

# Trabajando en el laboratorio el diseño físico de bases de datos y la optimización de consultas

Arturo Jaime Elizondo, César Domínguez Pérez

Dpto. de Matemáticas y Computación  
Universidad de la Rioja  
Ed. Vives, Luis de Ulloa s/n, 26004 Logroño  
{arturo.jaime, cesar.dominguez}@unirioja.es

## Resumen

En este trabajo explicamos el tratamiento práctico que hacemos en la enseñanza de conceptos de diseño físico de bases de datos. Describimos nuestra experiencia en la preparación y puesta a punto de un laboratorio cuyo objetivo principal es aprender a mejorar la ejecución de consultas mediante la creación de índices adecuados. Incluimos además las dificultades que nos hemos encontrado y los medios que hemos utilizado para resolverlas.

## 1. Introducción

Las técnicas activas de aprendizaje tienen una gran aceptación como método de enseñanza en informática. Éstas incluyen actividades como lecturas, resolución de problemas, etc. La enseñanza de informática es “por su propia naturaleza una disciplina de aprendizaje activa” [13]. Esta idea tiene aún más fuerza en una carrera de ciclo corto [2]. Este trabajo pretende explicar el carácter práctico que intentamos dar a la enseñanza de un tema como es el diseño físico de bases de datos en la titulación de Ingeniería Técnica de Informática de Gestión de la Universidad de La Rioja.

El diseño físico de una base de datos (BDs) es la última etapa en el proceso de diseño de la misma. En ella se describen las estructuras de almacenamiento y los métodos de acceso, tratando de mejorar aspectos como el rendimiento y la disponibilidad de los datos. Este aspecto de las BDs está cobrando cada vez más importancia. Esto se debe fundamentalmente al incremento en el volumen de datos que se necesitan almacenar y la rapidez requerida en el acceso a los mismos, de modo que las estructuras físicas sobre las que se apoya dicho almacenamiento y acceso se convierten en factores críticos. De hecho,

actualmente existen profesionales que han constituido consultorías dedicadas a ofrecer ayuda al diseño de índices para tablas [12, 8].

A pesar de esta aparente importancia, un sentimiento que parece que se ha instalado en la empresa es que los titulados en informática carecen de destrezas suficientes para el tratamiento del diseño físico de BDs [3]. Por otro lado, la literatura básica sobre diseño de BDs no suele dedicar mucho espacio al diseño físico. Además, no es difícil leer proyectos de fin de carrera donde se hace un pobre tratamiento de este aspecto (si es que se ha tenido en cuenta).

Debemos tener presente que los diseños conceptual y lógico de BDs tienen un carácter más abstracto y se puede obviar el sistema gestor de bases de datos (SGBD) concreto sobre el que se implantará la BD. Sin embargo, el diseño físico está más cerca de un sistema concreto y es más propicio para tratarlo en el laboratorio. Existen muchos sistemas, cada uno con sus propias características y se podrían estudiar los más utilizados. Las restricciones de tiempo nos obligan a ser selectivos y elegir uno: SQL Server 2005 y explicar al alumno cómo desenvolverse en él.

El diseño físico tiene suficiente entidad como para crear una asignatura completa. Sin embargo, nuestro propósito no es ése, sino sacar partido a las horas que dedicamos a explicar temas que son preliminares (organizaciones de datos y optimización de consultas). La parte más sencilla consiste en crear BDs y asignar dominios a las columnas, pero no ocurre lo mismo con la creación de índices o de vistas materializadas que aceleren consultas concretas. Estos últimos aspectos conllevan entender estructuras y algoritmos más o menos complejos. Nuestro laboratorio va destinado a facilitar la adquisición de estos conceptos.

La idea que presentamos para este laboratorio surge de una ponencia impartida por Kimberly

Tripp [19]. De hecho, cuando comenzamos a impartir los laboratorios en el curso 2003/04 utilizamos casi directamente el material suministrado en la ponencia. Desde entonces se han hecho continuas mejoras en la exposición, el tratamiento de los ejemplos y se han introducido ejercicios.

