



UNIVERSIDAD AUTÓNOMA DE
QUERÉTARO

FACULTAD DE INFORMÁTICA

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA



DATA WAREHOUSE EN ORACLE 8I
TESINA

Que para obtener el título de
Licenciado en Informática

Presenta

Yolanda Alvarado Aguilar

Asesor

I.S.C Jabel Resendíz González

Santiago de Querétaro, Qro., Abril del 2004



TS
005.75
A472d

F06908

TS
005.75
A472d

F06908



UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA

No. Adq. F06908
Clasif. TS 005.75
Cutter A472d

UNIVERSIDAD AUTÓNOMA DE QUERÉTARO
BIBLIOTECA
FACULTAD DE INFORMÁTICA



CARTA DE ACEPTACIÓN

Por este medio, se otorga constancia de aceptación de tesina para obtener el título de Licenciado en Informática, que presenta la pasante **YOLANDA ALVARADO AGUILAR**, con el tema denominado "DATA WAREHOUSE EN ORACLE 8i".

Este trabajo fué desarrollado como una investigación derivada del curso de titulación "ADMINISTRACIÓN ORACLE 8i", dando cumplimiento a uno de los requisitos contemplados en el artículo 34 del reglamento de titulación vigente, en lo referente a la opción de titulación por realización y aprobación de cursos de actualización.

Se extiende la presente para los fines legales a que haya lugar y para su inclusión en todos los ejemplares impresos de la tesina, a los doce días del mes de enero del 2005.

ATENTAMENTE

I.S.C. JABEL RESÉNDIZ GONZÁLEZ
INSTRUCTOR DEL CURSO

Quiero agradecer primeramente a Dios
por darme esta vida y estar siempre en ella.

A mis padres por todo su apoyo incondicional
por educarme con responsabilidad
y amor.

A mi hermana Edith por estar siempre conmigo
a mi cuñado y sobrino por sus sonrisas y cariño.

A mi Universidad Autónoma de Querétaro
por su educación incanzable
la cual me ha dado en mi vida.

A mis maestros por dar lo mejor para transmitir
sus conocimientos y ayudarme a ser mejor
persona.

A toda mi familia por quererme y apoyarme.

A mi novio y mis amigos por su cariño y
su gran amistad.

RESUMEN

El Data warehouse es considerada la tendencia más grande dentro de la administración de información, se describe como una tecnología de almacenamiento, una base de datos relacionada, diseñada para búsquedas y análisis en lugar de procesamientos de transacciones, teniendo como ventaja el apoyo a la toma de decisiones. Un Data warehouse usualmente contiene datos históricos que son obtenidos de una transacción de datos, pero que pueden incluir datos de otras fuentes. Esto separa el trabajo de análisis del trabajo de la transacción y permite a un negocio consolidar los datos de varias fuentes.

El Data warehouse cuenta con un ciclo de desarrollo al igual que un software, compuesto por varias etapas, otro punto importante dentro del desarrollo del Data warehouse es considerar las diversas estrategias que hay que tomar en cuenta para cada organización, ya que la tecnología Data warehouse basa sus conceptos y diferencias entre dos tipos fundamentales de sistemas de información en todas las organizaciones: los Sistemas Técnico - Operacionales y los Sistemas de Soporte de Decisiones. La arquitectura Data warehouse consiste de una alimentación de datos operacionales y datos externos, pasando a la etapa de almacén donde se encuentra un resumen de datos, el metadata y los datos en crudo para después seguir a la etapa de generación de reportes, de los cuales hace uso la empresa o los usuarios destinados al manejo del Data warehouse.

Data warehouse dentro de Oracle 8i cuenta con una amplia variedad de componentes físicos y lógicos al ser desarrollados dentro de este software de manipulación y administración de las bases de datos. En el diseño lógico se encuentran los esquemas, los objetos, Fact-tables, dimensiones, etc., y dentro del diseño físico se encuentra el paralelismo y particionismo, índices, constraints, vistas materializadas, dimensiones, así como un manejo en el ambiente de almacén que se considera la extracción, transportación, transformación, etc.

El Data Mart, es considerado un Data warehouse que está diseñado para una línea particular del negocio, tales como ventas, mercadotecnia o finanzas. En un data mart independiente, los datos pueden ser derivados a un Data warehouse empresarial y en un data mart independiente, los datos pueden ser derivados directamente de las fuentes.

Al ser comparada el Data warehouse con versiones anteriores muestra la evolución de Oracle 8i y los beneficios al desarrollar e implementar esta técnica de almacenamiento de datos dentro de una organización, así como las ventajas de Oracle 8i trabajando conjuntamente con Data warehouse para obtener una mayor rapidez y simplicidad para los usuarios y los datos.

ÍNDICE

<i>Resumen</i>	3
<i>Índice</i>	4
<i>Introducción</i>	8
Parte I. DATA WAREHOUSE.	
1. Definición de Data warehouse.	9
2. Sistemas de información.	10
2.1 Sistema estratégico.	10
2.2 Sistema táctico.	10
2.3 Sistema técnico- operativo.	11
2.4 Sistema interinstitucional.	11
3. Características de un Data warehouse.	11
3.1 Orientado a temas.	11
3.2 Integración.	12
3.2.1 Codificación.	12
3.2.2 Medida de atributos.	12
3.2.3 Fuentes múltiples.	12
3.3 De tiempo variante.	13
4. Ciclo de desarrollo del Data warehouse.	15
4.1 Planeación.	15
4.2 Requerimientos.	16
4.3 Análisis.	16
4.4 Diseño.	18
4.5 Construcción.	18
4.6 Despliegue.	19
4.7 Expansión.	20
5. Estructura del Data warehouse.	20
6. Arquitectura típica del Data warehouse.	22
6.1 Elementos constituyentes de una arquitectura Data warehouse.	23
6.1.1 Base de datos operacional / Nivel de base de datos externo.	23
6.1.2 Nivel de acceso a la información.	23
6.1.3 Nivel de acceso a los datos.	24
6.1.4 Nivel de directorio de datos (metadata).	24
6.1.5 Nivel de gestión de procesos.	25
6.1.6 Nivel de mensaje de la aplicación.	25
6.1.7 Nivel Data warehouse (físico).	25
6.1.8 Nivel de organización de datos.	25
6.2 Operaciones en un Data warehouse.	25
6.2.1 Sistemas operacionales.	25
6.2.2 Extracción, transformación y carga de los datos.	26
6.2.3 Metadata.	26
6.2.4 Acceso de usuario final.	26
6.2.5 Plataforma del Data warehouse.	26
6.2.6 Datos externos.	26
6.3 Evolución del depósito.	27
7. Transformación de datos y metadata.	27
7.1 Transformación de datos.	27
7.2 Metadata.	27

8. Flujo de datos.	28
9. Usos del Data warehouse.	29
10. Consideraciones adicionales y excepciones en el Data warehouse.	30
11. Organización de un proyecto.	31
11.1 Factores en la planificación de un Data warehouse.	31
11.2 Estrategias para el desarrollo de un Data warehouse.	32
11.3 Estrategias para el diseño de un Data warehouse.	33
11.4 Estrategias para el gestión de un Data warehouse.	33
12. Desarrollo de un proyecto.	34
12.1 Consideraciones previas al desarrollo de un Data warehouse.	34
12.2 Elementos claves para el desarrollo de un Data warehouse.	35
12.3 Confiabilidad de los datos.	38
 Parte II. DATA WAREHOUSE EN ORACLE 8i	
13. Introducción de Data warehouse en Oracle 8i.	39
13.1 Esquemas de Data warehousing.	39
13.2 Objetos de Data warehousing.	40
13.3 Fact- tables.	40
13.4 Dimensiones.	40
13.5 Jerarquías.	41
14. Diseño físico.	42
14.1 Estructura del diseño.	42
14.1.1 Tablespace.	42
14.1.2 Particiones.	43
14.1.3 Índices.	43
14.1.4 Constraints.	43
15. Hardware y Entrada/Salida	43
15.1 Striping data.	43
15.2 Consideraciones de E/S.	44
16. Paralelismo y particionismo.	44
16.1 Tipos de paralelismo.	44
16.1.1 Block range granules.	45
16.1.2 Partition granules.	45
16.2 Tipos de particiones.	46
16.2.1 Range.	46
16.2.2 Hash.	46
16.2.3 Composite.	46
16.3 Particionar índices.	47
16.4 Partition-wise join.	47
17. Índices.	48
17.1 Bitmap indexes.	48
17.2 Índices b-tree.	49
18. Constraints.	50
18.1 Utilidad de los constraints en Data warehouse.	50
18.2 Típicos constraints del Data warehouse.	50
18.3 Constraints y paralelismo.	51
18.4 Constraints y particiones.	51

19. Vistas materializadas.	51
19.1 Vistas materializadas para Data warehouse.	51
19.2 La necesidad de materializar vistas.	51
19.3 Tipos de vistas materializadas.	52
19.4 Creación de una vista materializada.	53
19.5 Jerarquizar vistas materializadas.	53
19.6 Registrar una vista materializada existente.	54
19.7 Particionar una vista materializada.	54
19.8 Selección de indexación para vistas materializadas.	54
19.9 Invalidar una vista materializada.	55
19.10 Pautas para usar vistas materializadas en un Data warehouse.	55
19.11 Alterar una vista materializada.	55
19.12 Borrar una vista materializada.	56
20. Dimensiones.	56
20.1 Definición de dimensión.	56
20.2 Crear una dimensión.	57
20.3 Dimension wizard.	57
20.4 Vistas de dimensiones.	58
20.5 Dimensiones y constraints.	58
20.6 Validar una dimensión.	58
20.7 Alterar una dimensión.	59
20.8 Borrar una dimensión.	59
21. Tablespace transportables.	59
22. Rollup.	60
23. Cube.	60

Parte III. DATA WAREHOUSE PARA MÁS DATOS Y MÁS USUARIOS

24. Rápido y simple para mayores datos y más usuarios.	61
25. Oracle 8i más rápido para los usuarios.	62
25.1 Índices y métodos join.	63
25.2 Bitmap indexes.	63
25.3 Bitmap indexes dinámicos.	63
25.4 Bitmap star join.	63
25.5 Function based.	63
25.6 Fast full index scan.	65
25.7 Index join.	65
25.8 Hash join.	65
25.9 Partition-wise join.	65
25.10 Vistas materializadas y manejo de resúmenes.	66
25.11 Oracle 8i manejo de resúmenes.	67
25.12 Reescritura de resúmenes.	67
25.13 Particionando.	69
25.14 Paralelismo.	69
25.15 Capacidades analíticas.	72
25.16 Cube y rollup.	72
25.17 Incremento para búsquedas en base a N.	73
25.18 Sampling.	73
25.19 Optimización de la búsqueda.	73

26. Oracle simple para más usuarios.	74
26.1 Asesor de resúmenes.	75
26.2 Incremento a los índices wizard.	75
26.3 Progreso del monitor.	76
26.4 Manejabilidad en el desarrollo de búsquedas paralelas.	76
26.5 Manejo de fuentes de la base de datos.	76
26.6 Fine-grained access control.	77
27. Oracle 8i simple para más datos.	78
27.1 Particionamiento.	79
27.2 Soporte del esquema de cargado del rolling-window.	79
27.3 Nuevas técnicas de particionamiento.	81
27.4 Nuevas características adicionales en Oracle 8i más fácil para más datos.	82
27.5 Manejo de recuperación y respaldos en el servidor.	82
27.6 Paquete nuevo para el manejo de estadísticas.	83
27.7 Generación automática del optimizador de estadística.	83
28. Oracle 8i rapidez para más datos.	84
28.1 Paralelización de operaciones administrativas.	84
28.2 Constraints mejorados.	85
28.3 Resúmenes actuales.	85
28.4 Direct-path load API.	86
28.5 Transportable.	86
Parte IV. DATA MARTS	
29. ¿Qué es un data mart?	87
30. ¿En que difiere de un Data warehouse?	88
31. Data marts híbridos, independientes y dependientes.	88
32. Extracción, transformación y transportación (ETT).	90
33. OLAP (Procesamiento analítico en línea).	91
34. Oracle Discovered.	91
Conclusión	92
Bibliografía	93
Glosario	94

INTRODUCCIÓN

Las organizaciones siempre han usado los datos desde sus sistemas operacionales para atender sus necesidades de información. Los métodos han evolucionado a través del tiempo y ahora las organizaciones manejan datos sobre los cuales se toman decisiones importantes.

El **Data warehouse (DW)**¹ se considera la tendencia más grande dentro de la administración de información, la tecnología que finalmente puede llevar al sueño que perseguían los teóricos del tema desde la década de los sesenta, ya que representa un proceso de reunir información histórica de una organización en un depósito central, y se ha convertido en una tecnología común y fundamental.

Hoy en día la tecnología del DW o "Almacenamiento de datos" es considerada un ingrediente esencial en el conjunto de soluciones para el soporte de decisiones en una empresa, ya que logra reunir la información trascendente de ésta y puede convertirse en el punto de éxito de la empresa.

La importancia del DW es que no solamente se limita a la información interna sino que también considera todos aquellos factores externos que llegan a intervenir en la empresa, su objetivo es facilitar la toma de decisiones de una empresa y su eficacia a través de la recopilación de información y elementos que por sí solos no significan nada, son extraídos para eliminar inconsistencias y unidos bajo ciertas estructuras e integración se convierten en información esencial para la empresa.

El DW, es actualmente, el centro de atención de las grandes instituciones, porque provee un ambiente para que las organizaciones hagan un mejor uso de la información que esta siendo administrada por diversas aplicaciones operacionales.

La innovación de la Tecnología de información dentro de un ambiente DW, puede permitir a cualquier organización hacer un uso más óptimo de los datos, como un ingrediente clave para un proceso de toma de decisiones efectivo, al evolucionar estas técnicas de manejo de información, también ha ido evolucionando la información y desarrollo acerca de estas técnicas así como el software que permite una mayor facilidad en el manejo de base de datos, orientándonos en este caso a Oracle 8i.

Parte I. DATA WAREHOUSE

DEFINICIÓN DE DW

W.H. Inmon, quien es considerado el padre del DW, lo define: "Un DW es un conjunto de datos integrados orientados a una materia, que varían con el tiempo y que no son transitorios, los cuales soportan el proceso de toma de decisiones de una administración".

De acuerdo con algunas organizaciones, el DW es una arquitectura o depósito de datos, una colección de datos orientado a temas, integrado, no volátil y de tiempo variante, que se usa para el soporte del proceso de toma de decisiones gerenciales.

DW es el centro de la arquitectura para los sistemas de información en la década de los 90's, soporta el procesamiento informático al proveer una plataforma sólida, a partir de los datos históricos para hacer el análisis y facilitar la integración de sistemas de aplicación no integrados.

El DW es siempre un almacén de datos transformados y separados físicamente de la aplicación donde se encontraron los datos en el ambiente operacional.

El DW está orientado a una materia ya que organiza y orienta los datos desde la perspectiva del último usuario, administra grandes cantidades de información ya que contiene información histórica que se retira con frecuencia de los sistemas operativos porque ya no es necesaria para las aplicaciones operacionales y de producción, maneja información a diferentes niveles de granularidad, es mucho mayor que las Bases de Datos (BD)² Operacionales, guarda información en diversos medios de almacenamiento, comprende múltiples versiones de un esquema de BD, condensa y agrega información, integra y asocia información de muchas fuentes de información, DW recopila y organiza en un solo lugar la información que las aplicaciones han acumulado al paso de los años.

Base de Datos Operacional	Data Warehouse
Datos Operacionales	Datos del negocio para Información
Orientado a la aplicación	Orientado al sujeto
Actual	Actual + histórico
Detallada	Detallada + más resumida
Cambia continuamente	Estable

Tabla 1.1 Diferencia BD Operacional – DW.

SISTEMAS DE INFORMACIÓN

Los sistemas de información como muestra la figura 1.1 se han dividido de acuerdo a niveles.

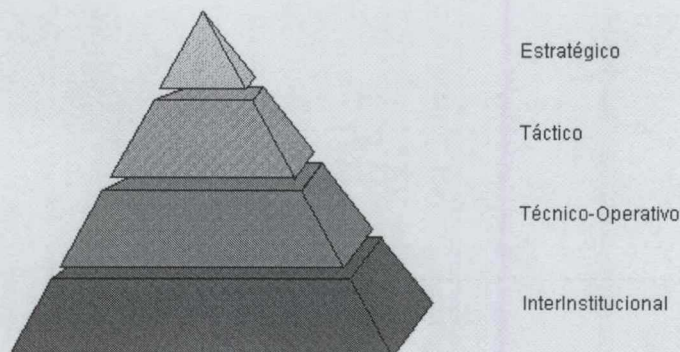


Fig. 1.1 Sistemas de Información.

- **Sistemas Estratégicos**, orientados a soportar la toma de decisiones, facilitan la labor de la dirección proporcionándole un soporte básico, son sistemas sin carga periódica de trabajo.
- **Sistemas Tácticos**, diseñados para soportar las actividades de coordinación de actividades y manejo de documentación, facilitan la consulta sobre información almacenada en el sistema y proporcionan informes.

- **Sistemas Técnico - Operativos**, cubren el núcleo de operaciones tradicionales de captura masiva de datos y servicios básicos de tratamiento de datos. Estos sistemas están evolucionando con la irrupción de sensores, autómatas, sistemas multimedia, BD Relacionales más avanzadas y *DW*.
- **Sistemas Interinstitucionales**, es consecuencia del desarrollo organizacional orientado a un mercado de carácter global, obliga a pensar e implementar estructuras de comunicación más estrechas entre la organización y el mercado.

La tecnología DW basa sus conceptos y diferencias entre dos tipos fundamentales de sistemas de información en todas las organizaciones: los Sistemas Técnico - Operacionales y los Sistemas de Soporte de Decisiones (SSD)³. SSD es la base de un DW.

CARACTERÍSTICAS DE UN DW

ORIENTADO A TEMAS

Una primera característica del DW es que la información se clasifica en base a los aspectos que son de interés para la empresa. Los datos tomados están en contraste con los clásicos procesos orientados a las aplicaciones.

El ambiente operacional se diseña alrededor de las aplicaciones y funciones mientras que el ambiente DW organiza toda la información alrededor de sujetos. (Ejemplo: cliente, vendedor, producto y actividad). La alineación alrededor de las áreas de los temas afecta el diseño y la implementación de los datos encontrados en el DW.

Todas las aplicaciones están relacionadas con el diseño de la BD y del proceso. El DW se enfoca al modelo de datos y el diseño de la BD.

El DW excluye la información que no será usada por el proceso de SSD, mientras que la información de las orientadas a las aplicaciones, contiene datos para satisfacer de inmediato los requerimientos funcionales y de proceso, que pueden ser usados o no por el analista de soporte de decisiones.

INTEGRACIÓN

El aspecto más importante del ambiente DW es que la información encontrada al interior está siempre integrada. La integración de datos se muestra de muchas maneras: en convenciones de nombres consistentes, en la medida uniforme de variables, en la codificación de estructuras consistentes, en atributos físicos de los datos consistentes, fuentes múltiples y otros.

Codificación. Los diseñadores de aplicaciones codifican la información, por ejemplo, el campo "Género" se puede codificar en varias formas. Un diseñador representa "Género" con una "M" y una "F", otros con un "1" y un "0", otros con una "X" y una "Y" e inclusive, como "Masculino" y "Femenino". No importa cómo el "Género" llega al DW, lo importante es que de cualquier fuente de donde venga el "Género" debe llegar al DW en un estado integrado uniforme.

Medida de Atributos. Los diseñadores de aplicaciones miden las unidades en una variedad de formas. Al dar medidas a los atributos, la transformación traduce las diversas unidades de medida usadas en las diferentes BD para transformarlas en una medida estándar común. Cuando la información llegue al DW necesitará ser medida de la misma manera.

Fuentes Múltiples. El mismo elemento puede derivarse desde fuentes múltiples. El proceso de transformación debe asegurar que la fuente apropiada sea usada, documentada y movida al depósito. La información necesita ser almacenada en el DW, en un modelo globalmente aceptable y singular, aun cuando los sistemas operacionales subyacentes almacenen los datos de manera diferente. Cuando el analista de SSD observe el DW, su enfoque deberá estar en el uso de los datos que se encuentre en el depósito, antes que preguntarse sobre la confiabilidad o consistencia de los datos.

DE TIEMPO VARIANTE

Toda la información del DW es requerida en algún momento. En el ambiente operacional, cuando accede a una unidad de información, usted espera que los valores requeridos se obtengan a partir del momento de acceso. En el DW es solicitada en cualquier momento, los datos encontrados en el depósito se llaman de "tiempo variante".

