

Herramienta docente para consultas en Oracle

M^a Belén Vaquerizo García, Antonio Eduardo Renedo Mena

Área de Lenguajes y Sistemas Informáticos

Universidad de Burgos

09006 Burgos

e-mail: {belvagar,arenedo}@ubu.es.es

Resumen

En este trabajo se plantea el desarrollo de una herramienta docente, cuyo objetivo principal es ayudar a una mejor docencia en cuanto a Bases de Datos se refiere, y más concretamente que sirva de soporte a la asignatura de Administración de Bases de Datos, ayudando al alumno en el manejo y aprendizaje de consultas en Oracle, y en la profundización en las tareas propias del administrador de bases de datos, a través de la Web.

1. Introducción

En todo currículo en Informática [8], la docencia en bases de datos abarca una materia con una gran componente práctica, y en los contenidos de la asignatura sobre Administración de Bases de Datos, el conjunto de clases de prácticas van a estar destinadas a poner de manifiesto cuáles son las tareas específicas del administrador [10], tratando de que el alumno sepa implementar un gestor y administrar la base de datos.

Por ello, se ha de introducir al alumno en aquellos aspectos de implementación de la base de datos. Estos aspectos inciden en el ajuste e integridad de la misma, pues tienen una gran importancia en el estudio de la planificación y optimización de consultas. De esta forma, se ayuda al alumno a comprender el alcance de las tareas de monitorización y ajuste asociadas a la base de datos.

Siguiendo estas premisas, en el presente trabajo se plantea el desarrollo de una

herramienta docente que sirva de ayuda en la labor de aprendizaje del alumno y le ayude a adentrarse en la administración de bases de datos [11], enseñándole las herramientas que están a disposición del administrador de bases de datos para llevar a cabo su función, y, por otra parte, le facilite el manejo y aprendizaje de Consultas en Oracle a través de la Web.

Dicho sistema constará de diferentes herramientas implementadas en la Web, permitiendo que tras realizar una consulta en SQL desde un host local, ésta se mande a un servidor web, para ser procesada y optimizada hasta de tres maneras posibles: basada en costes, basada en reglas o mediante hints. Se podrán presentar cualquiera de las optimizaciones anteriores a través de la visualización de sus correspondientes planes de ejecución desarrollados en forma de árbol y de la presentación de comparativas gráficas entre los distintos tipos de optimización. También se mostrarán estadísticas referentes al coste de cada consulta; todo ello orientado a poder comparar y determinar cuál de las optimizaciones ha sido la mejor

Además, permitirá realizar una traducción del lenguaje SQL al Álgebra Relacional de una forma transparente al usuario, pasando además este álgebra a un grafo, que muestre el orden en el que Oracle accede a cada una de las partes de que se compone la consulta en su forma canónica, es decir, sin ningún tipo de optimización. Mostrando, así, su relación de forma gráfica y clara se le ayuda al alumno en el aprendizaje de este lenguaje, así como también se le permite profundizar al alumno en el conocimiento total del comando Analyze de Oracle [6], [7].

Y, finalmente, se le presenta un trabajo con índices para ayudarle a comprender el correcto uso de los índices a la hora de formular las

consultas, mostrándole los resultados que se deducen de la presencia o ausencia de estos.

Por consiguiente, se va a reflejar el proceso de desarrollo a seguir para la realización de este sistema, y, en primer lugar se exponen los diferentes objetivos planteados en la construcción de este sistema, para, de esta manera, poder comprender la elección de las herramientas utilizadas y el método de trabajo a seguir, que se mostrará en el tercer apartado, en el cuarto apartado se indican las interfaces gráficas utilizadas para cubrir estas funcionalidades, y, para finalizar, se reflejan las conclusiones obtenidas.