Este trabajo se estructura de la siguiente forma. En la sección siguiente incluimos el contexto en el que se ubica el tema a desarrollar. En la sección 3 presentamos un ejemplo que sirve para ilustrar la metodología que seguimos en el laboratorio. En la sección 4 presentamos prerrequisitos, objetivos y casos de estudio. En la sección 5 se detallan los problemas detectados y cómo se han resuelto y en la sección 6 se hace una breve descripción de trabajos similares. El artículo termina con una sección de conclusiones.

## 2. Contexto

El diseño físico en BDs aparece reflejado explícitamente en el primer punto de las competencias específicas del Perfil de Sistemas, dentro de los perfiles profesionales de grado, en el “Libro Blanco del título de grado en Ingeniería Informática” [2]. También se refleja su importancia en las recomendaciones de la ACM sobre el currículo en informática [1]. Todo ello, junto con el interés real que tiene en la práctica el diseño físico de BDs, hace recomendable su inclusión en el currículo de las actuales titulaciones en informática.

La asignatura donde se trata el diseño físico de BDs en la titulación de Ingeniería Técnica de Informática de Gestión de la Universidad de La Rioja se llama *Diseño de Bases de Datos*. Se imparte en el 1<sup>er</sup> cuatrimestre del 2<sup>o</sup> curso. Es una asignatura obligatoria que consta de 3 créditos teóricos y 3 prácticos de aula o laboratorio. La titulación cuenta con otras dos asignaturas obligatorias relacionadas directamente con las BDs: Bases de Datos (2<sup>o</sup> cuatrimestre, 1<sup>er</sup> curso) y Programación de Bases de Datos (2<sup>o</sup> cuatrimestre, 2<sup>o</sup> curso) y una optativa: Administración de Bases de Datos (3<sup>er</sup> curso). Todas ellas constan de 6 créditos. Conviene comentar que la asignatura donde se estudian las estructuras de datos está ubicada en 2<sup>o</sup> cuatrimestre del 1<sup>er</sup> curso. Una de las pretensiones de esta nueva titulación (comenzó a impartirse en el curso 2002/2003) es la

enseñanza práctica de los conceptos. En esta filosofía se enmarca el presente trabajo: el desarrollo de laboratorios para trabajar conceptos de diseño físico de BDs.

El diseño físico abarca varios temas. En concreto, en la metodología de diseño físico presentada en [4] se distinguen los siguientes pasos: traducir el modelo lógico al SGBD seleccionado, diseñar la organización de los archivos y los índices, diseñar las vistas y mecanismos de seguridad, considerar la introducción de redundancia controlada y monitorizar y ajustar el sistema final. Teniendo en cuenta la disponibilidad de tiempo (y que no es una asignatura específica de diseño físico), nos centramos en el segundo paso trabajando la optimización de consultas a través de índices. Otros pasos son tratados en otros temas y laboratorios de la asignatura. Por ejemplo, la construcción de la BD o la desnormalización. Los mecanismos de seguridad, la monitorización y ajuste del sistema final se delegan a otras asignaturas posteriores como *Administración de Bases de Datos*.

En la Tabla 1 incluimos una revisión del tratamiento que se hace de capítulos relacionados con el diseño físico en aquellos libros básicos sobre BDs que entendemos pueden ser los más utilizados<sup>1</sup>. Reflejamos el número de páginas dedicadas a tres temas: (1) la descripción de las diferentes organizaciones de datos y estructuras de acceso, (2) la optimización de consultas y (3) el diseño físico de BD. Sobre este último tema se contabilizan el número de ejemplos y de ejercicios recogidos en cada obra.

El análisis de esta tabla permite observar que se dedica en general poco espacio a ejercitar estos temas, proporcionándose muy pocos ejemplos, a diferencia del tratamiento que se da a los diseños conceptual (por ejemplo, con diagramas E/R) y lógico (transformación al modelo relacional). Como consecuencia, puede dar la sensación de ser un asunto menos importante. Además, estos temas se presentan desde una perspectiva más bien teórica. Se dedica mucho espacio a temas

<sup>1</sup> Existen libros específicos de diseño físico [16, 8, 12, 3, 14] que obviamente dan un tratamiento singular del tema, que no son objeto de nuestro estudio.