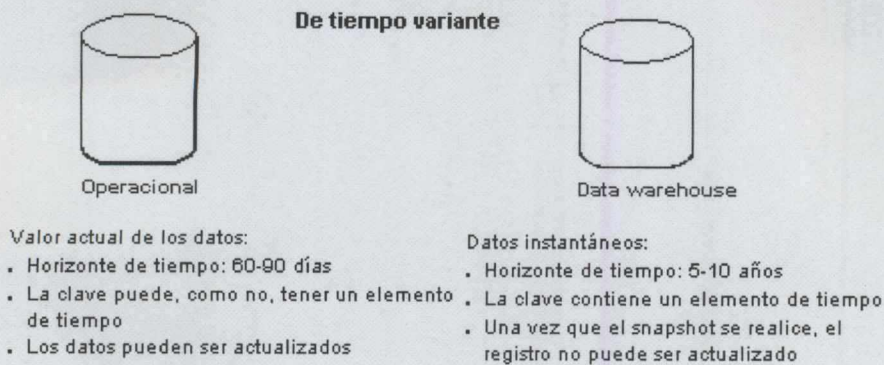


Fig. 2.3.1 Tiempo variante.

El tiempo variante se muestra de varias maneras:

- La información representa los datos sobre un horizonte largo de tiempo - desde cinco a diez años. El horizonte de tiempo representado para el ambiente operacional es mucho más corto.
- En el DW está la estructura clave. Cada estructura clave en el DW contiene, implícita o explícitamente, un elemento de tiempo como día, semana, mes, etc. El elemento de tiempo está casi siempre al pie de la clave concatenada.
- Cuando la información del DW, es registrada correctamente, no puede ser actualizada. La información del DW es, para todos los propósitos prácticos, una serie larga de "Snapshots" .

La información es útil sólo cuando es estable. Los datos operacionales cambian sobre una base momento a momento. Para el análisis y la toma de decisiones, requiere una BD estable. La manipulación básica de los datos que ocurre en el DW es mucho más simple.

Hay dos únicos tipos de operaciones: la carga inicial de datos y el acceso a los mismos. No hay actualización de datos (en el sentido general de actualización).

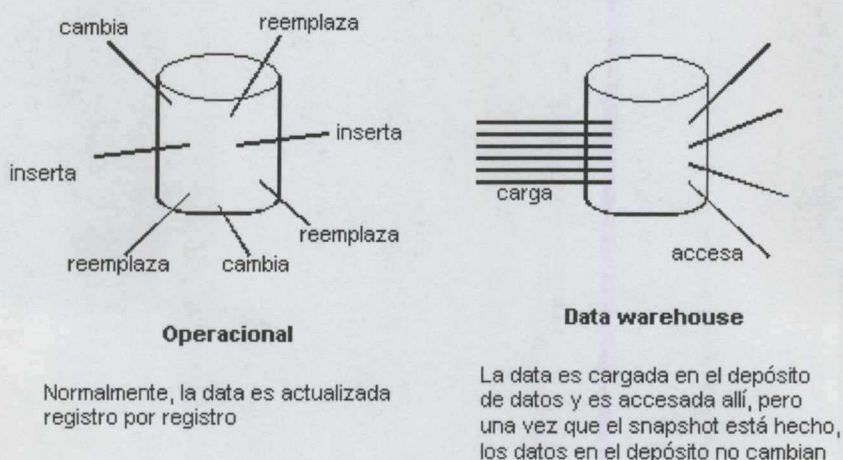


Fig. 2.3.2 BD Operacional – DW.

La tecnología permite realizar copias de seguridad y recuperación, transacciones e integridad de los datos, la detección y solución al estancamiento que es más complejo. En el DW no es necesario el procesamiento. La fuente de casi toda la información del DW es el ambiente operacional.

Los datos se filtran cuando pasan desde el ambiente operacional al de depósito, sólo los datos que realmente se necesitan ingresarán al ambiente de DW. La información en el ambiente operacional es más reciente con respecto a la del DW. El DW contiene un resumen de la información que no se encuentra en el ambiente operacional.

Los datos experimentan una transformación fundamental cuando pasa al DW ya que la mayoría de los datos se alteran física y radicalmente cuando se mueven al depósito.

CICLO DE DESARROLLO DEL DW

El DW sigue el mismo ciclo de perfeccionamiento que todos los desarrollos de software.

- Planeación.
- Requerimientos.
- Análisis.
- Diseño.
- Construcción.
- Despliegue.
- Expansión.

PLANEACIÓN

Es una fase importante de la implementación del DW. Las decisiones tomadas durante la fase de planeación tienen mucho impacto significativo en el ámbito de implementación y en la magnitud de esfuerzo. Decidir la estrategia general de implementación tiene mucho que ver con la cultura de la organización y se basa en cómo se llevan a cabo las tareas dentro de la organización.

Estos pasos son indispensables en la planeación para el sistema de DW:

- Selección de la estrategia de implementación.
 - Enfoque de arriba hacia abajo.
 - Enfoque de abajo hacia arriba.
 - Una combinación de los anteriores.
- Selección de la metodología de desarrollo.
 - Método de análisis y diseño estructurado (en cascada).
 - Método de desarrollo espiral, se presta principalmente al desarrollo de aplicaciones de BD al desarrollo de un DW y al desarrollo de sistemas Orientados a Objetos.

- Selección del ámbito de implementación.
 - Ámbito determinado a partir de la perspectiva del usuario empresarial del DW.
 - Determinación del ámbito con base en consideraciones tecnológicas.
- Selección del enfoque arquitectónico.
 - El almacenamiento operacional en comparación con el uso de copias de datos operacionales.
 - Sólo DW.
 - Sólo mercados de datos.
 - DW y mercados de datos.
 - Separación de plataforma e infraestructura.
 - Arquitectura cliente/servidor de dos hileras.
 - Arquitectura cliente/servidor de tres hileras.
- Desarrollo de un programa y del presupuesto del proyecto.
 - Articular tanto un plan de programa como un conjunto de planes de proyecto.
 - Reservar un presupuesto adecuado para el programa al tiempo que se compromete el gasto para proyectos específicos.
 - Proporcionar medidas para la estimación de la retribución del DW.
- Desarrollo de escenarios de uso empresarial.
 - Un usuario empresarial claramente identificado, como un gerente de producción.
 - Un área funcional que respalde el DW o el mercado de datos.
 - Una o más consultas empresariales de interés crucial para el área funcional.
- Recopilación de metadatos .

REQUERIMIENTOS

Esta fase es una especificación precisa de las funciones que se obtendrán del DW. Los requerimientos describirán con claridad el ambiente operativo en el que se entregará el DW, es una transición de la visión del propietario, del marco de referencia a la visión de la arquitectura. Los requerimientos son conducidos por el negocio y por la tecnología. La cuidadosa selección y especificación de requerimientos en esta etapa proporciona un proyecto cimentado que arroja resultados con rapidez.

Los requerimientos para la solución del DW desde varias perspectivas de la gente involucrada en su planeación e implementación, están compuestos por:

- Definir los requerimientos del propietario.
- Definir los requerimientos del arquitecto.
- Definir los requerimientos del desarrollador.
- Definir los requerimientos de los usuarios finales.

ANÁLISIS

Significa convertir los requerimientos mencionados en la fase anterior, en un conjunto de especificaciones que puedan apoyar el diseño. Esta fase es importante, ya que determina la forma en que se cubrirán los requerimientos, estas especificaciones se usan para generar extractores del DW y software de transformación, integración, resumen y adición. Hay tres especificaciones principales de entrada para el DW, son:

- Requerimientos de enfoque empresarial que delimitan las fronteras de la información que debe comprender el DW.
- Especificaciones de requerimientos de fuentes de datos que delimitan las fronteras de información disponible en las fuentes de datos actuales.
- Especificaciones de requerimientos de usuario final y acceso, utilizará la información del DW.

DISEÑO

Los modelos lógicos desarrollados en la fase de análisis se convierten en modelos físicos. Los procesos identificados en la fase de análisis para conectar las fuentes de datos con el DW, el DW con los mercados de datos y el DW/mercado de datos con las herramientas de estación de trabajo del usuario final, se convierten en diseños para programas que realizarán las tareas requeridas por los procesos.

En la fase de diseño se encuentran las siguientes dos actividades principales:

- Diseño detallado de la arquitectura de datos.
- Diseño detallado de la arquitectura de aplicación.

La arquitectura de datos comprenden las siguientes actividades:

- Desarrollo de modelos físicos de datos para la BD de almacenamiento del DW y mercado de datos.
- Correspondencia de los modelos físicos de datos de las fuentes de datos con los modelos físicos del DW/ mercados de datos.

CONSTRUCCIÓN

Es responsable de implementar físicamente los diseños desarrollados durante la fase de diseño. Es similar a la construcción de un sistema de BD, se necesita construir las siguientes aplicaciones:

- Programas que creen y modifiquen las BD para el DW y los mercados de datos.
- Programas que extraigan datos de fuentes relacionales y no relacionales.
- Programas que realicen transformaciones de datos, integración, resumen y adición.
- Programas que realicen actualizaciones de BD Relacionales.
- Programas que efectúen búsquedas en BD muy grandes.

Los principales retos durante la fase de construcción son los siguientes:

- Entender cómo incorporar las inversiones existentes en plataformas, tecnología y habilidades.
- La toma de decisiones inteligentes.
- La selección y evaluación adecuada de componentes suministrados por el fabricante.
- La capacidad de integración de sistemas.
- La administración de metadatos.

La fase de la construcción resalta los diversos intercambios “construir en comparación con comprar”

DESPLIEGUE

Tiene que ver principalmente con los retos de instalación, puesta en servicio y uso de la solución de DW. El sistema de DW tiene los siguientes requerimientos:

- La mayoría de los usuarios finales del DW no son técnicos. Necesitan ver definiciones de la información contenida en el DW en términos y lenguajes que comprendan.
- El DW se percibe principalmente como una capacidad de apoyo a las decisiones, no una capacidad de misión crítica. Existe una necesidad de promover activamente entre los usuarios finales la información que proporciona el DW.

El despliegue del DW requiere las siguientes capacidades adicionales, que normalmente se asocian con los sistemas de información empresarial:

- Comercialización de la información.
- Directorio/catálogo de información.
- Examinadores (browsers) de información.

La fase de despliegue en el ciclo, tiene un componente único denominado "Comercialización de Información", esto reconoce que la mercancía que suministra el DW a sus usuarios finales es la propia información.

EXPANSIÓN

El método de desarrollo espiral se describió como un ingrediente clave para el despliegue rápido inicial de un DW, este método requiere también de una rápida evolución de características y funciones basadas en las lecciones aprendidas en anteriores despliegues. Para explotar las ventajas que ofrece el modelo espiral, es importante comenzar a reunir requerimientos para la siguiente iteración del ciclo de desarrollo.

ESTRUCTURA DEL DW

Hay niveles diferentes de esquematización y detalle que delimitan el DW. La estructura de un DW se muestra en la siguiente figura:

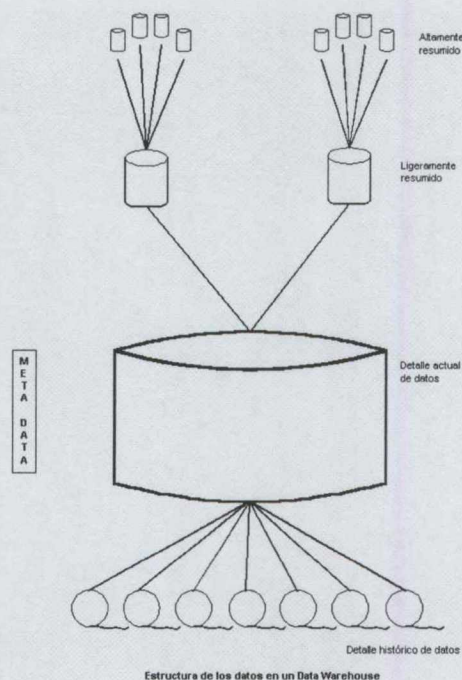


Fig. 5.1 Estructura de los datos en un DW.

Detalle De Datos Actuales

- Refleja las ocurrencias más recientes, las cuales son de gran interés.
- Es voluminoso, ya que se almacena al más bajo nivel de granularidad.
- Casi siempre se almacena en disco, el cual es de fácil acceso.

Detalle De Datos Antiguos. La data antigua es aquella que se almacena sobre alguna forma de almacenamiento masivo. No es frecuentemente su acceso y se almacena a un nivel de detalle, consistente con los datos detallados actuales.

Datos Ligeramente Resumidos. La data ligeramente resumida es aquella que proviene desde un bajo nivel de detalle encontrado al nivel de detalle actual.

Metadata. El componente final del DW es el de la metadata. La metadata juega un rol especial y muy importante en el DW y es usada como:

- Un directorio para ayudar al analista a ubicar los contenidos del DW.
- Una guía para la trazabilidad de los datos, de cómo se transforma, del ambiente operacional al de DW.
- Una guía de los algoritmos usados para la esquematización entre el detalle de datos actual, con los datos ligeramente resumidos y éstos, con los datos completamente resumidos, etc.

La metadata contiene:

- La estructura de los datos.
- Los algoritmos usados para la esquematización.
- La trazabilidad desde el ambiente operacional al DW.

La información adicional que no se esquematiza es almacenada en el DW, allí se hará el análisis y se producirá un tipo de resumen. El único tipo de esquematización que se almacena permanentemente en el DW, es el de los datos que son usados frecuentemente.

ARQUITECTURA TÍPICA DEL DW

Una de las razones por las que el desarrollo de un DW crece rápidamente, es que realmente es una tecnología muy entendible. A fin de comprender cómo se relacionan todos los componentes involucrados en una estrategia DW, es esencial tener una Arquitectura DW.

El DW y su arquitectura, puede variar dependiendo de las especificaciones de cada organización. La figura 6.1 muestra la arquitectura más básica para un DW. En este, un DW es alimentado de uno o más sistemas fuente, y el usuario final accesa directamente al DW.

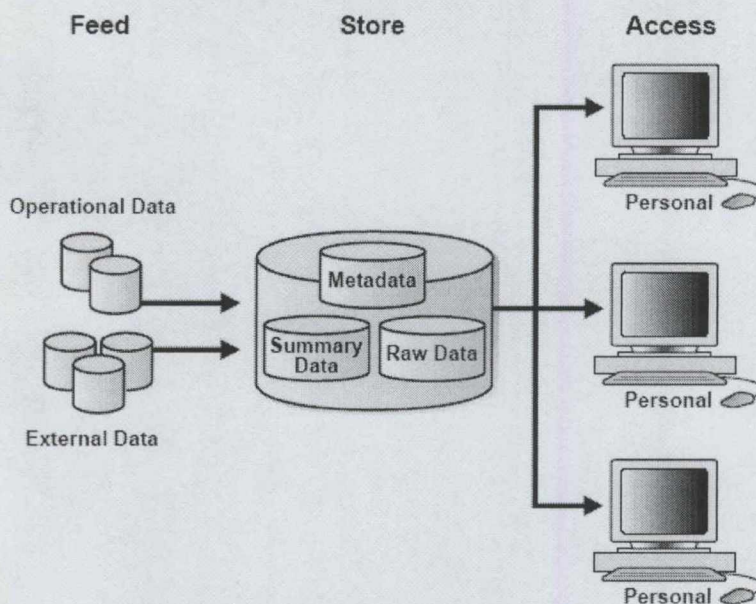


Fig. 6.1 Arquitectura Típica del DW.

La figura 6.2 ilustra un ambiente DW más complejo. En suma a una BD central, hay un sistema estacionado usado para limpiar e integrar datos, tan bueno como un data-mart múltiple, el cual es un sistema diseñado para una línea particular de trabajo.

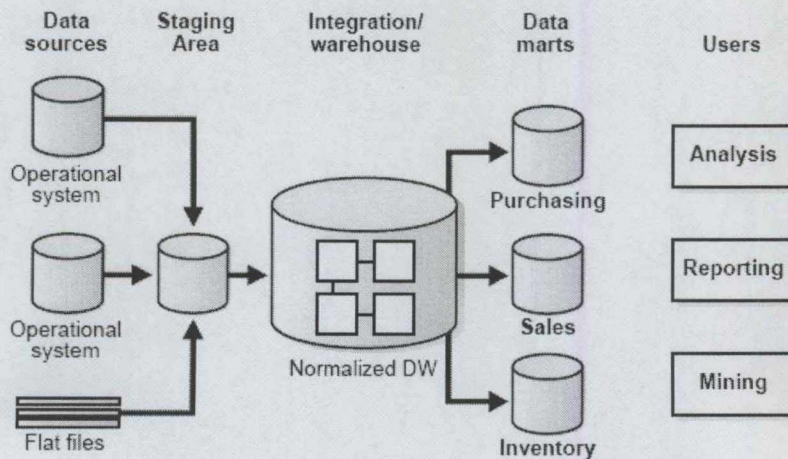


Fig. 6.2 Arquitectura típica para un complejo DW.

Una Arquitectura DW (DWA⁷) es una forma de representar la estructura total de datos, comunicación, procesamiento y presentación, que existe para los usuarios finales que disponen de una computadora dentro de la empresa.

ELEMENTOS CONSTITUYENTES DE UNA ARQUITECTURA DATA WAREHOUSE

BD Operacional / Nivel de BD Externo

Los sistemas operacionales procesan datos para apoyar las necesidades operacionales críticas. Las BD Operacionales históricas proveen una estructura de procesamiento eficiente, para un número relativamente pequeño de transacciones comerciales bien definidas. Las BD Diseñadas para soportar estos sistemas, tienen dificultad al acceder a los datos para otra gestión o propósitos informáticos. La meta del DW es liberar la información que es almacenada en BD Operacionales y combinarla con la información desde otra fuente de datos, generalmente externa.

Nivel de acceso a la Información

Es el nivel del que el usuario final se encarga directamente. Representa las herramientas que el usuario final normalmente usa día a día. Este nivel también incluye el hardware y software involucrados en mostrar información en pantalla y emitir reportes de impresión, hojas de cálculo, gráficos y diagramas para el análisis y presentación.

Actualmente, existen herramientas más sofisticadas para manipular, analizar y presentar los datos, sin embargo, hay problemas significativos al tratar de convertir los datos tal como han sido recolectados y que se encuentran contenidos en los sistemas operacionales, una de las claves para este problema es encontrar un lenguaje de datos común que puede usarse a través de toda la empresa.

Nivel de acceso a los Datos

El nivel de acceso a los datos de DWA está involucrado con el nivel de acceso a la información para conversar en el nivel operacional. El lenguaje de datos común que ha surgido es SQL.

Una de las claves de una estrategia DW es proveer a los usuarios finales con "Acceso a Datos Universales". Los usuarios finales sin tener en cuenta la herramienta de acceso a la información o ubicación, deberían ser capaces de acceder a cualquier o todos los datos en la empresa que es necesaria para ellos, para hacer su trabajo. El nivel de acceso a los datos entonces es responsable de las interfaces entre las herramientas de acceso a la información y las BD Operacionales.

Nivel de directorio de datos (Metadata)

A fin de proveer el acceso a los datos universales, es necesario mantener alguna forma de directorio de datos o repositorio de la información metadata. La metadata es la información alrededor de los datos dentro de la empresa.

Es necesario tener una variedad de metadata disponibles, información sobre las vistas de datos de los usuarios finales e información sobre las BD Operacionales. Idealmente, los usuarios finales deberían de acceder a los datos desde el DW, sin tener que conocer dónde residen los datos o la forma en que se han almacenados.

Nivel de gestión de procesos

El nivel de gestión de procesos tiene que ver con la programación de diversas tareas que deben realizarse para construir y mantener el DW y la información del directorio de datos.

Nivel de mensaje de la aplicación

El nivel de mensaje de la aplicación tiene que ver con el transporte de información alrededor de la red de la empresa. Se refiere también como "subproducto".

Nivel DW (Físico)

Es donde ocurre la data actual, usada principalmente para fines estratégicos. En un DW físico (muchas copias) de datos operacionales y/o externos, son almacenados realmente en una forma que es fácil de acceder y es altamente flexible. Los DW son almacenados sobre plataformas cliente/servidor, pero por lo general se almacenan sobre mainframes .

Nivel de organización de datos

El componente final de DWA es la organización de los datos o gestión de copia o réplica, incluye todos los procesos necesarios como seleccionar, editar, resumir, combinar y cargar datos en el depósito y acceder a la información desde BD Operacionales y/o Externas. Involucra una programación compleja, pero cada vez más, están creándose las herramientas DW para ayudar en este proceso.