2. Objetivos

El objetivo general es proporcionar una interfaz web que con estructura de aplicación docente e interactiva [3], permita a los alumnos familiarizarse con el manejo de consultas SQL y su relación con el Álgebra Relacional, además de ver los distintos planes de consultas dependiendo del tipo de optimización empleada y la posibilidad de poder emplear herramientas de análisis y manipulación de índices.

La página web dinámica constará de un manual de ayuda en línea, útil para comprender y utilizar correctamente la aplicación, así como para aclarar diversos conceptos teóricos necesarios en el entendimiento de los resultados que se puedan obtener.

Los objetivos a lograr serían los siguientes:

1. Permitir al alumno introducir la consulta de maneras diferentes, bien sea de manera directa, bien permitiéndole usar un asistente a través del cual formulará su consulta, o bien, eligiendo una consulta tipo de entre varios tipos de consultas mostradas.
2. Facilitar el conocimiento y manejo de los planes de consulta de Oracle a través de la web. Se permite, a partir de una consulta SQL que se introduzca desde el navegador, obtener su plan de ejecución y mostrarlo por pantalla en un entorno gráfico amigable.
3. Optimizar las consultas, donde el alumno elegirá el tipo de optimización deseada: - basada en Costes, - basada en Reglas, - mediante Hints. Se permite ver cómo se realizan estas diferentes optimizaciones en Oracle, presentando los resultados obtenidos mediante una serie de gráficos y estadísticas que faciliten su comparación y determinar, así, para cada consulta, qué optimización ha sido la mejor.
4. Pasar la consulta de SQL al Álgebra Relacional. De una forma transparente al usuario, que muestre el orden en cómo accede Oracle a cada una de las partes de la consulta en su forma canónica, es decir, sin optimización, para que ésta se ejecute de la manera más eficiente posible, y mostrando, además, su relación de forma gráfica y clara, mediante el diagrama en árbol o grafo correspondiente generado a partir de esa consulta. También se mostrarán otros diversos gráficos y datos estadísticos sobre su ejecución.
5. Permitir profundizar en el conocimiento del comando Analyze de Oracle, mediante un asistente que permite seleccionar el tipo de análisis a realizar (index, table, cluster), el modo de hacerla (compute statistics, estimate statistics, delete statistics, validate structure, list chained rows), y otros parámetros. Todo representado mediante gráficos.
6. Estudiar el correcto uso de los índices al formular las consultas, mediante un asistente para crear, modificar y borrar índices, permitiendo comparar los resultados que se deducen de la presencia o ausencia de éstos.

Para lograr esta finalidad es necesario llevar a cabo los siguientes objetivos intermedios:

- Especificar una gramática que relacione las numerosas posibilidades que ofrece SQL con las reglas del Álgebra Relacional.
- Y, desarrollar una página web dinámica que permita introducir consultas SQL, así como distintos comandos de análisis y manipulación de índices de manera sencilla e intuitiva, y que muestre por pantalla los resultados obtenidos.

3. Desarrollo y Técnicas a utilizar

En el desarrollo de esta herramienta se utilizarán el paradigma orientado a objetos, UML y patrones de diseño [4], [5].

Como lenguaje de programación se utilizará Java [9], donde tras compilar la gramática (con extensión .jj), se obtendrán varios ficheros con extensión .java, y con compilar el archivo NombreGramatica.java (javac NombreGramatica.java) mediante el JDK 1.3., se obtendrá el fichero .class, que se ejecutará con el JDK 1.3. (java NombreGramatica). El servlet también será compilado y ejecutado con el JDK 1.3, [2].

En el desarrollo de la gramática que relacione las numerosas posibilidades que ofrece SQL con las reglas del Álgebra Relacional y permita pasar del primero al segundo, se implementará un parser con el generador de compiladores JavaCC, como generador de parsers y de analizadores léxicos, y con él se generará y se compilará la gramática. En la etapa de análisis comprenderá: el analizador léxico (scanner), el analizador sintáctico (parser), y el analizador semántico, y en la etapa de síntesis incluirá: el optimizador de código intermedio, el generador de código, el optimizador de código final.