Libro	Organización y acceso	Optimización	Diseño Físico	Ejemplos	Ejercicios
Elmasri y Navathe 2007 [7]	73	39	11	0	0
Connolly y Begg 2007 [4]	23	47	22	1	4
Date 2001 [5]	0	46	0	0	0
Silberschatz y otros 2007 [18]	73	0	0	0	0
De Miguel y otros 1999 [6]	18	0	0	0	0
Rob y Coronel 2004 [17]	1	0	1	1	0
Pons y otros 2005 [15]	51	0	0	0	0
Kroenke 2003 [10]	22	0	0	0	0
<b>Promedio</b>	<b>32,6</b>	<b>16,5</b>	<b>4,25</b>	<b>0,12</b>	<b>0,5</b>

Tabla 1. Tratamiento bibliográfico de temas relacionados con el diseño físico

preliminares sobre ficheros, optimización y recomendaciones de diseño físico sin rematar todo ello con al menos un ejemplo práctico. Tampoco se proponen laboratorios (de prácticamente nada). La causa probablemente radica en tenerse que decantar por un sistema concreto o tener que incluir cómo hacer estas tareas en un conjunto de sistemas. La mayoría de los libros específicos sobre este tema [16, 8, 3] optan por la primera opción.

Hemos realizado también un pequeño estudio del tratamiento del tema en algunas universidades españolas. No se trata de un estudio formal (que puede ser muy interesante) sino que hemos consultado los programas de estudios y de asignaturas publicadas en la webs oficiales de universidades con todas las limitaciones que ello conlleva. La impresión obtenida es que este tema suele ser tratado en asignaturas optativas de cursos superiores, utilizando en los laboratorios (si es que existen) herramientas automatizadas de diseño físico. De todos modos, el material de los laboratorios no suele estar disponible.

### 3. Ejemplo

En esta sección presentamos un ejemplo de utilización de la herramienta *Management Studio* de *SQL Server 2005* para la elección de índices adecuados que aceleren consultas concretas.

El material proporcionado a los alumnos [20] consta de un fichero con código T-SQL (*script*), y una presentación *powerpoint* donde se introduce el uso de la herramienta, se explican algunos elementos de los planes de ejecución y se presenta cada uno de los casos de estudio presentes en el fichero de código.

Como ejemplo ilustrativo presentamos la consulta de la figura 1. Ésta accede únicamente a

la tabla *cliente*, que consta de 10.000 filas organizadas en 94 bloques de disco y el resultado obtenido consta de 106 filas. Inicialmente *cliente* cuenta con un solo índice denominado PK\_cliente, que es un IP (índice primario) sobre el atributo clave primaria *id*.

```
select id, nombre, region
from cliente
where nombre LIKE 'K%' and
region between 'La Rioja'
and 'Navarra' and id > 5000
```

Figura 1. Consulta para mejorar el rendimiento

Para ver el *plan de ejecución* utilizado en el procesamiento de la consulta, hay que elegir previamente la opción “Incluir plan de ejecución real” del menú “Consulta”. Dicho plan de ejecución es una representación gráfica de los algoritmos seleccionados por el optimizador para resolver la consulta. La figura 2 muestra el plan de ejecución seguido si sólo contamos con el índice PK\_cliente. Las flechas indican el orden de ejecución (de derecha a izquierda). Cuando situamos el puntero del ratón sobre un icono o una flecha de este gráfico se muestra información sobre el mismo en una ventana emergente.

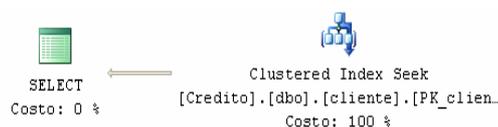


Figura 2. Plan de ejecución sobre PK\_cliente.

