UNIVERSIDAD NACIONAL AUTONOMA DE MEXICO

UNIDAD ACADEMICA DE LOS CICLOS PROFESIONALES
Y DE POSGRADO DEL CCH

EL MODELO RELACIONAL: EL PROBLEMA DE LA EFICIENCIA EN LOS MANEJADORES DE BASES DE DATOS RELACIONALES (SMBDR).

T E S I S

QUE PARA OBTENER EL GRADO DE
MAESTRO EN CIENCIAS DE LA COMPUTACION
P R E S E N T A
MARIA CLOTILDE DEMESA SALGADO

ASESOR: DR. SERGIO MARCELLIN JACQUES

MEXICO, D.F. 1996

INDICE

Introducción

Es un hecho que los sistemas de información han cobrado una gran importancia en el mundo de hoy. La mayoría han sido desarrollados teniendo en su base varias capas de software que realizan ciertos servicios standard. Como es el caso de aquellos sistemas que tienen como base algún manejador de base de datos. Si bien se ha realizado una importante investigación en el campo de bases de datos, este sigue siendo un campo de la informática en el que queda mucho por realizar. Debido a la demanda de nuevas facilidades que exigen los sistemas para el manejo de la información. Estas facilidades van desde la explotación de nuevos tipos de datos, fuentes de datos de largo alcance, fuentes de datos heterogeneas entre las que se requiere cooperación, para lo cual se requiere homogeneizar la manera de compartir los datos.

El objetivo del presente trabajo es dar un panorama de las consideraciones a tomar al intentar mejorar o mantener el desempeño de un manejador de bases de datos relacional en un ambiente cambiante. Para ilustrar el proceso que se pretende seguir partiré del título de la presente tésis: El modelo relacional: El problema de la eficiencia en los manejadores de bases de datos relacionales (MBDR). Lo primero que se hace necesario es revisar las reglas que forman el modelo relacional, para poder establecer un marco imparcial bajo el cual se examinaran tres diferentes manejadores. Esto con el fin de establecer las diferencias y carencias de los manejadores con respecto a la definición señalada en el modelo relacional. Una vez obtenida esta comparación se pretende contar con la información suficiente para abordar el problema de la eficiencia en los manejadores de bases de datos. Entendiendo por eficiencia: el resultado esperado en el tiempo esperado con los recursos esperados.

El problema de la eficiencia se traduce frecuentemente en un mal desempeño de los sistemas. En mi caso me interesan en particular aquellos que tienen en su base un manejador de bases de datos relacional. Cuando se tiene un mal desempeño es que se hace necesario afinar el sistema.

¿Por qué afinar?

En la mayoría de los casos el proceso de afinación se hace necesario al liberar paulatinamente modulos del sistema. El interés principal en la liberación de un sistema es algunas veces, el de cumplir con los tiempos comprometidos. De esta manera el tiempo de respuesta del sistema no siempre se evalua en la fase de pruebas y aun cuando se incluyen pruebas de volúmen no siempre se pueden controlar todas las variables que impactarán. Un sistema que no está afinado puede proporciona deficientemente los servicios para los que fué creado, sin embargo las quejas de los usuarios finales no se hacen esperar, también lleva aparejados altos costos para su funcionamiento, derivados de un mal empleo de los recursos de la computadora.

¿Porqué una metodología de afinación?

La necesidad de contar con un metodología de afinación se deriva de la necesidad de controlar los cambios que se lleven a cabo sobre el sistema una vez que ha sido liberado. Al ser el objetivo de la presente tesis el problema del desempeño en los manejadores de bases de datos relacionales, el alcance de la metodología se restringe exclusivamente a estos manejadores. Por lo tanto en el contenido de la presente tesis revisaremos el tipo de cambios que se pueden aplicar y hasta que estos se implantan se puede saber el impacto real sobre el sistema. Es por ello que se requiere llevar un control de los cambios que se realizarán, así como un registro de lo acontecido en cada caso. Esto es importante ya que ni todos los sistemas son iguales, ni todas las aplicaciones proporcionan los mismos servicios.

De estos puntos se deriva la necesidad de contar con pasos claves para el éxito del proceso de afinación, estos puntos los englobaremos en una metodología de afinación.

Para lograr esto, se requiere conocer las limitaciones con que se topan los manejadores de bases de datos relacionales. Determinar si el modelo relacional, que sirve como sustento teórico de dichos manejadores ha sido superado. Para lo cual se hace una comparación entre los manejadores más comunes en el mercado, con el fin de determinar su apego al modelo relacional. Una vez conocidas las diferencias substanciales entre cada manejador se propone obtener una arquitectura general que sirva como base para describir la metodología que auxiliará en mejorar el desempeño de los manejadores.

El contenido del trabajo es el siguiente:

Capítulo 1. Inicia con un poco de historia acerca de la evolución de los manejadores de base de datos. Plantea la problemática actual de los manejadores de bases de datos relacionales. Ilustra algunas tendencias en el campo de bases de datos.

Capítulo 2. Nos introduce en el modelo relacional, definiendo las reglas de CODD dandoles la estructura del modelo ANSI. La importancia de este capítulo, se tiene en el hecho de que es necesario conocer el modelo en que se sustentan los manejadores a comparar. Se define una arquitectura general de una manejador de base de datos relacional.

Capítulo 3.- En base a las reglas que se destacan en el capítulo 2 se realiza la comparación de los tres manejadores de bases de datos escogidos como alcanze de la presente tésis: ORACLE 7.0,SYBASE 10.0 y DB2 V2R3.

Capítulo 4.- Realiza una descripción de la metodología para el mejoramiento del desempeño, tomando como base la arquitectura general descrita en el capítulo 2.

Capítulo 5.- Se toma el manejador DB2 como el caso para ilustrar la metodología para mejorar el desempeño.

Conclusiones.

Bibliografía

MANEJADORES DE BASES DE DATOS RELACIONALES: DEL INICIO A NUESTROS DIAS

El objetivo del presente capítulo es dar un panorama de la evolución de los manejadores de base de datos, cuál es la situación actual y cuáles son los retos a los que se enfrentan los desarrolladores de manejadores de bases de datos relacionales.

I Historia

La investigación en base de datos surge a finales de los 60's. Se ha llevado a cabo en gran medida en la industria, en donde se han identificado 4 requerimientos principales en el área de Administración de Sistemas en Bases de Datos SMBDR(Sistemas Manejadores de Bases de Datos):

- Eficiencia en el acceso: para consulta y modificación de grandes volúmenes de datos.
- Elasticidad: Habilidad de la información para sobrevivir a fallas de hardware o software
- Control de accesos: Acceso concurrente a información consistente por parte de varios usuarios que cuenten con la autoridad pertinente.
- Persistencia: Asegurar la información por largos períodos de tiempo

Las direcciones en que se han enfocado los esfuerzos se destacan las siguientes:

I.I Sistema Manejador de Bases de Datos Relacionales (SMBDR)
I.II Manejadores de transacciones
I.III Sistemas de base de datos distribuidos

I.I Manejadores de Bases de Datos Relaciónales

En 1970 había 2 tendencias para la construcción de sistemas basados en manejadores de bases de datos:

- El IMS de IBM manejador de base de datos jerárquico
- La norma CODASYL manejadores de bases de datos reticulares (tal fue el caso del IDMS de Cullinet)

Las limitaciones fundamentales de ambas soluciones eran:

1. Se requería ser un experto para la navegación en la base de datos.
2. Frecuentes cambios en la estructura de la base de datos implicaban reescritura de programas completos.

El modelo de datos relacional liberado por E.F. Codd entre los años de 1970 y 1972, ofrecía un acercamiento fundamentalmente diferente. Una discusión más amplia de dicho modelo es el propósito del capítulo 2.

Durante los años 70's la comunidad de bases de datos investigó extensivamente el concepto de SMBDR, como frutos de esta investigación se:

- Inventaron lenguajes de consulta de alto nivel.
- Desarrollo la teoría y algoritmos necesarios para la optimización de las consultas
- Formuló una teoría para la normalización, con el propósito de auxiliar el diseño de bases de datos
- Construyeron algoritmos para agrupar las tuplas[1] en bloques con el propósito de minimizar los costos de acceso a la información
- Construyeron técnicas de indexación para proveer a los sistemas con un acceso rápido a los datos
- Instrumentaron prototipos de SMBDR que fueron precusores de los manejadores actuales.

I.II Manejadores de transacciones

Son piezas de software encargadas del manejo de lo que llamaremos transacción (conjunto de operaciones englobados en una supraoperación) que debe parecer atómica. Es decir se debe garantizar que la ejecución de una transacción lleva a la base de datos de un estado consistente a otro. Para asegurar esto se requiere:

1.- La ejecución concurrente de transacciones debe ser tal que parezca que la transacción se ejecuta en aislamiento . El control de concurrencia es la técnica empleada para asegurar esto.

2.- Las fallas en el sistema , sean estas causadas por problemas en el hardware o en el software, no deben desembocar en estados inconsistentes en la base de datos. Una transacción se debe ejecutar toda completa o nada de esta. La técnica empleada para asegurar esto se llama recuperación.

[1] Tupla: Un miembro de una relación

Relación: Subconjunto del producto cartesiano $D_1XD_2X...XD_k$. En donde D_1 a D_k son dominios.Las tuplas de esta relación tienen k componentes, se emplea $(v_1,v_2,...,v_k)$ para denotar una tupla

Durante la decada de los 70's a los 80's se trabajó exhaustivamente sobre el modelo transaccional. En un principio se evaluó la serialización de las transacciones. Sin embargo surgieron otros algoritmos en los que se emplearon técnicas como las siguientes:

- Mantener varias versiones de los datos que se accesan

- Poner candados a la información para evitar colisiones o conflictos en el acceso a la infurmación. Se establecieron diferentes protocolos para el manejo de estos candados como son:

1. Poner una marca de tiempo de tal manera que el manejador no tenga problemas en serializar los accesos. Estos accesos se manejan mediante un esquema de gráfica que puede reconstruirse cada vez que:
   Se inserta o elimina una petición de espera.
   Se cumple un lapso de tiempo
   Se invoque el algoritmo de detección de ciclos por el coordinador

2. En un ambiente distribuido, cada una de las localidades conserva una gráfica local, y se encuentran conectadas entre ellas a travez de otro arco, si este no está, la localidad está bloqueada, en caso contrario se invoca el algoritmo de localización de bloqueos global.

Este trabajo de detección de bloqueos lo lleva a cabo el coordinador, que puede ser un nodo fijo o puede ser determinado al momento de operación.

En los manejadores de transacciones se tienen conceptos importantes como:

Concurrencia: Posibilidad de que cualquier número de transacciones accesen una Base de Datos.

Seguridad: Protección de la información de eventos como: uso no autorizado, destrucción o alteración.

Integridad: Se refiere a la confiabilidad o validez de la información

Recuperación: Capacidad de llevar la Base de Datos de un punto presente a un punto de sincronía en el pasado.

Punto de sincronía: Estado válido de la base de datos.

La recuperación es un punto central en el manejo de transacciones por ello se tienen técnicas como:

1. Escritura previa en la bitácora. Es decir los cambios se registran en una bitácora antes de afectar a la base de datos, de manera tal que si una transacción termina anormalmente la bitácora es empleada para recuperar la base de datos al estado previo a la ejecución de la transacción.

2. Archivos sombra, copias de los datos se crean para reflejar los efectos de una transacción , cuando esta termina se copian las nuevas versiones de lo contrario se descarta la nueva versión.

I.III Bases de Datos Distribuidas

La comunidad de investigación en base de datos a finales de los 70's emitió la reflexión de que las organizaciones son primordialmente descentralizadas por lo que requieren bases de datos en diferentes localidades. La necesidad de compartir información lleva al concepto de Base de Datos distribuida. Que consiste de una colección de instalaciones que:

1. Tienen una Base de Datos Propia
2. Las instalaciones han decidido trabajar juntas de tal manera que cualquier usuario autorizado pueda accesar los datos de cualquier base de datos como si esta formase parte de la base de datos local.

Los problemas que se pretenden resolver con el uso de bases de datos distribuidas son los siguientes:

- Proceso de Consultas
- Administración del catálogo
- Propagación de las modificaciones
- Control de las recuperaciones
- Control de concurrencia.

El resolver estos problemas requiere a su vez nuevos algoritmos que se resumen en doce reglas para bases de datos distribuidas:

1.- Autonomía local
2.- No dependencia de una instalación central
3.- Operación continua
4.- Independencia de las instalaciones
5.- Independencia en los fragmentos
6.- Independencia en la replicación
7.- Procesamiento de consultas distribuidas
8.- Manejo de transacciones distribuidas
9.- Independencia del Hardware
10.- Independencia del sistema operativo
11.- Independencia de la red
12.- Independencia del tipo de SMBD

La técnica de 2 phase commit es importante al hablar de bases de datos distribuidas. Esta técnica es empleada cuando una transacción interactua con varios manejadores de datos independientes. En este escenario una transacción debe asegurar que los cambios se efectuan o no en todos los manejadores. Para asegurar esto existe un componente del sistema llamado el coordinador, la manera en que trabaja se describe a continuación:

Suponiendo que el cambio tenga que ser efectuado.

1. Se requiere que todos los manejadores guarden su log en un dispositivo no volátil. Al terminar contestar exitoso o retorno al coordinador.

2. El coordinador recibe todas las respuestas y escribe en su log la decisión que sea pertinente. Exitoso si todas las respuestas recibidas fueron: exitoso. Retorno si alguno de los manejadores contestó: retorno. El coordinador en la fase dos se encarga de propagar su decisión a todos los manejadores y estos deben tomar las acciones pertinentes. Si por alguna razón hay una falla el manejador puede encontrar en el coordinador la decisión, si no se encuentra entonces asume que la respuesta fué retorno.

II Nuevas Generaciones

Hay quienes opinan que los manejadores de bases de datos son una tecnología madura y es por ello conveniente enfocar la investigación sobre nuevos tópicos. Si bien esto es cierto en parte también es igualmente cierto que los requerimientos en manejo de información que demandan las nuevas aplicaciones no pueden ser cubiertos por los manejadores de bases de datos existentes. Las nuevas generaciones de SMBD se enfrentan a nuevos requerimientos como son:

II.I Nuevos tipos de datos

Las aplicaciones cada vez requieren no sólo almacenar grandes volúmenes de datos, sino que a la vez la complejidad interna de los mismos crece. Por lo tanto será necesario encontrar solución a los problemas que se derivan de esta situación como son:

Problema de encontrar nuevas técnicas para el acceso a datos: Las aplicaciones actuales manejan usualmente registros pequeños, con los nuevos tipos de datos este paradigma seguramente no se mantendrá, tal es el caso del almacenamiento de imágenes. Es por ello que serán necesarias nuevas técnicas para el manejo de estos tipos de datos, tales como procedimientos de compactación de la información, nuevos lenguajes interrogadores, así como técnicas para emplear adecuadamente el almacenamiento a fin de proporcionar un buen desempeño en el acceso a estos datos.

Problema de la inclusión de reglas para la administración de los tipos de datos: Se deben proporcionar ayudas a los programadores para que ellos puedan construir los tipos de datos que juzguen adecuados. Sin embargo estas nuevas facilidades hacen necesario el surgimiento de nuevas disciplinas que permitan la combinación adecuada de manejo dinámico de los tipos de datos con la persistencia de la información en la base de datos.

Problema del proceso de Reglas: La siguiente generación de aplicaciones requerirá almacenar reglas de integridad que se deben aplicar sobre los datos, agregandole una característica extra a los mismos, como sería significado, forma de operar y acciones que se pueden tomar sobre el mismo. Es necesaria la investigación para obtener técnicas para especificar las reglas así como para obtener los algoritmos necesarios para manejar de manera eficiente sistemas basados en reglas.

II.II Nuevos conceptos en los modelos de datos

En realidad muchos de estos conceptos se encuentran en varias de las aplicaciones viejas, mas no en todas y son conceptos que se encuentran entretejidos en las aplicaciones, pero no definidos claramente. Tal es el caso de:

1. Espacio de Datos: Se refiere al manejo que se hace de los datos para satisfacer consultas del tipo: encontrar los diez datos vecinos más cercanos a determinado dato. Las estructuras necesarias para responder estas interrogaciones funcionan muy bien en memoria pero no se han logrado llevar exitosamente a tipos de almacenamiento persistentes.

2. Tiempo: Para algunas aplicaciones es de suma importancia el poder contar con una historia de la base de datos, además de ser posible accesarla, para rastrear el comportamiento de determinado dato.

3. Incertidumbre Cuando una conclusión debe ser obtenida a partir de información parcial o poco confiable. El problema en este caso es que información es la relevante.

4. Escalabilidad Los algoritmos para el manejo y mantenimiento de los datos deberán funcionar de igual manera para un alto o un bajo volumen. Ya que es la tendencia a contar con bases de datos de gran volumen que estén disponibles en una ventana 24 horas x 7 días, por lo que las técnicas para mantenimiento,respaldo y recuperación deben ser desarrolladas con el fin de cumplir con los requerimientos de volumen y disponibilidad de la información.

5. Paralelismo: El pensar en bases de datos de alto volumen , hace necesario que el manejo de la información sea ágil, lo cual con un proceso secuencial sería imposible, por ello es que se piensa en la solución de consultas empleando procesos paralelos.

6. Almacenamiento terciario y transacciones de larga duración: Para aplicaciones de alto volumen se va a requerir tanto un almacenamiento secundario (disco) como la integración de un almacenamiento terciario (información archivada). Los futuros optimizadores tendrán que lidiar con consultas en donde se combine la información que está en almacenamiento secundario y en el terciario. Así mismo cuando las transacciones sean de alta duración es necesario desarrollar técnicas para mantener la integridad, compartir y recuperar la información.

7. Versiones y Configuraciones: Las aplicaciones de la siguiente generación requerirán versiones que representen estados alternados o sucesivos de una entidad conceptual.

8. Browsing: Se debe poder obtener información sobre la información así como los pasos necesarios para responder a determinada consulta.

9. Procesos más que transacciones: La orientación que se ha dado a los manejadores es hacia el manejo de transacciones en tiempo real, más que el manejo de largos procesos. Estos procesos son conjuntos de transacciones que se deben ejecutar en su totalidad o de lo contrario debe regresar la base de datos al estado previo a la ejecución de los mismos. Sin embargo este tipo de manejo sólo se lleva a cabo a nivel de transacción y no de proceso. Esta es una cualidad de la que carecen los manejadores actuales y es deseable en las nuevas generaciones.

10.Bases de Datos Distribuidas y Heterogéneas: Se puede pensar en una base de datos mundial, al igual que se logró tener una red telefónica mundial o una red de computadoras. La tecnología necesaria para lograr esto se debe empezar a desarrollar. Las implicaciones de que las bases de datos sean heterogéneas son:

No todos los manejadores hacen las mismas Asunciones o supuestos acerca de la manera en que manejan la información., así como normalmente si bien interoperan cada base de datos tiene un dominio local de la información

Para lograr el objetivo de compartir la información que se tiene almacenada en diferentes plataformas es necesario cubrir las limitaciones mencionadas en los siguientes puntos.

I Incompletitud e Inconsistencia: Como las bases de datos no fueron diseñadas con miras a compartir información el unir información de diferentes fuentes puede acarrear inconsistencias semánticas. Por lo que el modelo de datos debe ser extendido con el fin de obtener información del significado de la información en cada base de datos.

II Mediadores: Como los problemas de fusión de la información e inconsistencia semántica son tan severos, es necesario una parte intermedia que sirva como fuente de información y se encuentre entre el usuario y la base de datos heterogénea.

III. Servicios de Nomenclatura: Se debe tener un lugar en donde se pueda consultar la ubicación y nombre de las bases de datos así como de los mediadores de interés.

IV. Seguridad: Un tópico débil en los manejadores tradicionales, se vuelve aún más complejo cuando se pretende trabajar con bases de datos distribuidas y heterogéneas.

V. Ruta de acceso: Al aumentar la capacidad de los manejadores de base de datos en cuanto a conectividad y almacenamiento, se vuelve cada vez más costosa la evaluación de la mejor ruta de acceso, se deben encontrar nuevos algoritmos para resolver este problema.

VI. Globalización: La tendencia mundial en cuanto al proceso de información es contar con ventanas de tiempo de 24 horas los 365 días del año. Por lo que los viejos esquemas para procesos nocturnos o procesos de fin de semana deben ser superados. Esto impone la solución de nuevos problemas: como evitar que procesos como replicación de información,respaldar o dar mantenimiento a las bases de datos interfieran con la disponibilidad de la misma.

VII. Interoperabilidad: La información se encuentra residente en diferentes plataformas. El reto para las herramientas de consulta es proporcionar la información de manera tan simple como sería una consulta jerárquica en un único manejador, de igual manera se pretende que se resuelva una consulta que implique navegación entre manejadores. Esto manteniendo la independencia de plataforma o formato.

VIII. Independencia del manejador de bases de datos:
En nuestros días normalmente la elección de un manejador de bases de datos restringe la forma de uso del espacio físico, de memoria,de métodos de acceso, del lenguaje interrogador así como las interfaces de aplicación. Uno de los retos para los futuros manejadores es el de estar compuestos por varios módulos, lo que les permita flexibilidad y no ser como ahora un universo en sí mismos.

IX. Manejo de Transacciones: En un ambiente heterogéneo se topa con el problema de que cada manejador tiene diferencias en los protocolos que emplea para el manejo de la concurrencia.

II.III Aplicaciones Heredadas

En la actualidad existen multitud de aplicaciones con las que operan normalmente las corporaciones y se encuentran desarrolladas en lenguajes de tercera o segunda generación. Para emplear las nuevas tecnologías o cubrir sus necesidades de información la empresa actual necesita nuevos desarrollos. Esto podría lograrse a través de tomar el código ya existente y simplemente recodificar toda la aplicación tal y como está en la actualidad. También se podría prescindir tanto de código, como de la información actual y desarrollar un sistema nuevo. Por otro lado se podría hacer uso de las técnicas de la ingeniería de la información para encontrar la convergencia entre procesos y datos; esto lleva a una reducción en el esfuerzo de desarrollo que se requiere. Esto se logra cuando la descomposición de procesos tuvo efecto y se pueden mapear las relaciones con los procesos de tal manera que se puede identificar los procesos convergentes. Como resultado de este análisis se tiene información tanto de los procesos como de los datos, con esta se puede construir el modelo corporativo de datos así como alimentar a un repositorio de datos, el cuál nos permitirá el acceso en forma ordenada de la información acerca de los datos.

El contar con modelos corporativos de datos, así como también con software que permita manejar la información acerca de datos de los datos, es un paso necesario para abordar proyectos como:

1. Desarrollo de aplicaciones cliente-servidor

   Debido a los altos costos que implica el mantener grandes sistemas centralizados es que se está buscando desarrollar aplicaciones tipo cliente-servidor. Esta estrategia de descargar al computador central de trabajo que no es medular para el trabajo de la corporación y dejar que servidores más pequeños en áreas específicas del negocio cuenten con la información y recursos que el area necesita recibe el nombre de downsizing.

   Existen varios pasos que hay que tomar para llegar a una estrategia cliente/servidor:

   a) Establecer un ambiente relacional
   b) Adquirir estaciones de trabajo
   c) Reescribir los programas de aplicación para substituir las interfaces de mainframe por pantallas más amigables
   d) Moverse a una filosofía de servidores en donde las estaciones de trabajo serán servidores de aplicaciones y los hosts compartidos serán servidores de peticiones SQL.

2. Construcción del almacén de información.

   El concepto de almacén de información surge a partir de varios requerimientos de información como son:
   - Información histórica
   - Información resumida
   - Información consolidada de diferentes fuentes

   Esto se debe lograr cumpliendo las siguientes condiciones:
   - Evitar impactar a las transacciones y procesos de la base de datos operativa.
   - Rediseñar la base de datos para poder almacenar grandes volúmenes de datos.
   - Generar nuevas entidades y mantenerlas actualizadas.

Por ello, la solución que se propone es contar con una base de datos aparte de la operativa, que recibe el nombre de almacén de información. Nuevos problemas se presentan como:

- Replicacion y propagación de la información
- Extracción, refinamiento y población del almacén de información
- Mantenimiento a la estructura del almacén debido a cambios en el diseño de la base de datos operativa
- Estrategia para recuperar información histórica, reduciendo los costos de almacenaje.

3. Desarrollo empleando cúdigo reusable.
El desarrollo empleando código reusable se hace más popular en nuestros días, sin embargo todavía presenta problemas a ser resueltos. Se mencionan brevemente 3 técnicas para desarrollar programas empleando código reusable:

a) Empezar por el todo: Se parte de una aplicación general que se modificará hasta tener la aplicación específica en este caso un repositorio y un modelo de datos son de gran utilidad ya que permiten llevar un control de la evolución del sistema.

b) Empezar por las partes:Se parte de una colección de bloques o módulos los cuales pueden modificarse. En este caso el modelo de datos va a fungir como la liga entre bloques y el repositorio como la garantía de que las reglas del manejo de los datos no se quebranten en ningún módulo.

c) Es una combinación de ambos.

II.IV Bases de Datos Orientadas a Objetos

Las bases de datos orientadas a objetos pretenden cubrir la brecha semántica entre el dominio de la aplicación y su representación en almacenamiento persistente. Las Bases de Datos Orientadas a Objetos (BDOO) obtienen su capacidad de modelado de los conceptos de orientación a objetos como:

- Objeto: Cualquier abstracción de una entidad del mundo real
- Atributos y Métodos: Un objeto tiene uno o más atributos y uno o más métodos que operan sobre los valores de los atributos.
- Encapsulamiento y paso de mensajes: El acceso a los valores de los atributos se requiere a través de mensajes, así como también a través de los métodos ya que ambos se encuentran encapsulados en el objeto.
- Clase: Todos los objetos que comparten los mismos atributos y métodos deben formar parte de una misma clase.

Un modelo de datos bajo los conceptos de orientación a objetos es una organización lógica de los objetos del mundo real, restricciones de los mismos y la relación entre ellos. Una base de datos orientada a objetos es una colección de objetos cuyo comportamiento y estado , así como sus relaciones se encuentran definidos en concoordancia con un modelo de datos orientado a objetos. Un sistema manejador de bases de datos orientado a objetos es aquel que permite la manipulación de bases de datos orientadas a objetos.

Si bien las BDOO no pretenden cubrir todas las limitaciones de los manejadores tradicionales, si son una buena aproximación para cuestiones como manejo de datos mas complejos que los tradicionalmente definidos en los manejadores como veremos en el siguiente capítulo. Tambien pretende cubrir la brecha entre el lenguaje empleado para accesar la base de datos y el lenguaje empleado para el proceso de datos.

II.V Bases de Datos Basadas en Lógica

Dentro de las propuestas con que se cuenta en bases de datos se tienen aquellas que estudian la relación que existe entre la teoría de bases de datos relacionales y la lógica de predicados de primer orden. Si bien los estudios sobre bases de datos y lógica, se remontan a finales de los 70's es en 1984 con un artículo escrito por REITER que el tema se retoma.