OPERACIONES EN UN DW

Sistemas Operacionales

Los datos administrados por los sistemas de aplicación operacionales son la fuente principal de datos para el DW. Las BD Operacionales se organizan como archivos indexados, BD de redes/jerárquicas o sistemas de BD Relacionales (DB2, ORACLE, INFORMIX, etc.).

Extracción, Transformación y Carga de los datos

Se requieren herramientas de gestión de datos para extraer datos desde BD y/o archivos operacionales, luego es necesario manipular o transformar los datos antes de cargar los resultados en el DW.

Metadata

La metadata consiste de definiciones de los elementos de datos en el depósito, sistemas de los elementos fuente. Como la data, se integra y transforma antes de ser almacenada en información similar.

Acceso de usuario final

Los usuarios acceden al DW por medio de herramientas de productividad basadas en GUI ⁹ (Interface gráfica de usuario). Software de consultas, generadores de reportes, procesamiento analítico en línea, herramientas data/visual mining, etc., una sola herramienta no satisface todos los requerimientos, por lo que es necesaria la integración de una serie de herramientas.

Plataforma del DW

La plataforma para el DW es casi siempre un servidor de BD Relacional. Los extractos de la data integrada/transformada se cargan en el DW. Muchas de las organizaciones escogen una plataforma por diversas razones. El sistema de depósito ejecuta las consultas que se pasa a los datos por el software de acceso a los datos del usuario. Las consultas típicamente se formulan como pedidos SQL ¹⁰ (es un lenguaje universal y estándar para el acceso a datos).

Datos Externos

El alcance del DW puede extenderse por la capacidad de acceder a la data externa.

EVOLUCIÓN DEL DEPÓSITO

Construir un DW es una tarea grande. Se recomienda que los requerimientos de una serie de fases se desarrollen e implementen en modelos consecutivos que permitan un proceso de implementación más gradual e iterativo. Los datos en el DW no son volátiles y es un repositorio de datos de sólo lectura. Uno de los desafíos de mantener un DW, es idear métodos para identificar datos nuevos o modificados en las BD Operacionales.

TRANSFORMACIÓN DE DATOS Y METADATA

TRANSFORMACIÓN DE DATOS

Uno de los desafíos de cualquier implementación de DW, es el de transformar los datos, ya que se encarga de las inconsistencias en los formatos de datos y la codificación, que existe dentro de una BD única y que casi siempre existen cuando múltiples BD contribuyen al DW. Para llevar a cabo la transformación se requiere una planificación cuidadosa y detallada que permite transformar datos inconsistentes en conjuntos de datos conciliables y consistentes para cargarlos en el DW.

METADATA

Metadata es la información sobre los datos que se alimenta, se transforma y existe en el DW. Es un concepto genérico para cada implementación.

La metadata incluye los siguientes campos:

- Las estructuras de datos que dan una visión de los datos al administrador de datos.
- Las definiciones del sistema de registro desde el cual se construye el DW.
- Las especificaciones de transformaciones de datos que ocurren tal como la fuente de datos se replica al DW.

Algunas implementaciones de la metadata incluyen definiciones de las vistas presentadas a los usuarios del DW.

Los esquemas y subesquemas para BD Operacionales, forman una fuente óptima de entrada cuando se crea la metadata. La documentación existente acelera el proceso de definición de la metadata en el ambiente DW. La metadata se considera el corazón del ambiente DW.

FLUJO DE DATOS

Existe un flujo de datos normal dentro del DW. Los datos ingresan al DW desde el ambiente operacional. Al ingresar al DW, la información va al nivel de detalle actual, se queda ahí y se usa hasta que ocurra uno de los tres eventos siguientes:

- Sea eliminado.
- Sea resumido.
- Sea archivado.

Con el proceso de des-actualización en un DW se mueve el detalle de la data actual a data antigua.

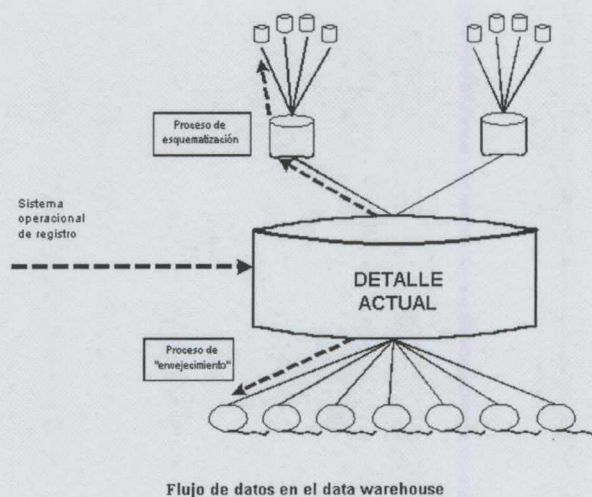


Fig. 8.1 Flujo de datos en el DW.

USOS DEL DW

Maneras diferentes de uso de datos

Los usuarios de un DW necesitan acceder a los datos complejos, frecuentemente desde fuentes múltiples y de formas no predecibles. Los usuarios efectúan tareas que requieren acceso a un conjunto de datos desde fuentes múltiples y frecuentemente no son predecibles. Por ejemplo, un especialista en el cuidado de la salud podría necesitar acceder a los datos actuales e históricos para analizar las tendencias de costos, usando un conjunto de consultas predefinidas.

Maneras diferentes de uso de datos

Sólo pocos usuarios acceden a los datos concurrentemente.

Los usuarios generan un procesamiento no predecible complejo

Los usuarios del DW generan consultas complejas y las respuestas a una consulta conducen a una formulación de otras preguntas más detalladas, es un proceso llamado *Drilling Down*¹¹. Los usuarios frecuentemente comienzan buscando en los datos resumidos y terminan accediendo al conjunto de datos detallado.

Las consultas de los usuarios accedan a cantidades grandes de datos

Las consultas al DW permiten acceder a volúmenes muy grandes tanto de data detallada como resumida.

Las consultas de los usuarios no tienen tiempos de respuesta críticos

En el DW, las transacciones operacionales tienen un requerimiento de respuesta no crítico porque el resultado frecuentemente se usa en un proceso de análisis y toma de decisiones. Los diferentes niveles de datos dentro del DW reciben diferentes usos.

Hay un mayor uso de los datos completamente resumidos, a diferencia de la información antigua que apenas es usada, la razón es que la data más resumida, permite capturar los datos en forma más rápida y eficiente.

La seguridad de la información de detalle se consigue de muchas maneras, aun cuando estén disponibles otros niveles de esquematización. Una de las actividades del diseñador de datos es el de desconectar al usuario del SSD del uso constante de datos con un detalle más bajo.

Preparación de un reporte complejo. Al crear un DW y combinar todos los datos requeridos, se obtienen los siguientes beneficios: Las inconsistencias de los datos se resuelven automáticamente cuando los elementos de datos se cargan en el DW, no manualmente, cada vez que se prepara un reporte. Los errores que ocurrieron durante el proceso complejo de la preparación del informe, se minimizan porque el proceso es ahora mucho más simple. Los elementos de datos son fácilmente accesibles para otros usos, no sólo para un reporte particular. Se crea una sola fuente.

CONSIDERACIONES ADICIONALES Y EXCEPCIONES EN EL DW

Existen consideraciones adicionales que deben tenerse en cuenta al construir y administrar el DW.

- La primera consideración es respecto al índice. La información de los niveles de esquematización más altos pueden ser libremente indexados y los niveles más bajos de detalle pueden ser indexados moderadamente.
- Los datos en los niveles más altos de detalle pueden ser reestructurados fácilmente.
- Los datos en los niveles más inferiores son tan grandes, que los datos no pueden ser fácilmente reestructurados.

- Estructural, el nivel de detalle actual es casi siempre particionado de dos maneras: al nivel de DBMS y al nivel de la aplicación.

Excepciones dentro del DW. La data resumida es calculada fuera del DW pero es usada a través de la corporación. El detalle de datos permanentes comparte muchas de las mismas consideraciones como otro DW, excepto que:

- El medio donde se almacena la data debe ser tan seguro como sea posible.
- Los datos deben permitir ser restaurados.
- Los datos necesitan un tratamiento especial en su indexación, ya que de otra manera los datos pueden no ser accesibles aunque se haya almacenado con mucha seguridad.

ORGANIZACIÓN DE UN PROYECTO

Factores en la Planificación de un DW

Puntos claves que deben considerarse en la planificación de un DW:

- Establecer una asociación de usuarios, gestión y grupos.

Es esencial involucrar tanto a los usuarios como a la gestión para asegurar que el DW contenga información que satisfaga los requerimientos de la empresa.

- Seleccionar una aplicación piloto con una alta probabilidad de éxito.
- Construir prototipos rápida y frecuentemente.

La única manera para asegurar que el DW reúna las necesidades de los usuarios, es hacer el prototipo a lo largo del proceso de implementación.

- Implementación incremental.

Reduce riesgos y asegura que el tamaño del proyecto permanezca manejable en cada fase.

- Reportar activamente y publicar los casos exitosos.

La retroalimentación de los usuarios ofrece una excelente oportunidad para publicar los hechos exitosos dentro de una organización.

Estrategias para el Desarrollo de un DW

- Primera.

Establecer un ambiente "DW virtual", el cual puede ser creado por:

- Instalación de un conjunto de facilidades para acceso a datos, directorio de datos y gestión de proceso.
- Entrenamiento de usuarios finales.
- Control de cómo se usan realmente las instalaciones del DW.
- Basados en el uso actual, crear un DW físico para soportar los pedidos de alta frecuencia.

- Segunda.

Construir una copia de los datos operacionales y posibilitar al DW de una serie de herramientas de acceso a la información. Esta estrategia tiene la ventaja de ser simple y rápida.

- Tercera.

La estrategia DW óptima es seleccionar el número de usuarios basados en el valor de la empresa y hacer un análisis de sus puntos, preguntas y necesidades de acceso a datos.

De acuerdo a estas necesidades, se construyen los prototipos DW y se prueban para que los usuarios finales puedan experimentar y modificar sus requerimientos.

- En conclusión.

Como la tecnología DW va evolucionando, se aprende cada vez más y más sobre el desarrollo de DW, que resulta en que el único enfoque práctico para el almacenamiento de datos es la evolución de uno mismo.

Estrategias para el Diseño de un DW

1. Los usuarios de DW usualmente no conocen mucho sobre sus requerimientos y necesidades como los usuarios operacionales.
2. El diseño de un DW, involucra lo que se piensa en términos más amplios y con conceptos del negocio más difíciles de definir que en el diseño de un sistema operacional. Un DW está bastante cerca a Reingeniería de los Procesos del Negocio (Business Process Reengineering).
3. La estrategia de diseño ideal para un DW es generalmente de afuera hacia adentro.

Estrategias para el Gestión de un DW

1. Un DW es una inversión buena sólo si los usuarios finales realmente pueden conseguir información vital más rápida y más barata de lo que obtienen con la tecnología actual.
2. La administración debe reconocer que el mantenimiento de la estructura del DW es tan crítico como el mantenimiento de cualquier otra aplicación de misión crítica.
3. La gestión debe comprender también que si se embarcan sobre un programa DW, se crearán nuevas demandas sobre sus sistemas operacionales, como: demandas para mejorar datos, demandas para una data consistente, demandas para diferentes tipos de datos, etc.

DESARROLLO DE UN PROYECTO

Lo primero para ampliar un negocio, se necesita que la información sea comprensible. La combinación de dos factores ha ayudado a la difusión del DW.

1. Se ha reconocido los beneficios del procesamiento analítico en línea (OLAP¹²), más allá de las áreas tradicionales de marketing y finanzas.
2. El crecimiento de la computación cliente/servidor, ha creado servidores de hardware y software más poderosos y sofisticados que nunca.

Consideraciones Previas al Desarrollo de un DW

- Alcance de un DW.

El alcance de un DW puede ser tan amplio como toda la información estratégica de la empresa desde su inicio, o puede ser tan limitado como un DW personal para un solo gerente.

- Redundancia de Datos.

Hay tres niveles esenciales de redundancia de datos

- DW "virtual" o "Point to Point".
- DW "centrales".
- DW "distribuidos".

DW "VIRTUAL" O "POINT TO POINT"

Significa que los usuarios finales pueden acceder a BD Operacionales directamente, usando cualquier herramienta que posibilite "la red de acceso de datos". Este enfoque provee flexibilidad así como también la cantidad mínima de datos redundantes que deben cargarse y mantenerse. El almacenamiento virtual es una estrategia inicial en las organizaciones.

DW "CENTRALES"

Es una única BD Física, que contiene todos los datos para un área funcional específica. Se seleccionan por lo general donde hay una necesidad común de los datos informáticos y un número grande de usuarios finales ya conectados a una red o computadora central.

DW "DISTRIBUIDOS"

Son aquellos en los cuales ciertos componentes del depósito se distribuyen a través de un número de BD Físicas diferentes.

Comúnmente involucran la mayoría de los datos redundantes y como consecuencia de ello, se tienen procesos de actualización y carga más complejos.

- Tipo de Usuario Final.

De la misma forma que hay una gran cantidad de maneras para organizar un DW, es importante notar que también hay una gama cada vez más amplia de usuarios finales.

Elementos Claves para el Desarrollo de un DW

Los DW exitosos comienzan cuando se escogen e integran satisfactoriamente tres elementos claves. Cuando se escoge incorrectamente, el DW se convierte la empresa en una gran empresa con problemas difíciles de trabajar en su entorno, costoso para arreglar y difícil de justificar. Para conseguir que la implementación del depósito tenga un inicio exitoso, se necesita enfocar hacia tres bloques claves de construcción:

- Arquitectura total del depósito.
- Arquitecturas del servidor.
- Sistemas de Gestión de BD.

ARQUITECTURA DEL DEPÓSITO

El desarrollo del DW comienza con la estructura lógica y física de la base de datos del depósito más los servicios requeridos para operar y mantenerlo.

ARQUITECTURA DEL SERVIDOR

Al decidir sobre una estructura de depósito distribuida o centralizada, también se necesita considerar los servidores que retendrán y entregarán los datos. El tamaño de su implementación influirá en la elección de la arquitectura del servidor.

1. Servidores de un solo procesador: Los servidores de un sólo procesador son los más fáciles de administrar, pero ofrecen limitada potencia de procesamiento y escalabilidad.
 2. Multiprocesamiento simétrico: Las máquinas de multiprocesamiento simétrico (SMP¹³) aumentan mediante la adición de procesadores que comparten la memoria interna de los servidores y los dispositivos de almacenamiento de disco.
 3. Procesamiento en paralelo masivo: Una máquina de procesamiento en paralelo masivo (MPP¹⁴), conecta un conjunto de procesadores por medio de un enlace de banda ancha y de alta velocidad. Cada nodo es un servidor, completo con su propio procesador y memoria interna.
 4. Acceso de memoria no uniforme: NUMA¹⁵ crea una sola gran máquina SMP al conectar múltiples nodos SMP en un solo (aunque físicamente distribuida) banco de memoria.
- Sistemas de Gestión de BD.

La gran parte de los sistemas operacionales fueron resultados de aplicaciones basadas en antiguas estructuras de datos, los depósitos y SSD aprovecharon el RDBMS¹⁶ por su flexibilidad y capacidad para efectuar consultas con un único objetivo concreto. Los RDBMS son muy flexibles cuando se usan con una estructura de datos normalizada. En una BD normalizada, las estructuras de datos son no redundantes y representan las entidades básicas y las relaciones descritas por los datos. Muchas de las herramientas de acceso a los DW explotan la naturaleza multidimensional del DW. La estructura de los datos en una BD Relacional tradicional, facilita consultas y análisis a lo largo de dimensiones diferentes que han llegado a ser comunes.

Los esquemas podrían usar tablas múltiples e indicadores para simular una estructura multidimensional. Las BD Multidimensionales (MDDBs) ayudan directamente a manipular los objetos de datos multidimensionales (por ejemplo, la rotación fácil de los datos para verlos entre dimensiones diferentes.

- Nuevas Dimensiones.

Una limitación de un RDBMS y un MDDB, es la carencia de soporte para tipos de datos no tradicionales como imágenes, documentos y clips de vídeo / audio. Muchos RDBMS almacenan los datos complejos como objetos grandes binarios (BLOBs). En este formato, los objetos no pueden ser indexados, clasificados, o buscados por el servidor. Los DBMS relacional - objeto, almacenan los datos complejos como objetos nativos y pueden soportar las grandes estructuras de datos encontradas en un ambiente orientado a objetos. Una desventaja del enfoque relacional - objeto, es que la encapsulación de los datos dentro de los tipos especiales de datos, requiere de operadores especializados para que hagan búsquedas simples previamente.

- Combinación de la Arquitectura con el Sistema de Gestión de BD.

Para seleccionar la combinación correcta de la arquitectura del servidor y el DBMS, primero es necesario comprender los requerimientos comerciales de su compañía, su población de usuarios y las habilidades del personal de soporte. Por regla general, a mayor área del depósito, se requiere mayor potencia y funcionalidad del servidor y el DBMS. El análisis complejo, que es típico de los ambientes de decisión - soporte, requiere más poder y flexibilidad de todos los componentes del servidor y los ambientes dinámicos, con sus requerimientos siempre cambiantes, se adaptan mejor a una arquitectura de datos simple.

El valor de la data fresca requerida indica cuán importante es para el DW renovar y cambiar los datos.

- Planes de Expansión.

Su depósito evoluciona y los datos que contiene llegan a ser más accesible, por ejemplo: al enlazar su DW a otros sistemas, se puede compartir información con otras entidades comerciales con poco o sin desarrollo. Al crecer el DW, llegarán a ser más organizados, más interconectados, más accesibles y, en general, más disponibles a más empleados. El resultado de esta evolución, será la obtención de mejores decisiones en el negocio, más oportunidades y más claridad de trabajo.

Confiabilidad de los Datos.

La data "sucia" es peligrosa. Las herramientas de limpieza especializadas y las formas de programar de los clientes proporcionan redes de seguridad. Si se alimenta mal la información, se obtendrá resultados incorrectos o falsos. Los datos "sucios" pueden presentarse al ingresar información en una entrada de datos y daña la credibilidad de la implementación del depósito completo.

Decidir qué herramienta usar es importante y no solamente para la integridad de los datos. Si se equivoca, se podría malgastar semanas en recursos de programación o cientos de miles de dólares en costos de herramientas. La limpieza de una data "sucia" es un proceso multifacético y complejo. Los pasos a seguir son los siguientes:

1. Analizar sus datos corporativos para descubrir inexactitudes, anomalías y otros problemas.
2. Transformar los datos para asegurar que sean precisos y coherentes.
3. Asegurar la integridad referencial, que es la capacidad del DW, para identificar correctamente al instante cada objeto del negocio, tales como un producto, un cliente o un empleado.
4. Validar los datos que usa la aplicación del DW.

Parte II. DATA WAREHOUSE EN ORACLE 8i

INTRODUCCIÓN DE DW EN ORACLE 8i

Un DW es una BD que esta diseñada para consultas y análisis en lugar de procesamientos de transacciones. Usualmente contiene datos históricos que son derivados de una transacción, pero pueden incluir datos de otras fuentes. Separa la carga de trabajo de análisis de la carga de trabajo de transacción y permite a la organización consolidar los datos de varias fuentes.

En adición a la BD, un ambiente DW frecuentemente consiste de una solución de extracción, transportación y transformación (ETT¹⁹), un proceso analítico en línea (OLAP), herramientas de análisis para el cliente, y otras aplicaciones que manejan el proceso de reunir los datos y lo entrega a los usuarios.

ESQUEMAS DE DATA WAREHOUSING

Un esquema es una colección de BD, incluyendo tablas, vistas, índices y sinónimos. Hay una variedad de arreglos en el modelo del esquema diseñado para DW-ing. La mayoría de los DW usan un modelo dimensional.

Esquemas de Estrella (Star Schemas)

El esquema de estrella es el esquema más simple del DW. Es llamado esquema de estrella porque el diagrama de un esquema de estrella se asemeja a una estrella, con puntos que se irradian, salen a partir de un punto. El centro de la estrella consiste de una o más tablas de eventos y los puntos de la estrella son las dimensiones de la tabla.

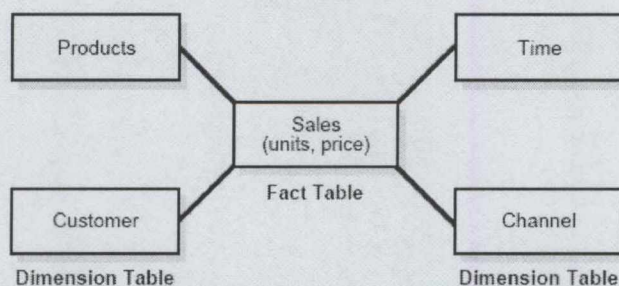


Fig. 13.1.1 Esquema de Estella.