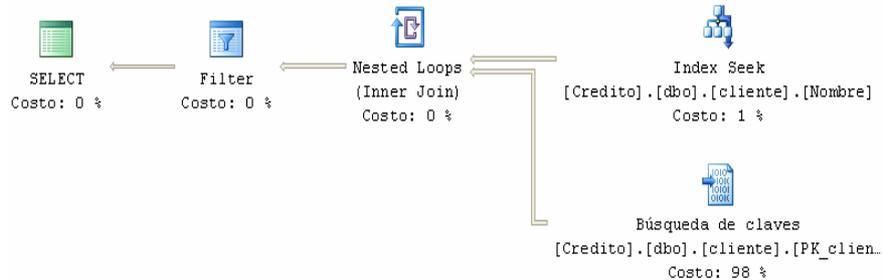


Figura 3. Plan de ejecución sobre el índice Nombre

El algoritmo *Clustered Index Seek*, mencionado en el plan de la figura 2, realiza una búsqueda desde la raíz a la hoja del índice *PK\_cliente* donde se encuentra (o debería estar, en caso de no existir) el valor 5000 de *id*. Como se trata de un *IP* las filas se mantienen ordenadas por el valor de *id* (*Clustered*). Además de la búsqueda anterior, el algoritmo recorre las filas posteriores a la de *id=5000* (si la hubiere). Una vez llevadas a memoria las filas con *id>5000*, *Clustered Index Seek* descarta las filas cuyo nombre no comience por *K* o cuya región no pertenezca al rango comprendido entre 'La Rioja' y 'Navarra'.

En el caso de *BDs* grandes el objetivo principal de la optimización es minimizar el coste de acceso a disco, es decir, el número de bloques transferidos entre disco y memoria. El optimizador estima este coste para cada algoritmo posible y ejecuta el que se presupone que será más barato.

El número de bloques transferidos real se puede conocer con esta herramienta a partir de la ejecución de la instrucción `set statistics io on`. Al volver a realizar la consulta podemos ver en la solapa "mensajes" información sobre la ejecución. Dicha información incluye el dato de 48 "Lecturas lógicas", que serían los bloques transferidos<sup>2</sup>.

La consulta anterior incluye otros atributos además de *id*. Creamos un *IS* (índice secundario) llamado *Nombre*, para el atributo no clave *nombre* con la instrucción `create index Nombre on cliente(nombre)` o usando la interfaz de la herramienta. Aproximadamente la mitad de las filas tiene *id>5000* pero sólo 1/26 de la tabla

contiene nombres que empiezan por *K*. Este índice promete dar mejores resultados. Sin embargo la ejecución de la consulta nos muestra de nuevo el plan de ejecución de la figura 2.

¿Por qué no ha elegido el índice *Nombre*? Para averiguarlo, forzamos el uso del índice *Nombre* en la ejecución de la consulta. Esto se consigue cambiando la cláusula `from` de la consulta por: `from cliente with(index(Nombre))`. El plan de ejecución mostrado es el de la figura 3 y el número de bloques transferido aumenta a 623, lo cual empeora el simple recorrido secuencial de los 94 bloques de la tabla cliente.

Para entender el algoritmo representado en el plan de la figura 3 hay que tener presente que en *SQL Server* los índices *clustered*<sup>3</sup> son físicos, pero los no *clustered* (*IS*) sólo son físicos si la tabla no tiene índice *clustered*. En caso contrario los *IS* son índices lógicos. La figura 4 muestra una pequeña tabla con un *IP* físico a la izquierda y un *IS* lógico a la derecha. Las hojas del índice lógico contienen valores del *IP* en lugar de punteros a bloques de la tabla.

Los índices lógicos de *SQL Server* también incluyen los valores del índice *clustered* en los nodos intermedios y las hojas están encadenadas en ambos sentidos.

El algoritmo de la figura 3 explica que se hace una búsqueda desde la raíz hasta la hoja del índice *Nombre* (*Index Seek*), pudiendo recorrerse varias hojas del árbol (recorre todas las entradas que

<sup>2</sup> *SQL Server* dispone en memoria de gran parte de la *BD* en uso. Por dicha razón se habla de lecturas lógicas.