De acuerdo al planteamiento de esta tendencia se habla de:

Axiomas base: que tienen una correspondencia con las denominadas tuplas en el modelo relacional.

Axiomas deducibles: conjunto de reglas mediante los cuales dados ciertos hechos es posible deducir nuevos hechos.

A continuación se da una breve explicación de como estos conceptos, llevan a la definición de una nueva clase de manejadores de base de datos.

Para poder hablar de manejadores de bases de datos deductivos se establece que estos deben soportar lo que llamaremos, vista de prueba teórica, la cual consiste de:

1.- Axiomas base que corresponden a lo que conocemos como relaciones base del modelo relacional. Recibe a su vez el nombre de Base de datos extensional (BDE).

2.- Axiomas de completitud en donde se establece que no existe ninguna tupla existente perteneciente a X relación fuera de aquellas implícitamente inscritas. De manera que se cumple con la asunción de mundo cerrado, la cual establece que la omisión de una tupla en una relación implica que la asserción correspondiente a esa tupla es falsa.

3.- Axioma de nombre único el cual postula que toda constante es diferenciable de las demás.

4.- Axioma de cerradura de dominio el cual establece que no existe ninguna constante fuera de aquellas incluidas en los dominios de la base de datos.

5.- Axiomas para definir el predicado "="

Además es capaz de deducir hechos adicionales a partir de la BDE aplicando axiomas deductivos. Tanto estos como las reglas de integridad constituye lo que se denomina base de datos intencional (BDI). Por último hay que agregar que la base de datos deductiva se conforma de la BDE y BDI.

Consultas Recursivas

Es un tópico que se incluye al estudiar las bases de datos deductivas, ya que este tipo de consultas no se tenían en el enfoque relacional.
Para la solución de este tipo de consultas se requiere desarrollar nuevas técnicas algunas de estas son:

1.- La técnica standard de Prolog en cuanto a unificación y Resolución.

2.- Evaluación inocente se emplea la técnica de encadenamiento hacia atrás, empleando como entrada la BDE.

3.- Evaluación semi-inocente se emplea la técnica de encadenamiento hacia atrás, empleando como entrada los valores no evaluados previamente de la BDE.

4.- Filtrado estático consiste en reducir el conjunto de hechos relevantes como en la optimización tradicional y sobre este conjunto emplear la técnica de encadenamiento hacia atrás.

Acerca de las aportaciones de la investigación sobre bases de datos deductivas se tiene la de poder desarrollar una liga entre la base de datos y el lenguaje de programación sin la necesidad de contar con un lenguaje intermedio y propietario de la base de datos. Que permita el trabajo con información de la base de datos sólo a través de código embebido.

III SITUACION ACTUAL

Se puede pensar que la tecnología en bases de datos esta madura, y de hecho es común en la actualidad. con aplicaciones desarrolladas teniendo como base algún manejador de base de datos relacional. En el mercado de software se encuentran varios manejadores que ofrecen los mismos servicios ya que todos tienen como sustento el modelo relacional, sin embargo difieren en su instrumentación, operación y la(s) plataforma(s) en que operan. Para el alcance de la presente tesis se eligieron los manejadores ORACLE,SYBASE y DB2 para llevar a cabo una comparación entre ellos, con el objeto de que al conocer sus semejanzas y diferencias se pueda establecer una metodología que sirviese como guía en el proceso de afinación de aplicaciones.Las bases sobre las que se llevó a cabo la comparación fueron: La arquitectura ANSI/SPARC para manejadores de base de datos y el modelo relacional.

Sin embargo en la actualidad se tienen mejoras en el hardware con respecto a los años anteriores, de igual manera los precios en el hardware y software son mas bajos y las características de ambos, superiores a las que se tenían. Sin embargo ello no quiere decir que los SMBDR se deban quedar como están, sino que deben cambiar para cubrir las necesidades futuras.

Se tienen en la actualidad SMBDR tanto centralizados como distribuidos, trabajando. De igual manera los esfuerzos en orientación a objetos y sistemas deductivos o basados en lógica, trabajan si bien no a gran escala, si en propuestas de punta.

El mayor exito de las bases de datos relacionales, es tal vez el haber llevado un concepto manejado por algunos investigadores, a un concepto que en la actualidad se encuentra difundido gracias a la gran variedad de aplicaciones, que en la actualidad tienen como sustento un manejador de bases de datos relacional.

"If only it weren't for the people, the godammed people," said Finnerty, "always getting tangled up in the machinery. If it weren't for them, earth would be an engineer's paradise."
From Player Piano by Kurt Vonnegut, Jr.

EL MODELO RELACIONAL: ARQUITECTURA

El objetivo del presente capítulo es proporcionar un panorama del modelo relacional, así como un marco de referencia de algunos conceptos manejados en las diferentes instrumentaciones del modelo relacional.

I Antecedentes

Los modelos de datos no son todos relacionales, el modelo de datos relacional surge alrededor del año 1968, el responsable de estas ideas es E.F CODD. Como el mismo dice, a lo largo de casi 20 años el modelo relacional ha tenido que ser enriquecido. En un principio se tuvo una serie de artículos que vieron su culminación en la versión 1 del modelo relacional (1979). A este primer modelo le siguió la versión 2 en (1988), que en esencia contiene todas las características de la versión 1, sin embargo en esta versión se explican puntos que en la anterior se dieron por hecho, pero que condujeron a los usuarios y a los investigadores a interpretaciones equivocadas.

I.1 Introducción

Al empezar a hablar de modelo relacional hay que poner en claro dos conceptos:

Modelo.

"Un modelo es una representación abstracta de algo real. La abstracción resalta características importantes e ignora otras.
Los modelos nos ayudan a comprender problemas y tomar decisiones de manera informada"[1]

[1] Yourdon,
Information Systems Analysis Workshop
1991.

Relación

Dos definiciones:

"El modo particular en que una cosa se piensa en conexión con otra; cualquier conexión, correspondencia o asociación, que puede concebirse como naturalmente existente entre las cosas."

"Dados los conjuntos S1,S2,...,Sn (no necesariamente distintos), R es una relación de estos n conjuntos si existe un conjunto de m tuplas de las cuales el primer componente se obtiene de S1, el segundo de S2 y así sucesivamente"[2]

Después de estas dos definiciones podemos continuar con una breve definición de modelo relacional, definiéndolo como un conjunto de reglas que establecen las características que los manejadores que se dicen relacionales deben guardar al estructurar, administrar y resguardar la integridad de los datos que conforman una base de datos.

[2] E.F.Codd,
The relational Model for Database Management
Versión 2,
Sin lugar, Addison-Wesley, Julio 1990
pag 1.

II Las Doce Reglas de Codd

Originalmente se tenían doce reglas(Modelo relacional versión 1), que definían adecuadamente un manejador de bases de datos relacional, válidas a principios de los ochentas. Estas reglas se enuncian a continuación:(DATE: P.391)

1 - Regla de la información.- Simplemente requiere que toda la información en la base de datos se represente en uno y solo un sentido, etiquetando por valores en las posiciones de las columnas dentro de los renglones de las tablas. Este requerimiento se conoce como "El principio básico del modelo relacional."

2.- Regla de la garantía de acceso.- Esta regla es escencialmente un replanteamiento del requerimiento fundamental para llaves primarias. Dice que cualquier valor individual escalar en la base de datos debe ser lógicamente direccionable, especificando el nombre de la tabla que lo contiene, el nombre de la columna y el valor de la llave primaria que identifica al renglón que lo contiene.

3.- Tratamiento sistemático de los valores nulos.- El SMBD requiere soportar una representación de información faltante o información no aplicable, que es sistemáticamente distinta de valores regulares e independiente del tipo de dato. Se implica además que la representación debe ser manipulada por el SMBD de manera sistemática.

4.- Catálogo en línea activo basado en el modelo relacional.- El sistema requiere proporcionar un catálogo relacional en línea, accesible para usuarios autorizados por medio de su lenguaje regular de consulta.

5.- Sublenguaje de datos comprensible.- El sistema debe soportar al menos un lenguaje relacional que tenga una sintáxis lineal; que puede ser utilizado de 2 maneras: interactiva y embebido en el progama aplicativo y soportar operaciones de definición,manipulación de datos, seguridad, control de la integridad y manejo de operaciones transaccionales.

6.- Regla de actualización de vistas.- Todas las vistas que son teóricamente modificables, deben ser modificadas a través del sistema.

7.- Inserciones,borrados y modificaciones de alto nivel.- El sistema debe soportar a un tiempo operaciones de inserción, borrado y modificación.

8.- Independencia física de los datos

9.- Independencia lógica de los datos

10.-Independencia de la integridad.- Las relaciones de integridad deben ser especificadas de manera separada de los programas aplicativos y guardadas en el catálogo. Debe ser posible cambiar dichas aplicaciones como y cuando sea apropiado sin sacarles de línea.

11.- Independencia de la distribución.- Las aplicaciones existentes deben continuar operando con éxito a) cuando una versión distribuidad del SMBD se introduce por primera vez; b) cuando información existente, ya distribuida se redistribuye en el sistema.

12.- Regla de no subversión.- Si el sistema provée una interfaz de bajo nivel, entonces dicha interfaz no puede ser usada para subvertir el sistema pasando por alto las reglas de seguridad e integridad.

Este conjunto de reglas aparentemente eran suficientes para determinar si un SMBD se podía considerar relacional o no, sin embargo debido a las diferentes interpretaciones o alcances que se podían hacer de cada regla, ya que esta no presentaba límites contundentes, surge la revisión del mismo, motivo por el cual las doce reglas originales se convierten en trescientos treinta y tres reglas, de las cuales me ocuparé en las páginas siguientes.

III Las Trescientos Treinta y tres reglas de Codd

Todo este conjunto de reglas (Modelo relacional versión 2), se incluye en las categorías que se enuncian a continuación

Distintivo	Clase	Incluida	Modelo ANSI
A	Autorización	SI	C
F	Funciones	SI	C
I	Integridad	SI	C
J	Indicadores	SI	C
Q	Calificadores	SI	C
T	Tipos de Datos	SI	C
B	Operadores Básicos	SI	E
M	Manipulación	SI	E
V	Vistas	SI	E
Z	Operadores Avanzados	SI	E
C	Catálogo	SI	I
E	Comandos para el DBA	SI	I
N	Nombramiento	SI	I
S	Estructura	SI	I
D	Principios de Diseño de SMBDR	NO	
L	Principios de Diseño de lenguajes	NO	
P	Protección de la inversión	NO	
X	Bases de Datos distribuidas	NO	

En la tabla anterior:
Distintivo: se refiere a la letra con que se distingue a las reglas de determinado grupo.
Clase: es el nombre del grupo de reglas
Incluida: indica si el conjunto de reglas se incluyó entre los conjuntos de reglas que se plantearán en la comparación.(Ver Capítulo 3)
Modelo ANSI: se refiere a la capa del modelo ANSI en donde, se encuentra el conjunto de reglas.

En el anexo A se puede encontrar una breve descripción de cada una de las reglas fundamentales del grupo que se emplearán en la comparación de los manejadores. Es necesario aclarar que en este anexo se incluyen únicamente las reglas fundamentales porque son aquellas que en la definición del modelo relacional CODD marca como indispensables, entonces si un manejador no cumple con estas,¿ Cómo puede llamarse relacional? Aún cuando cubra varias de las reglas básicas.

IV Arquitectura ANSI -- SPARC

Está dividida en tres niveles:

A) INTERNO: Es el nivel más cercano al almacenamiento físico. En él se van a manejar los diferentes tipos de registros a almacenar y los índices existentes y como se representan los archivos, que secuencias manejan. Sólo en algunos casos se permitirá trabajar a este nivel, sobre todo a los procesos para mantenimiento de la Base de Datos.

B) EXTERNO: Es el nivel más cercano al usuario. El cual generalmente se encuentra interesado en sólo una parte de la base de datos. Normalmente se tendrá una vista abstracta que no necesariamente corresponde con la manera en que están almacenados los datos, esto se denomina vista externa. El conjunto de definiciones de registros externos forma el esquema externo

C) CONCEPTUAL: Es el nivel que sirve de enlace entre los otros dos niveles, es una representación de todo el contenido de la Base de Datos, al igual que en el nivel anterior el almacenamiento físico no tiene porque concordar, con la manera en que los datos se encuentran almacenados. Las definiciones en el esquema conceptual pretenden almacenar características como chequeo de seguridad, integridad, etc.

Como se puede apreciar en el punto III las reglas se asociaran a un determinado de acuerdo a la naturaleza de los servicios que pretenden regular.

V Arquitectura General de un SMBDR

Para la instrumentación de los SMBDR, Codd enuncia ciertos principios bajo las siguientes reglas:

- No violar ninguna regla fundamental de la matemática
- Representación y acceso a datos transparente para el usuario
- Separar características para manejo semántico y de desempeño
- Independencia en la concurrencia (Ningún usuario puede requerir un tipo de candado específico)
- Protección para cualquier tipo de candado de largo-tiempo
- No sobrecargar estructuras
- Indices basados en dominios
- Estadísticas de la base de datos
- Consulta de las estadísticas
- Cambio del dispositivo de almacenamiento o cambio en el tipo de acceso sin impactar a la aplicación
- Protección automática en caso de mal funcionamiento
- Recuperación inmediata en caso de mal funcionamiento
- Ejecución automática de comandos relacionales
- Respaldo automático
- Evitar el producto cartesiano completo al resolver operaciones
- Responsabilidad de la encriptación y desencriptación

Se pretende que los principios englobados en estas reglas sean respetados por cualquier intrumentación del modelo relacional. Esto debido a que estas reglas de instrumentación se encuentran en concordancia con las reglas que forman el modelo relacional. Para el desarrollo de esta tésis fué necesario conocer la arquitectura y servicios que brinda cada uno de los manejadores a evaluar. En el anexo A se puede encontrar un resúmen de la arquitectura de cada manejador. Basada en estas arquitecturas es que se propone una arquitectura general de SMBDR. Esta arquitectura general se retomará en el capítulo 4 para explicar la metodología de afinación.

V.1 Descripción de los procesos de la arquitectura general.

Corazón del SMBDR

- Controla requerimientos al SMBDR
- Maneja los componentes de recuperación (Buffers y bitácora de la actividad)

Atención de Peticiones

- Manejo de las diferentes operaciones (Inserción, borrado, actualizaciones, así como consultas planeadas o no planeadas)
- Requerimientos de datos (Abrir archivos, manejo de buffers de datos y I/O)
- Cálculo de trayectorias de acceso

Atención del tipo de acceso

- Tipo de Petición:
  Exclusiva
  Compartida
  Exclusiva debida a procesos de mantenimiento

Manejador de requerimientos distribuidos

- Responsable de establecer y monitorear la conexión
- Responsable de manejar la petición y su resultado.

Componentes empleados por los diferentes procesos:

- Bitácora de la actividad: Por lo general se trata de una serie de archivos en los que se registran todas las operaciones que se llevan a cabo sobre la información. Su propósito principal es el de ser usados durante los procesos de recuperación de información debida a fallas, sean estas del propio manejador o el ambiente, así como de fallas debidas al usuario final.

- Buffers de Datos: Area en memoria dedicada a almacenar de manera temporal la información necesaria para atender una petición.

- Información de Objetos Abiertos: Area de memoria en la que se almacenan las definiciones de los objetos que se han requerido por alguna aplicación. La información que reside en esta area se obtiene del catálogo y directorio pero el hacerlo implica un mayor tiempo de respuesta es por ello que es preferible almacenar esta información en la memoria principal.

- Areas para almacenamiento físico: En los discos se encuentra toda la información del SMBDR, por lo general se encuentra asignada en areas lógicas asignadas a determinado dispositivo.

V.II Operación

El SMBD es un software que maneja todos los accesos a la base de datos. Conceptualmente el proceso es:

1.- Un usuario realiza un requerimiento de acceso a los datos
2.- El SMBD lo intercepta y lo analiza
3.- El SMBD inspecciona el esquema externo para el usuario, realiza el mapeo del nivel externo al conceptual, el esquema conceptual, realiza el mapeo del nivel conceptual al interno y definición de la estructura de almacenamiento
4.- El SMBD ejecuta las operaciones necesarias sobre la BD almacenada

El flujo se describe en el siguiente esquema:

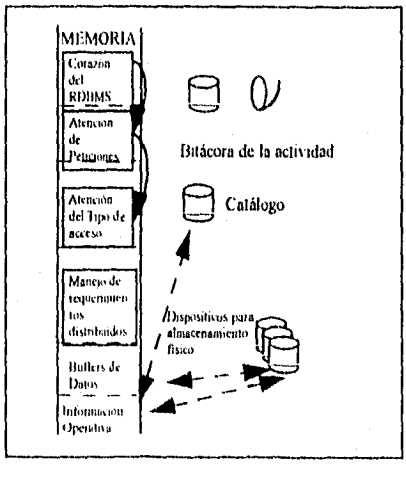


Fig 1

Otras funciones en la operación del SMBD

- Definición de datos
- Seguridad e integridad de datos
- Recuperación y manejo de concurrencia
- Diccionario de datos
- Utilerías
- Proceso Distribuido

V.III Preparación de un requerimiento al SMBD

El modulo de atención de peticiones requiere para atender a las peticiones. Elaborar un plan de acceso a los datos para lo que requiere conocer:

1) Predicados (Tipo y necesidad de proceso)
2) Indices y tablas (Diseño y características de la información que almacenan)
3) Nivel de orden de los datos (Desorden)
4) Estadísticas (Frecuencia de actualización)

En general el optimizador emplea para el cálculo de la ruta de acceso los I/Os estimados ,costo de CPU para resolver cada bloque de consulta(resolución con tablas intermedias,sorts,etc). Cada optimizador emplea esta información de manera diferente. Si bien el proceso de generación del plan de acceso se lleva a cabo tanto en consultas planeadas como no planeadas. Pero en el caso de consulta planeada con código del lenguaje relacional embebido en un programa se requiere prepararlo para que pueda solicitar los servicios del manejador.

En general los manejadores cuentan con una facilidad que llaman EXPLAIN y que permite conocer las decisiones que tomó el optimizador en cuanto al plan de acceso a datos.

Preparando un programa

En este caso estamos hablando de Consulta Planeada y el flujo es el siguiente:

1. Se extraen los postulados de SQL embebidos en un programa, agrupandolos y asignandoles un nombre. Con lo que se genera un programa sin código SQL substituyendo los predicados de SQL por llamados al manejador , además de un módulo con todos los predicados SQL.

2. El programa puede ser compilado y linkeditado

3. El módulo de predicados SQL es procesado por el optimizador, el cual calcula los planes de acceso y los almacena para ser empleados al momento de ejecución.

4. En el momento de ejecución, el programa y los planes de acceso de los predicados, trabajan en conjunto para proporcionar el servicio solicitado por el usuario final.

Preparando una consulta AD-HOC

En el caso de consultas no planeadas el cálculo del plan de acceso por parte del optimizador, se lleva a cabo en el momento de ejecución.

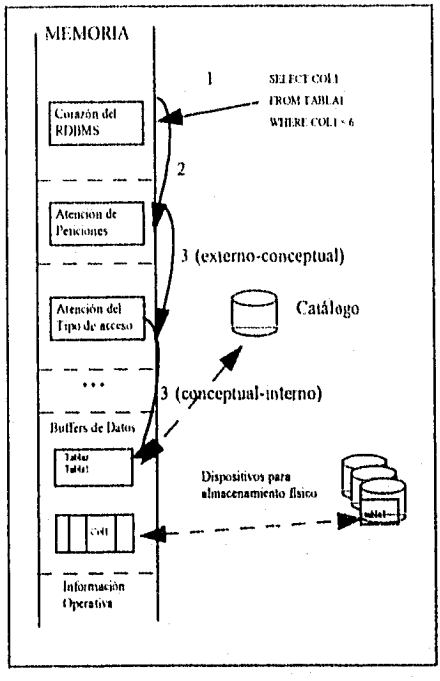


Fig 2

VI Optimización

En la actualidad los manejadores de base de datos cuentan con una pieza de software encargada de determinar la mejor forma de accesar los datos orientada a obtener el mejor desempeño.

Para lograr esto se requiere de cierta información como:

- Tipo de consulta
- Estadísticas de los datos almacenados
- Estructuras orientadas al performance con que cuenta la base de datos

Dependiendo de las variables anteriores, se pueden tener diferentes estimaciones de costos, las cuales se basan en el número de registros accesar y el número de registros estimados a recuperar. Es aquí donde el tipo de consulta y las estructuras orientadas al performance con que cuenta el manejador se conjugan en la solución de peticiones al manejador. Los diferentes tipos de escenarios de optimización se presentan a continuación:

a) Cuando no se cuenta con índices entonces se hace un filtrado suponiendo:

n: número de tuplas
V(A,r): número de valores distintos en la relación para el atributo A.

Y una distribución uniforme de los valores, el número de registros que califican con A es n/V(A,r)

b) Cuando se cuenta con índices. El filtrado es prácticamente el mismo, sólo que hay que recordar que un bloque de índice almacena: apuntadores a los datos y explícitamente la llave, lo que implica menor número de registros índice por bloque y finalmente menos operaciones de lectura y por consiguiente un mejor desempeño.

c) Cuando el tipo de operación implica un producto cartesiano,en general el optimizador contará con varias estrategias:

- Empleo de índices
- Trabajo a nivel de bloque más que a nivel de registro
- Ordenar los registros en almacenamiento temporal

d) Dividir la consulta en varias subconsultas
e) Generar un consulta equivalente

El número y la naturaleza de las estrategias de optimización que contiene un optimizador varía dependiendo del manejador.

VII Tipos de almacenamiento

Como se enunció en el punto anterior parte de las variables que pueden influenciar la elección de la ruta de acceso por parte del optimizador, son los tipos de almacenamiento con que se cuenta. Es por ello que se enuncian algunos de estos y se subraya el número de accesos necesarios dependiendo de la instrumentación. Ya que los accesos a disco consumen más tiempo que cualquier otra operación. A continuación se enuncian diferentes tipos de organización de archivos empleados por los manejadores.

Archivos secuenciales con registros de longitúd fija
Archivos secuenciales con registros de longitúd variable
Archivos secuenciales indexados
Archivos con índices secundarios
Archivos indexados de arbol B^+
Archivos indexados de arbol B

Manejo de Bloques

Uno de los objetivos de la organización de bloques es reducir el número de accesos que se tienen que hacer a disco. A una mejor instrumentación de esta organización menor es el número de accesos a disco.

Manejo de Buffer

El objetivo de un manejador de buffers es que se maximize la probabilidad de que cuando se necesite un bloque de datos, este ya se encuentre en la memoria principal, evitando un acceso a disco. Para lograr esto se pueden emplear diferentes técnicas como:

- Estrategia de reemplazo
- Bloques sujetos
- Salida forzada de bloques

Parámetros para medir la mejor técnica de acceso.

1. Tiempo de acceso: Tiempo que se lleva en localizar un dato determinado

2. Tiempo de inserción: Tiempo que se lleva en insertar un dato determinado

3. Tiempo de eliminación: Tiempo que se lleva en eliminar un dato determinado

4. Espacio extra: El espacio adicional que ocupa la estructura para su manejo.

VIII SQL

Al igual que para la instrumentacion Codd enuncia algunas reglas para el lenguaje relacional como:

- Debe estar al alcance de una variedad de usuarios, tanto programadores como no programadores.
- El DBMS debe aceptar la compilacion de los comandos del LR, incluso la recompilación automática en el caso de cambios en la ruta de acceso, métodos de acceso o indexacion
- Debe poder interactuar con el lenguaje anfitrion
- Los comandos embebidos deben poder compilarse inmediatamente o diferir la compilación
- Debe poder ser un lenguaje fuente (Codificado fácilmente por un usuario) y lenguaje objeto (codificado por algún producto de mayor nivel)
- El entorno de operadores,comparadores, funciones,conectores lógicos, calificadores e indicadores debe concordar con una regla simple y comprensible
- Cuando se trate de un bloque de comandos este debe delimitarse por una cláusula de BEGIN y otra de END.
- Si una característica semántica es soportada en un contexto debe ser soportada en todos.
- El LR debe estar ligado al cálculo relacional
- Debe soportar el manejo de comando a nivel conjunto
- Uso de queries anidados y definición de constantes a nivel comando debe permitir definirlas
- Debe existir sólo una manera de expresar un requerimiento
- Debe realizarse sólo una optimización global
- Si se puede expresar un requerimiento lógicamente equivalente a otro el optimizador traduce la ruta de acceso a una forma canónica equivalente
- Se debe poder reemplazar constantes en comandos del lenguaje relacional por variables del lenguaje anfitrion o funciones
- Debe soportar condicionesen que su validez dependa del tiempo
- Permitir la subordinación de comandos por ejemplo subordinar un UNION a un JOIN.

De todos los lenguajes interrogadores propuestos como: QBE, ISBL, SQUARE, SEQUEL o QUEL

El SQL (Structured Query Language) es el que ha tenido mayor éxito a nivel comercial. Ya que la mayoría de los manejadores cuentan con su versión de SQL. Si bien se han hecho esfuerzos por contar con un standard de SQL, cada manejador mantiene una versión propia del lenguaje. Sin embargo en general las instrumentaciones de SQL presentan las siguientes deficiencias:

a) Permite que existan tuplas repetidas en las relaciones
b) Expresiones equivalentes en su lógica, que en teoría deberían arrojar el mismo conjunto de resultados difieren en estos.
c) No soportan de manera adecuada el manejo de lógica de tres y cuatro valores.

IX Plataformas

Por último se ilustran dos tipos de variables que pueden determinar el tipo de ambiente en que se desempeñe el manejador.

HARDWARE

MAINFRAME	MINI	PC
DB2	DB2/6000	DB2/2
SYBASE	SYBASE	SYBASE
ORACLE	ORACLE	ORACLE

SISTEMA OPERATIVO

MVS	UNIX	OS/2- DOS
DB2	DB2/6000	DB2/2
	SYBASE	SYBASE
	ORACLE	ORACLE

COMPARACION DE TRES MANEJADORES DE BASES DE DATOS RELACIONALES

El presente capítulo mostrará las deficiencias existentes en cuanto al cumplimiento de las reglas de Codd de los siguientes manejadores de base de datos:

ORACLE 7.0
SYBASE 10.0
DB2 V2.3

Con el propósito de señalar que las necesidades de información han rebasado a las instrumentaciones del modelo relacional, mas no así al modelo mismo. Además de mostrar que el pobre desempeño se debe a su vez a este distanciamiento del modelo relacional.

I Descripción de la forma de comparación.

