / NAVATHE, *Sistemas de Bases de Datos.* México. Segunda edición. Addison Wesley Iberoamericana. 1996. 887pp

GABILLAUD, Jérôme. *SQL Server 2005.* Barcelona, ENI, 2006, 440pp

GABILLAUD, Jérôme. *Oracle 10g: SQL, PL/SQL, SQL *PLUS.* Barcelona, ENI, 2006. 496pp.

GILFILLAN, Ian. *La Biblia de MySQL.* Barcelona. Anaya Multimedia. 879pp

GONZALES, ROJAS. *Comparación entre sistemas de gestión de bases de datos (SGBD) bajo licenciamiento libre y comercial.* UNIVERSIDAD CATÓLICA DE COLOMBIA, Bogotá, 2005. 177pp

HANSEN, HANSEN. *Diseño y Administración de Bases de Datos.* 2da Edición Madrid. Prentice Hall, 2001. 569pp.

LONEY, Kevin y BRYLA, Bob. *Oracle Database 10g Manual del Administrador,* Madrid, Oracle Press, 2004. 768pp.

MARTIN, Daniel. *Técnicas avanzadas para las bases de datos.* Barcelona. Omega, 1987. 354pp.

SARAVIA ROGER. *Modelo de Bases de Datos para Estructuras Planas.* Maestría en Ingeniería del Software. La Paz - Bolivia. Universidad Mayor de San Andrés. 2007. 16pp

SILBERSCHATZ, KORTH, SUDARSHAN. *Fundamentos de Bases de Datos 4ta Edición.* Madrid. Mc Graw Hill, 2002. 787pp

VIEIRA, Robert. *Programación con SQL Server 2005.* Madrid, Anaya Multimedia, 2007. 768pp.

REFERENCIAS WEB

ALEGSA. *Definición de Rendimiento.* [ON-LINE]. Diccionario Informático. <http://www.alegsa.com.ar/dic/rendimiento.php>

DESCONOCIDO. *VmWare.* [ON-LINE] [Fecha de Consulta: Enero 2009]. Disponible en <http://es.wikipedia.org/wiki/VMware>

DESCONOCIDO. *Catálogo del Sistema.* [PDF]. Instituto Tecnológico Autónomo de México. 2002. [Fecha de Consulta: Agosto del 2008]. Disponible en: <http://cursos.itam.mx/akuri/2002/S12002/basesdedatos/CHAPTER6.PDF>

DESCONOCIDO. *Adquisición de un Sistema de gestión de Bases de Datos.* [ON-LINE]. Consejo Superior de Administración Electrónica. 2005. [Fecha de Consulta: Febrero del 2008]. Disponible en: <http://www.csi.map.es/csi/silice/Sgbd2.html>

MARQUEZ, María. *Apuntes de Ficheros y Bases de Datos.* [PDF]. 2001. [Fecha de Consulta: Agosto del 2008] Disponible en: <http://www3.uji.es/~mmarques/f47/apun/apun.pdf>

MANUEL, José. *El Catálogo de Oracle.* [On-Line] Iniciación a Oracle. 2001. [Fecha de Consulta: Octubre del 2008] Disponible en: http://www.wikilearning.com/curso_gratis/iniciacion_a_oracle/

MySQL AB y Sun Microsystems. INFORMATION SCHEMA. [ON-LINE]. MySQL 5.0 Reference Manual. [Consultado: Octubre 2008]. Disponible en: <http://dev.mysql.com/doc/refman/5.0/es/information-schema-tables.html>

VARGAS Magali, y CANO Yuri, *Concurrencia en las Bases de Datos.* [ON-LINE]. Corporación Universitaria Minuto de Dios. 2005. [Fecha de Consulta: Septiembre del 2008]. Disponible en: <http://www.monografias.com/trabajos24/concurrencia-base-datos/concurrencia-base-datos.shtml>

VÉLEZ, Ignacio. *Sensitivity Analysis.* [PDF]. Documento electrónico adquirido en Internet. 2003. [Fecha de Consulta: 20 de Marzo del 2009] Disponible en SSRN: <http://ssrn.com/abstract=986887>

GLOSARIO

ARQUITECTURA DE DATOS: La arquitectura de datos identifica y define las mejores clases de datos que apoyan las funciones del negocio. La DA consiste de entidades de datos, cada una de las cuales tiene atributos y relaciones con otras entidades de datos.