<sup>3</sup> Índice *Clustered* es el que mantiene las filas de la tabla ordenadas por el atributo o atributos sobre los que se define. Corresponden a los índices primarios, si no admiten repeticiones de valor, o a los agrupados en caso contrario.

empiezan por K). Pero para obtener la columna región, presente en where y select, se debe completar la búsqueda utilizando el IP. Esto último se explica mediante los iconos *Búsqueda de claves* e *inner join*. Esta última acción es la que dispara el número de bloques transferidos.

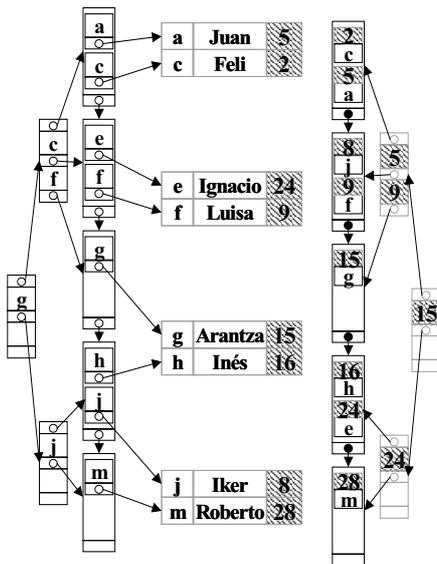


Figura 4. Índices físico (izda.) y lógico (dcha.)

En lugar del IS Nombre podríamos haber definido uno para el atributo región. En ese caso el número de bloques transferidos sería de 6.246 (de aquí se desprende que la subconsulta sobre región es menos selectiva que la de nombre). Sin embargo, usando ambos índices se obtiene el plan de la figura 5 que reduce el número de bloques a 21, aunque el optimizador sigue seleccionando el

primer plan (figura 2). En este caso no hace falta acceder al IP ya que los valores del atributo id (clave primaria) están presentes en los IS. El icono *hash/match* representa el cálculo de la intersección de filas obtenidas por ambos índices mediante un fichero intermedio de direccionamiento calculado (*hash*).

El efecto combinado de los IS ha reducido los bloques a transferir respecto a la situación inicial (figura 2) en un 56%. ¿Se puede mejorar? La respuesta es afirmativa: con `create index NomReg on cliente(nombre, region)`.

El índice compuesto produce un plan de ejecución sencillo con un recorrido *Index Seek* sobre el mismo y el correspondiente *Select* final. En este caso los bloques a transferir se reducen a sólo 4, logrando un ahorro nada menos que del 91% respecto a la situación de partida.

A la vista de lo experimentado podemos enunciar algunas *pautas de actuación* que se sugieren en la literatura para casos similares [19, 8]:

- Como cada subcondición conectada con `and` va limitando el resultado, podemos empezar creando un índice sobre una columna de `where` que obtenga pocas filas (selectiva). Si hay un índice así, podría valernos.
- Si no es suficiente, se crean índices para más columnas de `where` que sean selectivas. No tienen por qué ser todas.
- Si no hay columnas suficientemente selectivas probamos a crear un índice compuesto que incluya todos los atributos de `where` seguidos de los de `select`.

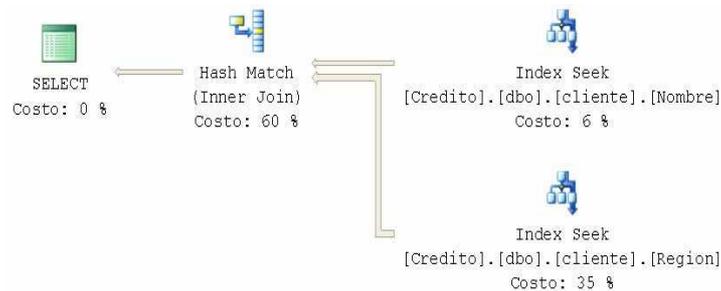


Figura 5. Plan de ejecución sobre los índices Nombre y Región

#### 4. Prerrequisitos, objetivos y casos de estudio

Como se puede apreciar tras la lectura de la sección anterior, los alumnos deben dominar bastantes conceptos para comprender los planes de ejecución. Deben tener presente las distintas formas en las que se puede recorrer un árbol B o B<sup>+</sup>, qué algoritmos de reunión (join) utilizan los SGBD, qué son índices lógicos, qué son ficheros de direccionamiento calculado (hash), qué es un proceso de fusión (merge) o cómo manejan los optimizadores las funciones de coste de los algoritmos.