La comparación se llevará a cabo en base a los grupos de reglas enunciados en el capítulo anterior. Siendo las reglas fundamentales los puntos de comparación entre manejadores, las tablas que se incluyen a continuación tienen por objeto mostrar de manera simple las diferencias y semejanzas en los manejadores. En cada una de las tablas se enuncia el tipo de reglas a comparar, bajo esta columna se nombra el tipo de regla de que se trata. La nomenclatura de las mismas se puede encontrar en el capítulo anterior. Bajo el nombre de cada manejador se especifica un SI, si en base a las características que el manejador presenta se cumple la regla, se especifica un NO en caso contrario. Como en algunos casos no se puede decir que la regla no se contemple o se cubra en su totalidad, se incluyen comentarios en donde se explican las limitaciones en que la regla es respetada o no, estos se señalan como ° para DB2, * para ORACLE y ¢ para SYBASE Así también se incluyen conclusiones parciales por conjunto de reglas.

La clasificación en torno al modelo ANSI se retoma del capítulo anterior. Por esta razón el orden a seguir en la descripción de las reglas es nivel EXTERNO, CONCEPTUAL e INTERNO.

II Nivel Externo

En cuanto a la instrumentación esta es la parte del modelo que se encuentra más cercana al usuario final. Es importante señalar que los tres manejadores que se compáran emplean como lenguaje relacional el SQL. Por lo que las deficiencias del lenguaje se reflejan en el incumplimiento de reglas para operadores, manipulación y vistas.

II.I Operadores Básicos

Operadores Básicos	DB2	ORACLE	SYBASE
RB-1	SI	SI	SI
RB-2	NO"	NO*	NO¿
RB-3 RB-12	NO"	NO*	NO¿
RB-14 RB-23	SI	SI	SI
RB-25	SI	SI	SI
RB-26	SI	SI	SI
RB-27	SI	SI	SI
RB-28	SI	SI	SI
RB-29	SI	SI	SI
RB-30	NO"	NO*	NO¿
RB-31	SI	SI	SI
RB-32	SI	SI	SI
RB-35	SI	SI	SI

Comentarios:

"* **RB-2** Se aceptan duplicados, por lo que la regla no se contempla al no arrojar una relación como resultado

"* **RB-3 RB-12** Se aceptan duplicados, por lo que se tiene la misma situación que en el caso anterior.

*¿ **RB-30** La persistencia de las tablas con resultados intermedios se limita a la duración de la sesión, por lo que no se puede hablar de persistencia, así como tampoco se almacena la definición de la tabla en el catálogo.

" **RB-30** La asignación de resultados intermedios normalmente no se realiza mediante comando del RSMBD, sino como parte de las extensiones de software del mismo.

Conclusiones:

El hecho de que los operadores básicos trabajen en un ambiente en el que se permiten los duplicados, constituye una de las principales fallas en la instrumentación. Esto implica que en algunas ocasiones una interpretación correcta de los resultados que arroja una operación, requerirá de un conocimiento profundo de la manera en que el manejador desarrolla una operación. Bajo el esquema del modelo ANSI en el nivel externo el usuario no tiene que preocuparse por la manera interna de trabajar del manejador. Bajo el punto de vista del modelo relacional al usuario estas consideraciones le deben ser transparentes.

II.II Operadores Avanzados

Operadores Avanzados	DB2	ORACLE	SYBASE
RZ-1	NO °	NO *	NO ¢
RZ-2	NO	SI	NO

Comentarios:

°*¢ **RZ-1** Cubre con GROUP BY pero no mas de un agrupamiento y no numera

RZ-2 No es una función soportada ya que en el caso de columnas repetidas, califica ambas con el nombre de la relación, en el caso de una extensión.

Conclusiones:

En general se puede decir que no se soportan los operadores avanzados descritos en las reglas fundamentales. Ya que se encuentran restringidos por las condiciones que deben cumplir los datos sobre los que se va a operar. Con el propósito de obtener el resultado esperado. Por ejemplo se debería incluir una columna que fuese numérica para poder tener un identificador único ya que no se tiene una columna que se agregue y que cumpla con el propósito del FID(Frame Identifier) Identificador del marco.

II.III Manipulación

Manipulación	DB2	ORACLE	SYBASE
RM-1	SI	SI	SI
RM-2	NO °	NO *	NO ¢
RM-4	SI	SI	SI
RM-5	NO°	NO	NO ¢
RM-6	SI	SI	SI
RM-8	NO °	NO *	NO ¢
RM-9	SI	SI	SI
RM-10	NO	NO	NO
RM-11	NO °	NO *	NO ¢
RM-12	SI	SI*	SI
RM-13	SI	SI	SI
RM-14	NO °	NO *	NO ¢
RM-15	SI	SI	SI
RM-18	SI	SI	SI

Comentarios:

°*¢ **RM-2** El resultado no es una relación, por lo demás el lenguaje de manipulación es analizable sintácticamente

°*¢ **RM-5** Al romper la regla de estructura RS-3, la cerradura en los operadores que son soportados por el RSMBD , no se mantiene ya que

al efectuar una operación entre dos relaciones no se obtiene necesariamente una relación.

°*¢ **RM-8** No permite creación de dominios, si bien para la creación de tablas no requiere tener al manejador fuera de línea , no se pueden accesar datos en tanto el proceso de creación se lleva a cabo. La reconstrucción de rutas de acceso tiene efecto si bien de manera transparente para el usuario.

°*¢ **RM-11** Puede representar un valor faltante pero no lógica de cuatro valores

* **RM-12** No se hace distinción de valores nulos. Emplea una función denominada DECODE

°*¢**RM-10** Ninguno de los manejadores evaluados maneja lógica de cuatro valores.

°*¢**RM-14** Al no contar con definiciones de dominios el tipo de datos básico, ocupa el lugar del dominio y una operación DCO (Domain Check Override) carece de sentido.

Conclusiones:

Las carencias en cuanto al manejo de la información son las más importantes en todos los manejadores comparados. El manejo de renglones duplicados interfiere con la buena instrumentación de las reglas de manipulación. El mal manejo de los tipos de datos restringe al manejador, ya que los únicos tipos de datos con los que se puede trabajar son aquellos que están soportados por el código del manejador, sin permitir la adhesión de nuevos tipos de datos definidos por el usuario. El manejo de lógica de cuatro valores y la falta de representación adecuada para la información faltante son puntos importantes que mantienen apartadas a las instrumentaciones del modelo.

II.IV Vistas

Vistas	DB2	ORACLE	SYBASE
RV-1	SI°	SI *	SI ¢
RV-2	SI	SI	SI
RV-3	NO°	NO *	NO ¢
RV-4	NO °	NO *	NO ¢
RV-5	SI	SI	SI
RV-6	SI	SI	SI
RV-7	NO°	NO *	NO ¢
RV-8	NO °	NO *	NO ¢

Comentarios:

"*¢ RV-1 Sólo acepta definición sobre tablas R-base
"*¢ RV-3 No acepta lógica de cuatro valores
"*¢ RV-4 No todas las operaciones se permiten en las vistas
"*¢RV-7 No permite renombrar ninguna columna en la vista
"*¢RV-8 No se almacena en el catálogo ninguna información acerca del dominio del cual deriva sus valores la columna de la vista.

Conclusiones Nivel Externo:

El hecho de aceptar duplicados, no tener un manejo adecuado de la lógica de cuatro valores y el no manejar de manera transparente relaciones base y vistas; ocasiona que un usuario final no cuente con una vista externa es decir una vista de la base de datos en que no se requiere conocer la forma de almacenamiento de los datos o contar con información de como realizará el manejador ciertas operaciones, o de los tipos de datos que pueden manejar que operaciones, etc.

En lo tocante al desempeño todo el proceso extra que se requiere para cubrir las deficiencias del manejador, como es el hecho de que los resultados intermedios no se pueden hacer persistentes por lo que el programador requiere guardar estos para emplearlos en un proceso posterior.

III Nivel Conceptual

En este nivel se agruparon las reglas que están ligadas a las especificaciones de las entidades. Como autoridades, reglas de integridad, funciones, calificadores, indicadores y tipos de datos. La instrumentación de cada conjunto de reglas varia considerablemente de acuerdo al manejador.

III.I Autorización

Autorización	DB2	ORACLE	SYBASE
RA-1	SI	SI	SI
RA-2	NO°	NO ^	NO ¢
RA-6	NO°	NO *	NO ¢
RA-7	NO °	NO *	NO ¢
RA-8	SI	SI ¢	SI ¢
RA-10	NO°	SI ¢	SI

Comentarios:

*°¢ **RA-2** No emplea lógica de cuatro valores. No hay ninguna facilidad para otorgar autoridades de manera temporal.

*°¢**RA-6** Los manejadores al efectuar un DROP no permiten una recuperación posterior , ni del objeto ni de los objetos dependientes.

*°¢ **RA-7** No soporta todos los tipos de autoridades que exige el modelo relacional. Al no soportar toda la creación de objetos que exige el modelo relacional, no son necesarias todo el conjunto de autoridades que exige el modelo relacional.

* **RA-8** Si excepto restricciones de cambiar valores nulos y modificar la llave primaria , no las soporta

° **RA-10** No se pueden definir grupos de usuarios en DB2

* **RA-10** Si bien en ORACLE no se pueden definir grupos de usuarios se pueden definir perfiles comunes.

Conclusiones:

Las reglas de autorización no se encuentran cubiertas en su totalidad. La mayoría de las fallas se dan en torno a que no se soportan todos los tipos de autoridades que presupone el modelo relacional. Tal es el caso de no poder asignar autoridades de manera temporal. Así como el de asignar autoridades sobre estructuras no soportadas por las instrumentaciones como dominios.

III.II Funciones

Funciones	DB2	ORACLE	SYBASE
RF-1	SI	SI	SI
RF-2	NO°	NO *	NO ¢
RF-8	SI	SI	SI

Comentarios:

***°¢RF-2** El (DOD) Grado de Duplicidad de los datos no es una característica de los manejadores analizados.

Conclusiones:

El hecho de soportar renglones duplicados, hace irrelevante para la instrumentación el soportar el (DOD) Grado de Duplicidad de los datos.

III.III Restricciones de integridad

Restricciones de integrilad	DB2	ORACLE	SYBASE
RI-1 RI-5	NO°	NO *	NO¢
RI-6	NO	SI*	SI¢
RI-11	SI°	SI*	SI¢
RI-12	NO°	NO *	NO¢
RI-16	SI	SI	SI
RI-19	NO°	NO *	NO¢
RI-22	SI	SI	SI

Comentarios:

°*¢ **RI-1 RI-5** No se tienen todos los tipos de restricciones de integridad

*¢ **RI-6** La integridad referencial se puede deshabilitar

°*¢ **RI-11** La validación tiene sus restricciones en cuanto a dominios y su dependencia con respecto a los indicadores.

°*¢ **RI-12** No se tienen las marcas de A o I. Es decir se puede restringir NOT NULL pero o se puede restringir por NOT A o NOT I

°*¢ **RI-19** Permite reemplazar nulos en donde había valores, sin embargo si una columna se redefine como no nulo se rechaza durante la carga toda la información que sea nulo

Conclusiones:

La mayoría de las reglas fundamentales están cubiertas, si bien en algunos casos de manera parcial. En el caso de DB2 la regla **RI-6** no se cumple porque la temporalización no se tiene como una característica del manejador. Así como tampoco aquellas que se refieren a la lógica de cuatro valores.

III.IV Restricciones de integridad definidas por el usuario

Restricciones de integridad definidas por el usuario	DB2	ORACLE	SYBASE
RI-32	SI°	SI	SI
RI-33	SI	SI	SI
RI-34	SI	NO	NO

Comentarios:

" **RI-32** No hay chequeo temporalizado de restricciones de integridad El DB2 es el único manejador que maneja marcas cuando la regla de intergridad se viola

Conclusiones:

La mayoría de las reglas de integridad se encuentran incorporadas a la instrumentación, sin embargo no se puede asegurar que alguno de los manejadores las cubra completamente, tan sólo la integridad referencial se encuentra soportada y aún esta con ciertas limitantes, tal es el caso de las reglas de borrado o actualización en el manejo de integridad, así como el manejo de la temporalización de la integridad

III.V Indicadores

Las diferencias existentes a nivel de indicadores para los diferentes manejadores no se incluyen como conjunto de reglas a comparar en la presente tésis. Ya que los indicadores que utilizan los manejadores varían considerablemente de acuerdo a la intrumentactón. Las reglas enunciadas para indicadores no se cumplen en su mayoría excepto RJ-1, RJ-3 y RJ-4 que todos los manejadores incluyen. Las reglas no se cumplen ya que se refieren a estructuras no soportadas como dominios, a valores de la lógica de cuatro valores, manejo de ciertas operaciones sobre vistas o violación del principio de no-renglones duplicados.

III VI Calificadores

Calificadores	DB2	ORACLE	SYBASE
RQ-3	NO	NO	NO¢
RQ-7	SI	SI	SI
RQ-11	NO	NO	NO

Comentarios

¢ **RQ-3** Asigna una leyenda o valor que el usuario determine para los nulos

RQ-3 Los manejadores no soportan lógica de cuatro valores, por lo que el calificador MAYBE podría encontrar su equivalente con IS NULL en la lógica de tres valores que soportan los manejadores.

RQ-11 No se cuenta con el DOD (Degree of Duplication) que indica el número de veces que se repite un renglón para las relaciones derivadas.

Conclusiones

Sólo el calificador de orden es soportado en su totalidad. Los calificadores ligados con la representación de información faltante y tuplas repetidas no son soportados por ninguno de los manejadores.

III.VII Dominios y Tipos de datos extendidos

Dominios y Tipos de Datos Extendidos	DB2	ORACLE	SYBASE
RT-1	SI	SI	SI
RT-2	[a]	*	SI
RT-3	NO	NO	NO

Comentarios:

[a]* **RT-2** No se acepta el tipo monetario(currency) como tipo de dato
RT-2 No se tienen las definiciones de Tipo de Dato Abstracto (TDA) que el modelo requiere, como se puede apreciar en el conjunto de reglas de estructura. Ni aun como parte del sistema
RT-3 No se permite definir TDA por parte del usuario

Conclusiones:

La definición de tipos de datos abstractos no se contempla como parte de ningún manejador. Los únicos tipos de datos que el manejador permite utilizar son los que este proporciona como parte del mismo. El usuario no puede definir nuevos tipos de datos o dominios.

Conclusiones Nivel Conceptual:

En la parte del nivel conceptual queda aún mucho por hacer, ya que la vista de la base de datos a este nivel es prácticamente la misma que en el nivel interno. Salvo las características de integridad y autorización que se manejan en este nivel, con las limitantes que ya se expusieron. Sin embargo el mapeo en el caso de tipos de datos sigue siendo el mismo que en el nivel interno.

En resumen tenemos que se cuenta con una pobre instrumentación de las restricciones de integridad ya que estas no se encuentran instrumentadas en su totalidad al igual que la capacidad de definir tipos de datos abstractos. Estas dos restricciones hacen ver como si el modelo relacional ya hubiese sido rebasado. Sin embargo el hecho de no soportar el DOD acarrea preproceso de datos y ciclos extra, así como manejo de las funciones para obtener una información que el manejador debería proporcionar. Por otro lado hay que señalar que las instrumentaciones actuales de las reglas de integridad referencial en la mayoría de los casos impactan de manera negativa al desempeño.

IV Nivel Interno:

En este nivel se lleva a cabo el almacenamiento físico, las reglas que se dedujo están ligadas a este nivel son las reglas del catálogo, en donde se almacena toda la información de las estructuras con que trabaja el SMBD para almacenar la información Así también tenemos las reglas de nombramiento que pretenden regular los nombres que se empleen para cada uno de los objetos que se crean en el manejador con el fin de evitar inconsistencias debidos a duplicidad de nombres de objetos. En cuanto a los archivos en que residen los objetos del manejador , se requiere contar con procesos especiales para su mantenimiento.

IV.I Catálogo

Catálogo	DB2	ORACLE	SYBASE
RC-1	SI	SI	SI
RC-2	SI	SI	SI
RC-3	NO	NO	NO
RC-4	SI	SI	SI
RC-6	SI	SI	SI
RC-8	NO°	SI*	SI
RC-10	SI	SI	SI
RC-11	SI	SI	SI

Comentarios:

* **RC-2** Puede ocasionar degradación del desempeño el hacer consultas simultáneas sobre el catálogo

° **RC-8** No hay manejo de trigger

* **RC-8** Almacena las restricciones de integridad referencial pero los triggers los almacena en archivos independientes

RC-3 Al no contar con la regla RS-3 el almacenar información de dominios en el catálogo carece de sentido.

Conclusiones:

Casi todas las reglas del catálogo se cubren , excepto aquellas ligadas a la definición de dominios y de restricciones de integridad, sobre todo las ligadas a triggers y manejo de excepciones. En las reglas del catálogo no se estipula que se deba guardar información relativa al espacio físico de la base de datos sin embargo algunos manejadores si lo hacen.

IV.II Comandos para el DBA

Comandos para el DBA	DB2	ORACLE	SYBASE
RE-3	NO	NO	NO
RE-6	NO	NO	NO
RE-7	SI	SI	SI
RE-9	SI	SI*	SI
RE-10	NO "	NO*	NO ¢
RE-13	NO	NO	NO
RE-14	SI	SI	SI
RE-16	SI	SI	SI

Comentarios:

¢°* RE-10 No existe la diferencia entre faltante aplicable o no

¢°*RE-3 Al no tener la regla de estructura RS-6 el comando de creación de dominio no existe ya que no tiene razón de existir.

¢°*RE-6 De igual manera el comando de borrado de dominio tampoco tiene sentido.

¢°*RE-13 No se cuenta con un comando para borrar una columna, cuando esto es requerido es necesario borrar la entidad y volverla a crear.

Conclusiones:

Estos comandos se orientan al mantenimiento de la base de datos. Es decir a realizar cambios tales como: cambio en línea de la estructura de las tablas. Sin embargo estos comandos no se encuentran instrumentados en un cien por ciento, ya que el manejo de dominios no es una función soportada por ninguno de los manejadores evaluados. Así también todos los cambios relativos a la estructura de las tablas no son soportados por ninguno de los manejadores evaluados.

IV.III Nombramiento

Nombramiento	DB2	ORACLE	SYBASE
RN-1	°	*	¢
RN-2	SI	SI	SI
RN-3	SI	SI	SI
RN-4	NO	NO	NO
RN-5	SI	SI	SI
RN-6	NO	SI	NO
RN-7	NO"	NO	NO¢

Comentarios:

"¢* **RN-1** Sólo se tienen los dominios ya definidos

"*¢ **RN-7** Se requiere calificar la columna para distinguir identidades

RN-4 Todos los manejadores permiten desprender el nombre de la columna de la relación lo que conduce a ambigüedades.

RN-7 Cuando se involucran entidades con nombres comunes en las columnas , es necesario calificar las columnas para mantener la conmutatividad.

Conclusiones:

Al igual que otros conjuntos de reglas el hecho de que los manejadores no soporten dominios, implica que las reglas no se cumplen en su totalidad. El nombramiento tiene fallas, tal es el caso de las columnas al no identificar a las columnas como *tabla.columna*, además de no soportar la conmutatividad al momento de solucionar interrogaciones.

IV.IV Estructura

Estructura	DB2 V2.3	ORACLE 7	SYBASE
RS-1	SI	SI	SI
RS-2	SI	SI	SI
RS-3	NO	NO	NO
RS-4	SI	SI	SI
RS-6	NO	NO	NO
RS-7	SI°	SI*	SI¢
RS-8	SI	SI	SI¢
RS-10	SI	SI	SI
RS-13	SI	SI	SI

Comentarios:

"¢ **RS-7** No es posible establecer restricciones de integridad a nivel columna, sin embargo existen algunas restricciones preestablecidas como UNIQUE, CHECK.

"*¢ **RS-8** Se pueden definir tablas base sin especificar llave primaria

RS-3 No se cuenta con tablas tanto base como derivadas que no tengan renglones duplicados

RS-6 No se tiene la facilidad de contar con definiciones de dominio (TDA: Tipo de dato abstracto)

Conclusiones:

Este conjunto de reglas se encuentra más ligado al almacenamiento físico y a la naturaleza de la información que se almacena en las tablas. Nuevamente las dos grandes fallas en la instrumentación de los manejadores, se refleja en las fallas en la instrumentación de las reglas de estructura.

Conclusiones Nivel Interno:

Las reglas englobadas como parte del nivel interno, no describen como se debe llevar a cabo la instrumentación, es decir el almacenamiento no se delimita en estas reglas. Sin embargo si regula el tipo de información que se debe almacenar de las relaciones así como la naturaleza de las mismas

Si bien en todas las instrumentaciones se pueden guardar estadísticas de los datos almacenados en las tablas, estas no se tienen actualizadas ya que es necesario correr procesos extra para hacerlo. El hecho de no contar con estadísticas actualizadas puede llevar al optimizador a elegir una ruta de acceso no óptima e impactar directamente al desempeño. Por otro lado los cambios en las estructuras de las tablas en la mayoría de los casos requiere sacar de línea las tablas y en el peor de los casos borrar y volver a crear las estructuras, con todos los controles que ello implica.
En las violaciones a las reglas de estructura se puede resumir dos de las fallas principales de las instrumentaciones:

1.- Admitir renglones duplicados
2.- No poder definir dominios

Si bien sólo el manejo de renglones duplicados se reconoce como una causa de degradación del desempeño.

V Conclusión General de la comparación

Como se puede observar, en las tablas comparativas las consideraciones en cuanto a las reglas soportadas por uno u otro manejador no tienen una variación considerable. Es importante recordar que el objetivo de la presente tesis , no es determinar si un manejador es mejor que otro en base a número de reglas que cubra, sino mostrar las diferencias existentes entre las diferentes instrumentaciones y el modelo relacional. Todo esto con el fin de poder observar las implicaciones que estas desviaciones acarrean sobre el desempeño de los manejadores y aplicaciones.

Si retomamos los módulos de la arquitectura general de un SMBDR, veremos que cada uno de los componentes se ven afectados por el incumplimiento de las reglas del modelo relacional como es el caso de:

Corazón del SMBDR al recibir como entrada peticiones en SQL, ya se vió que una de las principales fallas es el responder con diferentes conjuntos de resultados a peticiones lógicamente equivalentes.

Atención de peticiones es probablemente el componente más afectado, ya que tiene que lidiar con registros duplicados, cálculo de trayectorias basadas en estadísticas antiguas, reglas de integridad referencial que consumen muchos recursos y proceso extra para resultados intermedios.

Atención del tipo de acceso debe ser capaz de manejar dos tipos de peticiones las que se marcan claramente en el modelo relacional en cuanto a candados, pero además debe ser capaz de asignar un tipo de candado especial sobre todo para procesos de mantenimiento o que requieran cambios a las estructuras. Lo cual va encontra del modelo relacional e impacta a las aplicaciones en cuanto a la disponibilidad de la información.

Manejador de requerimientos distribuidos, el cumplimiento de las reglas para proceso distribuido se dejaron de lado ya que los manejadores analizados se encuentran realizando cambios en vista de soportar procesos distribuidos.

Componentes empleados por los diferentes procesos en general la probabilidad de impacto que se aprecia es en relación con el volúmen y el manejo de registros duplicados.

El porque se realizó esta comparación, más que mostrar el impacto al desempeño por la falta de apego a la instrumentación, pretende establecer que las instrumentaciones no son tan diferentes entre si, y que es posible desarrollar una metodología para mejorar el desempeño de manera general, como lo veremos en el siguiente capitulo.

En la literatura revisada para futuras instrumentaciones de los manejadores antes mencionados se contempla:

- Dominios
- Integridad a nivel de columna
- Triggers
- Instrumentación de algunas de las reglas para el manejo de bases de datos distribuidas
- Tolerancia en el manejo de registros repetidos y no se observa ningún esfuerzo orientado a desaparecerlo.

"Speed,quality,price. Pick any two."
James M. Wallace

LA METODOLOGÍA DE AFINACIÓN PARA LOS MANEJADORES DE BASES DE DATOS RELACIONALES

El objetivo de este capítulo es obtener una metodología de afinación que pueda emplearse para afinar un ambiente de bases de datos relacional. La metodología se ilustrará tomando como base la arquitectura general descrita en el capítulo 2.

1 Metodología de afinación enfocada a la arquitectura general

Retomando la definición de eficiencia: el resultado esperado en el tiempo esperado con los recursos esperados. En el entorno de la metodología de afinación diremos que eficiencia es obtener el mejor desempeño al más bajo precio. Por ello es importante hacer notar la importancia de contar con una metodología de afinación y un proceso para llevar a cabo la afinación. Dada la complejidad de los sistemas proporcionados al dia de hoy no es tan sencillo determinar con exactitud cuales son las causas de que un sistema consuma más recursos de los necesarios, lo que representa pérdidas tanto monetarias como de credibilidad en las soluciones al emplear nuevas tecnologías. Es por ello que es imperativo contar con un metodología que nos permita enfrentarnos a la diversidad de variables que pueden afectar el desempeño de un RDBMS.

Para empezar a hablar de una metodología para eficientar el desempeño de un manejador (mayor información se tiene en la referencia (10)):

1.Enunciar las metas y definir el sistema. Es decir delimitar el objeto a medir y el ambiente a medir. Identificar los parámetros significativos

2.Listar servicios y salidas. Por ejemplo un SMBDR puede contestar una consulta correctamente, incorrectamente o no contestarla.

3.Seleccionar las medidas que se tomarán. En general se relacionan con velocidad, confiabilidad y disponibilidad del servicio.

4.Listar los parámetros que afectan al desempeño estos pueden ser parámetros del sistema (software y hardware) o parámetros de carga del sistema (requerimientos del usuario)

5.Seleccionar los factores de estudio. Los parámetros se pueden dividir en dos aquellos que variarán durante la evaluación y aquellos que no. Los parámetros que varian se denominan factores y sus valores niveles. Los mejores factores son aquellos que se espera impacten fuertemente al

sistema. Al seleccionar los factores es importante considerar las restricciones económicas, políticas y tecnológicas. Así como también aquellas impuestas por personas que toman la decisión y el tiempo disponible. El tomar en cuenta estas consideraciones incrementa la probabilidad de encontrar una solución aceptable e instrumentable.

6.Seleccionar la técnica de evaluación. Las tres técnicas son modelado analítico, simulación y medición de un sistema real. La selección de la técnica adecuada depende de la cantidad de tiempo y servicios con que se cuente.

7.Seleccionar la carga de trabajo: consiste de una lista de peticiones al sistema. Ej comparar una serie de queries bajo ciertas cantidades de memoria dependiendo de la técnica que se emplee se obtendrán determinados resultados.

8.Diseños experimentales. En la primera fase se cubren más factores con menos niveles. En la segunda fase se cubren factores más significativos con más niveles.