INFORMACIÓN: La información es un conjunto organizado de datos que constituye un mensaje sobre un cierto fenómeno o ente. La información permite resolver problemas y tomar decisiones ya que su uso racional, es la base del conocimiento.

CONCURRENCIA: La concurrencia se da, cuando varios usuarios intentan modificar datos al mismo tiempo, por lo que es necesario establecer controles para impedir que las modificaciones de un usuario influyan negativamente en las de otros. El sistema mediante el cual se controla lo que sucede en esta situación se denomina control de concurrencia.

INTERPOLACIÓN: El método de interpolación es un método científico lógico que consiste en suponer que el curso de los acontecimientos continuará en el futuro, convirtiéndose en las reglas que utilizamos para llegar a una nueva conclusión.

ÍNDEXACIÓN: Es la acción de registrar ordenadamente información para elaborar su índice. En informática, tiene como propósito la elaboración de un índice que contenga de forma ordenada la información, esto con la finalidad de obtener resultados de forma sustancialmente más rápida y relevante al momento de realizar una búsqueda. Es por ello que la indexación es un elemento fundamental de elementos como los motores de búsqueda y las bases de datos .

DESEMPEÑO: El concepto de *desempeño* ha sido tomado del inglés *performance* o de *perform*. Aunque admite también la traducción como *rendimiento*, será importante conocer que su alcance original tiene que ver directamente con el logro de objetivos (o tareas asignadas).

NIVELES DE AISLAMIENTO: definen el grado en que se debe aislar una transacción de las modificaciones de recursos o datos realizadas por otras transacciones. Los niveles de aislamiento se describen en función de los efectos secundarios de la simultaneidad que se permiten, como las lecturas de datos sucios o las lecturas fantasmas.

MODELO RELACIONAL: El modelo relacional para la gestión de una base de datos es un modelo de datos basado en la lógica de predicado y en la teoría de conjuntos. Es el modelo más utilizado en la actualidad para modelar problemas reales y administrar datos dinámicamente.

PARTICIONAMIENTO: El particionamiento de tablas es un procedimiento mediante el cual se crean múltiples cadenas de páginas para una misma tabla, las cuales son accedidas por los worker processes, que son sub-tareas que pueden tener acceso simultáneo a estas cadenas de páginas y su propósito es mejorar el rendimiento.

PORTABILIDAD: Se define como la característica que posee un software para ejecutarse en diferentes plataformas, el código fuente del software es capaz de reutilizarse en vez de crearse un nuevo código cuando el software pasa de una plataforma a otra. A mayor portabilidad menor es la dependencia del software con respecto a la plataforma.

PROTOCOLO DE BLOQUEO: indica cuando una transacción puede bloquear y desbloquear elementos.

SGBD: Sistema de Gestión de Bases de Datos

SQL: es un lenguaje declarativo de acceso a bases de datos relacionales que permite especificar diversos tipos de operaciones sobre las mismas. Una de sus características es el manejo del álgebra y el cálculo relacional permitiendo lanzar consultas con el fin de recuperar -de una forma sencilla- información de interés de una base de datos, así como también hacer cambios sobre la misma. Es un lenguaje de cuarta generación (4GL).

SISTEMAS REMOTOS: Tipo de instalación capaz de trabajar sin la presencia humana. Suele emplazarse en lugares alejados o de difícil acceso y estar encargado de distintas tomas de control en instalaciones o del acopio y emisión de datos a una central.

TRANSACCIÓN: conjunto de órdenes que se ejecutan formando una unidad de trabajo, es decir, en forma indivisible o atómica.