3.1. Tecnologías de páginas Web dinámicas

Se utilizarán tecnologías de generación de páginas web dinámicas con:

- Servlets: Para obtener el plan de ejecución de una consulta, las pantallas de errores y las posibles opciones de optimización.
- Mediante HTML y JavaScript se hará el interfaz gráfico de la aplicación, como una serie de ventanas enlazadas entre ellas, donde todas tienen su ayuda y la posibilidad de ir a la ventana anterior. JavaScript para la implementación del ajuste de los applets a la resolución de la pantalla, y HTML como lenguaje estandarizado de control de navegadores, que les indica cómo colocar la información disponible.
- Diversos Applets [1] que permitan la introducción de los distintos comandos de análisis y manipulación de índices de una manera sencilla e intuitiva y que muestren por pantalla los resultados obtenidos.

Como servidor de base de datos se utilizará Oracle.

Como servidor de páginas web se utilizará Tomcat.

Como procesadores de textos, se usarán FreeJava, de libre distribución, para la edición de la gramática y del servlet, y, Notepad, como editor de Windows, para la edición del código HTML y JavaScript.

Y, el browser elegido para la visión de las páginas web, será Internet Explorer 5.5.

4. Interfaces Gráficas implementados

4.1. Pantalla de presentación

Se muestra al iniciar la sesión en el servidor. Presenta un botón “Consulta”, para acceder a las diferentes pantallas con las funcionalidades de cada una, y, al igual que en todas las pantallas de la aplicación, un botón “Ayuda”, informando del uso de la aplicación y de las diferentes funcionalidades que contiene.

4.2. Pantalla de opción de consulta

Aparece tras pulsar el botón “Consulta” de la pantalla anterior, y permite escribir una consulta SQL de una de las tres formas siguientes:

- Consulta Directa: Se introduce directamente, sin ayuda, implica un conocimiento de las tablas, sus campos y del lenguaje SQL, para no cometer ningún tipo de error.
- Consulta Asistida: Es un asistente de ayuda al usuario para elegir la información deseada: BD, Tablas, Campos, y Cláusula Where. También permite modificarla manualmente, con diferentes criterios y operaciones que no figuren en el asistente.
- Consulta Tipo: Permite elegir ejemplos de diferentes tipos de consulta SQL: Puntual, Multipunto, de Rango, de Prefijo, de Valor Extremo, Ordenada, de agrupamiento, de Join; además de consultas con subconsultas en su interior. Estas consultas podrán ser modificadas por el usuario, si éste lo desea.

4.3. Pantalla de elección de funcionalidad

Desde cualquiera de las pantallas anteriores, siempre y cuando la consulta se haya procesado correctamente, pulsando en su botón “Procesar Consulta” se puede pasar a ésta que se muestra

en la Figura 1, desde aquí se pasará a la ventana correspondiente de la función elegida.

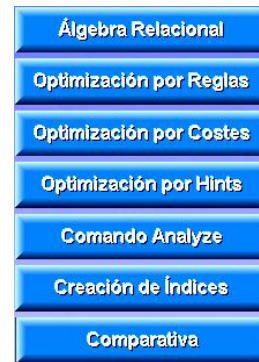


Figura 1: Pantalla de elección de funcionalidad

- Álgebra Relacional:

Muestra el árbol correspondiente al álgebra relacional de la consulta que se ha mandado procesar.
- Optimización por Reglas:

Oracle realiza este tipo de optimización por defecto. Conduce al plan de ejecución, en forma de árbol, de la consulta así optimizada. Se aplican un conjunto de reglas a la sentencia SQL, basadas en la sintaxis y construcción de las sentencias SQL, para determinar el mejor plan de ejecución, y se clasifica el camino de acceso usando los predicados en la cláusula WHERE y el orden de clasificación determina el plan de ejecución. Por defecto, usará un índice si está disponible, prescinde de factores como el tamaño de las tablas, por este motivo será aconsejable influir en la ejecución manualmente, indicando el orden de las tablas de la cláusula FROM. La tabla a la que se accederá en primer lugar será referida como tabla guía, por defecto, el orden de las tablas de la cláusula FROM será de derecha a izquierda.