9.Analizar e interpretar los datos: El resultado provee las bases sobre las cuales se elaborarán conclusiones

10.Presentar resultados

En nuestro caso lo que nos interesa medir es la eficiencia de un SMBDR, que es el encargado de resguardar y proporcionar la información que se le solicite. Los servicios del manejador deben ser proporcionados en un mínimo de tiempo con un consumo mínimo de recursos en un sistema ideal, sin embargo sabemos que existen variables que afectan el desempeño.

Variables como:

I. HARDWARE
   a) Modelos de máquina
      (1) Capacidad de memoria
      (2) Velocidad de proceso
      (3) Marca / velocidad del reloj
   b) Modelos de dispositivos de almacenamiento
      (1) Capacidad de almacenamiento
      (2) Velocidad de transferencia de información

2 SOFTWARE
   a) Sistema Operativo
      (1) Interfaz de usuario
      (2) Manejador de operaciones de I/O
      (3) Control de Seguridad
      (4) Control de concurrencia
      (5) Tipo de servicios de almacenamiento
      (6) Organizaciones de archivos
3 OPERATIVAS
   a) Nuevos requerimientos
   b) Operaciones en linea
   c) Operaciones en lote
   d) Mezcla de ambas
   e) Volumen
4. TIEMPO
   a) Madurez y evolución del sistema
   b) Tiempo comprometido de disponibilidad de la aplicación
   c) Tiempo comprometido de respuesta
5. Características de los componentes del manejador de base de datos
   a) Manejo de memoria y buffers
   b) Tipo de organización de archivos soportada
   c) Tiempos promedio de atención de peticiones en condiciones ideales

Bajo este esquema es importante hacer notar que en todo sistema vivo estas variables pueden cambiar en cualquier momento. Es por ello que los procesos de afinación son parte de la vida útil del sistema. Si bien un sistema puede trabajar sin ser afinado, generalmente los servicios que presta están por abajo de las expectativas de los usuarios. En la literatura revisada encontramos diferentes estrategias de afinación las cuales se enfocan a un manejador en particular. Este capítulo pretende realizar generalizaciones independientes del tipo de manejador de que se trate. En los anexos B y C se pueden encontrar extractos de las metodologías revisadas para ORACLE y SYBASE. Con el propósito de evitar obtener una metodología que sólo aplique para cierto manejador es que se decidió enfocarla sobre la arquitectura general. En el siguiente capítulo se ilustra el uso de la metodologia sobre el manejador DB2.

Para delimitar aún más el entorno la afinación la describiré a dos niveles, además hay que aclarar que este no pretende ser un proceso de única vez , sino que será un proceso cíclico:

Subsistema : El manejador como tal.

Aplicaciones : Cada uno de los sistemas que requieren servicios del manejador.

II Afinación de Subsistema

Cuando hablamos de afinación a nivel de subsistema nos referimos a todos los componentes que integran el manejador, así como los recursos del sistema operativo que el manejador requiere como:

- MEMORIA
- MANEJO DE I/O
- SEGURIDAD
- CONCURRENCIA
- ALMACENAMIENTO FISICO

Estos recursos en su mayoría se pueden dimensionar al momento de inicio del sistema indicando:

- Tamaño de Buffers para:
  1. Bitácora (LOG)
  2. Datos
  3. Información de datos
  4. Operación
- Estimados del número de:
  5. Usuarios concurrentes
  6. Objetos abiertos empleados concurrentemente
- Frecuencias de:
  7. Puntos de sincronía que tomará el manejador
  8. Captura de la Información para monitoreo

Algunos otros parámetros se dimensionarán durante la vida del manejador, estas variables no podrán ser restringidas por algún valor que asigne el manejador. Las variables que se identificaron son las siguientes:

- Tiempo de respuesta de discos, se ve afectado por:
  1. Distribución de la información en los diferentes dispositivos
  2. Acceso óptimo a las áreas de trabajo
  3. Tipo de dispositivos a emplear
- Seguridad
  4. En general el sistema operativo permitirá el acceso o no al manejador, y el manejo de los objetos definidos en el manejador queda controlado por el mismo.
  5. El sistema limitará el acceso a los componentes del manejador como archivos de bitácora, catálogo, etc.
- Concurrencia
  6. Asegurar que se tienen los recursos para atender a varios requerimientos del manejador.

II.1 Análisis a nivel de subsistema.

Una vez que se han determinado las variables con que estan trabajando los diferentes componentes del subsistema. Hay que determinar cual de estos parámetros requiere cambios.

Buffers

Si el manejador de aplicaciones requiere bajar de memoria varios bloques de información. con el fin de tener espácio para los bloques relativos a nuevos requerimientos. Entonces es probable que se requiera incrementar el tamaño de los buffers.

Bitácora

Si el manejador envia el mensaje que indica bitácora llena durante el tiempo en que las transacciones en línea están siendo procesadas entonces tal vez sea necesario incrementar el tamaño del archivo de bitácora para evitar espaciar el evento de que el manejador requiera vaciar su bitácora.

Puntos de sincronía

Son acciones que lleva a cabo el manejador en donde toda la información que pudiese estar en riesgo por no estar respaldada físicamente, es respaldada en almacenamiento persistente y este evento se registra en la bitácora. Mientras más frecuentemente se lleven a cabo estas acciones los procesos de recuperación serán más confiables, pero cada vez que se toma un punto de sincronía hay que recordar que se tiene prioridad en el manejador y por lo tanto hay un impacto al desempeño. Por ello dependiendo de la estrategia de recuperación de información que se tenga, convendrá espaciar los puntos de sincronía.

Obtener infomación para el monitoreo

Los procesos que capturan información del subsistema para monitorearlo, no deben afectar el desempeño del manejador. De ser así hay que llevar un control extricto de los procesos que lo impacten y evaluar en que casos es conveniente activarlos, dependiendo de la información que se pretende obtener de ellos.

Usuarios concurrentes

Si existe una manera por parte del manejador de limitar la cantidad de usuarios concurrentes cuando los eventos de deadlock son muy altos, esto es conveniente hacerlo como una medida temporal en tanto se descubre la manera en que los procesos que incurren en deadlock puedan convivir. De esta manera lo que se hace es encolar los requerimientos hacia un mismo recurso de tal manera que el proceso que se desarrolle sea efectivo y no consuma ciclos obsoletos que terminarán en deadlock.

II.II Instrumentación

Es recomendable cambiar los parámetros asociados al tipo de servicio que proporciona el manejador todos a la vez. Por ejemplo

Si se detecta que los buffers permanecen en un 90% de ocupación durante las horas pico y que la actividad de I/O se incrementa en un 50%. La siguiente accion a tomar es incrementar el tamaño de los buffers de datos, pero hay que tomar en cuenta que se requiere memoria también para los buffers de bitácora, información de datos y de operación. Por lo que habrá que estudiar a nivel de sistema operativo si todavía se cuenta con memoria que pueda ser asignada al manejador.

III Afinación de aplicaciones

Una vez que se ha realizado una vuelta en la afinación del manejador es necesario comenzar con la afinación de aplicaciones. Como se mencionó anteriormente, el cambio en el entorno traerá cambios en las aplicaciones. Debido a esto tal vez sea necesario regresar sobre el proceso de afinación del subsistema y aplicaciones varias veces.

Flujo general de afinación de aplicaciones:

I. Dimensionamiento
II. Análisis
III.Instrumentación
IV.Mantenimiento

III.I Dimensionamiento

El primer paso será formar un equipo de trabajo que se integre por soporte técnico del sistema, administrador de bases de datos, coordinador de cambios hacia el area de desarrollo, usuario final.

Pasada la integración del equipo este tiene que decidir:

1. ¿Cuales son las aplicaciones por las que se va a comenzar la afinación?
2. ¿Cuales son las aplicaciones más criticas?
3. ¿Cuales representan mayor volumen?

Para contestar estas preguntas es necesario conocer
A) ¿Cual es el área del negocio que representa mayor riesgo de pérdida?
B) ¿Qué servicios presta el área de sistemas a esta área?
C) Resultados de monitorear el desempeño de las aplicaciones tanto de trabajo en línea como batch en horas pico.
D) Las aplicaciones que sean requeridas por el área más crítica del negocio
E) Aquellas aplicaciones que requieran recursos en común con el área más crítica del negocio.

Una vez que se tiene información sobre las aplicaciones y se ha decidido cuales son las de mayor riesgo para el negocio, es necesario:

I. Conocer el tipo de aplicación de que se habla:
- Trabajo en lote (batch)
- Trabajo en línea(online)
- Mixta (aplicación online dispara reporte batch)
- Consultas no planeadas

II. El tiempo de respuesta requerido por el usuario. Tanto para el servicio en línea, como el horario requerido para reportes.
III. Ventana de tiempo comprometida para la aplicación
IV. Conocer el tiempo en que se espera obtener resultados de la afinación
V. Problema principal reportado

Al terminar esta primera fase de dimensionamiento. Se continúa con el análisis de cada una de las aplicaciones seleccionadas. El flujo de afinación es independiente del tipo de aplicación del que se trate. Sin embargo las recomendaciones, instrumentación de cambios y pruebas varían.

III.II Análisis

La información necesaria para llevar a cabo un análisis de desempeño es:
a) Distribución de datos en los dispositivos de almacenamiento físico
b) Reporte del consumo de recursos por aplicación
c) Aplicaciones con problemas de deadlock o timeout
d) Factores de crecimiento

Al conocer esta información se ubica el problema a atacar:

% Memoria	% Requerimientos de I/O
% Tiempo de Ordenamiento	# Situaciones de deadlock
% Consumo CPU	

Una vez que se han obtenido estos datos se tiene una estimación general de los problemas que impactan el desempeño. Dependiendo del parámetro que arroje un tiempo más alto se puede requerir información extra para realizar un análisis de mayor profundidad. Es importante señalar el hecho de que un valor alto no siempre apunta directamente a la causa del problema sino es más bien un síntoma secundario. El atacar a su vez la raíz del problema puede hace notorias otras carencias de recursos que se tienen en el sistema.

Memoria

Si el problema se manifiesta en un consumo alto de memoria, generalmente se trata de problemas en el area de buffers de datos o información operativa (empleada para guardar temporalmente la información que se transfiere del almacenamiento físico). En la mayoría de los casos se piensa que con incrementar estas áreas de memoria se tendrá un beneficio. Y de hecho esto puede ocurrir, sin embargo incrementar la memoria en este punto no nos da la pauta para identificar y resolver el problema. Este consumo de recursos puede o no ser justificado. Para determinar el problema real se debe conocer si el volumen de requerimientos a las aplicaciones ha crecido o si nuevas aplicaciones han sido liberadas. Si esto es verdad se debe contar con área extra para poder atender a las peticiones. Si por el contrario ni el volumen ni la cantidad de aplicaciones ha crecido, un consumo alto de memoria se debe a que las aplicaciones están manteniendo en memoria más información de la necesaria.

Ordenamiento

Si el problema se detecta en el tiempo de ordenamiento. La primera causa posible y la más obvia se refiere al volumen de datos que la aplicación solicite que se ordene. En algunas ocasiones estos ordenamientos son innecesarios ya que debido al método de acceso empleado por el manejador los datos ya se encuentran en orden. Para determinar esto es necesario conocer el diseño de la base de datos, y la manera en que se encuentran almacenados los datos. Es conveniente recordar que dadas las limitaciones que se ilustraron en el capítulo 3 acerca de las instrumentaciones del modelo relacional, en una base de datos normalizada hasta la forma normal de Boyce-Codd, las operaciones de JOIN (junta) que se requieren son muy costosas en las instrumentaciones actuales , por lo que hay que intentar disminuir la necesidad de dichas operaciones.

Consumo de CPU

Consumo elevado de CPU se puede deber al empleo de funciones muy costosas o postulados de SQL que requieran obtener resultados parciales. Como se explicó en el capitulo 2 ya sea que los postulados sean estáticos o dinámicos, se cuenta con herramientas para conocer:

- Proceso necesario para obtener los resultados.
- Objetos que están siendo empleados para la solución del postulado.
- Dispersión,orden y cardinalidad de los datos

Un consumo elevado de CPU se puede deber a que es necesario realizar mucho I/O, esto está ligado al metodo de acceso que se emplea, así como la necesidad de llevar a cabo SORT la cual va ligada al tipo de postulado que se pide.

Requerimientos de I/O

Que el postulado de SQL requiera de gran cantidad de información para resolver el requerimiento, puede ser un comportamiento necesario para reportes detallados, sin embargo para consultas, inserciones, borrados y modificaciones que trabajen con unos cuantos registros , es necesario revisar las trayectorias de acceso a la información.

Si fuese el caso que una cantidad muy grande de información requiere ser procesada, habrá que evaluar las alternativas que se tengan para auxiliar al manejador, las cuales pueden ser:
a) Incrementar la cantidad de memoria cache asignada al dispositivo en que se encuentran los datos.
b) Cambiar de organización de archivos si el manejador lo permite, o de técnica de acceso a los mismos.

Situaciones de Deadlock

Para situaciones de deadlock lo esencial es esquematizar las aplicaciones que necesitan el recurso ,así como identificar que otros recursos tienen bloqueados que estén ocasionando problemas con otras aplicaciones como timeout. Una vez identificado este escenario es necesario realizar el análisis de las demás variables para obtener la causa por la que una aplicación bloquea tanto tiempo un objeto de tal manera que afecta el desempeño de las demás aplicaciones.

Una vez que se obtiene la información inicial, se puede identificar el problema. Tomando como guía las variables antes mencionadas, se puede obtener la entrada a la instrumentación realizando los siguientes pasos:

1. Identificar el punto de mayor consumo.
2. Identificar la o las causas del mayor consumo
3. Generar alternativas para disminuir el consumo
4. Priorizarlas y enumerar los posibles impactos, así como la dificultad de su instrumentación

III.III Instrumentación

Al terminar la fase de análisis , ya conocemos las aplicaciones críticas, conocemos las causas de los problemas de las aplicaciones , tenemos diferentes alternativas y sus implicaciones. Es en este momento cuando el tiempo que se comprometió para obtener resultados, va a ayudar en la decisión de la instrumentación de los cambios. Generalmente si se tiene premura de tiempo se instrumenta el cambio más sencillo aún cuando las implicaciones puedan ser mayores, sin embargo las propuestas de cambio más complejas no deben ser descartadas. Los cambios deben ser instrumentados en el ambiente de pruebas y evaluados. Para ello será necesario obtener la información que se requirió para empezar el análisis. Comparar estos resultados con los obtenidos en la medición original del entorno productivo. Si se tienen resultados favorables, realizar un plan para instrumentar el cambio en producción.

1.- Actividades y tiempo para realizar el cambio
2.- Mediciones para determinar si se mantiene o se revierte el cambio
3.- Plan de retorno del cambio
4.- Plan de monitoreo y reporte de resultados
5.- Retomar el proceso de afinación

III.IV Mantenimiento

Se regresa al primer punto para comenzar determinar las aplicaciones a afinar. Puede no regresar al primer punto si todas las aplicaciones responden a las expectativas del usuario final, en cuyo caso la información que se obtenga tendrá una intención proactiva, para realizar estudios de capacidad. Así mismo los resultados pueden servir como punto de referencia para determinar cuando es necesario llevar a cabo otro proceso de afinación.

APLICACIÓN DE LA METODOLOGÍA DE AFINACIÓN

El objetivo de este capitulo es ilustrar el empleo de la metodologia descrita en el capitulo anterior sobre el SMBDR DB2. Algunos de los conceptos necesarios para compreder el funcionamiento del manejador DB2 a nivel fisico.

I Descripción del entorno del manejador

Como punto número uno de la metodologia se debe definir el entorno, para lo cual se tomará la descripción general de un ambiente mainframe, en cuanto a los componentes que interactuan con el manejador.

1. HARDWARE
   a) Modelos de máquina serie 9121 IBM
      (1) Capacidad de memoria = 2G
      (2) Velocidad de proceso = (x) * número de procesadores
   b) Modelos de dispositivos de almacenamiento
      (1) Capacidad de almacenamiento = 2.838 Gbytes (DASD 3390 Modelo 3)
      (2) Velocidad de transferencia de información = n bytes * seg
2. SOFTWARE
   a) Sistema Operativo
      (1) Interfaz de usuario (CICS,TSO,CAF)
      (2) Manejador de operaciones de Entrada/Salida (VSAM)
      (3) Control de Seguridad (RACF)
      (4) Control de concurrencia
      (5) Tipo de servicios de almacenamiento
      (6) Organizaciones de archivos
3. OPERATIVAS
   a) Nuevos requerimientos
   b) Operaciones en línea
   c) Operaciones en lote
   d) Mezcla de ambas
   e) Volúmen
4. TIEMPO
   a) Madurez y evolución del sistema
   b) Tiempo comprometido de disponibilidad de la aplicación
   c) Tiempo comprometido de respuesta

II Metodología de afinación para un subsistema DB2

Bajo el control de un sistema operativo pueden existir varios subsistemas. En donde un subsistema lo describiremos como una instancia completa y autónoma de un manejador.

II.1 Análisis de la afinación del subsistema

Retomando la arquitectura general de DB2 descrita en el anexo B, y recordando las variables del sistema a medir:

- MEMORIA
- MANEJO DE Entrada/Salida
- SEGURIDAD
- CONCURRENCIA
- ALMACENAMIENTO FISICO

Estas variables encuentran su correspondencia en el subsistema de la siguiente manera:

Memoria

Los siguientes componentes van a estar en memoria y se identifican con los componentes que describimos en el capítulo dos como corazón del SMBDR, atención de peticiones y atención del tipo de acceso:

SYSTEM SERVICES ADDRESS SPACE (DSNMSTR)
Controla las diferentes conexiones con el subsistema,tareas necesarias para inicio y paro del sistema, los componentes de recuperación y recaba estadísticas del comportamiento del sistema. El consumo aproximado de memoria oscila entre 5000K y 2M dependiendo de los recursos disponibles.

DATABASE SERVICES ADDRESS SPACE (DSNDBM1)
Encargado de responder a las peticiones del sistema. Su consumo aproximado de memoria es de 16M con un mínimo de 14M.

IMS RESOURCE LOCK MANAGER (IRLMPROC)
Controla el acceso concurrente a los datos. Se calcula aproximadamente 250 bytes por candado lo que es aproximadamente 5M

Además de estos modulos, en memoria se tienen algunas estructuras como:

i) Buffer Pool: Grupo de buffers que almacenan la información de tablas aplicativas, para calcular su tamaño adecuado se tienen la siguiente formula:

BP0: # Buffers * 4K = X_{BP0}
BP1: # Buffers * 4K = X_{BP1}

.
.
.

BP32: # Buffers * 32K = X_{BP32}

TOTAL $X_{BP0} + X_{BP1} + \ldots + X_{BP32}$

Nota: Se cuenta con un buffer que maneja páginas de 32K para aquellas tablas que requieran esta longitud de registro o para el caso de algunas juntas.

ii) Sort Pool: Area en memoria en donde se tiene parte de la información a ordernar, se debe especificar de 60 páginas (4k por pag) o 10% del total del Buffer Pool.

iii) RID Pool: En esta area de memoria se lleva a cabo el sort de RID de los índices al llevarse a cabo un list prefetch su tamaño recomendado es de 200M o 50% del total del Buffer Pool.

iv) EDM (Environment Description Manager) Pool: Area en memoria en donde se almacenan las trayectorias de acceso (planes) , algunos apuntadores y status de los datos que estan siendo usados por X aplicación su tamaño aproximado se calcula mediante la fórmula:
((Número de planes concurrentes + total planes/4) *
tamaño del plan promedio +
(Número de DB concurrentes * tamaño de DBD) +
50K overhead) * 1.25 pérdida por fragmentación)

v) Working Storage: Area de memoria principal empleada por los *servicios de bases de datos*(DSNDBM1) para almacenar datos de manera temporal (resultados intermedios). El tamaño recomendado se da:
300K + (número de usuarios concurrentes* 40)

Manejo de Entrada Salida

Los siguientes componentes de almacenamiento físico son aquellos que contienen la información para atender a los requerimientos de datos del manejador Por lo que:

A) La razón de uso de los discos debe ser inferior a la especificada por el fabricante, para los discos que contienen:
i) Directorio y Catálogo: Información referente a tablas, indices, autoridades, consultas planeadas, estadísticas,etc.
ii) Logs- Bootstrap: Información referente a la bitácora e información bital para recuperaciones del subsistema o aplicativas.
iii) Base de datos DSNDB07: Area de trabajo para realizar toda la actividad de ordenamiento, debe estar distribuida se recomienda solo un tablespace por disco.
iv) Bases de datos aplicativas: En la definición de la Base de datos de las aplicaciones, separar en volúmenes diferentes los tablespaces de los índices.

Seguridad

La seguridad en DB2 se administra mediante dos comandos muy sencillos grant(otorga) y revoke (revoca). Se puede otorgar la autoridad a nivel de usuario o a nivel de grupo para lo cual se requiere que el DB2 trabaje en conjunto con otro subsistema del MVS llamado RACF. Para lograr el manejo de autoridades secundarias se requiere de una exit de seguridad la cual se señala en el momento de arranque del DB2. En lo tocante a desempeño el uso de autoridades secundarias puede hacer más rápido la busqueda, ya que se tendrán menos autoridades que empleando autoridad primaria. Además de que la administración de la autoridad será más sencilla.

Concurrencia

En lo que respecta al manejo de concurrencia se tienen algunas variables en el conjunto de parámetros para inicio del sistema (DSNZPARM) y una tabla que controla las conexiones denominada RCT (Resource Control Table), en donde por cada aplicación se especifica el número de ligas a mantener con el DB2, así como el tipo de las mismas, es decir si serán ligas dedicadas o se reestablecerán con cada nueva petición, o son ligas que encolan las transacciones en espera " las asignan a un buffer".

Almacenamiento Físico

El tipo de archivos manejados en DB2 estan manejados por los servicios VSAM (Virtual Storage Access Method) componente del sistema operativo. El tipo de archivos que maneja son secuenciales vistos como VSAM pero el DB2 le da un tipo de formato interno dependiendo de si se trata de tablespaces simples, segmentados o particionados. De igual manera trabaja con los indices, sin embargo estos tienen una estructura de arbol.

Todas los servicios anteriores pueden ser controlados desde el arranque del DB2, modificando los parámetros que se encuentran en la DSNZPARM. En ella se requiere especificar el número de:

Bases de Datos
Tablas
Columnas
Vistas
Tablespaces
Planes
Enunciados por Plan
Paquetes
Enunciados por paquete
Enunciados ejecutados
Tablas por enunciado
Archivos para area temporal
Bases de Datos abiertas
Threads (ligas)
  Remotas, TSO, Batch
Número mínimo de buffers en el BP0
Número máximo de buffers en el BP0
Número mínimo de buffers en el BP32k
Número máximo de buffers en el BP32k
Tipo de información que será recabada para efectos de monitoreo (traces)

II.II Instrumentación de la afinación del subsistema

A continuación se describen algunos de los síntomas que se pueden apreciar en el DB2, así como posibles causas y algunas medidas a tomar. Como se expresó en el capítulo 4, es conveniente hacer todos los cambios asociados a un servicio a la vez.

a) Contención en discos: Tiempo de respuesta del dispositivo excede la recomendación del fabricante

i) Contención en discos en que se encuentran el catálogo y directorio

Síntomas:
Las aplicaciones presentan problemas de deadlock o timeout por no poder obtener un candado sobre alguna tabla del catálogo o directorio
El monitor arroja tiempos de espera altos por espera en las peticiones de entrada/salida

Posibles Causas:
Los requerimientos de candado de algunas aplicaciones puede ser excesivamente largo
Peticiones de Entrada/Salida excesivas
Se estan llevando a cabo BINDS, creación o borrado de objetos.
Mala distribución en disco de los objetos de la BD
Indices y tablas del catálogo se encuentran en el mismo disco

Recomendaciones:
Revisar las opciones de lock de las aplicaciones que hacen acceso al catálogo
Revisar los métodos de acceso a los datos para evitar lecturas de más
Agendar la creación de objetos o BINDs , fuera del horario de mayor carga
Redistribuir las tablas con mucho acceso, si es posible particionarlas
No definir los índices de la tabla en el mismo volúmen en que se define la tabla.

ii) Contención en los discos en que se encuentran los logs y bootstraps

Síntomas:
Se detiene la actividad del subsistema
Bootstrap registra muchas entradas del log
Descargas frecuentes del log

Posibles Causas
Esperando por una descarga del log el sistema no atiende ninguna petición.
Tal vez el log es muy pequeño, lo que ocasiona descargas frecuentes

Recomendaciones:
Descargar oportunamente los logs, o al menos diferir las descargas sincronizadas, cuidando de no desbordar los logs
Crecer los logs para que el número de descargas disminuya

iii) Contención en discos en que se encuentra la base de datos DSNDB07

Síntomas:
Aplicaciones que involucran peticiones de ordenamiento, envían el mensaje de recurso no disponible
Entrada/Salida excesiva en las aplicaciones
Timeout por no poder obtener un tablespace de la base de datos temporal para realizar el ordenamiento

Posibles Causas:
Talvez los tablespaces de la base de datos DSNDB07 son muy pequeños
La base de datos DSNDB07 está mal distribuida y emplea sólo algunos tablespaces, desperdiciando el espacio en otros.
El número de areas definidas para la base de datos DSNDB07 es más pequeño que el número de aplicaciones concurrentes que requieren ordenamiento.

Recomendaciones:
Estimar un promedio del número de aplicaciones concurrentes y definir el número de tablespaces adecuados en la base de datos DSNDB07
Estimar el ordenamiento que involucre la mayor cantidad de registros, y definir un tablespace en DSNDB07 que pueda soportar ese volúmen.
Distribuir los tablespaces de la base de datos DSNDB07 de tal manera que se tenga un tablespace por volúmen unicamente.

iv) Tablespaces e indices

Síntomas:
Tiempo de espera por operación de Entrada/Salida alto
Número de peticiones de recuperar página alto

Posibles causas:
Demasiado fragmentado el tablespace
Método de acceso muy costoso en terminos de Entrada/Salida
Tablas e indices empleados por la aplicación en el mismo disco

Recomendaciones:
Tratar de influir el tipo de método de acceso, la manera de hacerlo se encuentra en la sección de afinación de aplicaciones.
Evitar que tablas e indices se

b) Uso de la memoria: La memoria disponible es insuficiente.

i) Memoria asignada para area de Buffer Pool insuficiente:

Síntomas:
Tiempo de espera por operación de Entrada/Salida alto y paginación excesiva, aparejado con un alto consumo de CPU
Tiempo de estancia alto
Hit ratio bajo. Se describe por la siguiente fórmula:

(1 - (synchio + Pfpagesrd)/getpages) *100 < 50%

En donde:
synchio: Número de peticiones de lectura de bloques de 32 páginas a la vez.
Pfpagesrd: Número de páginas consultadas.
getpages: Número de páginas acarreadas

Posibles Causas:
Posible saturación del Buffer Pool
Aplicación debería hacer uso del acceso denominado synchronous I/O

Recomendación:
Incrementar el tamaño del Buffer Pool de tal manera que se cumpla:
(1- (I-O)/readReq)) * 100 < 50

I-O: Número de peticiones de recuperar bloque
readReq: Número de peticiones de lectura
Analizar cual fué la causa del incremento en el uso del BP

ii) Memoria asignada al EDM Pool:

Síntoma:
La Entrada/Salida de planes y DBDs(Database Descriptors) es intensiva
Espacio EDM Pool desperdiciado, por que los objetos en la DBD no se emplean por la aplicación
Database Descriptors: Esqueletos de los objetos que se encuentran bajo la definición de una base de datos en DB2.