A diferencia de otras estructuras de la BD, en un esquema de estrella, las dimensiones son anormales o irregulares. Esto es, la dimensión de la tabla tiene redundancias, al eliminar las uniones múltiples en la tabla de dimensiones. En un esquema estrella, solamente una unión es necesaria para establecer la relación entre la tabla de eventos y cualquiera de las dimensiones de la tabla.

La principal ventaja de un esquema de estrella es optimizar los cambios. Un esquema de estrella se queda con las búsquedas simples y provee respuestas rápidas porque toda la información en cada nivel es almacenada en una línea.

OBJETOS DE DATA WAREHOUSING

Los siguientes tipos de objetos son comúnmente utilizados en DW:

- En el esquema warehouse la Fact-table²⁰ (ubicada al centro). La fact-table contiene típicamente eventos y llaves foráneas de la dimensión de la tabla. Usualmente representa datos numéricos y aditivos que pueden ser analizados y examinados. Ejemplo: incluye ventas, costos y beneficios.
- Dimensión de la tabla, también conocido como “operaciones de búsqueda” o “tablas de referencia”, contiene los datos relativamente estáticos en el warehouse. Ejemplo: el almacenamiento o los productos.

FACT-TABLES (TABLAS DE EVENTOS)

Una fact-table es una tabla en un esquema de estrella que contiene eventos. Una fact-table típicamente tiene dos tipos de columnas: la que contiene eventos y la que tiene llaves foráneas para la dimensión de la tabla. Una fact-table puede contener cada detalle del nivel de eventos o eventos que han sido agregados. Una fact-table que contiene eventos agregados es frecuentemente llamada “Tabla de resumen”. Una fact-table usualmente contiene eventos con el mismo nivel de agregación.

Los valores de los eventos o medidas son usualmente desconocidas; estos son observados y almacenados.

Una fact-table tiene las bases para la búsqueda de datos en las herramientas OLAP.

DIMENSIONES

Una dimensión es una estructura frecuentemente compuesta de una o más jerarquías, que categoriza los datos. Varias dimensiones distintas, combinadas con medida, son capaces de responder las preguntas de trabajo. Comúnmente se usan dimensiones para los clientes, los productos y las fechas como se muestra en la figura 13.4.1.

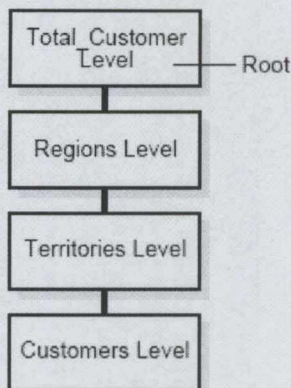


Fig. 13.4.1 Niveles típicos en una dimensión jerárquica.

La dimensión del dato es colectado típicamente en el nivel más bajo de detalle y luego es agregado en un nivel total más alto, lo cual es más eficiente para el análisis. Por ejemplo, en la dimensión de clientes totales hay 4 niveles: cliente total, regiones, territorios y clientes. La colección de datos en el nivel de clientes es agregada al nivel de territorios. Para la dimensión de regiones, la colección de datos para varias regiones tales como el este de Europa u oeste de Europa pueden ser agregadas como un evento en la tabla de eventos dentro del total para un área más grande como lo es Europa.

JERARQUÍAS

Las jerarquías son estructuras lógicas que se usan para ordenar niveles como una medida de organización de datos. Una jerarquía puede ser usada para definir una agregación de datos. Por ejemplo, en una dimensión de datos, una jerarquía puede ser usada para agregar datos del nivel mensual al nivel quincenal o al nivel anual. Una jerarquía también puede ser usada para definir una ruta de navegación y establecer una estructura familiar.

Dentro de una jerarquía, cada nivel está conectado lógicamente a los niveles superiores e inferiores; los valores de los datos que están en los niveles más bajos se agregan en los datos de los niveles más altos. Por ejemplo, en la dimensión producto, puede haber dos jerarquías (uno para la identificación del producto y uno para la responsabilidad del producto).

La dimensión de jerarquía también agrupa niveles desde muy generales hasta muy particulares. Las jerarquías son utilizadas como herramientas de búsqueda, permitiendo que usted modifique los datos de abajo a los diferentes niveles particulares (uno de los beneficios clave de un DW).

DISEÑO FÍSICO

Un diseño físico está cuando se cambia el esquema esperado en una estructura de BD actual. En ese momento se tiene un mapa:

- Entidades a las tablas.
- Relaciones a las llaves foráneas.
- Atributos a las columnas.
- Identificadores únicos primarios a la llave primaria.
- Identificadores únicos a las llaves únicas.

ESTRUCTURA DEL DISEÑO FÍSICO

Para cambiar el esquema en una estructura de BD actual se requiere crear lo siguiente:

- Tablespace.
- Particiones.
- Índices.
- Constraints.

Tablespace necesita ser separada por diferencias, por ejemplo, las tablas pueden estar separadas por índices y las tablas pequeñas pueden estar separadas dentro de grandes tablas.

Particiones las tablas de particiones grandes desarrollan cambios porque cada pieza particionada es más manejable. Típicamente, la partición esta basada en una transacción de fecha en un DW, por ejemplo, cada mes. Este trabajo mensual de datos puede ser asignado en su propia partición.

Índices un índice DW se asemeja a un índice OLTP. Un punto importante es que el índice bitmap tiene consultas comunes.

Constraints son un poco diferentes en el DW comparados con los ambientes OLTP porque la integridad de los datos esta razonablemente garantizada debido a las fuentes limitadas de datos y porque se puede verificar la integridad de los datos de grandes archivos de esa jornada.

HARDWARE Y ENTRADA/SALIDA (E/S)

STRIPING DATA

Para evitar el tráfico de E/S durante el procesamiento de paralelos o el acceso de búsquedas en curso, todas las tablespaces accesadas por operaciones paralelas deben ser rayadas (striping) como se muestra en la figura 15.1.1, la tablespace siempre debe rayar sobre al menos un dispositivo de CPU; en este ejemplo, hay 4 CPU's.

También se debe distribuir el dispositivo en los controladores, canales de E/S y o los buses internos.

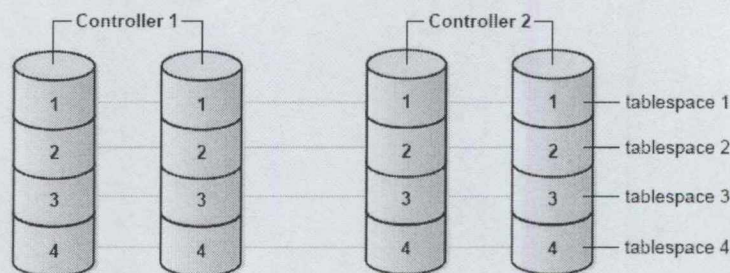


Fig. 15.1.1 Las rayas se oponen el excedente por lo menos tantos dispositivos como CPU's.

Esto también es importante para garantizar que el dato es distribuido a los archivos. Una forma de rayar el dato durante la carga, es usar el archivo que es igual a la cláusula del cargador de paralelo que sirven para cargar los datos en sesiones múltiples de carga dentro de archivos diferentes en la tablespace. Para ser un rayado efectivo, se debe garantizar que estén disponibles suficientes controladores y componentes de E/S para soportar el ancho de banda del movimiento del dato paralelo dentro y fuera de la tablespace stripe.

CONSIDERACIONES DE ENTRADA/SALIDA

Se debe ser cuidadoso cuando se configura el subsistema de E/S a través de la capacidad de poner en los discos que serán accedidos en un tiempo dado que no exceden la capacidad de poner los controladores de E/S. Se debe balancear el número de discos con el número total de controladores de E/S para prevenir un proceso lento del nivel de controlador.

PARALELISMO Y PARTICIONISMO

El DW con frecuencia contiene tablas grandes y requiere de técnicas para administrar estas tablas y proveer una buena consulta para el funcionamiento a través de estas tablas. Estas dos técnicas son necesarias para conseguir las direcciones.

La ejecución paralela reduce dramáticamente las respuestas para operaciones de datos intensivos en una BD grande típicamente asociada con los SSD. Se puede implementar una ejecución paralela en ciertos procesos de transacciones en línea y sistemas híbridos.

TIPOS DE PARALELISMO

Las diferentes operaciones en paralelo usan diferentes tipos de paralelismo. La capa óptima de la BD física depende de que operaciones en paralelo son más prevalecientes en la aplicación.

La unidad básica de paralelismo es llamada *granule*. La operación que está siendo paralelizada (por ejemplo: un escaneo de tablas, una modificación, la creación de un índice) es dividida en Oracle por gránulos. Los procesos de ejecución en paralelo ejecutan la operación de un gránulo a tiempo. El número de gránulos y su tamaño afecta el grado de paralelismo (DOP²¹) que se puede usar.

Esto también afecta como el trabajo es balanceado a través de consultas del servidor.

Block Range Granules

El block range granule es la unidad básica de la mayoría de la operaciones en paralelo. Esto es verdadero aún en tablas particionadas; esta es la razón de porque en Oracle, el grado de paralelismo no es relacionado con el número de particiones.

El block range granules son rangos de bloques físicos porque están basados en direcciones físicas de datos, Oracle puede medir los block range granules para permitir un mejor balance. El block range granules permite un paralelismo dinámico que no depende de una pre-asignación estática de tablas o índices. En sistema SMP, son localizados en diferentes dispositivos para enviarlos a tantos discos como sea posible. En muchos sistemas MPP, los block range granule son preferencialmente asignados a los procesos de consulta del servidor que tiene una proximidad física con los gránulos para el almacenamiento en discos.

Cuando un block range granule es usado predominantemente por un acceso paralelo para una tabla o un índice, las consideraciones administrativas podrían influenciar la capa de partición más que las consideraciones de funcionamiento. El número de discos que cubren las particiones deberían ser al menos igual a el valor del DOP de modo que la eficiente ejecución en paralelo no se reduce cuando se fractura una partición.

Partition Granules (Partición de gránulos)

Cuando se utiliza una partición de gránulos, un proceso de consulta en el servidor trabaja en una partición completa a una sub-partición de una tabla o índice. Porque la partición de gránulos son estáticamente determinados cuando una tabla o índice se crea, la partición de gránulos no permite tanta flexibilidad en una operación paralelizando. Esto quiere decir que es permitible el DOP en una fuerza limitada y que no carga la fuerza que debería balancear a través de los procesos de consulta al servidor.

La partición de gránulos es la unidad básica del rango del índice paralelo y de las operaciones paralelas que modifican múltiples particiones de una tabla o índice particionado. Estas operaciones incluyen modificación paralela, borrado paralelo, una carga directa paralela para insertar en tablas particionadas, creación paralela de índices particionados, y una creación paralela de tablas particionadas.

Cuando la partición de gránulos son usados por un acceso paralelo hacia una tabla o índices, esto es importante ya que hay una gran relatividad de particiones. Oracle puede balancear eficazmente el trabajo a través de los procesos de consulta del servidor.

TIPOS DE PARTICIONES

Oracle ofrece tres métodos de partición:

- Range
- Hash
- Composite

Partición Range

La partición range, mapea datos hacia particiones basadas en límites identificados por rangos de valores de columnas que se establecen para cada partición. Este método es frecuentemente útil para aplicaciones que administran datos históricos, especialmente DW.

Partición Hash

La partición hash mapea datos hacia particiones basadas en un algoritmo de hashing que Oracle aplica para una llave de partición identificada por el usuario. El algoritmo de hashing distribuye uniformemente filas entre particiones. Por lo tanto, el resultado de las particiones debería ser aproximadamente del mismo tamaño. Esto hace que la partición de hash sea ideal para distribuir datos uniformemente a través de dispositivos. La partición hash es una buena y fácil alternativa para la partición range, cuando los datos no se encuentran en el contenido histórico.

Partición Composite

La partición composite combina las características de las particiones hash y range. Con la partición composite, Oracle primero distribuye datos dentro de particiones de acuerdo a los límites establecidos por las particiones range.

Entonces Oracle divide datos en sub-particiones dentro de particiones range.

Oracle usa un algoritmo hashing para distribuir datos dentro de las sub-particiones.

PARTICIONAR ÍNDICES

Se pueden crear ambos índices globales y locales en una tabla particionada por particiones range, hash o composite. Los índices locales heredan los atributos de la partición de sus tablas relacionadas.

Por ejemplo, si se crea un índice local en una tabla composite, Oracle automáticamente particiona el índice local usando el método composite.

Oracle soporta solo particiones range para índices globales. No se puede particionar un índice global usando los métodos de partición hash o composite.

PARTITION-WISE JOIN

Una partition-wise join reduce el tiempo de respuesta de la consulta para minimizar la cantidad de datos intercambiados entre consultas al servidor cuando las uniones se ejecutan en paralelo. Esto significativamente reduce el tiempo de respuesta y la utilización de recursos, ambos en términos de CPU y memoria.

En ambientes Oracle Parallel Server (OPS²²), esto también se permite o al menos en los límites de tráfico de datos sobre la interconectividad, lo cual es la llave para una buena realización escalable para masivas operaciones de unión.

Hay dos variaciones de partition-wise join, completa y parcial.

Una partition-wise join completa divide una unión grande dentro de pequeñas particiones dentro de un par de particiones de dos tablas unidas, para usar estas características, se debe equi-particionar ambas tablas en sus uniones claves. Existen varios métodos para realizar esto, como lo son: hash-hash, composite-hash, composite-composite (hash dimension), range-range, range-composite, composite-composite (range dimension), los cuales son una variación de todos estos métodos.

Una partition-wise join parcial en Oracle solo se puede realizar una partition-wise join parcial en paralelo. A diferencia de la partition-wise join completa, ésta partición requiere que se particione solamente una tabla en la unión clave, no en ambas. La tabla particionada es referida como una tabla de referencia. La otra tabla podría o no podría ser particionada. La partition-wise join parcial es más común que una partition-wise join completa, porque esta requiere una sola partición de una partición de tablas particionadas en una unión clave. Existen varios métodos para realizar estas particiones, como lo son: hash, composite, range.

Los Beneficios de partition-wise join son, la reducción de comunicaciones encimadas y la reducción de requerimientos de memoria.

ÍNDICES

BITMAP INDEXES

Los índices bitmap²³ son usados extensamente en aplicaciones DW, el cual tiene grandes cantidades de datos y consultas pero un nivel bajo de transacciones concurrentes.

Para tales aplicaciones los índices bitmap proveen:

- Reduce el tiempo de respuesta para clases grandes de consultas.
- Una reducción sustancial de espacios en uso comparadas con otras técnicas de indexación.
- Un funcionamiento dramático gana incluso en el hardware con un pequeño número relativo de CPU o pequeñas cantidades de memoria.
- Un mantenimiento muy eficiente durante DML²⁴ paralelo y cargas.

Una indexación completa en una tabla muy grande con un índice tradicional b-tree puede prohibir un costo en términos de espacio, porque los índices pueden ser varias veces más largos que los datos en la tabla.

Los índices bitmap son típicamente solo una fracción del tamaño del dato indexado en la tabla.

El propósito de un índice es proveer puntos en las filas de una tabla que contiene un valor clave dado. En un índice regular, este es alcanzado por almacenar una lista de filas "ID" para cada clave correspondiente a la fila con ese valor clave.

Un índice bitmap, un bitmap para cada valor clave es usado en vez de una lista de filas "ID".

Cada bit en el bitmap corresponde a una posible fila "ID" y si se fija el bit, es decir que la fila con la fila "ID" correspondiente contiene el valor clave. Una función mapping²⁵ convierte la posición del bit a una fila "ID" actual, así que el índice bitmap provee la misma funcionalidad como un índice regular, aunque este se use en una representación interna diferente. Si el número de diferentes valores claves son pequeños los índices bitmap son muy eficientes en espacio.

Los índices bitmap son más eficientes para las consultas que contienen múltiples condiciones en la cláusula WHERE. Las filas satisfacen algo, pero no todas las condiciones son filtradas antes de que la tabla misma sea escaneada.

Esto mejora el tiempo de respuesta, a menudo dramáticamente.

Beneficios para aplicaciones DW

Los índices bitmap no son convenientes para aplicaciones OLTP con grandes números de transacciones concurrentes que modifican los datos. Estos índices son ante todo previstos por SSD en aplicaciones de almacenamiento donde los usuarios consultan los datos mejor modificados.

La consulta y el DML paralelo trabajan con índices bitmap así como con índices tradicionales. Crear índices paralelos y concatenados también son soportados.

ÍNDICES B-TREE

Un índice b-tree es organizado como un árbol arriba-abajo. El nivel más profundo del índice mantiene los valores actuales de los datos y apunta hacia las filas correspondientes como el índice en un libro que tiene un número de páginas asociadas con cada índice de entrada.

En general se pueden usar índices b-tree cuando se sabe que la consulta típica se requiere a columnas indexadas y recuperación de pocas filas. En estas consultas, esto es rápido para encontrar filas que se buscan en el índice.

Los índices b-tree son más comúnmente usados en DW para indexar llaves las cuales son únicas o casi únicas. En muchos casos, no sería necesario para el índice estas columnas en un DW porque los constraints únicos pueden ser mantenidos sin un índice, y desde las típicas consultas de DW no podrían producir mejor funcionamiento con tales índices. Los índices bitmap deberían ser más comunes que los índices b-tree en la mayoría de los ambientes DW.

CONSTRAINTS

UTILIDAD DE LOS CONSTRAINTS EN DW

Los constraints proveen un mecanismo para asegurarse que los datos sean especificados conformes a las pautas dadas por el administrador de la BD. Los tipos más comunes de constraints incluyen constraints único (asegura que una columna dada sea única), constraints no nulos y constraints de llaves foráneas.

Los constraints pueden ser usados por los siguientes propósitos en un DW:

- Data cleanliness²⁶: los constraints pueden ser usados para verificar que los datos en el DW conformen el nivel básico de la consistencia y corrección de los datos. Así, los constraints son frecuentemente utilizados en el DW para prevenir la introducción de datos sucios.
- Optimización del soporte a la consulta: la BD Oracle utilizara constraints cuando se optimicen consultas en SQL. Aunque los constraints pueden ser útiles en muchos aspectos para la optimización de la consulta, los constraints son particularmente importantes para re-escribir la consulta de vistas materializadas.

A diferencia de muchos ambientes de BD Relacional los datos en un DW son típicamente añadidos o modificados debajo de una circunstancia controlada durante el proceso ETT. Múltiples usuarios no modifican el DW directamente, esto es considerablemente diferente del uso de un sistema operacional.

Así que el uso específico de los constraints en el DW puede variar considerablemente desde el uso de los constraints en sistemas operacionales. Oracle 8i provee una ancha funcionalidad de constraints para direccionar las necesidades del DW así como las necesidades del sistema operacional.

TÍPICOS CONSTRAINTS DEL DW

Los típicos constraints del DW, son:

- Constraints únicos en DW: se encarga de usar un índice único en tablas que pueden ser extremadamente largas, este podría ser muy costoso en términos de procesamiento y de espacio en disco.

- Constraints de llaves foráneas en DW : al empezar un esquema en DW los constraint de llave foránea son típicamente creados en orden para validar la relación entre la tabla hecho y la dimensión de las tablas.
- Rely constraints: este es el menos común en el procesos ETT para verificar los constraints que aseguran que sean verdaderos, para crear uno de estos constraints es muy caro, ya que estos constraints no son validados y no hay un procesamiento de datos necesarios para crear tal constraint.

CONSTRAINTS Y PARALELISMO

Todos los constraints pueden ser validados en paralelo. Cuando se validan constraints en tablas muy largas el paralelismo es frecuentemente necesario para reunir las metas del funcionamiento.

El grado de paralelismo para una operación de un constraint dado es determinado por el grado de paralelismo de la tabla.

CONSTRAINTS Y PARTICIONES

Muchos aspectos de crear y mantener constraints pueden ser ejecutados en una partición.

VISTAS MATERIALIZADAS

VISTAS MATERIALIZADAS PARA DW

En DW las vistas materializadas pueden ser usadas para almacenar datos agregados tales como ventas. Las vistas materializadas en este tipo de ambientes son típicamente referidas a resúmenes porque estos almacenan los datos resumidos. Pueden también ser usadas para uniones con o sin agregación. Eliminan también asociaciones altas con uniones costosas y/o agregaciones muy grandes o importantes clases de consultas.