A continuación, en la Figura 2, se muestra un ejemplo de un plan de ejecución para la optimización basada en reglas.

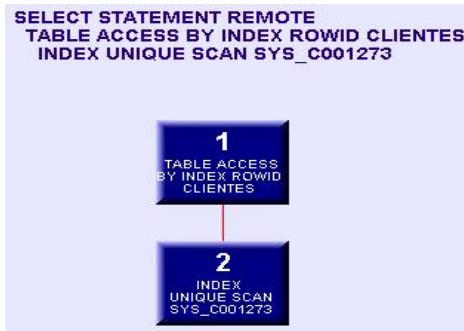


Figura 2: Ejemplo de un plan de ejecución, basado en Reglas

- Optimización por Costes:**
 Se basa en las estadísticas de la consulta para utilizar un tipo de índice u otro y, así, mejorar los tiempos en los que se realiza la consulta. Accede al plan de ejecución en forma de árbol de la consulta así optimizada. Genera un número de posibles planes de ejecución examinando diferentes permutaciones de los órdenes de unión de las tablas y, por lo tanto, los caminos de acceso a los datos. El coste de cada plan posible se determina usando estadísticas de tabla (columna) y de índices. Evalúa el coste de cada permutación para encontrar el mejor plan. El número de permutaciones depende del número de tablas implicadas; si es elevado, el tiempo de evaluar los distintos planes de ejecución será significativo, y, por tanto, se dará prioridad a la obtención de los datos de manera más rápida frente a la consecución del plan óptimo. Para que sea efectiva y garantizar un buen plan de ejecución, la información estadística debe ser mantenida y

actualizada, y el tamaño de la muestra debe ser adecuado para el total de los datos. En ausencia de estadísticas, se escoge el orden de guía de las tablas de izquierda a derecha en la cláusula FROM, al contrario de lo que sucedía en la basada en reglas.

A continuación, en la Figura 3, se muestra un ejemplo de un plan de ejecución para la optimización basada en costes.

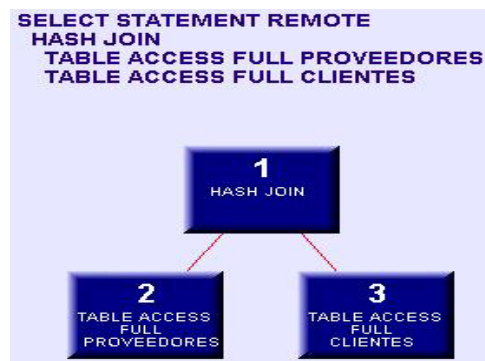


Figura 3: Ejemplo de un plan de ejecución, basado en Costes

- Optimización por Hints:**
 Requiere un conocimiento de la base de datos que se trata, ya que es el usuario el que aplica diferentes pasos para lograr una consulta óptima. Se selecciona una de las siguientes opciones: *-Acceso a tabla a través de Rowid, -Acceso a tabla a través de Índice, -Usar Nested Loops, -Usar Merge Join, -Usar Hash Join, -Reglas, -Todas las Filas, -Primeras Filas, -Usar Paralelismo.*
- Comando Analyze:**
 Accede a un asistente que nos permitirá seleccionar las diferentes opciones del comando "Analyze" de Oracle. En esta pantalla se podrán seleccionar los parámetros y opciones con los que se desea realizar el análisis, de manera semi-guiada, habilitando o inhabilitando opciones en función de los elementos seleccionados. Primero se selecciona qué análisis se desea realizar: *-INDEX, -TABLE, o -CLUSTER.* A continuación se debe seleccionar una de

las siguientes opciones: *-Compute Statistics*, *-Estimate Statistics*, *-Delete Statistics*, *-List Chained Rows*. Al seleccionar *compute statistics* o *estimate statistics* se activan las opciones de la cláusula *For*: *- For Table*, *- For All Indexed Columns*, *- For Columns*, *- For All Indexes*.