Posible Causa:
EDM Pool pequeño
Muchos objetos definidos en una Base de Datos sin relación entre si, desde el punto de vista del trabajo como aplicación.

Recomendación:
Incrementar el tamaño del EDM Pool
Definir en la Base de Datos sólo los objetos que se emplearán en conjunto por la aplicación

c) Problemas en la concurrencia

Síntoma:
Se tienen que cerrar Bases de Datos para abrir nuevas
Aplicación tiene que abrir thread
Mensaje indicando que la conexión no fue exitosa
Mensaje señalando recurso no disponible

Posible Causa:
Número de Base de Datos en DSNZPARM muy pequeño
Mala especificación de los ligas en la RCT. Por ejemplo algunas ligas que deben ser dedicadas no lo son y algunas que no deberían serlo lo son.
Número de ligas especificados por conexión insuficiente
Las ventanas de proceso aplicativo y mantenimiento de la base de datos se han traslapado

Recomendaciones:
Incrementar el parámetro Número de Bases de Datos en la DSNZPARM
Determinar las frecuencias de ejecución de las aplicaciones para hacer una especificación adecuada de la aplicación en la RCT
Incrementar el número de ligas asignadas al ambiente, es decir para conexiones CICS,TSO,CAF,etc.
Rediseñar el proceso de mantenimiento a la base de datos.

d) El recabar información para monitoreo puede ocasionar:

Síntoma
Consumo excesivo de CPU
Consumo excesivo de LOG

Posible causa
Trace activo, es decir se está recabando información para monitoreo ya sea de tiempo, consumo de recursos, etc.
Uso de opción DATA CAPTURE sobre los tablespaces
Correr las utilerías con la opción de LOG YES

Recomendaciones
Restringir los traces por horario y planes
Restringir el uso de la opción DATA CAPTURE a los tablespaces con información muy sensible, esto con el fin de poder auditar cambios
Correr las utilerías con opción LOG NO, para evitar uso excesivo de LOG.

III Metodología de afinación para las aplicaciones en DB2

De acuerdo a la metodología el flujo en afinación es:

III.I Dimensionamiento
III.II Análisis
III.III Instrumentación
III.IV Mantenimiento

III.I Dimensionamiento

En este punto el equipo de afinación debe estar ya integrado para poder contestar a las preguntas:

1.- ¿ Cuales son las aplicaciones por las que se va a comenzar la afinación?
2.- ¿ Cuales son las aplicaciones más críticas debidas al consumo de recursos?
3.- ¿ Cuales representan mayór volumen?

Se debe monitorear las aplicaciones con el fin de obtener por cada una de ellas:

Tiempo de estancia de la transacción o proceso en lote: Tiempo desde la petición de inicio hasta la obtención y despliege de resultados.
Tiempo en DB2 clase 1: Tiempo total de conexion con DB2
Tiempo en DB2 clase 2: Tiempo que se tarda en procesar un requerimiento
Consumo de CPU: Tiempo de CPU empleado
Uso de memoria: Cantidad de páginas de memoria empleadas
Peticiones de traer página (Get pages): Número de páginas requeridas
Peticiones de acceso Synchronous I/O: Número de lecturas de bloques de 32k
Número de transacciones en horario normal
Número de transacciones en horas pico

Para los casos de deadlock es importante contar con un reporte de dependencias como el siguiente:

Aplicacion	Recurso	
	Tabla1	Tabla2
Plan 1	X	S
Plan2	S	X

Una vez que se tiene esta información se pueden contestar las preguntas:

A) ¿Cual es el area del negocio que representa mayores ingresos?
B) ¿Qué servicios presta el area de sistemas a esta area?
C) ¿Cuales fueron los resultados de monitorear el desempeño de las aplicaciones tanto de trabajo en linea como batch?
D) ¿Cuales son las aplicaciones que son requeridas por el área más crítica del negocio?
E) ¿Cuales son aquellas aplicaciones que requieren recursos en común con el área más crítica del negocio?

Asi mismo es importante poder obtener los siguientes datos antes de comenzar con la sección de análisis. Aquí se muestra un ejemplo que se retomará mas tarde:

Lista de las aplicaciones más críticas

Plan	Tipo de aplicación	Tiempo de Estancia deseado	Tiempo de Estancia actual	Problema principal	Tiempo en que se esperan resultados
PXZ31B	BATCH	1hr	3 hrs	Tiempo de estancia exagerado	Proximo proceso mensual
PXZ45O	ONLINE	30seg	1m	Deadlock frecuentes	Dos semanas
PXZ34M	MIXTA	1m	2m	Respuesta lenta	Dos semanas

III.II Análisis

La información necesaria para llevar a cabo un análisis de desempeño es:

a) Distribución de datos en los dispositivos. Sólo de los objetos a emplear por las aplicaciones seleccionadas, además de tiempo de respuesta promedio del disco.
b) Reporte del consumo de recursos por aplicación
c) Factores de crecimiento
d) Aplicaciones con problemas de deadlock o timeout.

a) Aplicación 1	a) Aplicación 2	a) Recurso	a) Frecuencia
b) PXZ450	b) PXZ250	b) INDEX1	b)
c) PXZ450	c) PXZ350	c) INDEX2	c)

De acuerdo a la metodología el siguiente paso es identificar en que parte se encuentra el mayor consumo de recursos.

% Memoria	% Requerimientos de I/O
% Tiempo de Ordenamiento	# Situaciones de deadlock
% Consumo CPU	

Sin embargo cualquiera que sea el caso es necesario obtener un explain del plan para saber cual es la trayectoria de acceso que ha decidido tomar el manejador. Los resultados se guardan en la PLAN-TABLE, la estructura de la tabla y el significado de las columnas se explica a continuación:

COLUMNA	SIGNIFICADO
QUERYNO	Un número que identifica el enunciado al que se le dió EXPLAIN
QBLOCKNO	Un número que identifica diferentes bloques de un QUERY pej. SUBQUERIES,etc
APPLNAME	Nombre del PLAN
PROGNAME	Nombre del DBRM o PACKAGE
PLANNO	Orden de evaluación de los bloques del enunciado
METHOD	Método empleado en el SORT/JOIN del PLANNO (0,1,2,3 ó 4)[1]

0. [1] Tabla que se accesa primero en un JOIN
1. Nested Loop Join
2. Merge Scan Join
3. Sorts adicionales requeridos debido a ORDER BY, GROUP BY, SELECT DISTINCT,UNION o Predicado que incluya algún conteo.
4. Hybrid Join

COLUMNA	SIGNIFICADO
CREATOR	Creador de la tabla o vista empleada en este paso
TNAME	Nombre de la tabla empleada en este paso
TABNO	Un numero que identifica la tabla empleada en la clausula de FROM. Ayuda a distinguir varias referencias a la misma tabla
ACCESSTYPE	Método empleado para accesar la tabla (I, II, N, R, M, MX,MI,MU o blanco[2])
MATCHCOLS	El número de columnas del indice empleadas en accesos I, II, N ó MX
ACCESSCREATOR	El creador del indice empleado en un SCAN de indice I, II, N ó MX
ACCESSNAME	El nombre del indice empleado en un SCAN de indice I, II, N ó MX
INDEXONLY	Indica si sólo se accesa el indice en el paso
SORTN_UNIQ	Cuando el SORT se lleva a cabo en el paso para eliminar registros duplicados
SORTN_JOIN	Cuando el SORT se lleva a cabo en el paso por causa de algún JOIN
SORTN_ORDERBY	Cuando el SORT se lleva a cabo en el paso por causa de un ORDER_BY
SORTN_GROUPBY	Cuando el SORT se lleva a cabo en el paso por causa de un GROUP_BY
SORTC_UNIQ	Cuando el SORT se lleva a cabo en el paso para eliminar registros duplicados
SORTC_JOIN	Cuando el SORT se lleva a cabo en el paso por causa de algún JOIN
SORTC_ORDERBY	Cuando el SORT se lleva a cabo en el paso por causa de un ORDER_BY
SORTC_GROUPBY	Cuando el SORT se lleva a cabo en el paso por causa de un GROUP_BY
TSLOCKMODE	Tipo de LOCK empleado sobre la tabla (IS,IX,S,U,X ó SIX)
TIMESTAMP	Marca de tiempo en que se ejecutó el EXPLAIN
REMARKS	Comentarios del usuario
PREFETCH	Tipo de lectura empleado para transferir las páginas de datos a los buffers (Sequential,List o blanco)
COLUMN_FN_EVAL	Indica en qué momento se evalua la función de SQL (R,S o blanco)
MIXOPSEQ	Número que denota la secuencia empleada en operaciones que incluyen múltiples índices
VERSION	Indica la versión en el caso de paquetes
COLLID	Indica el identificador de la colección en el caso de paquetes
ACCESS_DEGREE	Indica el grado de paralelismo a emplear
ACCESS_PGROUP_ID	Identifica un conjunto de renglones asociados a un flujo de Entrada/Salida determinado
JOIN_DEGREE	Indica el grado de paralelismo a emplear durante un JOIN
JOIN_PGROUP_ID	Identifica un conjunto de renglones asociados a un flujo de Entrada/Salida determinado empleados en un JOIN paralelo

[2] I se usará el índice
II se realizará barrido de índice
N se realizará barrido de índice cuando el predicado contiene IN
R se realizará barrido de tablespace
M se utilizarán multiples índices
MX índice empleado
MI intersección de varios índices
MU Union de varios índices
Blanco accesará la tabla por el primer índice creado

PARAMETROS IMPLICADOS EN EL DESEMPEÑO

Con la PLAN-TABLE y el EXPLAIN ya sabemos cual es la ruta que decidió tomar el DB2. Para influir sobre la ruta de acceso es necesario saber que el DB2 la calcula en base a:

a) Tipo de Predicado
b) Indices
c) Nivel de orden de los datos
d) Estadísticas
e) Tipos de acceso con los que cuenta el DB2
f) Parámetros de BIND

Así como también es importante conocer la forma en que se evalúa la ruta de acceso.Las rutas de acceso son decididas en base a los Entrada/Salidas estimados ,costo de CPU por cada block de query. Los valores que calcula el optimizador son los siguientes:

F= Filter Factor (Porcentaje de los renglones que califican)
TN= Número total de renglones de todas las tablas en el FROM
NLEVD= El número de los niveles intermedios del indice
Predicate Cardinality= TN * F

Datos a tomar en cuenta:

Un factor de filtrado pequeño sugiere un porcentaje bajo de renglones que califican para la condición.Se calcula 1/FIRSTKEYCARD * CARD = F

En el tiempo de respuesta influyen:

1) Cantidad de datos a recuperar
2) Cantidad de SORTS a realizar
3) Frecuencia de uso del recurso

a) Tipo de Predicado

CONCEPTO

Se puede hacer una diferenciación entre los postulados del lenguaje que sirven para el manejo de datos, de aquellos que auxílian en la definición de las estructuras así como de los postulados orientados a permitir el acceso a los datos

DDL: Data Definition Language incluye los postulados de :
ALTER, CREATE y DROP

DML: Data Manipulation Language incluye los postulados de:
SELECT, INSERT, UPDATE y DELETE

DCL: Data Control Language incluye los postulados de:
GRANT y REVOKE

De estos postulados los que tienen un efecto determinante sobre el desempeño son los del DML. En particular el SELECT, ya que trabaja sobre conjuntos de datos , sea incluido como parte de un UPDATE o de un DELETE o como una consulta.

Los postulados se pueden clasificar en :

Indexables:	Cuando el manejador puede auxiliarse de un índice para agilizar el acceso a datos.
No indexables:	Cuando se requiere accesar directamente la tabla
Estado 1:	Cuando el manejador no requiere obtener resultados intermedios
Estado 2:	Cuando el manejador requiere obtener resultados intermedios

En la siguiente tabla se muestran el tipo de condiciones que determinan que un postulado sea de uno u otro tipo:

Tipo de predicado	Indexable	Estado 1
COL OP valor COL IS NULL COL BETWEEN valor1 AND valor2 COL LIKE 'cadena%'	SI	SI
COL NOT = valor . COL IS NOT NULL COL NOT IN (lista) COL NOT BETWEEN valor1 AND valor2 COL LIKE '%cadena' COL LIKE '_cadena' COL LIKE variable_host	NO	SI
COL OP(query correlacionado) COL NOT = (query correlacionado)	NO	NO

Tipo de predicado	Indexable	Estado I
T1.COL OP T2.COL T1.COL NOT = T2.COL[3]	SI	SI
T1.COL OP T1.COL T1.COL NOT = T1.COL[4]	NO	NO
COL IN (subquery) COL ...ANY (subquery) COL ...ALL (subquery) NOT EXISTS (subquery)	NO	NO
COL = expresión expresión OP valor expresión NOT= valor	NO	NO
predicado1 AND predicado2	TALVEZ	SI

b) Indices

En DB2 tenemos tres tipos de índices:

Cluster	Los datos de la tabla estan ordenados de acuerdo a las columnas del índice
Unique	Los datos de la tabla deben tener valores únicos en las columnas del índice
Normal	No tienen ninguna característica en particular

La estructura de los índices en DB2 es de árbol. Con esta información se deben diseñar los índices tomando en cuenta que no basta con codificar un SQL de determinada manera para asegurar el uso de índice. Así como no siempre el garantizar el uso de índice implicará buen desempeño.

Es importante al momento de realizar el diseño físico hacer consideraciones del tipo:

- Una tabla que tiene mucho trabajo de actualización,inserción o borrado no debe tener más de tres índices.
- Tablas muy grandes es recomendable que tengan índice
- Las llaves no deben ser demasiado largas
- Las llaves del índice deben incluir las columnas más accesadas
- Un índice cluster es mejor sobre columnas de datos redundantes

[3] Solo si son del mismo tipo y longitud las columnas a comparar, es indexable.
[4] Excepto para queries SELECT * FROM T1 A,T1 B WHERE A.C3 = B.C4

c) Nivel de orden de los datos

El trabajo sobre las tablas de datos hace que las condiciones iniciales de orden de los datos se pierdan y se requieran más accesos para obtener la misma información. Es por ello que es importante ordenar y respaldar los datos de manera periódica. Tomando en cuenta las siguientes variables para determinar cuando hacerlo:

- Cuando se tenga mucha fragmentación
- Porcentaje de páginas actualizadas desde el último respaldo excede un 25%
- Las páginas de los índices se encuentran dispersas
- El porcentaje de cluster-ratio reportado por las estadísticas es menor al 95%
- Los índices tienen más niveles que los requeridos

d) Estadísticas

CONCEPTO

Las estadísticas son empleadas por el optimizador para conocer la naturaleza de los datos que se almacenan en las tablas.

TIPOS DE ESTADISTICAS

TABLA DEL CATALOGO	COLUMNAS	DESCRIPCION
SYSTABLESPACE	NACTIVE	Número de páginas empleadas
SYSTABLES	CARD	Total de renglones en la tabla
	NPAGES	Número de páginas empleadas por la tabla
	PCTPAGES	Porcentaje de paginas en el tablespace que tienen información
SYSCOLUMNS Sólo se toman las estadísticas para la primera y última columna del IX	COLCARD	Número de valores distintos en la columna
	HIGH2KEY	Segundo valor mas alto en la columna si esta es parte de un índice
	LOW2KEY	Segundo valor mas bajo en la columna si esta es parte de un índice

TABLA DEL CATALOGO	COLUMNAS	DESCRIPCION
SYSINDEXES Si la distribución en FIRSTKEYCARD es baja en algunos casos le ocasiona al optimizador que ignore un índice NOCLUSTER	CLUSTERED FIRSTKEYCARD FULLKEYCARD NLEAF NLEVELS CLUSTERRATIO	Está la tabla ordenada de acuerdo al índice CLUSTER Número de valores distintos para la primera columna del índice Número de valores distintos para el índice Número de páginas hojas Número de niveles del árbol Porcentaje de renglones en la tabla que están en el orden de este índice
SYSFIELDS	EXITPARM EXITPARML	El valor actual de la columna del índice que se está contando El número de renglones que contienen ese valor en la columna expresado como porcentaje

e) Tipos de acceso con los que cuenta el DB2

A continuación se describen los diferentes métodos de acceso con que se cuenta, listados del método de acceso mas al menos eficiente:

1.- Compaginar con UNIQUE INDEX (predicados '=')
2.- Compaginar con CLUSTER INDEX
3.- Compaginar con NON-CLUSTER INDEX
4.- No compaginar con CLUSTER INDEX
5.- TABLESPACE SCAN
6.- NESTED LOOP JOIN (Se emplea un índice en la tabla que se selecciona como externa en la junta)
7.- MERGE SCAN JOIN (Se barren ambas tablas y los resultados entran a un sort)

El método de acceso a emplear no sólo depende del tipo de predicado que se emplée sino de la información que se requiere recuperar , de las columnas de la tabla sobre las que se imponen las condiciones así como las estadísticas que se tienen al respecto

f) Parámetros de BIND

Además de la ruta de acceso calculada por el manejador intervienen en la eficiencia del proceso los parámetros con los que se dió BIND al plan, es decir un requerimiento estático al manejador los cuales son:

Existen en la versión 2.3 dos tipos de requerimientos estáticos, el PLAN y el PAQUETE.

PARAMETROS IMPLICADOS EN EL DESEMPEÑO

De los parámetros de BIND sólo se incluyen los que tienen algun impacto sobre el desempeño:

BIND PLAN (plan-name)/OWNER (auth-id)/QUALIFIER (qualifier-name)
MEMBER(member-name) LIBRARY(lib-name)/
PKLIST(location-name./collection-id /pkg-id./*)
DEFER(PREPARE)/NODEFER(PREPARE)
VALIDATE(RUN/BIND)
ISOLATION(RR/CS)
ACQUIRE(USE/ALLOCATE)
RELEASE(COMMIT/DEALLOCATE)
EXPLAIN(NO/YES)

PLAN: Código SQL optimizado en el sentido de las rutas de acceso, se almacena en la DSNDB01.SCT02 en ocasiones recibe el nombre de SKCT. Consta de páginas de directorio y encabezado y una o más agrupaciones denominadas secciones.

DBRM: Es el conjunto de enunciados SQL embebidos en un programa

PACKAGE: Plan miniatura al que sólo le corresponde un sólo DBRM. El nombre consta de 4 calificadores: location_name,collection_id, package_id y version_id. Se almacena en la DSNDB01.SPT01

COLLECTION: Es el nombre que se le da a un agrupamiento lógico de programas. Se crea al darle BIND a un paquete por primera vez y asignarlo a una colección.

PKLIST: El empleo de paquetes tiene un impacto no sobre el desempeño sino sobre la confiabilidad con que los cambios se lleven a cabo.

ACQUIRE(ALLOCATE) se recomienda sobre *ACQUIRE(USE)* cuando se requiere asegurar un candado, sin embargo la concurrencia se ve afectada.*ACQUIRE(USE)* requiere que el EDMPOOL sea grande, sobre todo si los planes son muy grandes y el PLAN no está autorizado a público, entonces la memoria CACHE que se recomienda es mayor al los 1024 bytes, lo que implica un aumento en el uso del EDMPOOL.

VALIDATE (RUN/BIND) Es preferible que el chequeo de los objetos se lleve a cabo al momento de BIND y no al momento de correr la aplicación.

ACQUIRE (USE) RELEASE (COMMIT) Son las opciones que proporcionan mayor concurrencia.

EXPLAIN (YES) Si bien no impacta al desempeño, es recomendable emplear esta opción para conocer la ruta de acceso que tomara la aplicación.

Con toda esta información y el reporte de porcentajes de consumo, se pueden identificar en que parte del proceso de la petición se encuentran el o los problemas que ocasionan consumo en algunas de las variables marcadas como importantes(%memoria,%tiempo de ordenamiento, % consumo de CPU, %requerimientos de Entrada/Salida o situaciones de deadlock).

III.III Instrumentación

Plan para instrumentar el cambio en producción

1.- Actividades y tiempo para realizar el cambio
2.- Mediciónes para determinar si se mantiene o se revierte el cambio
3.- Plan de retorno del cambio
4.- Plan de monitoreo y presentación de resultados
5.- Retomar el proceso de afinación

III.IV Mantenimiento

Se regresa al primer punto para comenzar determinar las aplicaciones a afinar. Puede no regresar al primer punto si todas las aplicaciones responden a las expectativas del usuario final, en cuyo caso la información que se obtenga tendrá una intención proactiva, para realizar estudios de capacidad. El papel del usuario final en este caso será el de servir de supervisor de la calidad del servicio que se le brinda.

III.V Un ejemplo del proceso de afinación.

De la fase de dimensionamiento se obtiene que la aplicación PXZ450 determinada como crítica tiene las siguientes características:

Plan	Tipo de aplicación	Tiempo de Estancia deseado	Tiempo de Estancia actual	Problema principal	Tiempo en que se esperan resultados
PXZ450	ONLINE	30seg	1m	Deadlock frecuentes	Dos semanas

Tiempo de estancia de la transacción es de 1m
Tiempo en DB2 clase 1 es de 50 seg
Tiempo en DB2 clase 2 es de 40 seg
Consumo de CPU es de un 70% del total
Uso de memoria emplea cerca de 30000 paginas cada vez que se ejecuta
Peticiones de traer página (Get pages) cerca de 14000
Peticiones de acceso Synchronous I/O 5000
Número de transacciones en horario normal **50 en una hora**
Número de transacciones en horas pico **100 en una hora**

De estos datos se puede apreciar que el problema se encuentra en el proceso que se lleva a cabo en el DB2. Ya que el tiempo clase 2 es significativo al igual que el consumo de CPU y los requerimientos de Entrada/Salida. Además se tiene que los problemas de deadlock que ocasiona esta aplicación son:

Aplicación 1	Aplicación 2	Recurso	Frecuencia
PXZ45O	PXZ25O	INDEX1	20/100
PXZ45O	PXZ35O	INDEX2	30/100

Pasando a la fase de análisis las columnas relevantes de la PLAN_TABLE después del EXPLAIN del plan PXZ45O son:

QUERYNO	10	10
QBLOCKNO	1	1
APPLNAME	PXZ45O	PXZ45O
PROGNAME	PXZ45O	PXZ45O
PLANNO	1	2
METHOD	1	3
CREATOR	C1	C1
TNAME	T1	T2
ACCESSTYPE	I	
MATCHCOLS	1	

El enunciado analizado fué el siguiente:

```
SELECT COL1, COL2
FROM T1 A,T2 B
WHERE
A.COL1 = B.COL1 AND
B COL2 <> 'A'
ORDER BY COL2
```

Para conocer en donde está el problema es necesario continuar con el análisis estableciendo:

Tipo de predicado

Predicado de manipulación de datos. La cláusula A.COL1 = B.COL1 es indexable y estado 1. B.COL2 = 'A' no es indexable pero es estado 1

Indices

Mediante consulta al catálogo o al diseño físico se sabe que la tabla T1 tiene definido un índice unique sobre COL1 y esta se encuentra en primer lugar. T2 tiene índice CLUSTER que involucra COL2 pero está en segundo lugar, en primer lugar tiene a COL3.

Nivel de orden de los datos

Para tener una idea del orden de los datos se consulta la tabla SYSIBM.SYSINDEXES para los índice IX1 perteneciente a T1 e IX2 a T2.

INDICE	IX1	IX2
CLUSTERED	N	Y
FIRSTKEYCARD	2000	10000
NLEAF	100	1000
NLEVELS	2	3
CLUSTERRATIO	60%	95%

Estadísticas

Aparentemente los índices se encuentran en buen estado, pero al lanzar un proceso de extracción de estadísticas para IX1 se obtuvieron los siguientes resultados:

INDICE	IX1
CLUSTERED	N
FIRSTKEYCARD	500000
NLEAF	5000
NLEVELS	4
CLUSTERRATIO	50%

Con esto se hizo notar que el crecimiento de T1 no se había registrado. Así como también se hace notorio que no se ha reorganizado el índice, ya que 4 niveles es excesivo para un índice tan pequeño.

Tipos de acceso

El método de acceso seleccionado según el explain fue el NESTED-LOOP JOIN en donde T1 es accesada vía índice, pero dadas las condiciones en que se encuentra el índice es comprensible el número tan alto de peticiones de Entrada/Salida. T2 seleccionada como externa requiere un barrido secuencial ya que no se empleará el índice, por el tipo de predicado.

Parámetros de BIND

Al consultar el catálogo para el plan PXZ450 se descubrió que se le dió BIND con las opciones RELEASE(DEALLOCATE) y ACQUIRE(ALLOCATE), lo cual para una consulta es excesivo y se tiene bloqueada la tabla por más tiempo de lo necesario impactando la concurrencia sobre los recursos en conflicto.

Instrumentación

Como resultado del análisis se obtuvieron las siguientes recomendaciones:

Aplicación	Problema	Alternativa	Implicación
PXZ450	Provoca Deadlock con varias aplicaciones.	Ver si es posible incluir COL3 en la condición del SQL. Reorganizar y actualizar estadísticas de T1 con mayor frecuencia Dar BIND con los parámetros adecuados para maximizar la concurrencia.	Reducción en el tiempo de disponibilidad de T1. Afectar otras aplicaciones que empléen T1.

Plan para instrumentar el cambio en producción

1.- Todas las recomendaciones que se describieron se realizarán. Para poder revertir el cambio se tomarán los respaldos pertinentes. Los cambios se realizarán durante la noche y se realizarán algunas pruebas.

2.- Mediciánes para determinar si se mantiene o se revierte el cambio
Si el tiempo de respuesta de la aplicacion es de al menos el promedio en hora no pico el cambio permanece.
3.- Plan de retorno del cambio. Regresar los cambios mediante los respaldos y dar BIND con los viejos parámetros.
4.- Plan de monitoreo y presentación de resultados. Durante el periódo de horas pico se debe tomar las mediciones.
5.- Retomar el proceso de afinación

Mantenimiento

En esta fase si las medidas fueron exitosas se debe monitorear que los procesos de mantenimiento se corran con regularidad. De incurrir nuevamente en tiempos lentos, se debe revisar que las condiciones se mantengan antes de comezar otro esfuerzo de afinación.