LA NECESIDAD DE MATERIALIZAR VISTAS

Materializar vistas son usualmente usadas en DW para incrementar la velocidad de consultas en BD muy grandes. Las consultas en BD muy grandes frecuentemente implican uniones entre tablas o agregaciones tales como SUM, o ambas.

Estas operaciones son muy costosas en término de tiempo y poder de procesamiento. El tipo de vista materializada que es creada determina como esta puede ser actualizada y usada para reescribir consultas. Las vistas materializadas pueden ser usadas en un número de manera diferentes y casi siempre la sintaxis es usada idénticamente para realizar un número de roles diferentes. Las vistas materializadas mejoran el funcionamiento de las consultas para precalcular uniones costosas en operaciones de la agregación en la BD antes del tiempo de la ejecución en el almacenamiento de los resultados de la BD.

La optimización de las vistas usan vistas materializadas por un reconocimiento automático cuando una vista materializada existente puede y debería usarse para satisfacer un requerimiento.

Diseño del esquema para una vista materializada

Una definición de una vista materializada puede incluir cualquier número de agregaciones también como cualquier número de uniones en varias maneras una vista materializada se comporta como un índice:

- El propósito de la vista materializada es incrementar es incrementar el funcionamiento de la ejecución de peticiones.
- La existencia de una vista materializada es transparente para ejecuciones SQL así que un DBA puede crear o borrar vistas materializadas en cualquier tiempo sin afectar las aplicaciones validadas en SQL.
- Una vista materializada consume espacio de almacenamiento.
- El contenido de una vista materializada debería ser modificado cuando un detalle de una tabla subyacente es modificada.

TIPOS DE VISTAS MATERIALIZADAS

La cláusula SELECT es la creación de una vista materializada que define el dato que esta contiene. Hay solo unas pocas restricciones en las que podrían ser especificadas. Cualquier número de tablas pueden ser unidas juntas, sin embargo, no pueden ser tablas remotas si se desea tomar ventaja de reescribir consultas o de actualizar el warehouse fácilmente. Detrás de tablas, vistas en línea, consultas, subconsultas y vistas materializadas podrían ser todas unidas o referenciadas a la clase SELECT.

Los tipos de vistas materializadas son:

- Vistas materializadas con uniones y agregaciones.
- Tablas simples de vistas materializadas agregadas.
- Vistas materializadas que contienen solo uniones.

CREACIÓN DE UNA VISTA MATERIALIZADA

Una vista materializada puede ser creada con la declaración CREATE MATERIALIZED VIEW o usando Oracle Enterprise Manager. El siguiente comando crea una vista materializada store_sales_mv.

```
CREATE MATERIALIZED VIEW store_sales_mv
  PCTFREE 0 TABLESPACE mviews
  STORAGE (INITIAL 16k NEXT 16k PCTINCREASE 0)
  PARALLEL
  BUILD IMMEDIATE
  REFRESH COMPLETE
  ENABLE QUERY REWRITE
AS
SELECT
  s.store_name,
  SUM(dollar_sales) AS sum_dollar_sales
  FROM store s, fact f
  WHERE f.store_key = s.store_key
  GROUP BY s.store_name;
```

Ejemplo 19.4.1 Creación de una vista materializada.

JERARQUIZAR VISTAS MATERIALIZADAS

Una vista materializada jerarquizada es una definición de una vista materializada completa que esta basada en otra vista materializada. Una vista jerárquica o materializada puede referenciar otras relaciones en la BD además de vistas materializadas.

Las vistas materializadas se usan en un DW porque crean muchas vistas agregadas en una unión simple. El mantenimiento incremental de vistas materializadas agregadas distintas que pueden tomar largo tiempo porque las uniones adyacentes tienen que ser realizadas muchas veces. Al usar vistas materializadas jerarquizadas la unión es realizada una vez y el mantenimiento incremental de tablas simples en una vista materializada agregada es muy rápido debido al mantenimiento mismo de operaciones actualizadas en este tipo de vistas.

REGISTRAR UNA VISTA MATERIALIZADA EXISTENTE

Algunos DW tienen vistas materializadas implementadas en tablas ordinarias del usuario. Aunque esta solución provee el funcionamiento benéfico de vistas materializadas no:

- No provee consultas rescritas para todas las aplicaciones de SQL.
- No permite vistas materializadas definidas en una aplicación que es transparentemente accedida en otra aplicación.
- Generalmente soporta actualizaciones de vistas materializadas rápidas en paralelo o incrementales.

Debido a que estos problema, y debido a que existen vistas materializadas que podrían ser extremadamente grandes y costosas para reconstruir, se debería registrar las tablas existentes de vistas materializadas con Oracle siempre que sea posible. Se puede registrar una vista materializada de un usuario no definido con la declaración `CREATE MATERIALIZED VIEW... ON PREBUILT TABLE`. Una vez registrada, la vista materializada puede ser usada por consultas rescritas o el mantenimiento de estos métodos actualizados o ambos.

PARTICIONAR UNA VISTA MATERIALIZADA

Debido al volumen grande de datos mantenidos en DW particionar es una opción de utilidad que permite usar el diseño de BD. La partición de tablas hechas mejora la escalabilidad, simplifica sistemas de administración y permite la posibilidad de definir índices locales que pueden reconstruirse eficientemente.

La partición de una vista materializada también tiene beneficios para actualizar, desde el procedimiento actual puede usar un DML paralelo para mantener la vista materializada. Para realizar estos beneficios, la vista materializada debe estar definida como `PARALLEL` y el DML paralelo debe permitirse en la sesión.

SELECCIÓN DE INDEXACIÓN PARA VISTAS MATERIALIZADAS

Las dos principales operaciones en una vista materializada son la ejecución de la consulta y la actualización incremental, y cada operación tiene diferentes requerimientos de operación.

La ejecución de la consulta puede necesitar acceder a cualquier subconjunto de la vista materializada en columnas clave, y necesitar la unión y agregación sobre un subconjunto de estas columnas.

Consecuentemente, la ejecución de la consulta usualmente se realiza mejor si hay una columna simple de un índice bitmap definido en cada vista materializada de la columna clave.

En este caso la vista materializada contiene solo uniones usando la opción rápida de actualización, es altamente recomendado que los sub-índices se creen en las columnas que contienen la filas ID's para mejorar el funcionamiento de la operación actualizada.

Si la vista materializada usa uniones y agregaciones son rápidamente actualizables, entonces un índice es automáticamente creado y no puede ser desactivado.

INVALIDAR UNA VISTA MATERIALIZADA

Las dependencias relacionadas para una vista materializada son mantenidas para asegurar la operación correcta. Cuando una vista materializada es creada, la vista materializada depende de los detalles de las tablas referenciadas en esta definición. Cualquier operación DDL²⁷, tales como DROP o ALTER en cualquier dependencia de la vista materializada causarían una invalidación. Una vista materializada es automáticamente revalidada cuando esta es referenciada. En muchos casos la vista materializada será exitosa y transparentemente revalidada.

PAUTAS PARA USAR VISTAS MATERIALIZADAS EN UN DW

Determinar que vistas materializadas deberían ser mas benéficas para la ejecución por los analistas se usan las herramientas del paquete DBMS_OLAP (DBMS²⁸). Específicamente, se puede llamar al procedimiento DBMS_OLAP.RECOMMEND_MV para ver una lista de vistas materializadas que Oracle recomienda, basadas en la estadística y el uso de una BD.

ALTERAR UNA VISTA MATERIALIZADA

Hay cinco enmiendas que pueden ser hechas en una vista materializada:

- Cambios en opciones de actualización (FAST/FORCE/COMPLETE/NEVER).
- Cambios en un modo de actualización (ON COMMIT/ ON DEMAND).
- Recompilación.

- Permitir/no permitir se usa para reescribir consultas.
- Considerar actualización.

Todos los cambios son logrados por borrar y después recrear la vista materializada.

La cláusula `COMPILE` de la declaración `ALTER MATERIALIZED VIEW` puede ser usada cuando la vista materializada ha sido invalidada.

El proceso de la compilación es rápida y permite la vista materializada ser usada por consultas rescritas otra vez.

BORRAR UNA VISTA MATERIALIZADA

Utilizar la declaración `DROP MATERIALIZED VIEW` para borrar vistas materializadas.

Por ejemplo: `DROP MATERIALIZED VIEW sales_sum_mv;`

Este comando borra la vista materializada `sales_sum_mv`.

Si la vista materializada fue preconstruida en una tabla, entonces la tabla no es borrada pero no es tan larga para ser mantenida con el mecanismo de actualización, alternativamente se puede borrar una vista materializada usando Oracle Enterprise Manager²⁹.

DIMENSIONES

DEFINICIÓN DE DIMENSIÓN

Una dimensión es una estructura que categoriza datos en orden para permitir a usuarios finales responder cuestiones de negocios. Comúnmente las dimensiones usadas son clientes, productos y tiempo.

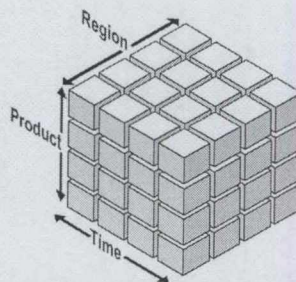


Fig. 20.1.1 Ejemplo de una Dimensión de Cubo.

Las dimensiones no tienen que ser definidas pero se gasta mucho tiempo crearlas, pueden producir muchos beneficios porque ayudan a realizar las consultas rescritas más completas.

Son obligatoria si se usa el consejero de resúmenes para recomendar cuales vistas materializadas son creadas, borradas o conservadas.

Los valores de la dimensión son usualmente usados dentro de jerarquías, van de un nivel en la jerarquía llamadas rolling up de datos a un nivel de jerarquía menor y es llamada drilling down de datos.

CREAR UNA DIMENSIÓN

Antes de que se pueda crear una dimensión las tablas deberán existir en la BD las cuales contienen la dimensión de los datos. Se crea una dimensión usando la declaración CREATE DIMENSION o Dimensión Wizard en Oracle Enterprise Manager. Dentro de la declaración CREATE DIMENSION, usar la cláusula LEVEL...IS para identificar los nombres de los niveles de la dimensión.

La localización de la dimensión contiene una jerarquía simple, con flechas dibujadas del nivel hijo al nivel padre. En lo más alto de la gráfica de dimensión el nivel especial ALL, que representa la agregación sobre todas las filas. Cada flecha en la gráfica indica que por cualquier hijo hay uno y solo un padre.

Una integridad jerárquica es necesaria para la operación correcta de las funciones administradas por las vistas materializadas que incluyen agregaciones.

Cada nivel en la dimensión debe corresponder a uno o más columnas en la tabla en la BD.

DIMENSIÓN WIZARD

La dimensión wizard es automáticamente invocada siempre y cuando una petición es hecha para crear un objeto de la dimensión en Oracle Enterprise Manager.

El usuario es entonces guiado paso a paso a través de la información requerida por una dimensión. Una dimensión creada vía wizard podría contener cualquiera de los atributos descritos en la "Creación de una dimensión" tales como uniones claves, múltiples jerarquías y atributos. Algunos usuarios prefieren usar el wizard debido a que es gráficamente guiado en la relación jerárquica como se va construyendo a dimensión.

VISTAS DE DIMENSIONES

Las dimensiones pueden ser vistas a través de uno o dos métodos:

- Utilizar el paquete DEMO_DIM.
- Utilizar Oracle Enterprise Manager.

Utilizar el DEMO_DIM Package

Dos procedimientos son disponibles para permitir desplegar las dimensiones que van a ser definidas:

- Para imprimir una dimensión específica DEMO_DIM.PINT_DIM.
- Para imprimir todas las dimensiones DEMO_DIM.PRINT_ALLDIMS.

Utilizar Oracle Enterprise Manager

Todas las dimensiones que existen en un DW pueden ser fácilmente vistas usando Oracle Enterprise Manager. Seleccionar el objeto de la dimensión desde dentro del icono del esquema mostrará todas las dimensiones. Seleccionar una dimensión específica mostrará gráficamente la jerarquía, los niveles y los atributos que han sido definidas.

DIMENSIONES Y CONSTRAINTS

Los constraints juegan un importante role con dimensiones. En la mayoría de los casos, la completa integridad referencial es hecha en la operación de la BD, y procedimientos operacionales pueden ser usados para asegurar que los datos fluyan dentro del DW, que nunca violan la integridad referencial; así que en la práctica los constraints en la integridad referencial pueden o no pueden estar activados en el DW.

VALIDAR UNA DIMENSIÓN

Si las relaciones descritas por las dimensiones son incorrectas, resultados incorrectos podrían ocurrir. Por lo tanto se debe verificar las relaciones específicas con CREATE DIMENSION usando el procedimiento DBMS_OLAP.VALIDATE_DIMENSION periódicamente.

Este procedimiento es fácil de usar, solo contiene cuatro parámetros:

- Nombre de la dimensión.
- Nombre del propietario.
- Colocar TRUE para checar solo las nuevas filas de las tablas en esta dimensión.
- Colocar TRUE para verificar que todas las columnas no sean nulas.

ALTERAR UNA DIMENSIÓN

Algunas modificaciones pueden ser hechas hacia la dimensión usando la declaración. Se puede añadir o borrar un nivel, jerarquía o atributo de la dimensión usando este comando.

BORRAR UNA DIMENSIÓN

La dimensión es removida o borrada usando el comando DROP DIMENSION.

Ejemplo: DROP DIMENSION time_dim;

Las dimensiones pueden ser borradas usando Oracle Enterprise Manager.

TABLESPACES TRANSPORTABLES

Oracle 8i introduce un importante mecanismo para transportar datos: Tablespaces transportables. Esta característica es el mecanismo más rápido para mover volúmenes grandes de datos entre dos BD Oracle. Estos mecanismos requieren que los datos sean recargados o exportados dentro de archivos de la BD fuente y, transportar estos archivos para después cargarlos o importarlos dentro de una BD fuente. Los tablespaces transportables son el puente de carga y descarga.

Usar tablespace transportables, en archivos de datos en Oracle pueden ser directamente transportados desde una BD a otra. Es parecido a importar y exportar, este mecanismo de transportar tablespace provee un mecanismo para transportar metadatos y datos de una manera rápida.

La aplicación más común de tablespace transportables en DW son el mover datos de una BD estacionada a un DW o mover datos de un DW a un Data Mart.

Un tablespace no puede ser transportado a menos de que no existe una transacción activa que modifique el tablespace.

ROLLUP

El Rollup permite a una declaración SELECT calcular múltiples niveles de sub-totales a través de un grupo específico de dimensiones. Esto también calcula totales. El Rollup es una extensión simple de la cláusula GROUP BY, así que la sintaxis es extremadamente fácil de usar. La extensión del Rollup es la más eficiente para añadir mínimas consultas. Ejemplo, la sintaxis sería de la siguiente forma:

```
SELECT... GROUP BY ROLLUP (grouping_column_reference_list).
```

Utilizar la extensión del Rollup en tareas que envuelven sub-totales, es muy útil para subtotalizar a lo largo de dimensiones jerárquicas tales como el tiempo o geografía, para los administradores del DW es muy útil utilizar tablas resumidas pero no utilizar vistas materializadas. El Rollup podría simplificar y acelerar el mantenimiento de tabla resumidas.

CUBE

La creación de sub-totales por Rollup representan solo una fracción de las posibles combinaciones del subtotal. El Cube activa una declaración SELECT para generar todas las posibles combinaciones de un grupo de dimensiones. Esto también calcula un total. Esto es un sistema de información típico necesario para todos los reportes cross-tabular, así que el cubo puede calcular reportes de cross-tabular con una simple declaración SELECT como el Rollup, el Cube es una extensión simple de la cláusula GROUP BY, la sintaxis es fácil.

El Cube aparece en la cláusula GROUP BY en la declaración SELECT por ejemplo:

```
SELECT ... GROUP BY  
CUBE (grouping_column_reference_list).
```

Utilizar Cube en cualquier situación requerida para reportes cross-tabular. La necesidad de datos para reportes cross-tabular pueden generarse con una cláusula SELECT usando Cube. Tanto el Rollup como el Cube pueden ser útiles en la generación de tablas resumidas. También el Cube es especialmente valioso en consultas que usan columnas de múltiples dimensiones mejor que columnas representadas en diferentes niveles de una dimensión simple.

Parte III. DATA WAREHOUSE PARA MÁS DATOS Y MÁS USUARIO

DW es la tecnología que facilita la decisión de hacer negocios. Los avances a través del tiempo propulsados por Data warehousing³⁰, que es el componente medular de cada organización en cada industria. Factores, tales como el incremento en la competencia, las oportunidades globales y las irregularidades gubernamentales han manejado la demandad de almacenamiento. Las organizaciones buscan como explotar la información como un arma estratégica para tomar mejores decisiones en los negocios. Oracle ha conocido las necesidades de estas organizaciones a través de un incremento significativo en el corazón del servidor de BD, Oracle 7 por medio de Oracle 8, es la plataforma más popular de Data warehousing.

RÁPIDO Y SIMPLE PARA MAYORES DATOS Y MÁS USUARIOS

El corazón de cada aplicación que soporta las decisiones de los negocios (para la empresa, DW es un mercado específico de datos) es el sistema de administración de la BD. Los requerimientos básicos de un servidor DW son:

- El DW debe ser rápido: los usuarios de un DW deben tener la disponibilidad de encontrar rápidamente la información en el warehouse para conocer las necesidades de su negocio. Estas necesidades pueden variar de simples búsquedas o consultas (Querys³¹) de reportes que están terminadas en segundos, hasta búsquedas analíticas complejas de datos que son terminados en horas. En suma, el mismo servidor DW debe tener la habilidad de cambiar eficientemente en este universo de búsquedas.
- El manejo del DW debe ser simple: el servidor DW debe proveer un mecanismo de uso fácil y completo para cargarse, indexando y sumalizando nuevos datos. Respaldar y recuperar operaciones que son escalabres, simples y libres de error. El DW debe tener la habilidad de proveer búsquedas óptimas cambiando con un tono mínimo, y debe tener la habilidad de manejar un gran número de usuarios garantizando que cada usuario reciba una parte apropiada de recursos.

Oracle 8i es una extensión de Oracle 8, la dirección tecnológica en Data warehousing. Para proveer el rango más ancho de capacidades de DW, Oracle 8i es único en la habilidad de soportar, cualquier sistema de toma de decisiones que ordena desde la construcción del mercado de datos hasta la más larga terabyte³² DW.

ORACLE 8i MÁS RÁPIDO PARA LOS USUARIOS

Un usuario de DW comúnmente no interactúa directamente con la BD relacionada. Utiliza una herramienta de SSD como lo es, Oracle DiscovererTM 33 que es usado comúnmente para recuperar los datos de la BD Relacional. Desde que el usuario accesa directamente en la BD, el único aspecto que éste puede observar del servidor de datos, es que tan rápido procesa la consulta. Por esta razón, los cambios en las consultas son frecuentemente considerados como el requerimiento más importante del servidor DW.

Oracle 8i direcciona este requerimiento para proveer a la industria el grupo más comprensivo de características de cambios en las consultas. Ésta extensión de las características de las consultas es especialmente importante para DW. En un ambiente de consulta, no hay una sola técnica que sea “La forma correcta” para ejecutar cualquier búsqueda. En vez de esto, Oracle busca proveer la técnica de procesamiento de búsqueda más apropiado para cada tipo de búsqueda.

Las características de los cambios de búsqueda de Oracle 8i caen en 6 categorías básicas:

- Índices (indexes) y métodos join³⁴
- Vistas materializadas (materialized views) y tablas resumidas (summary tables)
- Particionismo (Partitioning)
- Paralelismo (Parallelism)
- Capacidades analíticas
- Optimización de búsquedas

Las características de los cambios de búsqueda de Oracle 8i son integradas estrictamente, tanto que una sola búsqueda puede usar cualquier combinación de características de cambios de búsqueda de Oracle. Una búsqueda típica utiliza características de 3 ó 4 de éstas categorías.

ÍNDICES Y MÉTODOS JOIN

Uno de los incrementos más significativos de los cambios de búsqueda a sido la adición de nuevos índices y técnicas join que direccionan específicamente las necesidades del almacenamiento.

BITMAP INDEXES (Oracle 7 versión 7.3)

Tal vez es la técnica de indexado más importante para Data warehousing, los bitmap indexes podrían ser mucho más comunes que los b-tree indexes en un DW. La razón de la popularidad de estos bitmap indexes es que, estos índices tienen mucho mejor juego para búsquedas típicas de DW. Los bitmap indexes proveen un grupo base de manipulación de datos. Por ejemplo, un bitmap indexes provee un mecanismo extremadamente rápido para identificar el grupo de clientes de una compañía que tiene más de 40 años de edad, y combinar este resultado con el grupo de todos los clientes que viven en la región oeste.