- Creación de Índices:
Mediante un asistente permite seleccionar las diferentes opciones que permite Oracle de crear, modificar y borrar índices, sobre una columna de una tabla o sobre un cluster, único o bitmap, con o sin opción Nosort, indicando o no el tablespace al que el índice es asignado, los parámetros *ptfree* y *ftused*, así como *intrans* y *maxtrans*, y los parámetros de almacenamiento, tamaño inicial, siguiente incremento y máxima extensión.
- Comparativa:
En forma de gráfico de barras, generadas al usar uno u otro optimizador, se muestran los diferentes planes de ejecución de la consulta y las estadísticas, que son: *- Recursive Calls*, *-DB Block Gets*, *- Consistent Gets*, *-Physical Reads*, *-Redo Size*, *-Bytes Sent via SQL*NET to Client*, *- Bytes Received via SQL*NET from Client*, *- SQL*Net Roundtrips to/from Client*, *-Sorts (memory)*, *-Sorts (disk)*.

5. Conclusiones y Líneas Futuras

Se ha planteado el desarrollo de una herramienta docente, cuyo objetivo principal es ayudar a una mejor docencia en cuanto a Bases de Datos se refiere. Se trata la optimización de consultas de diferentes modos, para deducir cuál ha sido la más óptima. Y además permite realizar una traducción del lenguaje SQL al Álgebra Relacional de una forma transparente al usuario,

así como ayudar al alumno a comprender el correcto uso de los índices a la hora de formular las consultas.

Este trabajo, que está siendo desarrollado por alumnos en su proyecto de fin de carrera, supone el punto de partida en el desarrollo de una serie de herramientas docentes e interactivas que ayuden en la comprensión de la administración de la base de datos Oracle, las cuales se consideran muy útiles en las prácticas de la asignatura de Administración de Bases de Datos de 3º de I.T.I.G de la Universidad de Burgos;

Referencias

- [1] Anderson Loy, Jensen Cary & Stone Blake: *Manual de Oracle Jdeveloper: guía completa para construir potentes aplicaciones y applets en Java*. ,Mc Graw-Hill, 2002.
- [2] Cavaness, C., *Programming Jakarta Struts*, O'Reilly, 2003.
- [3] Dietrich, S. W., Eckert, E., Piscator, K., *WinRDBI: A Windows-Based Relational Database Educational Tool*, Proceedings of the 28th ACM SIGCSE Technical Symposium on Computer Science Education, San Jose, California, pp. 126-130, 1997.
- [4] García Molina, J. et al., *Un proceso basado en UML para aplicaciones de gestión*, Facultad de Informática, 2000.
- [5] Larman, C., "UML y Patrones: Una introducción al análisis y diseño orientado a objetos". Prentice-Hall, 1998.
- [6] Loney, K. & Koch, G.: "Oracle 8i. The Complete Reference", Mc Graw Hill (2000).
- [7] Pérez, C. Oracle 9i. *Administración y Análisis de Bases de Datos*, Rama, 2002.

- [8] Piattini Velthuis, M., Ruiz Gonzalez, F., Calero Muñoz, C., *Propuesta de actualización del curriculum de Bases de Datos*, Novática, 2002.
- [9] Rinehart., M., “Desarrollo de bases de datos en Java”, Mc Graw-Hill, 1998.
- [10] Silberschatz, A., Korth, H.F., Sudarshan, S., *Fundamentos de Bases de Datos (4th Edition)*, Mc Graw Hill, 2002.
- [11] W. Hansen, G., V. Hansen, J., *Diseño y Administración de Bases de Datos*, Prentice Hall.