CONCLUSIONES

La idea de realizar esta tesis surgió de algunas discusiones para decidir entre los SMBDR cual se apega más al modelo relacional. Dado el interes que surge por las nuevas propuestas para los ambientes de bases de datos. ¿Acaso la necesidad de nuevas propuestas implica que el modelo relacional ha sido rebasado? Como se expresó en el capitulo 1 , en la parte de generaciones futuras se espera que los SMBDR existentes en el mercado cubran una serie de necesidades. Pero de igual manera en el capítulo 3 se estableció que los SMBDR actuales incurren en una serie de fallas en la instrumentación con respecto a las reglas del modelo relacional. De estos puntos se desprende la premisa de que el modelo relacional no se encuentra superado, sino que el problema real es la falta de eficiencia en las instrumentaciones existentes de los SMBDR actuales. Retomando la definición de eficiencia como: el resultado esperado en el tiempo esperado con los recursos esperados, podemos llegar a la conclusión que en general los SMBDR actuales no son eficientes. Es decir por limitantes en las intrumentaciones en ocasiones los resultados pueden no ser los esperados, por fallas en el lenguaje relacional empleado, asi como en la tolerancia a los registros duplicados. Entonces o no se obtiene el resultado esperado, o los recursos empleados exceden a los estimados. Es este último problema el que se toma como punto central en esta tésis. En la literatura revisada encontré que en las nuevas versiones de los SMBDR, tomados como muestra para efectos de esta tésis, el esfuerzo principal se concentra en aceptar nuevos tipos de datos (dominios), bases de datos distribuidas y homogeneizar las representaciones de datos a fin de poder compartir información. Sin embargo problemas como: las restricciones del lenguaje relacional empleado, el hecho de aceptar registros duplicados, la necesidad de que el usuario final conozca las limitaciones del proceso al tratar con información faltante, no son tópicos que en las próximas versiones se traten de resolver. El problema de la eficiencia segun mi apreciación continuará en las próximas versiones, es por ello que la aportación de la presente tésis fué proporcionar una metodología de afinación. En un esfuerzo porque al menos los recursos empleados sean los estimados.

La manera de influir sobre los recursos empleados es tratando de influir sobre las rutas de acceso, si bien se han hecho esfuerzos por mantenerlas independientes de las decisiones de los usuarios, esto es parcialmente cierto, ya que en todos los SMBDR se encontró la necesidad de emplear las estadisticas como medio para decidir cual es la mejor ruta de acceso basandose en la naturaleza de los datos. Se encontró que todos los SMBDR son muy vulnerables a diseños fisicos inadecuados de la base de datos, ya que estos son elaborados en base a reglas que tienen como sustento un

SMBDR ideal en el cual todas las reglas del modelo relacional se encuentran instrumentadas en su totalidad. Para los SMBDs existentes hay que proporcionar diseños que tal vez no cumplan con todas las reglas de normalización, pero que pueden ofrecer un mejor desempeño

La metodología de afinación tanto de aplicaciones como del SMBDR, surge de la comparación entre SMBDR tomando como punto de referencia el modelo relacional, de esta comparación se obtiene la definición de una arquitectura general, sobre la que se desarrolla la metodología de afinación ya que si bien todos los SMBDR son diferentes en el detalle, todos tienen sorprendentes semejanzas en cuanto a la instrumentación, es por ello que fué posible desarrollar la metodología general de afinación.

Este trabajo puede tener como línea futura el aplicar la metodología a un caso de degradación de desempeño en la vida real. Así como plantear la posibilidad de desarrollar un sistema experto que pueda dar recomendaciones para mejorar el desempeño en general. Como se mostró en el capítulo 5 el análisis de la información podría ser resumido en un conjunto de hechos, los cuales formarían parte de la base de conocimiento del experto. El conjunto de reglas varían en cuanto al manejador pero pueden ser englobados en los cinco puntos de consumo de recursos que se mostraron como parte de la metodología en el análisis. Se tiene un ejemplo de estas reglas para DB2 en el capítulo 5. Pero de igual manera como se ve en los anexos C y D es posible recabar recomendaciones para cada ambiente dependiendo del SMBDR y alimentar con estas a un experto.

Los objetivos de la tésis se cumplieron ya que se logró ver las diferencias y limitaciones de los diferentes SMBDR, así como obtener la metodología de afinación. La pregunta inicial se contesta al darnos cuenta de que el modelo relacional si contempla los requerimientos de las nuevas generaciones pero las instrumentaciones no. Cabe preguntarse porqué aún faltan por instrumentarse algunas de las reglas del modelo relacional.

Finalmente el tratar de poner a un SMBDR por encima de otro u otros carece de sentido, dado el objetivo de la tésis. Se sostiene en base a que en cada una de las instrumentaciones de los SMBDR se priorizan de manera diferente, las características que han de ser incluidas en cada SMBDR. Creo que en este momento, los SMBDR ya no se encuentran como en el pasado intentando demostrar cual puede calificarse como relacional, dado su apego al modelo relacional. En nuestros días la competencia se ha vertido en satisfacer las necesidades de precio transportabilidad, conectividad, capacidades de explotación y presentación de la información, plataformas en que se encuentra disponible. Esta carrera se encamina para ver quien prevalece en el mercado.

BREVE DESCRIPCIÓN DE CADA REGLA FUNDAMENTAL DEL MODELO RELACIONAL DE ACUERDO AL MODELO ANSI

Las reglas se enuncian a continuación empezando por aquellas reglas de la capa conceptual

IV.1 Autorización

En este conjunto de reglas tenemos dieciseis reglas de las cuales seis son fundamentales y las otras diez son básicas. El propósito de estas reglas es permitir el control acerca de quien tiene el acceso y a que partes de la base de datos, así como limitar el alcance de las actividades que puede desempeñar sobre la información. Esto es para evitar el daño a la información compartida por muchos usuarios con necesidad y experiencia diferentes.

RA-1 Bases Afirmativas
Toda autorización debe ser otorgada sobre bases afirmativas; esto significa que a los usuarios se les debe otorgar el permiso explícito para accesar partes de la base de datos en lugar de negar el acceso de manera explícita.

RA-2 Otorgando Autoridad: Enfoque espacio-tiempo
Al otorgar autoridad todo el poder del I.R (incluyendo la lógica de cuatro valores de predicados de primer orden) debe aplicarse al definir (1) las partes de la base de datos y su descripción accesible para propósitos específicos (consulta, inserción o actualización de valores en la base de datos, archivado o borrado o cualquier combinación de estas actividades) y (2) en que horario el acceso es permitido (usando las funciones de fecha y hora del I.R).

RA-6 Retraso de borrado y destrucción debida al archivado
La ejecución del comando DROP RELATION da como resultado que la relación especificada se guarde por un periodo de como mínimo siete días. Los borrados en gran escala se retardan archivando la información de manera similar. El default es de siete días si no se especifica un periodo mayor.

RA-7 Autorización de actividades de control sobre la base de datos.
Existen al menos 13 actividades de control de la base de datos, a las que se les debe autorizar ya sea por separado o en combinación.

Las actividades son las siguientes:
1.- Crear y borrar un dominio
2.- Crear y borrar una tabla-R base
3.- Crear y borrar una de las columnas de una tabla-R base existente
4.- Crear y borrar una vista
5.- Crear y borrar una relación de integridad por tipo
6.- Crear y borrar una función definida por el usuario
7.- Crear y borrar una ruta de acceso para mejorar el desempeño
8.- Crear una llave foranea en una tabla-R, relacionando una llave primaria en otra tabla-R
9.- Requerir que una autoridad sea otorgada o descontinuada
10.- Requerir un snapshot
11.- Requerir de que un log de auditoria sea mantenido o descontinuado
12.- Establecer una condición para archivar con una nueva etiqueta
13.- Establecer el modo ARRIBA o ABAJO para redondear pseudo-fechas.

RA-8 Autorización de actividades de consulta o manipulación
Existen al menos 8 actividades de interrogación y manipulación, a las que se les debe autorizar ya sea por separado o en combinación.Y son las siguientes

1.- Recuperar de tablas-R específicas
2.- Insertar información en tablas-R específicas
3.- Modificar componentes específicos de renglones específicos en tablas específicas.
4.- Modificar la llave primaria de tablas-R específicas
5.- Archivar renglones de tablas-R específicas
6.- Borrado de renglones de tablas-R específicas
7.- Modificación de una valor marcado como I por un valor marcado como A, un valor específico o viceversa

RA-10 Otorgando y revocando autoridad
La autorización para el acceso o modificación de partes de la base de datos puede ser asignada a un usuario o a un grupo previamente definido, y más adelante esquematizar a partir del usuario o grupo empleando I.R. Ciclos en los cuales el usuario A otorga al usuario B autoridad ya sea directa o indirectamente están prohibidos.

IV.II Funciones

En este conjunto de reglas tenemos diez reglas de las cuales tres son fundamentales y las otras siete básicas. El propósito de las funciones se puede enunciar en los siguientes alcances:
- Transformar la información obtenida al ejecutar una consulta
- Formar parte de las condiciones que se plantean en las consultas.

RF-1 Funciones de agregación BUILT-IN
El DBMS provée al menos las cinco funciones de agregación - COUNT, SUM, AVERAGE, MAXIMUM, MINIMUM- como funciones BUILT-IN para ser usadas ya sea en la transformación de valores de bases de datos destino mediante una consulta, en el comando de definción de vistas ó en la determinación de la condición que debe ser satisfecha por la base de datos destino en recuperación y manipulación de datos.

RF-2 Las versiones DOD de las funciones estadísticas BUILT-IN
Para cada función estadística construida en el DBMS, hay también una función DOD (Degree of Duplication) Grado de Duplicación

RF-8 No-generación de valores marcados por funciones
La aplicación de funciones escalares a valores no marcados y de funciones de agregación a un conjunto de valores de la base de datos no marcados, nunca arroja un valor marcado.

IV.III Restricciones de integridad

En este conjunto de reglas tenemos veintidos reglas de las cuales once son fundamentales y las otras once básicas. El enfoque en el modelo relacional acerca de preservar la precisión de la información, es preventivo. Por ello se plantean métodos solamente para impedir el daño a la información por parte de los usuarios, más que contar con mecanismos que permitan recuperar la información una vez que el daño fué hecho. Con el propósito de asegurar la exactitud de la información es que se establecen las restricciones de integridad.

RI-1 a RI-5 Las restricciones de integridad son de cinco tipos
(1) D-type o Integridad de dominio, (2) C-type o Integridad de Columna, (3) E-type o Integridad de entidad, (4) R-type o Integridad Referencial y (5) U-type o integridad definida por el usuario

RI-6 Temporalización de pruebas para tipos R y U
Cada restricción de tipo R o U debe incluir un símbolo que especifique una condición de tiempo. De manera tal que el DBMS pueda determinar cuando una restricción es pertinente para un comando y simplemente la ejecute. Debe también examinar el símbolo de la temporalidad, para determinar si la restricción se debe evaluar (1) al término de la ejecución del comando o (2) como parte de la ejecución de un COMMIT para completar una transacción y antes de realizar los cambios en la BD.

RI-11 Violaciones a restricciones de integridad tipos D,C y E
Violaciones de restricciones de integridad tipos D,C y E no son permitidas. Si la fuente de un intento de violación es un programa de aplicación, el DBMS manda un mensaje indicando que no se efectuó el requerimiento. Entonces el programador puede elegir, el incluir comandos que lleven al programa a un estado de paro, cuando se encuentre dicho código. Si la fuente es un usuario en una terminal el DBMS simplemente niega el requerimiento y envía un mensaje explicando la negación.

RI-12 Prohibición definida por el usuario de valores faltantes
Para cualquier columna de una tabla-R base que no sea componente de la llave primaria de la tabla, el DBA puede especificar explícitamente que valores faltantes de un tipo específico se prohiben. Como resultado el DBMS rechaza por inaceptable cualquier ejecución de un comando LR que intente poner una marca A o I en tal columna.

RI-16 No-subversión Otros lenguajes además del LR pueden ser soportados por el DBMS para manipulación de datos. (El modelo relacional no prohibe dichos lenguajes) Si cualquiera de estos lenguajes es no relacional, debe existir una prueba rigurosa en que se muestre que es imposible pasar por alto las restricciones de integridad almacenadas en el catálogo, al emplear uno de estos lenguajes no relacionales.

RI-19 Introduciendo una restricción de integridad a nivel columna para no permitir valores faltantes Cuando un tipo C de restricción de integridad, que no permite la existencia de valores faltantes en una columna específica, se introduce en el catálogo, la base de datos en ese momento puede ser inconsistente con esta restricción. Es decir en la columna en cuestion puede contener varias ocurrencias de valores faltantes. El cumplimiento de dicha restricción se demora en el siguiente sentido. Todas las marcas de valor faltante, se permite que existan hasta que todas sean modificadas. Por otro lado el DBMS prohibe modificar un valor ya existente en la columna por una marca de valor faltante.

RI-22 Técnicas de en vuelo, fin de comando y fin de transacción Si el DBMS aplica técnicas de en vuelo para el chequeo de restricciones de integridad debe poder cambiar a técnicas de fin de comando y fin de transacción, cuando la técnica de en vuelo no funcione.

IV.IV Restricciones de Integridad definidas por el usuario

En este conjunto de reglas tenemos 11 reglas de las cuales 3 son fundamentales y las otras 8 básicas Estas reglas pretenden establecer cuales son los efectos para las reglas que puede definir el usuario, acerca del comportamiento de los mismos.

RI-32 El comando REJECT Si el chequeo es C-temporalizado, rechazar el comando y si el comando es parte de una transacción rechazar la transacción también. Si el chequeo es I-temporalizado, rechazar la transacción.

RI-33 El comando CASCADA Caso 1: Si una llave primaria está siendo modificada o borrada sin usar la opción de cascada en el comando, y el comando CASCADA es usado en respuesta a la violación, el DBMS lleva el efecto CASCADA de modificación o borrado a todas las llaves foráneas. Caso 2: Dejar que D sea el dominio primario. Si un valor de llave foránea de este dominio D se inserta en la base de datos como componente de un renglón y no hay un valor igual en una llave primaria definida sobre D, entonces un nuevo renglón se agrega a una relación que tenga una llave primaria definida sobre D y el valor de la llave primaria es igual al de la llave foránea mencionada con anterioridad. Esta acción tiene lugar sólo si todas los columnas no primarias aceptan marcas, el DBMS ejecuta un REJECT en caso contrario.

RI-34 El comando MARK Si la causa de un intento de violación puede ser impuesto a una(s) columna(s) no de la llave primaria y valores faltantes son aceptables en estas columnas . marcar los componentes como faltantes pero aplicables. Si el marcar falla ejecutar un REJECT

IV.V Indicadores

En este conjunto de reglas tenemos 14 reglas de las cuales todas son básicas. Su propósito es reglamentar los mensajes debidos a las condiciones excepcionales que se pueden presentar durante la evaluación de un comando.

RJ-1 Indicador de relación vacía

RJ-2 Indicador de división vacín

RJ-3 Indicador de información faltante

RJ-4 Indicador de Argumento Faltante

RJ-5 Indicador de dominio no declarado

RJ-6 Indicador de error en el chequeo de dominio

RJ-7 Indicador de dominio no borrable aún existe una columna.

RJ-8 Indicador de renglones duplicados

RJ-9 Indicador de llave primaria duplicada

RJ-10 Indicador de orden no redundante

RJ-11 Indicador de bloque de catálogo

RJ-12 Vista de tuplas no insertables

RJ-13 Vista de componentes no modificables

Rj-14 Vista de tuplas no borrables

IV.VICalificadores

En este conjunto de reglas tenemos 13 reglas de las cuales 3 son fundamentales y las otras 10 básicas. Reglas para las expresiones que pueden ser empleadas en un comando para alterar algún aspecto de la ejecución de dicho comando. Como en el caso de cláusulas MAYBE, ORDER-BY y contabilización de tuplas repetidas.

RQ-3 El calificador MAYBE Este calificador, basado en una lógica de cuatro valores que puede ser aplicada a cualquier expresión de verdad en un comando del LR. El DBMS se enfoca en aquellas características para las cuales la expresión tiene el valor de verdad *a* ó *t* (el cual denota tal vez pero aplicable ó inaplicable).Por ejemplo si el calificador MAYBE se aplica a toda la condición entonces el DBMS agrega al resultado final sólo aquellos elementos para los cuales toda la condición tenga el valor de verdad *a* ó *t*.

RQ-7 El calificador de ORDER-BY Una cláusula de ORDER-BY consiste de lo siguiente:

- Calificador ORDER-BY
- Nombres de las columnas de los operadores cuyos valores actuarán como bases del orden
- Un símbolo ASC o DESC indicando si el orden será ascendente o descendente

Una cláusula ORDER-BY puede ser agregada a un comando relacional que recupera datos. El DBMS proporciona entonces los datos en el orden especificado.

RQ-11 El Degree of Duplication (DOD) agregado Asumir que el calificador DOD ha sido agregado a la proyección de una relación o a la unión de dos relaciones union compatibles. Para cada renglón en el resultado, el DBMS calcula el número de ocurrencias de dicho renglón si se permiten duplicados en el resultado. Esta cuenta se agrega a cada renglón en el resultado actual como un componente extra. El resultado es una relación con una columna extra llamada DOD.

IV.VII Dominios y Tipos de Datos Extendidos

En este conjunto de reglas tenemos 3 reglas fundamentales y 6 basicas. Este conjunto de reglas se refiere a las caracteristicas que deben tener los manejadores de bases de datos en cuanto a la definición y uso de los tipos de datos. Distingue dominio de tipo de dato , por las confiuciones a que se presta el término, al identificar tipo de dato con las definiciones que se tienen en los lenguajes de programación, si se desea efectuar un simil es preferible compara dominio con tipo de dato extendido.

RT-1 Caracteristica de seguridad al comparar valores en la base de datos. Cuando se compara un valor de la base de datos en una columna con otro valor en otra, El DBMS checa que ambas columnas deriven sus valores de un dominio comun.

RT-2 Tipo de datos extendidos que forman parte del sistema. El DBMS soporta fechas, tiempos y campos monetarios como tipos de datos extendidos. Debe tener acceso a la fecha y tiempo actuales en todo momento.

RT-3 Tipo de datos extendidos definidos por el usuario El DBMS permite a los usuarios definir tipos de datos extendidos además de aquellos que provée el sistema.

Las siguientes reglas pertenecen a la capa Externa

IV.VIII Operadores Básicos

En este conjunto de reglas tenemos 31 reglas fundamentales y 6 básicas.Señala las operaciones básicas con que debe poder contar un usuario de cualquier objeto de la base de datos, para recuperar información sin tener que preocuparse de problemas de programación, siendo estos operadores flexibles y poderosos.

RB-1 El énfasis del producto cartesiano como un operador Un DBMS no debe soportar el producto cartesiano como un operador explícito separado, sin embargo un comando relacional puede tener un caso extremo en que se interprete como requerimiento de producto cartesiano.

RB-2 El operador de proyección Tiene como operando una tabla-R, se genera un resultado intermedio en el que se salvan las columnas listadas por nombre en el comando, las demás se descartan, y después se deja sólo una ocurrencia de aquellas que se encuentran duplicadas.

RB-3 a RB-12 El operador theta de selección El operador theta de selección originalmente theta restringido emplea una tabla R simple como su operando. En el uso normal el término theta select se abrevia como select y esto significa que el comparador de igualdad '=' se debe asumir a menos que se especifique de manera explícita otro comparador. Genera como resultado una tabla-R que contiene los renglones que cumplen la condición expresada en el comando. Como la tabla origen no contiene duplicados, el resultado tampoco debe contenerlos.

RB-14 a RB-23 El operador theta junta El operador theta junta emplea dos tablas-R y sus operandos, genera como resultado una tabla-R que contiene renglones de un operando digamos (S) concatenados con renglones de otro operando digamos (T) pero sólo en donde la condición se verifica como verdadera.

RB-25 El operador de junta natural Como se describe un equi-join genera un resultado en el cual 2 de las columnas tienen idénticos valores aunque diferentes nombres de columnas. Estas dos columnas son derivadas de las columnas comparables de los operandos, por supuesto las columnas pueden ser simples o compuestas. De los diez tipos de theta-junta, la equi-junta es la única que en su resultado las columnas de comparación son redundantes. La junta natural se comporta como un equi-join excepto que una de las columnas redundantes, simple o compuesta, es omitida del resultado. Para no afectar la conmutatividad, el nombre de la columna que permanece es aquel que se encuentre primero de acuerdo al orden alfabético.

RB-26 El operador de union El operador de union relacional es intencionalmente no tan general como el operador de union en matemáticas, mientras que este último permite la union de conjuntos que tienen una definición diferente, la union relacional sólo permite unir conjuntos con idéntica definición.

RB-27 El operador de intersección Si S y T son dos relaciones union-compatibles, son suficientemente compatibles para que se aplique el operador de intersección relacional entre ellas.Las columnas se deben alinear de la misma manera en que se alinean para la union. El resultado de la intersección de S y T es una relación que contiene los renglones de S que aparecen en T. El resultado no contiene renglones duplicados ya que los operandos no contenían renglones duplicados.

RB-28 El operador de diferencia Si S y T son dos relaciones union-compatibles, son suficientemente compatibles para que se aplique el operador de diferencia relacional entre ellas.Las columnas se deben alinear de la misma manera en que se alinean para la union. El resultado de la diferncia de S y T es una relación que contiene los renglones de S que no aparecen en T. El resultado no contiene renglones duplicados ya que los operandos no contenían renglones duplicados.

RB-29 El operador de división relacional La división relacional es semejante en algunos términos a la división aritmética de enteros . En la división relacional como en la división aritmética de enteros, existen: divisor, dividendo, cociente y residuo La división relacional tiene operandos y resultados similares.Sólo que en este caso los términos son relaciones. No es necesario que ninguna de ellas contenga datos numéricos, en caso de contenerlos esto no es relevante.

RB-30 Asignación Relacional Cuando se realiza una interrogación a una base de datos el usuario desea tener el resultado en una memoria bajo un nombre de su elección(una relación). Este requerimiento es cubierto por una extensión denominada asignación relacional. Este operador se denota por <__ en la expresión T <____ rve en donde 1) rve (relational value expression), es decir una expresión cuya evaluación da una relación y 2) T un nombre seleccionado por el usuario para la relación que es especificada por rve y debe ser retenida en memoria.

RB-31 Operador de inserción El operador de inserción permite que una colección de uno o mas registros se inserten en una relación. El usuario no tiene control del lugar en donde se inserten, sin embargo la inserción se ve afectada por el método de acceso que haya especificado el DBA para la relación.Se asume que para la inserción de registros en la relación T, el catálogo contiene una descripción detallada de T.

RB-32 Operador de actualización Al manejar una base de datos puede ser necesario modificar los valores en una base de datos, esto se distingue del proceso de inserción, de menos renglones porque el porcentaje de componentes a ser cambiados representan un pequeño porcentaje del número de componentes de cada renglón

RB-35 El operador de borrado El operador de borrado permite a un usuario borrar multiples renglones de una relación (ya sean casos como 0 o 1 renglón no reciben tratamiento especial). Es necesario para el usuario especificar la relación pertinente e identificar los renglones a borrar en cualquiera de los dos casos permitidos por el operador simple de modificación.

IV.IX Manipulación

En este conjunto de reglas tenemos 20 reglas de las cuales 14 son fundamentales y las otras 6 básicas. Describen estas reglas las propiedades generales y capacidades del lenguaje relacional (LR).

RM-1 Acceso Garantizado Cada uno de sus datos (valor atómico) almacenado en una base de datos relacional se garantiza que sea lógicamente accesible por una combinación de nombre de la tabla-R, valor de la llave primaria y nombre de la columna.

RM-2 Sublenguaje de datos analizable sintácticamente Hay al menos un lenguaje relacional (denotado por LR) soportado en el DBMS (no en un paquete adicional de software) tal que (1) los enunciados del LR deben ser capaces de ser representados como cadenas de caracteres, susceptibles de revisión sintáctica y por lo tanto pueden ser escritos o tecleados por un programador, (2) para cada operación de manipulación cada uno de los operandos es una relación y (3) para cada operación de manipulación cada resultado generado es una relación con los indicadores de resultados actuando como una fuente posible de información adicional.

RM-4 Inserción, borrado y modificación de alto nivel El lenguaje relacional soporta recuperación, inserción, modificación y borrado a un nivel uniformemente alto, nivel de conjuntos (muchos registros a un tiempo)

RM-5 Cerradura de las operaciones LR es matemáticamente cerrado con respecto a los operadores relacionales que soporta

RM-6 Bloque de transacción Los comandos BEGIN y COMMIT identifican el principio y fin de un bloque de comandos. Al menos uno de los comandos en el bloque debe ser expresado en el lenguaje relacional, otros pueden ser expresados en el lenguaje anfitrion, en el relacional o en ambos. Tal bloque constituye una transacción, durante su ejecución, ya sea que todas sus partes sean exitosas o no. ROLLBACK indica al DBMS (1) terminar esta ejecución de la transacción requerida por el programa y (2) evitar efectuar ninguno de los cambios que ya se habían solicitado durante la ejecución de la transacción.

RM-8 Modo Dinámico El DBMS soporta los siguientes tipos de cambios dinámicamente, es decir sin traer la actividad sobre información regular a un estado de alto, sin cambiar el código fuente de ningún programa de aplicación y sin recompilación offline de ningún enunciado fuente en LR.

1.- Creación y borrado de dominios, tablas- R y columnas para tablas-R ya definidas

2.- Creando la nueva y borrando la vieja representación de almacenamiento para partes de la base de datos

3.- Creando y borrando rutas de acceso orientadas al desempeño

4.- Cambiando la autoridad sobre la información en el catálogo

5.- Cambiando declaraciones en el catálogo (tipos de datos, funciones definidas por el usuario, integridad referencial)

RM-9 Modo Triple El mismo lenguaje LR puede ser empleado de tres maneras diferentes. 1.- puede ser usado de manera interactiva aún en terminales. 2.- enunciados en LR pueden ser incorporados en programas de aplicación. 3.- enunciado en LR pueden ser combinados para la acción a tomar en el caso de intentar violar la integridad referencial.

RM-10 Lógica de cuatro valores: Tablas de verdad El DBMS evalúa todas las expresiones usando la lógica de cuatro valores definida por las siguientes tablas:

			Q						Q				
P	not P	PVQ	t	a	i	f	P^Q		t	a	i	f
t	f		t	t	t	t	t		t	t	a	i	f
a	a	P	a	t	a	a	a	P	a	a	a	i	f
i	i		i	t	a	i	f		i	i	i	i	f
f	t		f	t	a	f	f		f	f	f	f	f

En estas tablas t representa a verdadero , a faltante pero aplicable, i faltante e inaplicable y f para falso. Notar que t,a,i y f son valores actuales y no deben ser vistos como valores marca. Porque los conjuntos recuperados y manipulados en el manejo de la base de datos son todos finitos, los cuantificadores existencial y universal pueden ser tratados como OR iterativo y AND iterativo respectivamente.