Para la construcción de bitmap indexes en columnas especificadas, un usuario DW tiene la capacidad de hacer consultas eficientes en millones o incluso en billones de registros.

BITMAP INDEXES DINÁMICOS (ÍNDICES DINÁMICOS DE MAPA DE BITS)

Un bitmap indexes dinámico, provee la habilidad de convertir dinámicamente la salida de un b-tree index en una representación de mapa de bits. Dentro de Oracle 8i, esto es seguido de una búsqueda simple para recibir los beneficios del procesamiento de grupo base, mientras se accedan ambos, el b-tree y el bitmap indexes.

BITMAP STAR JOIN (UNIÓN DE ESTRELLAS DE MAPA DE BITS)

Oracle 8i provee el procesamiento de almacenamiento con el bitmp star join. Esta técnica de innovación de unir libera los bitmap indexes y une efectivamente múltiples tablas dimensionadas al Fact-table en un solo paso. Los clientes de Oracle 8 utilizan esta característica extensamente dominante.

FUNCTION-BASED (FUNCIONES BÁSICAS DE LOS ÍNDICES) (Oracle 8i)

Mientras las versiones anteriores de Oracle 8 solamente permitían que los índices fueran creados en una o más columnas, los índices Function-based le permiten a un administrados warehouse construir un índice en una función.

Un uso común de los índices Function-based es la creación de índices case, los cuales pueden ser implementados con la creación de un índice de la función case aplicada a un carácter en una columna. Sin embargo, los índices Function-based pueden ser usados para muchos propósitos, específicamente para los requerimientos de un DW. Por ejemplo, los índices Function-based también proveen las medidas para el indexado de aplicaciones específicas. Una tabla puede contener columnas para costos y para ingresos, pero los usuarios frecuentemente pueden consultar o buscar en ingreso-costo (que es el beneficio). Un índice Function-based puede ser construido en la función ingreso-costo, lo cual mejoraría enormemente el cambio de la consulta en la función creada.

Un índice Function-based puede ser usado para soportar de forma más eficiente cambios de consulta de otros escenarios típicos de DW. Por ejemplo los usuarios de un DW pueden querer analizar como afecta la edad de sus clientes en su patrón de compras. En vez de buscar en edades específicas, los usuarios de un Data warehousing hacen búsquedas en categoría de edades, tales como "10-19", "20-29", "30-39" etc. Un índice Function-based puede ser creado para esta función de categoría de edades, lo cual permite a los usuarios buscar directamente en cada categoría.

Otra aplicación particular de uso de los índices Function-based es proveer una forma simple de comprensión de datos. En algunas aplicaciones DW una tabla grande, puede contener muchos atributos pequeños, por ejemplo un registro detallado de cada llamada en un warehouse de telecomunicaciones puede tener cientos de atributos. Muchos de estos atributos son pequeños (1 ó 2 caracteres largos, almacenados como un campo codificado). En algunos casos, un espacio importante puede ser salvado al hacer un grupo de estos pequeños atributos, y comprimiéndolo y/o concatenándolo en una sola columna dentro de la BD. Previamente, el problema asociado con esta técnica fue que no había una forma eficiente de consultar esta columna multi-atributos. El índice Function-based soluciona esta dificultad. Un índice Function-based puede ser construido en cada atributo de un campo comprimido. Las consultas fundadas en estos campos comprimidos están disponibles para utilizarse en el índice Function-based por lo tanto, los cambios en la búsqueda no se ven afectados por la compresión.

FAST FULL INDEX SCAN (Oracle 7 versión 7.3)

Oracle provee previamente mecanismos rápidos para scanear una tabla. Los mecanismos incluyen la capacidad de leer múltiples bloques de la BD en una sola operación de Entrada/Salida (lecturas de múltiples bloques), y esto evita el “buffer cache”³⁵ de Oracle para eficientar. Desde la versión 7.3, este mecanismo rápido de E/S ha sido extendido a los índices Scan. El índice Fast Full Scan son usados particularmente cuando un solo índice contiene una columna de la tabla que ha sido consultada. En vez de scanear la tabla, Oracle 8i simplifica el índice Scan ya que no hay almacenamiento temporal, el mecanismo de E/S no almacena nada temporal por eso es más eficiente. Este puede proveer cambios considerablemente más rápidos, porque el índice es mucho más pequeño que la tabla.

INDEX JOIN

También son conocidos como “índices cubiertos”, esta técnica posibilita a Oracle 8i para juntar múltiples índices. Esta característica complementa al Fast Full Index Scan. Mientras el Fast Full Index Scan es utilizado cuando un solo índice contiene una columna referenciada en una consulta, el Index Join puede ser utilizado cuando 2 o más índices contienen una columna referenciada. Estos índices pueden ser unidos para evaluar la consulta, mientras se accesa la tabla, al igual que el Fast Full Index Scan, el Index Join aumenta los cambios porque los índices son típicamente más pequeños que la tabla.

HASH JOIN (Oracle versión 7.3)

Hash join fue introducido específicamente para soportar búsquedas de decisiones complejas. Hash join provee un mecanismo escalable para unir tablas grandes e idealmente permite ejecuciones paralelas.

PARTITION-WISE JOINS (UNIONES PARTITION-WISE) (Oracle 8i)

Oracle 8i introduce las uniones “Partition-Wise”, un significativo mejoramiento para las tablas de unión que son particionadas idénticamente dentro de las llaves unidas (particionadas-equitativamente), la optimización de Oracle considera nuevos métodos de unión que influyen estas características de partición, y que pueden proveer significativos cambios en su desarrollo.

Por ejemplo, en la figura de abajo, los lineitems y los orders en la tabla están equi-particionados. Para conocer que las dos tablas están equi-particionadas, Oracle 8i solamente necesita unir la partición 1 de orders a la partición 1 de lineitems, y la partición 2 de orders a la partición 2 de lineitems, y así sucesivamente. Notablemente, las características de partición de la tabla garantizan que las filas de partición 1 de orders no se una a ninguna fila de partición 2 o partición 3 de lineitems. De esta manera las uniones Partition-Wise proveen los cambios para evitar las uniones de pares de particiones innecesarias Partition-Pair Joins.

Partition-Wise Joins puede ser efectivas en ambas búsquedas, seriales y paralelas, y son range, hash y composite partitioning schemes³⁶. Las uniones Partition-Wise también son particularmente efectivas en el ambiente del servidor paralelo de Oracle. Las uniones Partition-Wise no solamente reducen el tiempo de procesamiento de las uniones, también pueden reducir significativamente el volumen del tráfico de interconexiones en este ambiente.

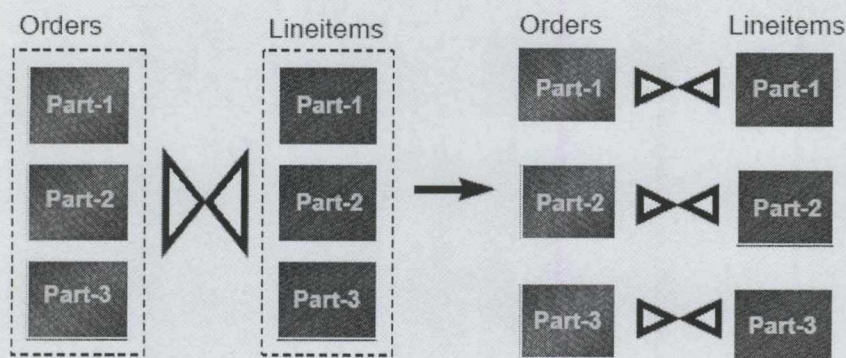


Fig. 25.9.1 Patition-wise join.

VISTAS MATERIALIZADAS Y MANEJO DE RESÚMENES

Oracle 8i introduce una facilidad completa para crear, mantener y usar tablas de resumen.

Esta facilidad es conocida como “vistas materializadas” porque las capacidades de Oracle se extienden más allá de los requerimientos de manejo de las tablas de resumen.

¿Qué es una tabla de resumen?

Muchos DW y data marts basados en Oracle 8, versión 8.0 (o versiones anteriores), ya utilizan tablas de resumen. Una tabla de resumen, la cual es también llamada tablas de agregación, almacén de resultados precalculados o datos agregados, (tales como sumas, conteos y promedios). Estas tablas son construidas para incrementar los cambios en las búsquedas en un DW. Esto es mucho más eficiente para acceder a un resultado pre-agregado, en vez de calcular un dato agregado cada vez que una búsqueda es ejecutada. Considere que un DW contiene información detallada de las ventas. El administrador warehouse puede elegir la construcción de una tabla de resumen que contenga las ventas totales por región-mes. Cuando un usuario requiere las ventas totales de la región, esta para Q1, Oracle 8i no calcula las ventas totales de una transacción individual de registros. En vez de eso, Oracle 8i recupera los datos pre-agregados de la tabla de ventas por región-mes. El uso de una tabla de resumen es completamente transparente para el usuario final. Oracle 8i automáticamente recupera una solicitud de SQL para usar una tabla de resumen. Para usar la tabla de resumen, los cambios en la búsqueda pueden ser desarrollados por orden de magnitud.

ORACLE 8I MANEJO DE RESÚMENES

Oracle provee una solución completa para el manejo de tablas de resumen. Las características del manejo de resúmenes en Oracle 8i, incluyen 3 componentes principales:

- Capacidad de reescribir búsquedas
- Mecanismos para mantener las tablas de resumen, incluyendo incrementos eficientes en las actualizaciones.
- Capacidades de asesoría que ayudan al administrador warehouse a crear y borrar resúmenes, basados en su uso.

La característica primaria es la capacidad de reescribir búsquedas.

REESCRITURA DE RESÚMENES

Como parte de la optimización de las búsquedas, cada búsqueda es analizada y, de forma apropiada, puede ser reescrita para usar una tabla de resumen. Reescribir resúmenes ha estado disponible por medio de otros productos de software tales como Oracle DiscovererTM.

Sin embargo, el servidor base reescribe resúmenes extendiendo las ventajas de las tablas de resumen, al proveer la capacidad de reescribir resúmenes para todas las búsquedas SQL que son emitidas de cualquier ambiente del usuario final.

Para integrar la reescritura de resúmenes en Oracle8i, las tablas de resumen llegan a ser similares a los índices en muchos aspectos:

- El propósito de una tabla de resumen es incrementar los cambios en las búsquedas.
- La existencia de una tabla de resumen es una aplicación transparente para SQL. Un administrador puede crear, modificar o eliminar tablas de resumen en cualquier momento sin afectar la exactitud de las aplicaciones SQL.
- Una tabla de resumen consume espacio de almacenamiento, y el contenido de la tabla de resumen debe mantenerse cuando los detalles de la tabla son modificados.

Justo como el administrador warehouse crea índices de rutina en sus esquemas, el administrador warehouse Oracle 8i usa manejo de resúmenes en todos los DW.

Las capacidades de reescribir resúmenes de Oracle 8i son extremadamente poderosos. Oracle 8i maneja definiciones de dimensiones y jerarquías y utiliza estos metadatos para incrementar las ventajas de reescribir las búsquedas. Oracle 8i reconoce, por ejemplo, si un usuario final requiere un reporte trimestral, y la tabla de resumen mensual puede ser usada para evaluar la búsqueda.

Más que solo tablas de resumen

El manejo de resúmenes es solo un subgrupo de una característica muy grande llamada "vistas materializadas". Esto también provee una estructura general para todas las formas locales de replicación de datos dentro de una BD de Oracle 8i. Solo como resúmenes se puede proveer incrementos en los cambios para las búsquedas agregadas, ciertamente las vistas materializadas pueden proveer cambios significativos en el desarrollo de las búsquedas para uniones no agregadas, y para otro tipo de búsquedas. De esta forma, las vistas materializadas proveen una capacidad también conocida como Indexes Join.

PARTICIONANDO

La partición también puede mejorar los cambios de la búsqueda para muchas búsquedas. Oracle 8 introdujo “el rango de particiones”. Una tabla dada puede ser entendida particionandola desde el principio en rangos de valores de la “columna de partición”. Por ejemplo, una tabla con una columna de fecha puede ser particionada por mes. Cada partición de la tabla debe contener toda la información de la línea que corresponde al mes.

Oracle 8i introduce dos nuevas técnicas de partición: hash y particiones compuestas (composite partitioning, una combinación de hash y particiones de rango).

Una forma intuitiva en la cual la partición mejora los cambios es a través del salto de particiones “Partition Skipping”³⁷. Suponga que una tabla de DW contiene un año de datos históricos y está particionada en rangos mensuales. Si una búsqueda solamente requiere datos para marzo, abril y mayo, entonces Oracle 8i solamente accederá las particiones de marzo, abril y mayo. Al saltar particiones puede proveer cambios del 75% de mejora para esta búsqueda, porque la BD solamente accesa a 3 de 12 particiones. Además, al igual que todas las otras características de búsqueda, el salto de particiones puede ser combinada con cualquier otra técnica de procesamiento de búsqueda de Oracle. Por ejemplo, las particiones de marzo, abril y mayo pueden ser accesadas por cualquier tipo de técnica de indexado, cualquier tipo de técnica de unión. Otra característica nueva de los cambios de búsqueda introducida en Oracle 8i es Partition-Wise Joins. Igual que Partition Skipping, Partition-Wise Join puede proveer un orden de magnitud en el desarrollo de los cambios para muchas búsquedas.

PARALELISMO

Búsquedas paralelas es la habilidad de aplicar múltiples fuentes de CPU y de Entrada/salida para la ejecución de una sola búsqueda SQL. La tecnología de búsqueda paralela de Oracle fue introducida en Oracle 7, versión 7.1 y está diseñada para llevar a cabo cambios altamente escalables en todo los hardware de arquitectura paralela: SMP, NUMA, CLUSTERS, MPP. La arquitectura de las búsquedas paralela de Oracle es única en está habilidad o en la habilidad de determinar dinámicamente el grado de paralelismo de cada búsqueda.

A diferencia de otras arquitecturas de BD, en las cuales el grado de paralelismo únicamente es determinado por el esquema de partición de las tablas fundamentales, las búsquedas paralelas dentro de Oracle están determinadas inteligentemente, basadas en el tamaño de las tablas, el número de CPU's, el número de archivos al ser accedados, y otras variables.

Oracle 8i promueve el incremento de las búsquedas paralelas con la introducción de un algoritmo adaptativo para determinar el grado necesario de paralelismo. Este algoritmo avanzado considera la carga actual en el DW cuando se elige el grado de paralelismo. Esta característica desarrolla el rendimiento de procesamiento de grandes DW ejecutando búsquedas paralelas concurrentes.

Por ejemplo, suponga que un usuario emita una búsqueda en un DW cuando ningún otro usuario está conectado. Oracle 8i puede elegir ejecutar la búsqueda usando un grado de paralelismo 4. Si 9 usuarios adicionales están dentro del DW y requieren búsquedas, entonces Oracle 8i puede elegir un grado de paralelismo 4. Conforme la mayoría de los usuarios emiten nuevas búsquedas (o conforme los primeros usuarios completan sus búsquedas), el grado de paralelismo para las nuevas búsquedas requeridas puede incrementar o decrementar basado en el sistema cargado. A cualquier hora, el DW está usando todas las fuentes disponibles, pero no a cualquier hora hay muchos procesos paralelos como para saturar el DW.

El algoritmo adaptativo de Oracle 8i también direcciona inteligentemente ambientes Clusters y MPP.

Inicialmente, con 2 búsquedas ejecutándose en el DW, Oracle 8i puede elegir un grado de paralelismo 12, con el proceso de paralelo 12, se distribuyen equitativamente las cruces de todos los nodos del cluster. Conforme la mayoría de las búsquedas son requeridas, el grado de paralelismo será escalado hacia atrás (retrocediendo). En el caso de 6 búsquedas concurrentes, no solamente habrá un grado de paralelismo sino 4 grados de paralelismo para cada búsqueda, pero cada búsqueda estará localizada en 1 solo nodo del cluster. En este caso el grado de paralelismo es lo suficientemente pequeño para cada nodo como para manejar toda la búsqueda paralela del proceso asociado en una sola búsqueda.

Esta habilidad para localizar una búsqueda minimiza la relación internodos durante la ejecución de la búsqueda, y desarrolla cambios en la búsqueda.

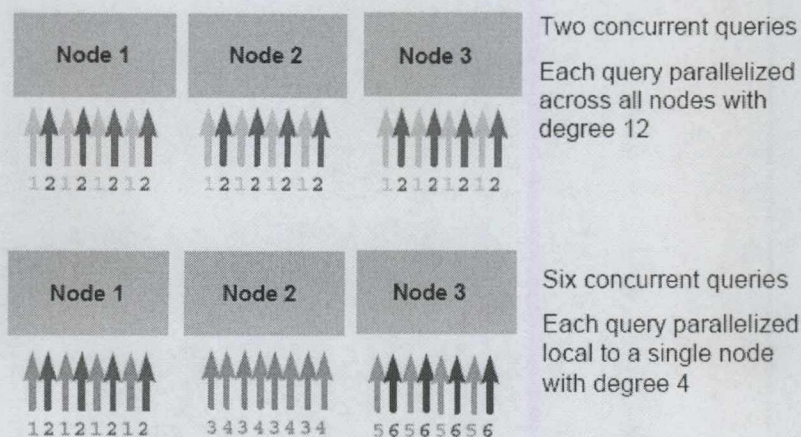


Fig. 25.14.1 Paralelismo.

Oracle es la única BD que provee una búsqueda paralela dinámica de arquitectura robusta con capacidades multinodos.

En resumen, los beneficios clave de la arquitectura de búsquedas paralelas del mundo real de Oracle son:

- El administrador de Oracle tiene la libertad de elegir el esquema de partición de los nodos para el mejor mapa o sus requerimientos de trabajo, en términos de disponibilidad y manejabilidad. La arquitectura paralela de Oracle reparte la escala de los cambios en la búsqueda, pero no está limitada por la partición de datos ni por la arquitectura del hardware multinodo.
- La utilización máxima de todas las fuentes de proceso disponibles. La ejecución dinámica del paralelo intrapartición de Oracle posibilita al servidor a extender la carga de trabajo a través de procesos múltiples o nodos aun en el caso de que una sola partición ha sido accedada.
- La fundación de una tecnología en común que libera los cambios, escalable y manejable en todos los hardware de arquitectura paralelo: SMP, NUMA, CLUSTERS, Sistema MPP.

CAPACIDADES ANALÍTICAS

Extendiendo las capacidades analíticas de los servidores de datos se pueden ofrecer muchos beneficios incluyendo:

- Programación simplificada, menos código SQL es necesario para realizar muchas tareas.
- Reducción de clientes procesando cargas, los cálculos analíticos son turnados a los servidores.
- Reduce el tráfico en la red, mucho menos niveles detallados de los datos es más barato para el cliente.
- Oportunidades para guardar agregaciones, búsquedas similares pueden influenciar el trabajo existente.

A parte de la programación simplificada, cada uno de los beneficios antes mencionados tiene el efecto neto de desarrollar los cambios de las búsquedas del DW.

CUBE Y ROLLUP

Los subtotales son una mezcla de muchas dimensiones que están en una parte esencial de los reportes que generan las herramientas de SSD. Oracle 8i provee nuevas capacidades significativas para el cálculo de subtotales con las operaciones Cube y Rollup de SQL.

Cube y Rollup son opciones nuevas para el operador GROUP BY de SQL que sigue una sola selección de cuentas para calcular niveles múltiples de subtotales. Por ejemplo, todos los datos que se muestran en la tabla cruzada de resultados pueden ser recuperados en una sola búsqueda SQL, usando la operación Cube.

1996			
Region	Department		
	Video Rental	Video Sales	Total
Central	82,000	85,000	167,000
East	101,000	137,000	238,000
West	96,000	97,000	193,000
Total	279,000	319,000	598,000

Tabla 25.16.1 Reporte Cross-Tab de resultados.

Sin el operador Cube, todos los subtotales adicionales (total por región, total por departamento, el gran total) son calculados en una sola búsqueda SQL, sin requerir cálculos del cliente y sin incurrir a la falta de múltiples búsquedas de SQL.

INCREMENTOS PARA BÚSQUEDAS EN BASE A N

Oracle 8i provee mejoras para otras búsquedas analíticas comunes, calcula en base a N de una lista de rangos. Una búsqueda típica puede ser “encontrar los 20 mejores vendedores en la región este”. En lugar de ordenar a todos los vendedores, Oracle solamente mantiene al día a los 20 mejores vendedores cuando procesa la búsqueda, esto provee un significativo desarrollo.