RM-11 Información Faltante: Manipulación A través de la base de datos la información faltante es manipulada por el DBMS de manera uniforme y sistemática y en particular independiente del tipo de dato.
Los usuarios deben ser capaces de explotar todo el poder expresivo de la lógica de predicados de cuatro valores en LR. En particular el calificador MAYBE debe aplicarse a cualquier expresión de verdad, ya sea a una condición lógica completa o sólo a una parte de dicha condición.

RM-12 Operadores Aritméticos: Efecto de valores faltantes Un valor marcado en una columna no puede ser aritméticamente incrementado o decrementado por el DBMS, los valores no marcados son susceptibles de tales operadores.

Si x denota un valor numérico en la base de datos, A una marca-*a* e I una marca-*i*.

x + x = 2x	x + A = A	A + x = A
A + A = A	A + I = I	I + A = I
I + I = I	x + I = I	I + x = I

RM-13 Concatenación: Efecto de valores casados Un valor marcado en una cadena de caracteres, no es sujeto de concatenación con ninguna otra cadena, por medio del DBMS, sin embargo los valores no marcados pueden hacerlo.

Sea ^ el operador de concatenación y *x* una cadena y *x* una cadena no marcada. Utilizando A e I como en RM-12

$x \hat{} x = x$	$x \hat{} A = A$	$A \hat{} x = A$
$A \hat{} A = A$	$A \hat{} I = I$	$I \hat{} A = I$
$I \hat{} I = I$	$x \hat{} I = I$	$I \hat{} x = I$

RM-14 Operadores restringidos al dominio y DOMAIN CHECK OVERRIDE Aquellos operadores relacionales que envuelven comparación de valores de la base de datos están normalmente restringidos a comparar pares de valores sí y solo si ambos son derivados del mismo dominio y por lo tanto tienen el mismo tipo de dato extendido.

Seguramente no habrá ninguna necesidad de sobreescribir esta restricción. Sin embargo si la necesidad se presenta el calificador DOMAIN CHECK OVERRIDE o (DCO) debe ser agregado al comando o a una expresión en el comando.

RM-15 Operadores restringidos por el tipo de datos básico Al usar un operador que normalmente compara pares de valores en la base de datos, al comparar un valor generado por una función con un valor en la base de datos u otro valor generado por función la característica enunciada en RM-14 de que los valores a ser generados deben pertenecer al mismo tipo de datos extendido se relaja, únicamente se requiere que pertenezcan al mismo tipo de datos básico.

RM-18 El sublenguaje de datos comprensivo El lenguaje relacional LR es comprehensivo con respecto a la administración de base de datos en cuanto a que soporta todos los requerimientos siguientes:
(1) Definición de información, (2) definición de vistas, (3) manipulación de información, (4) integridad referencial, (5) autorizaciones y (6) limitaciones de transacciones.

IV.X Vistas

En este conjunto de reglas tenemos 8 reglas de las cuales todas son fundamentales.Pretenden aislar a los usuarios de las relaciones base, permitiendo el cambio a las mismas a través de las vistas, así como en la definición de las vistas. Las vistas son una manera de restringir el acceso de un usuario a sólo la información que necesite.

RV-1 Definición de vistas: Qué son? Las vistas son relaciones virtuales representadas por su nombre y definición unicamente. El DBMS no almacena información extra. El DBMS almacena la definición de vistas en el catálogo, y soporta la defición de vistas en las tres alternativas siguientes: 1) tablas R-base solamente, 2) otras vistas solamente o 3) la mezcla de tablas R-base y vistas.

RV-2 Definición de vistas: Qué no son? Ninguna definición de vista es de naturaleza procedural, ni tiene información del almacenamiento, ruta de acceso o metodo de acceso, actualmente empleado por alguna parte de

la base de datos, ya sea que estas técnicas empléen relaciones o registros como operandos.

RV-3 Definición de vistas: Retención e interrogación Las vistas se definen utilizando LR y se almacenan sus definiciones en el catálogo. Deben ser requeridas empleando el mismo lenguaje LR que se emplea en las interrogaciones a la información regular. Tanto en la definición como en el manejo de la vista se debe poder emplear todo el poder del LR.

RV-4 Recuperación empleando vistas Ni el DBMS ni el LR hacen ninguna diferencia evidente al recuperar información, entre tablas R-base y vistas. Es más cualquier interrogación puede ser convertida en vista simplemente con prefijarla con CREATE VIEW.

RV-5 Manipulación empleando vistas Ni el DBMS ni el LR hacen ninguna diferencia evidente al recuperar información, entre tablas R-base y vistas, excepto que 1) algunas vistas no permitan inserción o borrado de datos así como actualización de ciertos valores y 2) que algunas vistas no tengan llave primaria y por ello no les sea posible aceptar estos operadores de manipulación.

RV-6 Actualización de vistas Para evaluar la facultad de actualizar de las vistas al tiempo de definición, el DBMS incluye una instrumentación del algoritmo VU-1 u otro más fuerte.Ni el DBMS ni el LR hacen ninguna diferencia evidente al recuperar información, entre tablas R-base y vistas, excepto que:

1.- Algunas vistas no permitan inserción o borrado de datos así como actualización de ciertos valores porque el algoritmo no soporte dicha acción y...
2.- Que algunas vistas no tengan llave primaria y por ello no les sea posible aceptar estos operadores de manipulación, ya que requieren llaves primarias en los mismos.

RV-7 Nombre de columnas y vistas
Fundamental

Al crear vistas el LR permite al usuario nombrar a las columnas de manera diferente con respecto al nombre de la columna fuente.El DBMS sin embargo retiene en el catálogo el nombre de la columna en la vista así como el correspondiente de la columna fuente.

RV-8 Dominios aplicables a columnas en vistas Una vista es creada empleando una definición que no estipula para cada columna el dominio del cual deriva sus valores. La identificación de dominios se lleva a cabo al momento de la definición de la vista y se almacena en el catálogo como parte de la definición. Sin embargo si los valores de la columna son derivados de algún cálculo se emplea el tipo de dato básico en vez del extendido.

IV.XI Operadores Avanzados

En este conjunto de reglas tenemos 44 reglas de las cuales 2 son fundamentales y las demás básicas. A diferencia de los demás conjuntos de reglas en este sólo describiremos las 2 reglas fundamentales. El conjunto de operadores avanzados tiene como propósito incrementar la flexibilidad y poderío de las operaciones que se pueden realizar sobre los datos de la Base de Datos sin necesidad de programar.

RZ-1 Enmarcando una relación Un marco separa el conjunto de renglones de la partición de cualquier otro miembro. Esta separación se logra al agregar una columna nueva a la relación y asignar un valor distinto para cada miembro distinto de la partición. El nombre estandar para esta columna es FID (Frame Identifier) ó identificador de marco (Group by de SQL)

RZ-2 Extendiendo la descripción de una relación para incluir todas las columnas de otra relación. La relación que aparece primero en el comando es la que primero se extiende para incluir todas las columnas de la segunda que no se encontraban en la primera. Entonces los valores introducidos se tendrán como A-marked a menos que el calificador de valor RQ-13 se aplique a un valor en particular.

El siguiente conjunto de reglas pertenece al nivel interno

IV.XII Nombramiento

En este conjunto de reglas tenemos 14 reglas de las cuales 7 son fundamentales y las otras 7 básicas.Este conjunto de reglas no pretende dar reglas para el DBA sobre los estándares que deben cubrir los nombres de sus objetos. Sino que pretende limitar el nombramiento para que la base de datos sea fácil de emplear y se eviten las ambigüedades.

RN-1 Nombramiento de dominios y tipos básicos Todos los dominios (tipos de datos extendidos) sean simples o compuestos, cuando sean incluidos o definidos por el usuario, se les debe asignar nombres que los distingan a unos de otros y diferentes a los nombres para relaciones o funciones.

RN-2 Nombramiento de relaciones y funciones Todas las relaciones sean base o derivadas y todas las funciones sean incluidas o definidas por el usuario, se les debe asignar nombres que sean distintos entre sí, así como tambien de los nombres de dominios, tipos de datos y columnas.

RN-3 Nombramiento de columnas Todas las columnas sean simples o compuestas, dentro de una relación se les debe asignar nombres que sean distintos unos de otros y distintos de los nombres de relaciones o funciones.

RN-4 Seleccionando columnas con comandos relacionales La combinación de nombre de relación y nombre de columna es una forma que no presenta ambigüedad en la selección de una columna en una base de datos en particular. El lenguaje relacional debe evitar separar los nombres de las relaciones de los de las columnas, lo que causa (1) dificultad de extender el lenguaje y (2) ya sea ambigüedad o dificultad innecesria por parte de los usuarios de extender los comandos relacionales.

RN-5 Libertad de nombramiento El éxito del DBMS en la ejecución de un comando del lenguaje relacional que incluya comparación de valores en la base de datos pertenecientes a diferentes columnas no debe depender de las columnas con nombres idénticos.

RN-6 Nombre de columnas que intervienen en la clase de operadores de union. En el lenguaje relacional cuando el usuario solicita
R UNION S, el o ella no tiene que especificar que columna de R se alinea con que columna de S excepto para aquellas columnas de R y S en donde dos ó mas columnas de R (o de S) deriven sus valores de un dominio

común. Lo mismo aplica para la intersección, la diferencia y sus contrapartes "outer", outer-union, outer-intersection y outer-difference

RN-7 No-afectación de la conmuntatividad Dados cualquiera de los operadores relacionales que tengan dos operandos y que sean conmuntativos, la regla construida en el DBMS para el nombramiento de columnas del resultado no debe afectar la conmuntividad. De manera similar no debe afectar cualquier otra identidad simple que aplique los operadores.

IV.XIIIComandos para el DBA

En este conjunto de reglas tenemos 22 reglas de las cuales 8 son fundamentales y las otras 14 básicas.El propósito de estas reglas es proporcionar los límites y funciones que el DBA debe tener a fin de poder desempeñar su trabajo. Si bien hay funciones parecidas en algunos manejadores ligadas a problemas de espacio o de evaluación de las trayectorias de acceso, estos servicios se encuentran más ligados a la instrumentación que a la definición del modelo.

RE-3 Comando de creación de un dominio Este comando establece un nuevo dominio como un tipo de dato extendido. La información que se proporciona como parte del comando incluye el nombre (seleccionado por el DBA) el tipo de datos básico, un rango de valores y si es significativo aplicar el operador < a estos valores.

RE-6 El comando de borrado de dominio Este comando borra un dominio, previendo que no existan columnas que deriven sus valores de este dominio. Si existiera una columna en este caso , un indicador se prende para señalar que este fué el caso y el comando fué abortado. Si existe un índice basado en tal dominio, si se borra el dominio se borra también el índice.

RE-7 Comando para la creación de una tabla-R Este comando guarda la definición de una tabla-R o una vista en el catálogo.Todos los dominios citados en el comando deben estar previamente definidos. De otro modo el comando aborta y el indicador de dominio no definido se prende.
La siguiente información se proporciona como parte del comando:

- El nombre de la tabla-R
- Si es una vista, su definición en terminos de la tabla- R base y otras vistas
- El nombre de cada columna
- Para cada columna el nombre del dominio del que deriva sus valores
- La combinación de valores que forman la llave primaria o el identificador débil
- Para cada llave foranea, que combinación de columnas constituyen la llave y que llaves primarias son el destino.Esta carncterística es vital en las tablas-R y menos importante en vistas.

RE-9 Comando para borrar una tabla-R Cuando una tabla-R base o una vista, sea S, es borrada varias de las partes de la descripción de la base de datos se ven afectadas: relaciones de integridad, vistas y relaciones de autoridad. Se debe recordar que una relación de integridad puede abarcar 2 o más tablas-R. De tal manera que no sólo se vea involucrada la tabla-R S, sino otras tablas. La definición de una vista tambien puede citar varias tablas de las cuales S sea sólo una. Será entonces necesario borrar las relaciones de autoridad basadas en la tabla-R.

RE-10 Comando para agregar una columna Este comando especifica el nombre de una tabla-R existente. El DBMS agrega a la descripción de dicha tabla en el catálogo el nombre de la nueva columna que deriva sus valores de un dominio predefinido, el nombre del dominio se proporciona como parte del comando. Cada renglón de dicha tabla debe incluir un valor para dicha columna. Hasta entonces se marca como A-marcado faltante a menos que el calificador RQ-13 se especifique en el comando

RE-13 Comando de borrado de columna Este comando hace que todos los valores de cada renglón que pertenezcan a la columna especificada sean inaccesibles a todos los usuarios. Estos componentes se remueven al tiempo de reorganización

RE-14 Comando para la creación de índice Este comando se pretende diseñado para un DBMS que explote los índices. Crea la descripción de un índice y la guarda en el catálogo. También crea el índice aunque no necesariamente de manera inmediata. Si el DBMS recibe varios requerimientos sucesivos de índices el DBMS intenta crearlos todos en una sala pasada de datos, con esto se pretende mejorar el desempeño.

RE-16 Comando para borrado de índices Este comando se pretende sea diseñado para un DBMS que explote los índices.Borra un índice cuyo nombre se le proporciona o reporta la no existencia de dicho índice.

IV.XIV Catálogo

En este conjunto de reglas tenemos 11 reglas de las cuales 8 son fundamentales y las otras 3 básicas. Una característica importante del modelo relacional es que tanto la base de datos como su descripción, se perciba por el usuario como una colección de relaciones. Salvo algunas excepciones el mismo lenguaje relacional empleado sobre las relaciones, debe ser empleado para consultar y modificar las definiciones de las mismas.

RC-1 Catálogo dinámico en Linea El DBMS soporta un catálogo dinámico en línea en el modelo relacional. La descripción de la base de datos (a nivel lógico) como datos ordinarios, permitiendo a los usuarios autorizados aplicar el mismo lenguaje relacional para interrogar sobre la descripción de la base de datos como datos regulares.

RC-2 Concurrencia El DBMS tienen un mecanismo de control de concurrencia lo suficientemente sofisticado que pueda soportar multiple actividad de recuperación y manipulación sobre el catálogo , en datos regulares o en ambos de manera concurrente.

RC-3 Descripción de Dominios Para cada dominio distinto sobre el cual la base de datos se construye, el catálogo contiene su nombre, su tipo de datos básico, el rango de valores permitido y si es el caso el comparador (<) MENOR QUE aplicable sobre los valores que se derivan de este dominio.

RC-4 Descripción de Tablas-R base Para cada tabla-R base, el catálogo contiene al menos las siguientes características: (1) el nombre de la tabla-R, (2) sinónimos para este nombre, si existe alguno (3) el nombre para cada columna, (4) para cada columna, el nombre de un dominio ya previamente declarado, (5) para cada columna, que tipo de valores faltantes son permitidos. (6) para cada columna , cuando los valores de la columna se requieren que sean distintos, (7) para cada columna, restricciones más allá de las declaradas para el dominio, (8) para cada

columna, el tipo de datos básico, si aplicable, (9) si la columna es un componente de la llave primaria para una tabla-R y (10) para cada llave foranea la secuencia de columnas de la cuales se compone y las llaves primarias destino en la base de datos.

RC-6 Descripción de vistas Para cada vista el catálogo contiene al menos una de las siguientes características: (1) nombre de la vista, (2) sinónimos para este nombre, si existe alguno (3) el nombre de cada columna simple, (4) para cada columna, el nombre de un dominio ya declarado (ha menos que la columna no se derive directamente de una columna con base simple), (5) cuando la columna sea un componente (posiblemente el único) de la llave primaria de la vista, (6) la expresion en LR que define la vista, (7) si se permiten inserciones en la nueva vista por el DBMS, (8) si se permiten borrados en la nueva vista por el DBMS y (9) para cada columna en la vista, si se permiten modificaciones en la nueva vista por el DBMS.

RC-8 Restricción de integridad referencial Para cada restricción de integridad de tipo R, el catálogo contiene una definición completa. Esto incluye su nombre, el evento de trigger, tipo de temporalización, las llaves que están involucradas y la respuesta a cualquier violación.

RC-10 Información de autorización El catálogo contiene toda la información de que usuarios interactivos, que terminales y que programas de aplicación están autorizados para accesar que partes de la base de datos empleando que tipo de operaciones bajo que condiciones.

RC-11 Estadísticas de la base de datos en el catálogo El catálogo contiene toda la información estadísitca acerca de la base de datos, que es empleada por el optimizador para compilar y precompilar comandos LR. Esto incluye al menos (1) el número de renglones en cada tabla R-base y (2) el número de valores distintos en cada columna de cada tabla-R base (no sólo columnas indexadas).

IV.XV Estructura

En este conjunto de reglas tenemos 9 reglas fundamentales y 5 básicas. Se puede efectuar la comparación entre una base de datos y una base de conocimiento , si limitamos a la primera a ser un conjunto de aserciones. La transparencia en la descripción de la base de datos es tambien una característica importante, es decir tener acceso a la información, características de columnas, manejo de información faltante así como trabajar bajo la arquitectura de 3 niveles ANSI. (Regla básica RS-5)

Las reglas fundamentales de estructura se enuncian a continuación así como la básica RS-5 :

RS-1 Característica informativa Es una regla que se refiere a que tanto la información que perciben los usuarios finales, como los programadores de aplicaciones está en las relaciones explícitamente, sean estas bases o no.

RS-2 Independencia de los conceptos posicionales El manejador debe evitarle tanto a los programadores de aplicaciones como a los usuarios finales, el tener que conocer la posición de un valor en la base de datos.

RS-3 Renglones duplicados están prohibidos en toda relación El DBMS debe prohibir la ocurrencia de renglones duplicados en toda relación ya sea esta base,derivada o vista.

RS-4 Portabilidad de la información Se refiere fundamentalmente a que la información debe ser la misma no importando el dispositivo en que se encuentre almacenada

RS-5 Arquitectura de tres niveles Se refiere a que un RDBMS debe contar con tres niveles consistentes de vistas, relaciones base y almacenamiento

RS-6 Declaración de los dominios como tipo de datos extendidos Cada dominio semánticamente distinto es distinguido por nombre y declarado en forma separada de las declaraciones de la tabla-R.

RS-7 Definición de columnas Para cada columna de una tabla-R debe existir la capacidad de: 1) declarar de que dominio la columna deriva sus valores y 2) que las restricciones existentes se apliquen a los valores de la columna.

RS-8 Llave primaria para cada tabla-R base Para todas las tablas el DBMS requiere una y sólo una llave primaria,requiere que los valores de la columna ya sean simples o compuestos sean diferentes en todo momento.

RS-10 Llave foranea El DBMS permite la declaración de una columna o la combinación de varias de una tabla-R como una llave foranea.

RS-13 Información Faltante: Representación. Através de la base de datos, la omisión de un valor se representa de manera uniforme y sistemática independientemente del tipo de dato a que corresponda el valor faltante.

ESQUEMAS COMPARATIVOS DE LOS DIFERENTES MANEJADORES

I Principios

Si bien para la instrumentación de los SMBDR Codd no proporciona una tecnica absoluta para la instrumentación, si enuncia ciertas reglas que se incluyeron en el capitulo 2. Al igual que para el lenguaje interrogador.

II Arquitectura General

A continuación se tiene una descripción de los diferentes componentes de cada uno de los manejadores analizados. Apartir de estos esquemas es que se definió el esquema de arquitectura general que se describió en el capitulo 2.

DB2

Trabaja bajo el concepto de espacio de direccionamiento (address space) y tiene por lo menos 3 epacios(address space) activos:

SYSTEM SERVICES ADDRESS SPACE (DSNMSTR)

Componente que:

- Controla las diferentes conexiones con el subsistema
- Activa las tareas necesarias para inicio y paro del sistema
- Maneja los componentes de recuperación (Bootstrap dataset y LOG)
- Provée estadísticas del comportamiento del sistema

DATABASE SERVICES ADDRESS SPACE (DSNDBM1)

Componente que está integrado por cinco subpartes:

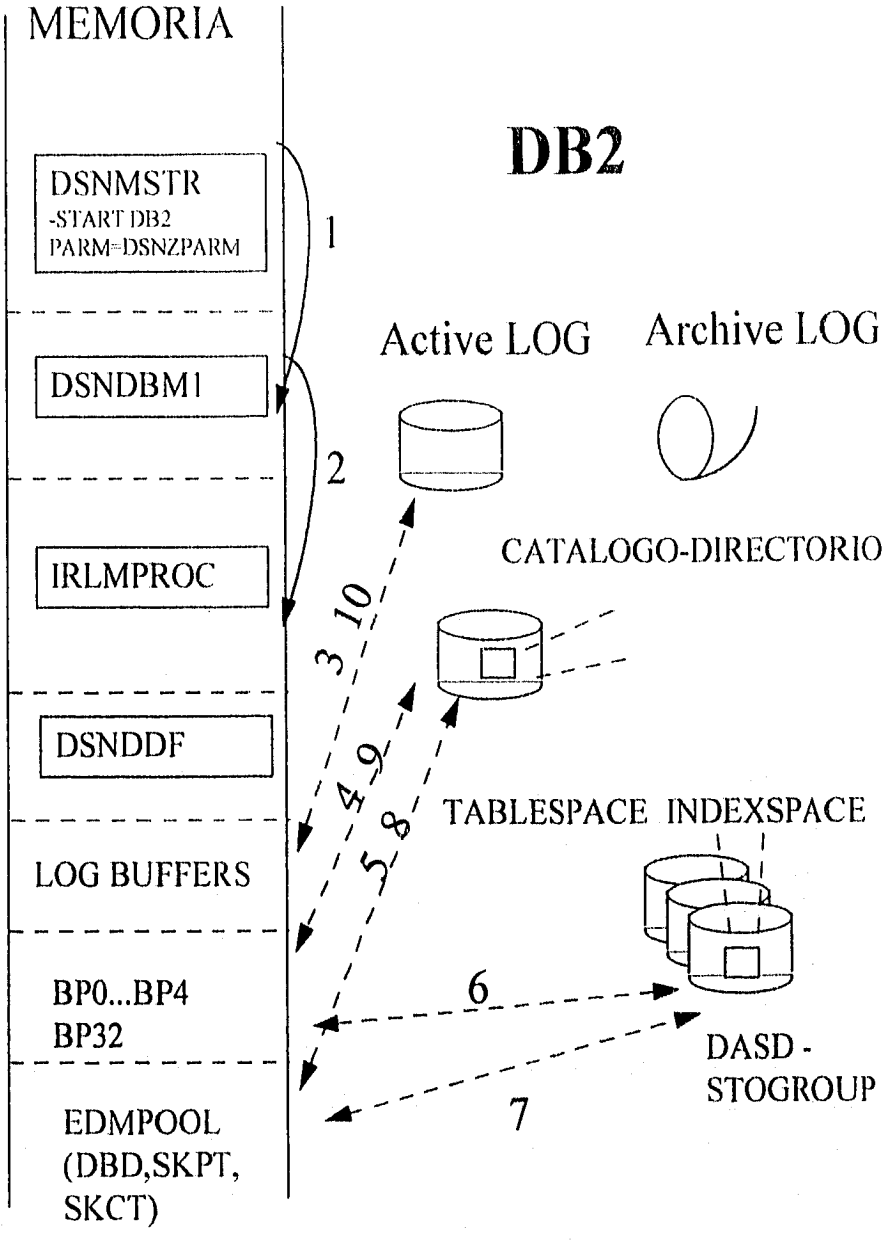
- PRECOMPILER: Procesa lenguajes HOST y reemplazat el código SQL por CALL. Construye un DBRM (Database Request Module)
- BIND: Compila uno o más DBRM para producir un plan
- RELATIONAL DATA SYSTEM(RDS):Determina el mejor mecanismo para satisfacer el requerimiento y lo manda al DATA MANAGER
- DATA MANAGER (DM): Analiza el requerimiento de información y después llama al Buffer Manager para satisfacerlo
- BUFFER MANAGER (BM): Busca si la información está en los buffers si no, hace uso de VSAM MEDIA MANAGER para recuperar la información.

IMS RESOURCE LOCK MANAGER (IRLMPROC)

Componente que controla el acceso concurrente a los datos.

Memoria

WORKING STORAGE: Area de trabajo propia de cada aplicación, en la que se manejan los resultados.
ENVIRONMENT DESCRIPTION MANAGER(EDM): Almacena las rutas de acceso de los planes y estructuras de datos como Cursor Tables y Database Descriptors (DBD)
BUFFER POOL Memoria en que se almacenan los datos de manera temporal
PARÁMETROS DE INICIALIZACIÓN: DSNZPARM es un archivo en el cual se establecen los parámetros del subsistema que van a dimensionar varios componemes ver Capitulo5


MEMORIA
DB2
DSNMSTR
-START DB2
PARM=DSNZPARM
1
DSNDBM1
2
IRLMPROC
DSNDDF
LOG BUFFERS
BP0...BP4
BP32
EDMPOOL
(DBD,SKPT,
SKCT)
Active LOG
Archive LOG
CATALOGO-DIRECTORIO
3 10
4 9
5 8
TABLESPACE INDEXSPACE
6
7
DASD -
STOGROUP

SYBASE

Trabaja bajo el concepto de memoria compartida para programas y datos

Se tiene en el SQL/SERVER siete componentes mayores:

EXTERNAL INTERFACES
Es el componente que maneja toda la comunicación entre el cliente y la base de datos servidor.Para lograr esto cuenta con:

- Manejo de un protocolo que delimita paquetes de datos, identifica el tipo y provée mecanismos de interrupción.
- Direcciona los requerimientos del usuario a : SQL Interface o Transaction Interface
- Realiza el chequeo de nombres de los objetos pedidos
- Realiza el chequeo de sintáxis de los requerimientos

SEQUENCER/DISTRIBUTOR

Supervisa la ejecución de una transacción, puede manejar varios sequence trees (planes de rutas de acceso definidos después de la precompilación, sql estático), así como también puede requerir los servicios necesarios para armar el sequence tree (sql dinámico). Es el responsable de coordinar el trabajo necesario para satisfacer un requerimiento.

QUERY PROCESSING

Es el responsable de convertir los requerimientos de alto nivel (enunciados SQL) en un formato optimizado e interno. Consta de dos subcomponentes: Decision/Compile. Los servicios que presta este componente son:

- Optimization: Trata de determinar cual es el acceso más eficiente de acuerdo a la naturaleza de los datos.
- Plan Generation: El encargado de generar los planes de queries. De manera tal que se encuentran los nombres de los objetos que se requieren para lograr un buen acceso, formando parte del plan.
- Authorization Requirements Determination: Proceso que determina que autoridades son requeridas para la ejecución.
- Plan Validation: Los planes que se han precompilado son checados a tiempo de ejecución para asegurar que los cambios a la base de datos no invalide los planes.
- Execution: Los planes se interpretan a tiempo de ejecución según sea necesario.

TRANSACT ACCESS METHODS /MANAGERS

Consta de varios componentes los dos primeros son. TRANSACTION MANAGER y ACCESS METHOD. El tercer componente MANAGER consta de seis subcomponentes

- Allocation Manager: Maneja la ubicación de espacio en disco
- Buffer Manager: Maneja el uso y disponibilidad de buffers
- Index Manager: Mantiene los índices Cluster y No-cluster
- Lock Manager: Encargado de coordinar los requerimientos de LOCK
- Sort Manager: Crea índices ,remueve duplicados y ordena resultados
- Text/Image Manager: Maneja los tipos de datos TEXTO e IMAGEN. realizando operaciones especiales sobre ellos.

KERNEL
Proporciona rutinas primitivas que proporcionan servicios como I/O de cinta y disco,comunicaciones en la red, control y monitoreo de procesos.

Memoria

Area de memoria para los componentes del servidor

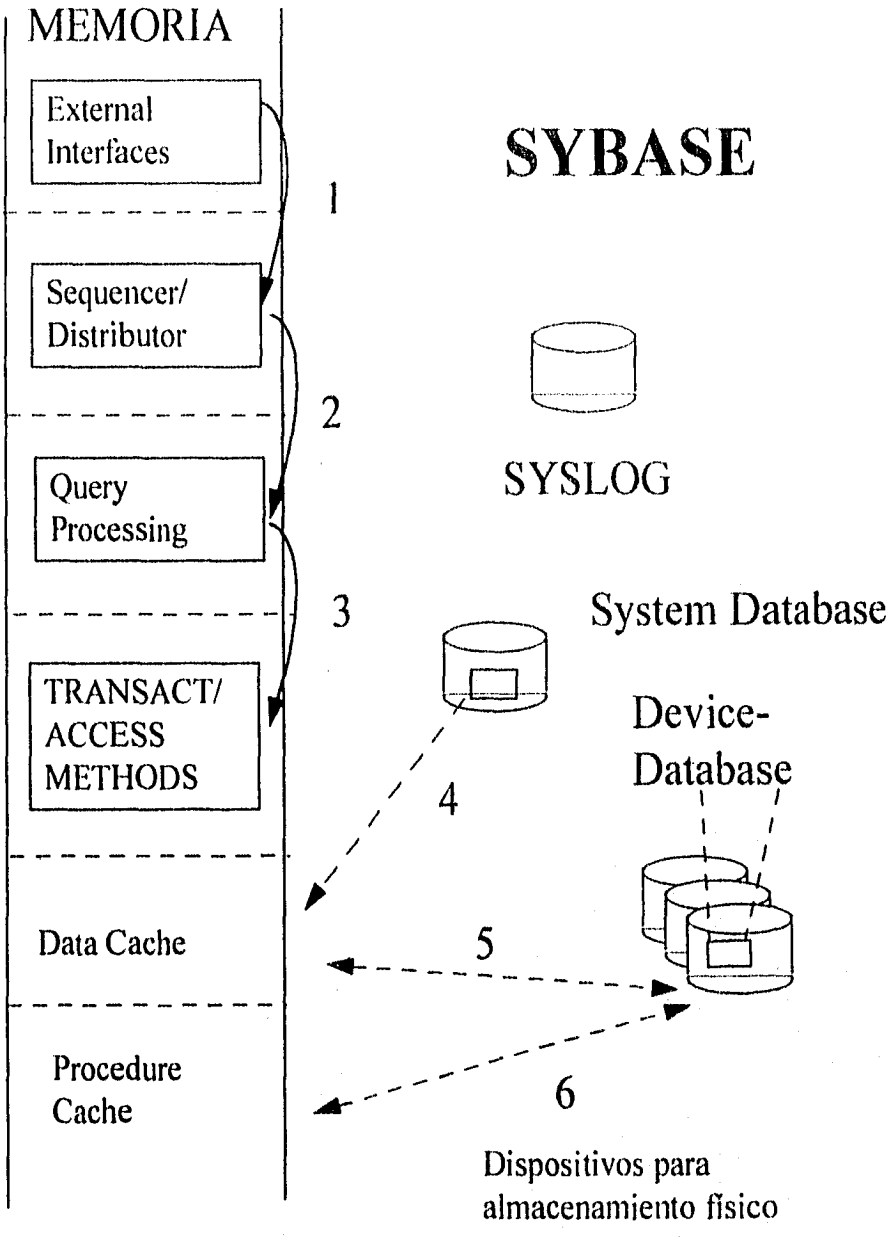
Data Cache: Area de trabajo para las bases de datos

Procedure Cache: Area de trabajo para la ejecución de procedimientos almacenados.

Parámetros de Inicialización

Algunos parámetros de configuración se pueden cambiar en línea sin necesidad de tirar y levantar el servidor. Las tablas del systema en que se almacenan estos parámetros son: sysconfigures y syscurconfigs. Esto se lleva a cabo mediante la ejecución del procedimiento sp_configure. Los parámetros de inicialización se pueden dividir en cuatro categorias:

- Resource allocation
- Memory and Disk allocation
- Operational Behavior
- Remote Server Access


MEMORIA
External Interfaces
Sequencer/ Distributor
Query Processing
TRANSACT/ ACCESS METHODS
Data Cache
Procedure Cache
SYBASE
1
2
3
4
5
6
SYSLOG
System Database
Device-Database
Dispositivos para almacenamiento físico

ORACLE

Trabaja bajo el concepto de instancia, cada una de ellas tiene

De 5 a 9 procesos ejecutables:

DATA BASE WRITER (DBWR)
Maneja la transferencia de datos en la System Global Area(SGA), a los archivos de datos

LOG WRITER (LGWR)
Transfiere información de los buffers de redo a los archivos de log database

SYSTEM MONITOR (SMON)
Realiza la recuperación de instancias o su inicialización. Es el responsable de limpiar segmentos temporales. En una ambiente de proceso paralelo, este componente recupera los nodos caídos.

PROCESS MONITOR (PMON)
Este proceso recupera los procesos fallidos de usuarios y limpia la memoria cache. Recupera los recursos atrapados por procesos fallidos.

ARCHIVER PROCESS (ARCH)
Si se activó archive loggind, este proceso tiene efecto. Guarda los redo logs, ya llenos en los archivos archive log.

Memoria

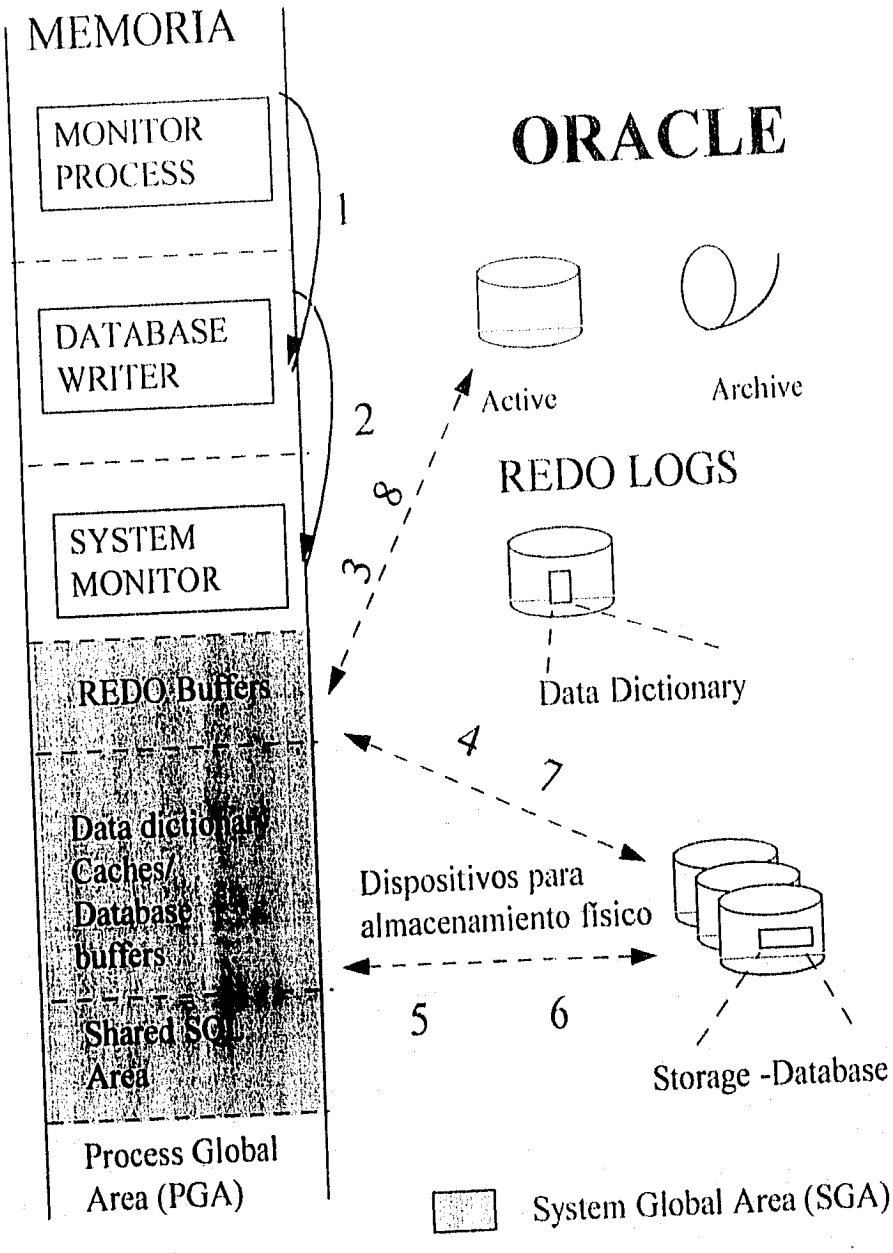
SYSTEM GLOBAL AREA (SGA)
Es un área de memoria central que contiene buffers para auxiliar en el performance y mantener la integridad y consistencia de los datos. Toda la información pasa por la SGA esta se divide en tres areas:

- Data dictionary caches/ database buffers
- Redo log buffers
- Colas de requerimiento y respuesta , así como un área de SQL compartida.

PROCESS GLOBAL AREA (PGA)
Area reservada para cada proceso que emplée ORACLE

Parámetros de inicialización

INIT.ORA


MEMORIA
MONITOR PROCESS
DATABASE WRITER
SYSTEM MONITOR
REDO Buffers
Data dictionary Caches/ Database buffers
Shared SQL Area
Process Global Area (PGA)
ORACLE
Active
Archive
REDO LOGS
Data Dictionary
Dispositivos para almacenamiento físico
Storage -Database
System Global Area (SGA)
1
2
3
4
5
6
7
8

III Operación

La siguiente tabla ilustra los diferentes componentes de operación y como se identifican en el manejador

Componentes:	DB2	SYBASE	ORACLE
Almacenamiento físico	Tablespaces		
Bitácora de la actividad	Log Activo Log Archive Log Buffers		REDO Log Buffer
Información de peticiones	SKPT EDM POOL	Procedure Cache Data Cache	SGA
Buffers de datos	BPn		Database Buffer Cache Shared Pool

IV Almacenamiento de las estructuras del SMBDR

En cada manejador hay diferencias en la instrumentación de catálogo y diccionario, estas se ilustran en la siguiente tabla

Catálogo y Diccionario		
DB2	SYBASE	ORACLE
Información sobre el subsistema se almacena en tablas del catálogo y directorio	Cada servidor tienen una master database que consta de 28 tablas	Se tiene un diccionario de datos por base de datos
Diccionario contiene DBD's y SKCT's	Algunas tablas del sistema están en cada base de datos y son 14 tablas	Las tablas base estan encriptadas, difíciles de accesar
Catálogo esta formado por cuarenta tablas	Se puede configurar para permitir updates	Vistas sobre el diccionario ALL_, USER_, DBA
Acceso a la información del catálogo es vía SQL (Excepto directorio)	Las tablas del sistema se crean al crear la Base de Datos	Vistas sobre la información para configuración
No se puede emplear DML para modificar la información directamente en el catálogo		Vistas DBA
Existe uno por subsistema		DML se puede emplear en el diccionario

VI Objetos de la Base de Datos

OBJETO	CARACTERISTICAS	CARACTERISTICAS	CARACTERISTICAS
	DB2	SYBASE	ORACLE
DATABASES	Máximo 65279	Contiene algunas tablas del sistema Superconjunto de la base de datos modelo Máximo 32767 Unidad de recuperación	Una por instancia Contiene tablas del diccionario y sistema Unidad de recuperación
TABLESPACES	Físico Asociado a un Storagegroup Unidad de recuperación Puede contener una o más tablas	N/A	Físico Creado en datafiles Almacenamiento de default Puede contener uno a muchos • tablas • indices • segmentos rollback unidad de recuperación
TABLES	Asociadas a un tablespace Máximo 750 columnas Longitud 4056 o 32714 bytes	Creada en BD o segmento Máximo 250 columnas Longitud 1962 bytes	Asociadas a un tablespace Almacenamiento específico al crear la tabla Máximo 254 columnas Longitud depende del tamaño del bloque
INDICES	Físico Asociado a un storagegroup	Creado en una BD o segmento Ignore duplicate - key Puede compartir espacio con su tabla Indices cluster no tiene páginas hoja Las páginas de datos son referenciadas por páginas no-hoja	Creado en un tablespace Puede compartir espacio con su tabla Create index no-sort Cada entrada tiene la llave de indice + RID Clusters se pueden construir en más de una tabla Almacenamiento especificado en CREATE INDEX

OBJETO	CARACTERISTICAS	CARACTERISTICAS	CARACTERISTICAS
	DB2	SYBASE	ORACLE
Columnas	Integer (5b),smallint(3b),float(n) o real (5b si 1<=n<=21), double precision o float(n) (9b si 22<=n<=53), decimal (x,y) o numeric,char(n) (1<= n<=2546),varchar (n) (3<=n+3<=longreg),long varchar,date(5b),time(4b),timestamp(11b) graphic(1<=n<127b) vargraphic (n<127) long vargraphic	nchar,char,varchar,nvarchar,binary, varbinary,int (4b), smallint(2b),tinyint(1b),float(8b),real(4b), money(8b),smallmoney(4b), datetime(8b),smalldtetime(4b),bit(1b),text(2064b),image(2064b),timestamp(8b)	char (n) (n <= 255b), varchar(2n) (n<2000b) number(p,s)(n<21b),date (7b), Long (2Gb) raw(n) (n<2000b),longraw(n)(n<2Gb) rowid(6b)
Integridad Referencial	Primaria-Foranea Reglas de borrado, cascada,set null y restrict	Referencial,triggers,reglas,default values	Referencial,triggers,check,default values
Planes Paquetes Procedimientos	Planes-paquetes (bound) Autoridades almacenadas Ruta de acceso almacenada SKCT copia para cada usuario PLAN-TABLE - EXPLAIN	Cache del procedimiento contiene el SQL y el T-SQL. El plan de ejecución es almacenado. Los procedimientos son compilados y almacenados en las tablas del sistema. EXPLAIN PLAN, PLAN-TABLE	Pool compartido contiene planes de ejecución y el diccionario de datos, el plan de ejecución contiene el arbol-parse,no hay scan de hojas. Procedimientos y paquetes son compilados y almacenados en las tablas del sistema. set show plan, no exec on , statistics id, plan-table
Estructura Física	Storagegroup,database,tablespaces-indexspaces,datasets,segments(N/A),tables, index	Databases,dispositivos o archivos,segmentos,tablas o indices	database,tablespaces-file(s)(system,user-data,rollback,temp),tables,index

OBJETO	CARACTERISTICAS	CARACTERISTICAS	CARACTERISTICAS
	DB2	SYBASE	ORACLE
Respaldo y Recuperacion	Respaldo a un archivo se registra la copia en el catálogo y LOG (Tipo de puntos de recuperación: Quiesce, Full e INCR)	Dump database dbname to dump-devname (DATA + LOG), Dump Transaction (LOG),Recover(Drop, recreate, load (a nivel BD)) Load DB & Transaction	Backup en line requiere BEGIN/END Backup. Full backup requiere BD abajo. Recuperación y backup vía O/S de otro modo debe correr en modo **archive log**
Seguridad	Grant-Revoke,autoridades administrativas, autoridades secundarias	Grant-Revoke,group-privileges, autoridades administrativas	Grant-Revoke,autoridades admin,create user...QUOTA,roles
LOCK	Lock a nivel de página, lock escalation, exclusive y share locks, IRLM	Lock a nivel de página, lock escalation, exclusive y share locks, no timeout. Se pueden desplegar los locks con el comando sp_lock	Lock a nivel de renglón,no-escalation, lock on-update, consistencia manejada por segmentos de rollback

Como se pudo observar también existen diferencias entre las diferentes estructuras que puede manejar el SMBDR.

VII Utilerías

Son procesos empleados por los administradores de la base de datos para realizar las tareas de carga, descarga, ordenamiento, respaldo, recuperación, redimensionamiento, etc. Que requieren las tablas base para tener un óptimo funcionamiento.

DB2	SYBASE	ORACLE
LOAD	Bulk copy (LOAD/UNLOAD	Export/import
COPY		SQL * LOADER
REORG		SQL*DBA
RUNSTATS		
RECOVER		
DSN1LOGP		
DSN1COPY		
DSN1CHK		

METODOLOGIA DE AFINACION PARA ORACLE

Tiene como corazón el SMBDR, sin embargo ofrece además productos CASE, lenguajes de 4a generación y paquetes aplicativos.

Cuando el modelo de datos es una realidad física.

1. Hay que monitorear la base de datos
2. Coordinar los cambios

Para tener un buen desempeño.

1. Buen diseño
2. Denormalización vs Normalización lo que implica redundancia y chequeo de información vs no tenerlas presentes.
3. Tipo de operaciones que pueden alentar la ejecución de una transacción:
   - El número de tablas o entidades que se necesita unir
   - Cálculo de valores derivados en donde los valores base no cambien
   - Junta con referencia a varias entidades
   - Cualquier recuperación de datos que no involucre índice
   - Contención : cuando se trata de accesar el mismo conjunto de renglones

Conceptos

PARTITION: Conjunto de uno o mas archivos, en cualquier combinación, se le debe asignar al menos un archivo antes de poder ser utilizada.

SPACE Las tablas se relacionan con una partición mediante este concepto, el cual es sólo una plantilla de como ORACLE ubica espacio para una tabla, esta asociación se lleva a cabo en el momento de creación de la tabla. Por ello si se requiere cambiar el almacenamiento se requiere:

1) Crear otra tabla
2) Copiar los datos
3) Borrar la tabla vieja
4) Renombrar la tabla

La relación PARTITION - SPACE es una a muchos pero no uno a muchas. Un space se divide en datapages e indexpages. Mediante los parámetros INITIAL e INCREMENTAL se toman los valores a usar de la partición.

INDICES Los árboles B para índices, se encuentran en el mismo space de la tabla para obtener un mejor performance. Se puede agregar el número de índices que se desée pero un aumento en la velocidad al momento de recuperar datos implica un detrimento en las operaciones de inserción y modificación. Se recomienda asignar índices sobre los identificadores únicos, llaves primarias y en las llaves foráneas.

TABLAS Provée la facilidad de clusterización intertablas de tal manera que las tablas a utilizar al mismo tiempo se encuentren cerca físicamente, ordenadas de acuerdo a una llave, se pueden unir así hasta 32 tablas.

LOCK Se puede dar de tres tipos shared,exclusive and shared update.Ademas del CURSOR STABILITY , prohibe escrituras pero no lecturas, REPEATABLE READ , permite las lecturas y escrituras sin embargo estas últimas no tienen efecto hasta que se libera el candado.

DATABASE

INSTANCE: Conjunto de procesos y memoria compartida que hace a una base de datos accesible al usuario.

El almacenamiento físico de una base de datos consiste de uno o varios archivos, cada uno es parte de un tablespace .

TABLESPACE: Se pueden guardar tablas, índices y otros objetos en un tablespace específico.

El espacio en un tablespace se divide en segmentos, los cuales contienen información de una tabla. Existen lo que llaman segmentos rollback para la información de transacciones en proceso.

SYSTEM es el tablespace en el que se encuentran las tablas e índices del diccionario de ORACLE

Consideraciones de almacenamiento físico

A un tablespace se el asignan y se le ubican archivos, si se requiere más espacio basta con concatenarle más archivos al tablespace

Contribuyendo al Performance

Revisar diseño de tablas
Revisar implementación física
Indices
Código SQL

Cómo escoge ORACLE los caminos de acceso?

ORACLE cuenta con una rutina que permite seleccionar el mejor método de acceso tomando en cuenta la naturaleza de los datos almacenados es decir se basa en el costo en que se incurre al ejecutar el query. Es recomendable ejecutar ANALYZE periódicamente sobre las tablas más volátiles.

METODOLOGIA DE AFINACION PARA SYBASE

El proceso de afinación en SYBASE incluye manipular:

I Opciones de proceso de cada Query
II Opciones de base de datos (sp_dboption)
III Monitoreo del servidor (sp_monitor)
IV Uso de estadísticas de update, para asegurar que se hace el mejor uso de los índices
V Cambio a las variables del sistema (sp_configure y reconfigure)
VI Redistribución de los objetos en los segmentos
VII Estimar y proyectar el tamaño de los objetos en la base de datos

I Opciones de proceso de cada Query

El tipo de opciones a ser usadas por el comando SET que permite cambiar la forma de proceso de un QUERY en cuanto a:
La forma de trabajar con la aritmética
Tipo de acciones permitidas por el QUERY
Restringir el número de renglones a recuperar
Permite capturar estadísticas de la ejecución (I/o - Tiempo)

II Opciones de base de datos (sp_dboption)

Opciones para modificar la escritura que se realiza en las bitácoras, acerca de las transacciones procesadas. El modificar estas opciones afecta los procesos de respaldo-recuperación, ya que se debe guardar la congruencia entre las opciones de respaldo y las de recuperación.

III Monitoreo del servidor (sp_monitor)

Con el llamado al proceso sp_monitor se puede ver el consumo de CPU y IO que se ha dado desde la última puesta en marcha del servidor. Tambien permite conocer el tipo de requerimientos que atendió el servidor, es decir si estos fueron de lectura o escritura, así como aquellos en los que se detectó algún error.

IV Uso de estadísticas de update, para asegurar que se hace el mejor uso de los índices

El tener estadísticas feacientes de los índices permite al optimizador emplearlos de manera eficiente. Es común que al terminar la actualización de estadísticas se emplée el recompile, para recompilar todos los programas que tienen efecto sobre determinada tabla.

V Cambio a las variables del sistema (sp_configure y reconfigure)

Los valores de default se establecen para trabajar bajo cierto escenario. Por lo que de acuerdo al escenario que se este manejando será necesario modificar algunas variables como

Recovery interval
User connections
Memory
Open databases
Locks
Open Objects
Procedure Cache
Fillfactor
Time slice
Database Size
Recovery Flags
Nested Triggers
Devices
Remote access
Remote logins
Remote sites
Remote connections
pre-read packets
max online engines
min online engines
engine adjust interval
cpu flush
i/o flush
stack size
additional net mem.
default network packet size
maximum network packet size
extent io/buffers
identify burning set factor

No todas las variables se pueden modificar en línea en algunos casos será necesario desactivar el manejador y volver a ponerlo en marcha.

Los servicios que se ven afectados al modificar estas variables son:

- Tiempo de recuperación así como frecuencia de checkpoint
- Facilidad de cambiar las tablas del sistema con queries AD-HOC y no sólo con los procesos tradicionales

- Número de conexiones permitidas por cada conexión se estima un overhead de alrededor de 50K de memoria
- Memoria especificada en unidades de 2K debe ser suficiente para:
    * Requerimientos del SQL-SERVER
    * Procedure Cache y Data Cache
    * Buffers de la red
    * Buffers para extended IO
* Número de bases de datos abiertas, locks, objetos abiertos, dispositivos a ser usados por el servidor
* Tamaño de rebanada de tiempo, base de datos periódo de retención de un respaldo
* Parámetros asociados a la transferencia de información en la red

VI Redistribución de los objetos en los segmentos

Cuando se tienen los volúmenes suficientes se debe tender a distribuir la información de manera adecuada. Por ejemplo la tabla en un dispositivo, IX-nacluster en otro y el log de transacciones en un tercer volúmen.
Tablas muy grandes se debe intentar que se encuentren en 2 segmentos diferentes

Deadlocks

Las recomendaciones van encaminadas a siempre actualizar las tablas en un mismo orden. Evitar obtener un lock muy largo.

Reducir la contención del LOCK

* Rediseñar las tablas para evitar accesos secuenciales o accesos costosos
* Asignar mas espacio libre en las páginas de los indices para incrementar la posibilidad de encontrar espacio

BIBLIOGRAFIA

1) Ullman, Jeffrey D.,
Principles of Database Systems
Computer Science Press, 1980

2) Date, C J.
An introduction to Database Systems,
Vol I,
Addison Wesley Systems Programming Series, 1991.

3) Vaskevitch, David.
Database in Crisis and Transition: A Technical agenda for the year 2001
Proceedings of the 1994 ACM Sigmod,
June 1994.

4) Melling, Wesley P.,
Enterprise Information Architectures --- They are finally changing;
Proceedings of the 1994 ACM Sigmod,
June 1994.

5) Martin, James,
Information Engineering: A trilogy by James Martin;
James Martin ADVANCE TECHNOLOGY LIBRARY

6) Gallaire, H., Minker, J. & Nicolas J M.,
Logic and Databases: a deductive approach
ACM, Computing Surveys, 1984.

7) Kint,Won,
Introduction to Object Oriented Databases;
MIT Press;1992

8) Setrag, Khoshafian.
Object Oriented Databases,
John Wiley & Sons, 1993

9) Stonebraker, Michael,
Multidatabase Systems: An advance Solution for Global Information Sharing
IEEE Computer Society Press, 1994

10) Raj, Jain;
The ART of computer systems performance analysis,
John Wiley & Sons Inc.; 1991

11) SYBASE 4.9.1,
System Administration Guide for SYBASE SQL Server;
October 15,1992.

12) SYBASE 10.0,
System Administration Guide for SYBASE SQL Server;
September 14,1993

13) McGoveran, D. and Date, C J.,
A guide to SYBASE and SQL Server,
Addison Wesley Publishing Company, Inc.; 1993.

14) Ault, Michael R ,
ORACLE 7.0 Administration & Management,
John Wiley & Sons, 1994

15) Corrigan, Peter and Gurry, Mark,
ORACLE Performance Tuning,
O'Reilly & Associates, Inc , 1994

16) Date, C J and White, Colin J .
A guide to DB2;
Addison Wesley Publishing Company, Inc , 1993

17) Werman, Aaron;
DB2 handbook for DBAs,
Mc Graw Hill; 1991.

18) Johnson, Robert H ;
MVS Concepts and Facilities;
Mc Graw Hill; 1989

19) Wiederhold, Gio;
File Organization for Database Design,
Mc Graw Hill, 1988.

20) Silberschatz,Avi, Stonebraker,Michael and Ullman,Jeff.
Database Systems: Achievements and Oportunities.
Communications of the ACM, October 1991.