SAMPLING (MUESTREO)

Oracle 8i introduce un operador de muestreo. Una búsqueda SQL estará disponible para recuperar una muestra aleatoria de filas de una tabla. Esta característica beneficia primariamente las herramientas estadísticas y las aplicaciones de datos incompletos de aplicaciones que siempre requieren muestreo y que cuidadosamente utilizan el muestreo para proveer resultados estadísticos válidos.

OPTIMIZACIÓN DE LA BÚSQUEDA

La optimización de la búsqueda es frecuentemente descrita como el “cerebro” de la ingeniería del procesamiento de búsquedas. El optimizador tiene la responsabilidad de determinar la técnica de procesamiento de búsqueda más eficiente para cada búsqueda, teniendo en cuenta:

- La característica de los datos al ser accedidos.
- La ejecución de la búsqueda dependiendo de las características del servidor.
- Las capacidades del hardware y su arquitectura.
- Las fuentes del sistema disponible.

La inversión de Oracle en un optimizador de búsquedas sofisticado para el almacenamiento inicia con la introducción del costo-base del optimizador de búsqueda en Oracle 7, versión 7.0 en 1992. El optimizador tiene desarrollados significativos en cada versión.

No solamente el optimizador de búsqueda ha sido extendido para soportar las nuevas características introducidas en Oracle 8, también ha llegado a incrementarse de forma mas inteligente en el procesamiento de tipos de búsqueda complejas que las herramientas que genera típicamente SSD.

El optimizador de búsqueda de Oracle 8i soporta un rango más ancho de funcionalidad de DW en la industria, y esta es la mayor ventaja disponible del optimizador. Algunas de las principales características del optimizador de búsquedas incluye:

- El uso de estadísticas avanzadas incluyendo histogramas.
- Provee la habilidad para reconocer y optimizar todas las búsquedas de inicio.
- El conocimiento de todas las características de búsqueda de Oracle 8i (tales como técnicas de indexación, particionamiento y paralelismo).
- Provee la habilidad de utilizar apropiadamente múltiples índices en una sola búsqueda.
- Sofisticada capacidad de reescribir búsquedas que soportan las vistas materializadas.
- Avanzadas transformaciones subquery, incluyendo semi-uniones y anti-uniones.
- Costo basado en la transformación de búsquedas.
- Capacidad de desarrollar vistas unidas.
- Optimización de búsquedas en lista.

ORACLE 8I SIMPLE PARA MÁS USUARIOS

Aunque los cambios de búsqueda pueden ser un requerimiento primario de una aplicación de toma de decisiones, es igualmente importante que las BD sea fácil de manejar y configurar, tanto que el usuario final puede realizar los máximos cambios en la búsqueda con un mínimo esfuerzo del administrador.

Oracle 8i provee un grupo robusto de características que mejoran de forma automática la administración para los usuarios de DW. Estas características incluyen:

- Asesor de resúmenes. (Summary advisor)
- Incrementa los índices wizard.
- Progreso del monitor.

- Manejabilidad en el desarrollo de búsquedas paralelas.
- Manejo de fuentes de la BD.
- Encontrar el acceso al control.

La importancia de estas características se basa en 2 puntos: Algunas características, tales como el asesor de resúmenes y el índice wizard buscan minimizar la administración de un DW o de un data-mart. Otras características, tales como el manejo de fuentes de la BD, provee la administración más fácil de cada una de las empresas más complejas de DW. Oracle 8i provee las características apropiadas de manejabilidad para direccional las aplicaciones enteras de almacenamiento.

ASESOR DE RESÚMENES

Un reto del administrador warehouse es que puede tener cientos o miles de tablas de resumen que pudieron ser creador de un esquema dado de warehouse. Oracle 8i conoce este reto al proveer sofisticadas capacidades de asesorías para las tablas de resumen. Basado en la carga de trabajo de búsqueda, este asesor wizard determina cual es el potencial que se puede agregar a las nuevas tablas de resumen. Esto también identifica cuales tablas de resumen existentes son sub-utilizadas y le suma un poco de valor. Las capacidades del asesor adicionalmente ponen los requerimientos de espacio de cada resumen en un contador, balanceando los beneficios potenciales de un resumen con los requerimientos de almacenamientos. La ventaja para el administrador warehouse es que solamente decide que administrador necesita para tener la cantidad de espacio en disco suficiente para los resúmenes. Usando estas metas el asesor de resumen puede recomendar el grupo más apropiado de tablas de resumen para un DW.

INCREMENTO DEL ÍNDICE WIZARD

Como el asesor de resúmenes hace recomendaciones de cuales resúmenes crear mejor, un índice wizard hace recomendaciones de cuales índices crear mejor. Aunque Oracle previamente provee un "Index Wizard" en el paquete de manejo de Oracle, el wizard Oracle 8i ha sido mejorado significativamente a través de las extensiones del optimizador de búsqueda. Por iniciar estrechamente integrados con el optimizador de búsqueda, el índice wizard puede proveer recomendaciones aún más exactas.

PROCESO DEL MONITOR

Para DW, ciertas operaciones, tales como búsquedas complejas, operaciones de recuperación de respaldos, construcción de índices, y otras operaciones DDL, pueden tomar mucho tiempo para ser completadas. Oracle 8i provee un progreso en el monitor, de tal forma que los usuarios y los administradores pueden ver el estatus de las operaciones grandes que están corriendo, y obtener información actual de su progreso. Oracle 8i mantiene estadísticas describiendo el “porcentaje de completado” de estas operaciones. Una interfase gráfica en el manejador de empresas de Oracle posibilita al administrador a ver desplegada una barra gráfica del “porcentaje de completado” de esa operación. Además, cualquier otra herramienta o cualquier administrador de BD puede también recuperar información en progreso directamente del servidor de datos Oracle 8i, vía sistema de vistas.

MANEJABILIDAD EN EL DESARROLLO DE BÚSQUEDAS PARALELAS

Oracle 8i simplifica la configuración de las búsquedas paralelas. Muchos de los parámetros de inicialización previa relacionados a las búsquedas paralelas han sido reemplazados por un pequeño número de parámetros muy intuitivos. El administrador warehouse necesita solamente especificar si la búsqueda será o no ejecutada en paralelo, y el número deseado de paralelos procesados que serían producidos en cualquier CPU dados.

El nuevo algoritmo adaptativo de Oracle 8i para determinar el grado de paralelismo (descrito previamente) también tiene el efecto de simplificar la manejabilidad de la búsqueda paralela. Oracle 8i automáticamente maneja el proceso de paralelos de forma tal que cada búsqueda recibe la cantidad apropiada de paralelismo, lo que hace que la plataforma del warehouse es maximizada.

Con esta mejora, cualquier administrador warehouse será capaz de tener las poderosas ventajas de las búsquedas paralelas, sin tener que entender los detalles del ambiente de las búsquedas paralelas.

MANEJO DE FUENTES DE LA BD

El DRM³⁸ provee un mecanismo para localizar las fuentes de un DW entre los grupos diferentes de usuarios finales.

Por ejemplo, suponga que el departamento de compras y el departamento de ventas comparten un DW. Usando el DRM, un administrador warehouse puede especificar que el departamento de compras reciba al menos el 60% de las fuentes de CPU de las máquinas, y que el departamento de ventas reciba el 40% de las fuentes de CPU. El administrador warehouse puede especificar los límites en el número total de sesiones activas, y el grado de paralelismo de búsquedas individuales para cada departamento.

El DRM provee un mecanismo muy flexible para localizar las fuentes de CPU. El ejemplo previo tiene un esquema de localización muy simple, en el cual, las fuentes de CPU fueron divididas entre los dos departamentos. Sin embargo, un administrador puede crear un esquema de localización múltiple. En este caso, las fuentes son localizadas inicialmente para los usuarios de "nivel uno" y las fuentes restantes se localizan para los usuarios del "nivel dos" y así sucesivamente.

Por ejemplo, un administrador puede localizar al 100% de los CPU para un "CASE CEO" de nivel uno. El nivel dos puede ser dividido 60/40 entre el departamento de compras y el departamento de ventas. Este esquema de localización puede seguirlo el usuario CEO para utilizar todas las fuentes en el DW. Las fuentes restantes pueden estar localizadas entre el grupo de compras y el de ventas. Esto sería un esquema típico de localización para un ambiente en el cual el CEO raramente requiere un reporte, pero los requerimientos del CEO siempre tienen prioridades muy altas.

Un administrador puede elegir también diferentes esquemas de localización para las diferentes horas del día. Una estrategia típica es continuar para la cantidad predominante de fuentes a ser localizadas para los usuarios en línea durante el día.

El DRM garantiza que las fuentes en un DW sean localizadas entre los usuarios. En lugar de requerir un administrador para los cambios en el monitor de la plataforma DW, el DRM puede reforzar proactivamente las políticas de utilización de fuentes dentro de DW.

FINE-GRAINED ACCESS CONTROL

El acceso a los datos y la seguridad son consideraciones importantes en un DW. Desde que un DW contiene información de múltiples organizaciones, no cualquier usuario DW puede tener acceso a todos los datos. Usando roles en la BD, un administrador warehouse ha sido capaz de especificar cuales usuarios accedan a tablas específicas.

En Oracle 8i, el acceso al control es promovido a través de un incremento en la seguridad del fine-grained. Cualquier tabla o vista puede ser asignada a una política de seguridad, una función que determina que línea en una tabla o vista dada puede acceder un usuario. Siempre que una búsqueda es emitida en una tabla, la política de seguridad aplicada. Por ejemplo, una simple política de seguridad puede hacer que los miembros de cada región de ventas solamente vean información de su región específica, aunque los usuarios cooperativos pueden ver toda la información de ventas. Las políticas están implementadas para descubrir dinámicamente las búsquedas entrantes. Para las bases de un nuevo anexo (en la cláusula where) para los requerimientos del usuario del SQL. La emisión de una búsqueda de un miembro de la región oeste de ventas puede tener la base "Región de ventas- oeste" 'Sales.Region='West'' anexo a su búsqueda. Para reforzar las políticas de seguridad en esta forma, estas políticas podrán tener un efecto mínimo en los cambios de las búsquedas, desde que las políticas base pueden ser optimizadas con los índices y con otras técnicas, es mejor tener cualquier base SQL.

Con la implementación de un acceso fine-grained en el servidor de datos, un administrador warehouse puede asegurar que las reglas de seguridad son reforzadas constantemente, a pesar de cómo el DW es accesado. Esta poderosa característica de seguridad facilita la administración de warehouse que contienen datos importantes, por permitir al administrador implementar las reglas de seguridad en un solo lugar, de una forma manejable.

ORACLE 8I: SIMPLE PARA MÁS DATOS

El DW típicamente almacena grandes cantidades de datos. Esta es una medida en que los servidores DW tienen la capacidad de hacer más eficiente el manejo de grandes cantidades de datos.

Oracle 8, versión 8.0, introdujo un host³⁹ de características que soportan grandes BD. Esto provee un enorme incremento en la escalabilidad para los DW de Oracle. Esta característica, es promovida en la versión 8i donde un administrador de DW de cualquier tamaño es mucho más simple.

PARTICIONAMIENTO

Particionamiento, es esencia, “divide-and-coquer” (divide y triunfaras) para el manejo de grandes BD. Las tablas y los índices pueden ser particionados en pequeñas unidades, tanto que un administrador de BD puede operar en estos pequeños objetos. Sin embargo, el particionamiento mantiene completa la transparencia de las herramientas del usuario final, de esta forma, una tabla particionada, puede ser buscada de la misma maneja que una tabla no particionada.

Esto aparentemente simplifica el concepto que beneficia de forma significativa el manejo en los datos:

- Cada partición puede ser almacenada en un espacio de tabla (tablespace) separado, consistente en un grupo distinto de almacenamiento de archivos de datos. Por lo tanto, el particionamiento provee un mecanismo simple para la distribución de datos en múltiples discos. Adicionalmente, la separación lógica y física de particiones tiene la capacidad de dar continuidad a los datos en forma viable en la fase de características parciales, desde que la perdida de un drive (unidad) solamente afecta la viabilidad de una sola partición o de un sub-grupo de particiones, y no a la tabla completa.
- Con el particionamiento, las operaciones del manejo de datos, tales como cargar los datos y construir índices, pueden ser combinadas al nivel de partición, lo opuesto a la tabla. Como resultado, esta tarea puede ser ahora combinada eficientemente dentro de una pequeña ventana operacional de mantenimiento, independientemente del tamaño actual de la tabla. La disponibilidad del resto de la tabla es mantenida aunque estas operaciones sean combinadas en un sub-grupo de particiones.
- El servidor de la BD está conciente del contenido de cada partición. Por lo tanto, este conocimiento puede ser aplicado efectivamente en el procesamiento de búsquedas para proveer un cambio escalable en la búsqueda, aún con el crecimiento substancial del volumen de los datos.

SOPORTE DEL ESQUEMA DE CARGADO DEL ROLLING-WINDOW

Las ventajas del particionamiento son más claras en el escenario de carga en el DW.

Suponga que un DW es cargado cada semana, y que la mayoría de los datos son cargados en una sola tabla (por ejemplo, la tabla en el esquema de inicio). En este ambiente, el administrador warehouse equitativamente elegirá el rango de partición de la tabla, de forma de que cada partición contenga una semana de los datos.

Este esquema de particionamiento simplifica enormemente la carga del DW: los nuevos datos son cargados en una sola partición, los cuales pueden ser indexados y respaldados. Entonces, las particiones nuevas pueden ser sumadas (o adheridas, agregadas) a la tabla particionada.

El beneficio de esta característica o tecnología (Rolling-window) tiene dos puntos:

1. Los datos nuevos han sido cargados con la mínima utilización de fuentes. Los índices y respaldos solamente necesitan ser creados o mantenidos por una sola partición, en lugar de por una tabla completa.
2. Los datos nuevos han sido cargados con el mínimo esfuerzo del usuario final. Cuando un índice necesita ser borrado o indisponible, la búsqueda que se está ejecutando actualmente no puede ser interrumpida.

Este esquema de particionamiento puede ser altamente amigable para un DW que contiene una cantidad variada de datos históricos.

Por ejemplo, si este DW contiene las últimas 52 semanas de los datos, entonces una partición de 52 semanas podrá ser borrada tan fácilmente como la nueva partición es agregada. Esto no hace necesario modificar los datos o los índices en la tabla particionada. Los datos viejos pueden ser movidos eliminando solamente una partición. Esta técnica es conocida como un "Paradigma Rolling-Window" y es capacitada por el método de particionamiento de rangos de Oracle.

Esta técnica rolling-window es frecuentemente la consideración primaria cuando se elige un esquema de partición. La basta mayoría de diseños de DW para Oracle 8 usan este tipo de partición. El particionamiento en rangos provee ventajas más importantes para los DW que tiene que cargar particionamientos hash que para otras BD. Un esquema de particionamiento hash es incapaz de soportar eficientemente un paradigma rolling-windows. En lugar de cargar los datos en una sola partición, el particionamiento hash necesita que los datos sean cargados en todas las particiones y que, consecuentemente los índices y los respaldos deben ser mantenidos en todas las particiones cuando cualquier dato nuevo es agregado.

NUEVAS TÉCNICAS DE PARTICIONAMIENTO

Oracle 8i introduce 2 nuevos métodos de particionamiento hash y composite (rango combinado con hash), lo cual provee opciones adicionales para que el administrador pueda lograr la mejor combinación de manejabilidad y eficiencia. Aunque Oracle espera que una basta mayoría de sistemas de almacenamiento utilicen el rango de particionamiento, los esquemas de particionamiento hash y composite ofrecen distintas ventajas en ciertos ambientes warehouse.

El particionamiento hash reduce la complejidad administrativa al proveer muchos beneficios de manejabilidad del particionamiento, con un esfuerzo mínimo de configuración. Con la implementación de particionamiento hash el administrador simplifica las opciones de una llave particionada y de un número de particiones. Oracle 8i automáticamente distribuye los datos en todas las particiones. El particionamiento hash es particularmente apropiada para las tablas que no tienen un rango natural de particionamiento, y tablas que no pueden ser cargadas usando una técnica rolling-window.

El particionamiento compuesto combina esquemas de particionamiento hash y rango. Una tabla particionada con el método composite es inicialmente particionada en rangos, y después subparticionadas usando el método hash. Aunque una tabla particionada con el método composite aún mantiene todos los beneficios de un particionamiento de rangos, el nivel adicional de un particionamiento hash provee un mayor control de la ubicación física de los datos y de las operaciones del manejo de datos finer-grainde.

Un particionamiento compuesto puede ser particularmente útil, por ejemplo, un DW carga una base mensualmente (esto sugiere una estrategia de particionamiento de rango), pero cada valor del mes es extremadamente grande y puede necesitar un particionamiento nuevo. Una estrategia de partición compuesta puede además ser efectiva para un ambiente MPP desde que la técnica de particionamiento provee un método muy directo para la distribución de cada dato mensual en cada nodo del ambiente MPP.

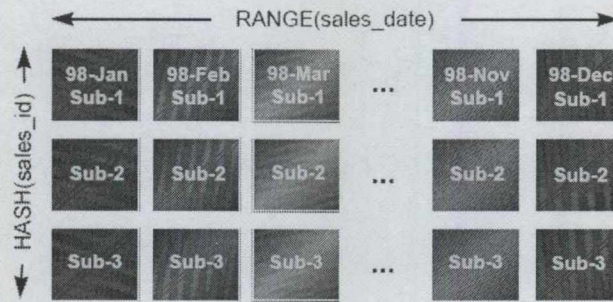


Fig. 27.3.1 Tabla particionada con el método composite.

NUEVAS CARACTERÍSTICAS ADICIONALES EN ORACLE 8I MÁS FÁCIL PARA MÁS DATOS

Aunque particionamiento es tal vez la característica más importante para el manejo de grandes cantidades de datos, muchas otras características de Oracle 8i también permite la manejabilidad de grandes DW.

MANEJO DE RECUPERACIÓN Y RESPALDOS EN EL SERVIDOR

Oracle 8 introduce una infraestructura inteligente de manejo de servidor para respaldo, restaurar y recuperar tareas que facilita la disponibilidad de operaciones seguras en una escala terabyte. Las ventajas de esta tecnología son:

- Los detalles relacionados con las operaciones de respaldo, restauración y recuperación son mantenidas por el servidor en un catálogo de recuperación, y automáticamente utilizadas como parte de estas operaciones. Esto reduce la carga administrativa y minimiza la posibilidad de errores humanos.
- Las operaciones de respaldo y recuperación están completamente integradas con el particionamiento. Particiones individuales, cuando se localizan en la misma tabla pueden ser respaldadas y restauradas independientemente de otras particiones de la tabla.
- Oracle 8 incluye el soporte para incrementar el respaldo y las recuperaciones, haciéndolas capaces de ser completamente eficientes en un tiempo proporcional a la cantidad de cambios, en lugar del tamaño total a la BD.

- La tecnología de respaldo/recuperación es altamente escalable, y provee interfaces estrechas para el manejo de subsistemas de media en la carga de datos para la industria. Esto provee operaciones eficientes que pueden almacenar grandes volúmenes de datos.

PAQUETE NUEVO PARA EL MANEJO DE ESTADÍSTICAS

Un paquete nuevo, DBMS_STATS simplifica mucho el manejo de optimizadores estadísticos. El optimizador de búsqueda utiliza estas estadísticas previamente generadas con el comando analizador de Oracle 8. Actualmente las estadísticas son importantes para realizar mejores búsquedas. Estas estadísticas contienen información tal como el número de líneas en cada tabla. El número de bloques en cada índice y la distribución de los datos en una columna dada. El paquete DBMS_STATS permite al administrador warehouse manejar más fácilmente el optimizador estático. Por ejemplo, las estadísticas pueden ser importadas y exportadas. Esto permite al administrador mover las estadísticas de una producción DW terabyte a una plataforma de 10-GB, y lo capacita para simular exactamente la conducta del optimizador en la plataforma. El paquete DBMS_STATS también provee un mecanismo para estadísticas específicas en un administrador, para casos en los cuales las características exactas de los datos pueden ser entendidas.

GENERACIÓN AUTOMÁTICA DEL OPTIMIZADOR DE ESTADÍSTICAS

Para ciertos comando DDL, tales como el CREATE INDEX (creador de índices) Oracle 8i reúne las estadísticas automáticamente. Esto elimina la necesidad para un administrador de reunir estadísticas como una operación separada siguiendo las operaciones DDL. Implícitamente Oracle 8i reúne estadísticas para muchos comando para los cuales, en general, la reunión de estadísticas es imposible. Esto permite al DW tener estadísticas más exactas, aunque minimizando las fuentes de utilización para reunir estas características. Lo más importante, es que estas características reducen el número de tareas que un administrador warehouse necesita para completar la tarea.

ORACLE 8i RAPIDEZ PARA MÁS DATOS

Así como el servidor de datos de Oracle siempre ha proveído mecanismos diferentes para las consultas de datos en un ambiente DW, Oracle 8i también provee mecanismos eficientes para la administración de grandes cantidades de datos en un ambiente DW.

Las características incluyen:

- Paralelización de operaciones administrativas.
- Contraints mejorados.
- Resúmenes actuales.
- Direct-path load API.
- Tablespace transportable.

PARALELIZACIÓN DE OPERACIONES ADMINISTRATIVAS

Oracle 8i es capaz de paralelizar todas las operaciones administrativas en un DW. Las operaciones básicas requeridas para cargar un DW, tales como el cargamento de datos en bruto y la creación de índices, son paralelizadas. Adicionalmente, todas las operaciones de partición están completamente paralelizadas, tanto que las particiones pueden ser movidas eficientemente, unidas y divididas como se requiera.

Una característica nueva en Oracle 8i es la completa paralelización para la reunión del optimizador estadísticos. Anteriormente esta paralelización, solamente era aplicable a tablas particionadas y ha sido extendida a ambas, tablas particionadas y tablas no particionadas. Esto permite al administrador hacer más exacto el optimizador estadístico de lo que podrían ser otras técnicas.

Particularmente un aspecto de Oracle 8i es la habilidad de paralelizar las inserciones, modificaciones y borrar tablas (insert, update and deletes). Esto, en conjunto con el operador paralelo de Oracle 8i llamado CREATE TABLE.... AS SELECT, provee un mecanismo eficiente y escalable para la implementación de la transformación del tamaño de los datos y la manipulación de operaciones dentro del DW. Muchas operaciones simples de transformación de datos pueden ser implementadas fácilmente dentro de SQL, previendo una solución robusta, escalable y fácil de programar.

CONSTRAINTS MEJORADOS

Dos nuevas características en Oracle 8i proveen una funcionalidad significativa en constraints para multimedia, para reunir los requerimientos de DW.

Oracle 8i soporta constraints únicos al confiar un único índice. Desde que los índices únicos son raramente usados por consultas de un warehouse, los espacios sustancialmente son salvados y realizar mejoras que pueden ser observadas por usar un constraint index-less.

Oracle 8i también soporta partiton-wise que son validaciones de integridad referencial y constraints únicos. Por lo tanto, cuando se usa un paradigma rolling-window, un administrador de DW puede validar el constraint en una partición simple, apenas como una simple partición puede ser indexada y respaldada independientemente de otras particiones.

Estos dos mejoramientos, a lo largo de Oracle 8i existen capacidades existentes de validación de constraints en paralelo, lo cual hará validación de constraints más eficientes en Oracle 8i en ambientes warehouse.

RESUMENES ACTUALES

Como parte de una nueva característica vistas materializadas, Oracle 8i administra la creación y mantenimiento de resúmenes de tablas. El administrador de warehouse puede especificar como y cuando Oracle 8i deberá restaurar el resumen de tablas. Resumen de tablas puede ser restaurado sincrónicamente, como un nuevo detalle de los datos que se cargan dentro del warehouse, o el administrador puede escoger para restaurar el resumen de tablas en menos tiempo. Algunos DW, por ejemplo, la restauración del resumen de tablas en una base semanal, mientras que los detalles de los datos son cargados en cada noche base. Una alta optimización, del mecanismo de restauración incremental se provee, así que el resumen de tablas son solamente modificadas para reflejar la carga diaria de renglones en el detalle de las tablas, más bien que reconstruyendo el resumen entero.

DIRECT-PATH LOAD API

Oracle 8i extiende a Oracle Call interface⁴⁰ para soportar una carga eficiente de los datos, estas extensiones de Call interface provee aplicaciones con acceso a Oracle 8i dirige las trayectorias para cargar las capacidades.

Las aplicaciones que usa Oracle Call interface ahora podrán cargar los datos a una BD Oracle tan eficientemente como las cargas del propio Oracle con el direct-path.

Esta característica permite una integración mas compacta entre Oracles 8 data-loading y productos de software de vendedores independientes (ISV). Por ejemplo, un motor de transformación podría cambiar y cargar datos directamente dentro de Oracle 8i. A través de muchos productos ISV's son integrados con Oracle SQL Loader, la nueva interfaz direct-path para cargar los niveles del funcionamiento y de la robustez del proceso de la carga usando estos productos, permitiendo que estos productos puenteen el paso inmediato de datos en un archivo plano.

TABLESPACE TRANSPORTABLES

Una de las más interesantes características en Oracle 8i usada para el DW es permitir tablespace transportables (incluyen contenidos) para ser movidos o copiados desde una BD Oracle 8i a otra. Esto provee un mecanismo extremadamente rápido para los volúmenes de movimiento de datos.

La transportación de los datos es un método que puede ordenar una magnitud rapidísima de datos export-import o cargar-descargar, desde la transportación tablespace que incluye solo la copia de archivo de datos y la integración apropiada de los metadatos. La transportación de tablespace podría mover también índices y constraints a lo largo de las tablas, esta característica tiene varias aplicaciones para el DW:

- Movimiento de datos desde el sistema operativo hacia un DW o entre un DW a un data-mart: puesto que los volúmenes de los datos que se transportarán son a menudo muy largos, esta característica puede proveer una considerable funcionalidad en los procesos de extracción, transportación, transformación (ETT).
- Publicación de datos: un proveedor de datos podría distribuir contenidos de CD's pre-cargados de tablespace de Oracle para sus clientes aún más que los archivos de datos brutos.

- Archivando datos, más bien archivando una exportación a una tabla, un administrador podrá archivar un tablespace de Oracle. Por lo que una pre-construcción de un tablespace puede ser tomada inactiva y traía a una activa mucho más rápido que una exportación de un archivo, esta estrategia provee una significativo beneficio en la funcionabilidad.
- Operaciones de carga de datos: es el ambiente más usado de la producción, no está disponible para fuentes de CPU que soportan la carga, la indexación y la sumarización de nuevos datos. Este proceso podría tomar lugar en una plataforma separada. Una vez que el nuevo dato ha sido cargado, el contenido del tablespace en el nuevo dato podría ser transportado al sistema de la producción. El nuevo dato podría estar disponible inmediatamente sin gastar una significativo fuente en el sistema de producción.

En el lanzamiento de Oracle 8i la capacidad de transportación de los tablespace esta limitada para BD que corren en el mismo sistema operativo. Es decir, un tablespace de una BD basada en NT puede ser copiado solamente para otra BD basada en NT, esta restricción podría ser distinta en el futuro lanzamiento.

Parte IV. DATA MARTS

¿QUE ES UN DATA MART?

Un Data Mart es una forma simple de un DW, que esta enfocada a un solo sujeto (o área funcional), tales como ventas, finanzas o mercadeo. Los Data Marts son frecuentemente contruidos y controlados por un solo departamento dentro de una organización. Dado su enfoque único, los Data Marts usualmente llevan datos de pocas fuentes. Las fuentes pueden ser sistemas operacionales internos, un DW central o datos externos.

¿EN QUE DIFIERE DE UN DW?

Un DW, se distribuye en múltiples áreas, y típicamente es implementado y controlado por una unidad organizacional central, tal como la corporación de información tecnológica. Frecuentemente, es llamado DW central o DW empresarial. Típicamente, un DW reúne datos de múltiples sistemas fuente.

Nada en esta definición básica, limita el tamaño del Data Mart o la complejidad de la toma de decisiones que contiene. Sin embargo, los Data Marts son típicamente pequeños y menos complejos que los DW; así que estos son fáciles de construir y mantener. La siguiente tabla resume las diferencias básicas entre un DW y un Data Mart:

	Data Warehouse	Data Mart
Scope	Corporate	Line-of-Business (LoB)
Subjects	Multiple	Single Subject
Data Sources	Many	Few
Size (typical)	100 GB-TB+	< 100GB
Implementation Time	Months to years	Months

Tabla 30.1 Diferencias entre DW y DM.

DATA MARTS HIBRIDOS, INDEPENDIENTES Y DEPENDIENTES

Existen tres tipos básicos de Data Marts, los dependientes, los independientes, y los híbridos. La categorización esta basada principalmente en la fuente de los datos que alimenta al Data Mart.

Data Mart dependiente distribuye los datos de algún DW central que haya sido creado.

Data Mart independiente, en contraste, es el único sistema construido para distribuir los datos directamente de una fuente operacional, una fuente externa de datos, o ambos.

Data Mart híbrido puede distribuir datos de un sistema operacional o de un DW.

DATA MART DEPENDIENTE

Un Data Mart dependiente te permite reunir los datos de tu organización en un DW. Esto te da las ventajas usuales de la centralización. La figura 31.1 ilustra un Data Mart dependiente.

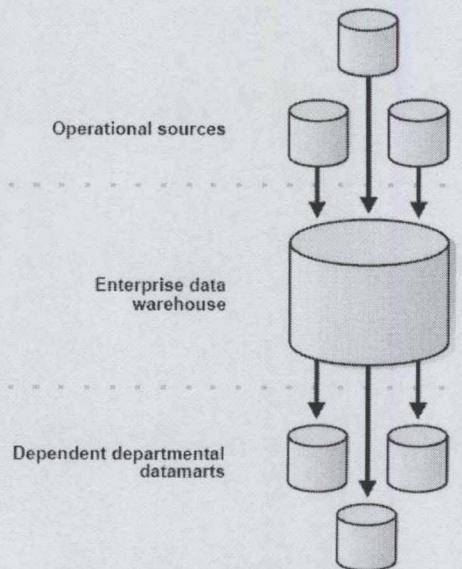


Fig. 31.1 Data Mart dependiente.

DATA MART INDEPENDIENTE

Un Data Mart independiente es creado sin el uso de un DW central. Esto puede ser deseable para grupos pequeños dentro de una organización. La figura 31.2 ilustra un Data Mart independiente.

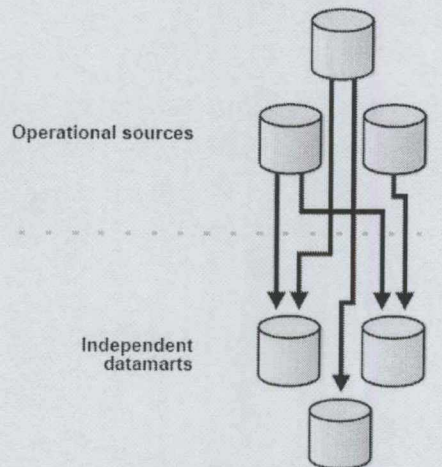


Fig. 31.2 Data Mart independiente.

DATA MART HÍBRIDO

Un Data Mart híbrido te permite combinar las entradas de otras fuentes como la de un DW. Esto puede ser útil para muchas situaciones, especialmente cuando necesitas una integración acorde, tal como la adición de un producto nuevo en la organización. La figura 31.3 ilustra un Data Mart híbrido.

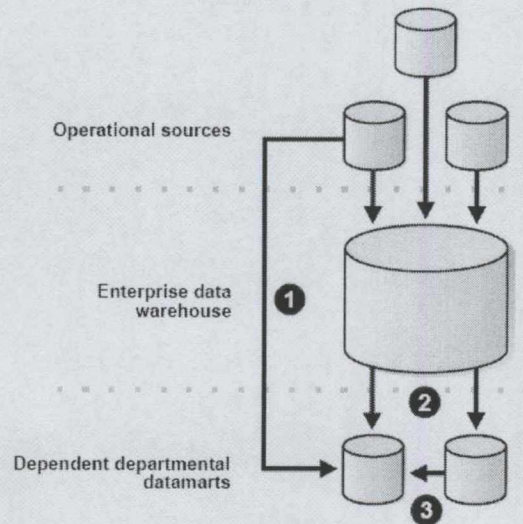


Fig. 31.3 Data Mart híbridos.

EXTRACCIÓN, TRANSFORMACIÓN Y TRANSPORTACIÓN (ETT)

La mayoría de las diferencias entre un Data Mart independiente y uno dependiente es como manejas el Data Mart, es decir, como obtienes los datos de las fuentes externas y los introduces al Data Mart. Este paso es llamado proceso de Extracción-Transformación-Transportación (ETT), involucra los movimientos de los datos de los sistemas operacionales, el filtro, el cargado y la introducción al Data Mart.

Con el Data Mart dependiente, este proceso está algo simplificado porque formatea y limpia los datos que han sido cargados en el DW central. El proceso ETT para Data Marts dependiente, es mayormente un proceso de identificación de subsets correctos de datos relevantes para la elección del área del Data Mart y para copiarlo, quizá en una forma limpia.

Con el Data Mart independiente, sin embargo, debes distribuirlo a todos los aspectos del proceso ETT, tal como haces con un DW central. El número de fuentes son igualmente pocas, y a cantidad de datos asociados con el Data Mart es menor que con el DW, dado que se enfoca a una sola área.

OLAP

Procesamiento analítico en líneas. En un DW se depositan datos para consulta, análisis y divulgación, a diferencia del procesamiento de transacciones en línea (OLTP), en donde los datos se reúnen y almacenan para operación y control. OLAP es una tecnología de procesamiento analítico que crea nueva información empresarial a partir de los datos existentes, por medio de un rico conjunto de transformaciones empresariales y cálculos numéricos. Se aplica en muchas funciones de empresa como: producción, ventas y análisis de rentabilidad de la comercialización, mezcla de manufacturas y análisis de logística, consolidaciones financieras, presupuestos y pronósticos, planeación de impuestos y contabilidad de costos.

ORACLE DISCOVERED

Se trata de una herramienta intuitiva y abierta, de consultas ad hoc, informes, análisis y publicación web, que permite a los usuarios de negocios a todos los niveles de la organización obtener acceso inmediato a información DW Relacional, centro de datos o sistemas de procesamiento de transacciones en línea. Utiliza el sistema de consultas SQL experto para generar dinámicamente consultas SQL de rendimiento optimizado.

CONCLUSIONES

El uso de herramientas para el manejo y administración de la información es evidente que en estos tiempos ha ido evolucionando así como la tecnología, los sistemas y la información, actualmente para las empresas es de suma importancia considerar estos conceptos y herramientas de trabajo para lograr una mayor eficiencia en la organización.

El Data warehouse como tecnología de almacenamiento y teniendo como ventaja el apoyo a toma de decisiones, el almacenamiento de datos históricos y actuales, pretende ser implementada en más empresas, teniendo más claro el concepto de Data warehouse así como sus componentes y arquitectura. Oracle, es considerado como uno de los más importantes softwares de manejo y administración de datos, al trabajar conjuntamente con Data warehouse tiene como finalidad una mejora en el desarrollo de la empresa y obtener una mayor utilidad en el desarrollo de sistemas, para llevar de esta manera a la empresa a dar un gran paso dentro del mercado y convertirse en empresas altamente competitivas y de gran alcance a nivel mundial, por lo que esta información acerca de Data warehouse es de gran ventaja tanto para el administrador de base de datos como para las organizaciones, siendo más claro el concepto y la estructura de la tecnología.

Todas aquellas organizaciones que trabajan con sistemas, ya sean pequeños o grandes, tienen la ventaja al implementar el Data warehouse o en caso de organizaciones y sistemas pequeños el Data mart, ya que no solo extraen información interna sino que explotan también al medio ambiente llamado factores externos, para utilizar toda esta información así como la información histórica también, en su propio bien y llegar a facilitar la toma de decisiones dentro de éstas.

Considerando que es de suma importancia la información dentro de cada organización, llamada base de datos, es momento de tomar en cuenta el concepto y desarrollo de Data warehouse e implementación dentro de éstas, pues se obtienen grandes beneficios y permite una evolución dentro de la organización. También se debe considerar la opción integridad con el medio de Internet.

Es muy interesante poder ver la capacidad de Data warehouse para un desarrollo e implementación dentro de una organización, ya que se puede trabajar satisfactoria con herramientas como Oracle DiscovererTM, Oracle Warehouse Builder, entre otros para obtener una interfaz que guía al usuario a desarrollar sus aplicaciones que desea.

Este tema aparte de amplio, es de suma importancia a todos aquellos que tienen interés en las Bases de datos, así como administradores y organizaciones, ya que cuenta con grandes ventajas y beneficios, es muy interesante saber en que consiste esta técnica y como puede desarrollarse, si una empresa cuenta con recursos suficientes para invertir en un desarrollo Data warehouse y no ha invertido, deberían enriquecerse más en este concepto y participar de estas herramientas y desarrollarse en éste ámbito.

BIBLIOGRAFÍA

INTERNET

- Oracle8i™ for Data Warehousing, Fast and Simple for More Data and More Users an Oracle Technical White Paper
<http://otn.oracle.com/products/oracle8i/index.html>
- Oracle 8i Data base <http://otn.oracle.com/documentation/oracle8i.html>
- Administración y optimización de base de datos de Oracle
<http://www.redcientifica.com/oracle/c0001p0006.html>
- Datawarehouse multidimensional <http://lsi.ugr.es/~bdf/Trabajos/lsi0006ugr.pdf>
- Tutorial Data warehouse <http://www.programacion.com/bbdd/tutorial/warehouse/>

LIBROS

- Claudio Casares, *Manual Data Warehousing*.
- Lane Paul, *Oracle8i, Data Warehousing Guide*, Oracle Corporation, Diciembre 1999.
- Harjinder S. Ginn, Praxash C. Rao, Cr. Ramón Tapia Moisen, *Data warehousing: la integridad de información para la mejor toma de decisiones*, México: Prentice Hall, 1996.

GLOSARIO

- ¹ DW Data Warehouse.
- ² BD Base de Datos.
- ³ SSD Sistemas de Soporte a la Decisión.
- ⁴ Snapshots, son vistas instantáneas.
- ⁵ Metadatos, datos acerca de los datos (información acerca de los datos).
- ⁶ Browsers, examinadores, buscadores.
- ⁷ DWA, Data Warehouse Architecture – Arquitectura del Data Warehouse.
- ⁸ Mainframes, marco principal.
- ⁹ GUI, Graphical User Interface, Interfaz gráfica del usuario.
- ¹⁰ SQL, Lenguaje Estructurado de consulta. Se considera un lenguaje estándar.
- ¹¹ Drilling down, proceso que le proporciona la posibilidad de descargar diferente grupos de datos en una sola vista. Le permite crear cualquier número de diferentes escenarios o estructuras que se acomoden a las necesidades de cada uno de los usuarios.
- ¹² OLAP On Line Analytical Processing, Procesamiento Analítico en línea.
- ¹³ SMP Symmetric MultiProcessing, Multiprocesamiento simétrico.
- ¹⁴ MPP Massively Parallel Processing, Procesamiento en paralelo masivo.
- ¹⁵ NUMA Non Uniform Memory Access, Acceso de memoria no uniforme.
- ¹⁶ RDBMS Relational Data Base Management Systems, Sistema de administración de BD Relacionales.
- ¹⁷ MDDBs MultiDimensional Databases, BD Multidimensionales.
- ¹⁸ BLOBs Binary Large Objects, Objetos grandes binarios.
- ¹⁹ ETT Extracción, transporte y transformación.
- ²⁰ Fact-table, Tabla de eventos.
- ²¹ DOP Grado de paralelismo.
- ²² OPS Oracle Parallel Server, Servidor Paralelo de Oracle.
- ²³ Bitmap Indexes, índices de mapa de bits.
- ²⁴ DML Data manipulation lenguaje, Lenguaje de manipulación de datos, ejemplo: insert, update, delete.
- ²⁵ Mapping, mapeando.
- ²⁶ Data cleanliness, limpieza de los datos.
- ²⁷ DDL Data definition lenguaje, definición del lenguaje de datos, ejemplo: create, alter, drop.
- ²⁸ DBMS System manipulation data base, sistema de manipulación de BD.
- ²⁹ Oracle Enterprise Management herramienta de Oracle.
- ³⁰ Data warehousing, almacenamiento.
- ³¹ Querys, consultas.
- ³² 1 Terabyte = 1024 Megabytes.
- ³³ Oracle Discoverer.
- ³⁴ Join, unión.
- ³⁵ Buffer cache, dispositivo de memoria.
- ³⁶ Partitioning schemes, esquemas de particiones.
- ³⁷ Partition skipping, saltando particiones.
- ³⁸ DRM Database resource manager, manejo de fuentes de la BD.
- ³⁹ Host raíz.
- ⁴⁰ Oracle Call Interface herramienta de Oracle, interfaz de llamada.