En la actualidad el laboratorio consta de dos sesiones de dos horas cada una. Los objetivos del mismo incluyen: el estudio de índices interesantes para diferentes tipos de consulta, el manejo de *SQL Server Management Studio* para el examen de planes de ejecución y para forzar el uso de índices concretos.

Se cubren los siguientes casos de estudio:

- Consultas sobre una tabla y sólo con and.
- Consultas sobre una tabla con or.
- Consultas con reuniones (join).
- Consultas con agregados.
- Uso de vistas materializadas (*indexed views*).

#### 5. Problemas detectados y cómo se han resuelto

El material utilizado inicialmente era una primera adaptación del presentado en [19]. Desde la primera experiencia detectamos que los alumnos tenían muchos problemas para la comprensión de los planes de ejecución y, como consecuencia, para entender qué se conseguía con cada índice creado.

La primera causa de estos problemas es el bagaje de conocimientos previos manejados. Éstos se han introducido durante aproximadamente 7 horas de clase de exposición teórica. El absentismo puntual, el tiempo transcurrido entre la introducción de los conceptos y su manejo en el laboratorio y el poco tiempo dedicado en general por los alumnos a llevar los estudios al día nos sitúan en un marco poco propicio para pasar rápidamente a la práctica. Como es lógico, la

solución no puede pasar por explicar de nuevo todos los conceptos ya vistos, pero nos fuerza a recordar conceptos de vez en cuando y a seguir un ritmo pausado. Durante el laboratorio procuramos obtener realimentación mediante preguntas sencillas. El número reducido de alumnos en las sesiones de laboratorio (máximo 20) facilita su participación.

En segundo lugar, debemos reconocer cierta complejidad en esta tarea. Es difícil anticipar qué algoritmo se elegirá e incluso, en casos concretos, cuesta deducir cómo consigue utilizar un determinado algoritmo sobre un índice o por qué el optimizador no habrá sido capaz de elegir la mejor opción. Como el tiempo de laboratorio es limitado evitamos el estudio de casos rebuscados. Se proponen como material de autoestudio otros ejemplos complementarios.

En tercer lugar, observamos que aunque los alumnos tengan una idea bastante aproximada de un árbol B<sup>+</sup>, explicarles un recorrido distinto al que visita nodos desde la raíz a una hoja les deja, en ocasiones, perplejos. Por esta razón construimos una representación gráfica para cada recorrido de árbol representado en los planes de ejecución y para algunos algoritmos formados por varios iconos. Un algoritmo sencillo es el recorrido *Index Seek* ilustrado en la figura 6 mediante flechas gruesas. Para otros recorridos más complicados, como el de la figura 3, utilizamos animaciones de powerpoint.

Otra cuestión interesante es dejar que los alumnos se enfrenten a un problema de cada caso estudiado, descubriendo por sí mismos los problemas que surgen y probando sus propias soluciones. Estos ejercicios les brindan la oportunidad de preguntar al profesor aspectos que no les hayan quedado claros.

También es necesario dedicar un tiempo de exposición a contar el uso de la herramienta. Qué opciones se deben activar, cómo acceder a la información que ofrecen los planes, cómo ejecutar trozos de script, cómo consultar, crear, eliminar índices, etc.

Hay que tener presente que las tablas pequeñas no obtienen, en general, beneficio de los índices en el proceso de consultas que acceden a las mismas. Necesitamos tablas que consten de un número razonablemente alto de filas. Para responder a esta necesidad hemos creado un script T-SQL que genera una BD con 5 tablas. Para

generar las filas utilizamos funciones aleatorias y tablas auxiliares que se recorren repetidas veces. El tamaño de las tablas obtenidas varía entre 100.000 y 3.000 filas.

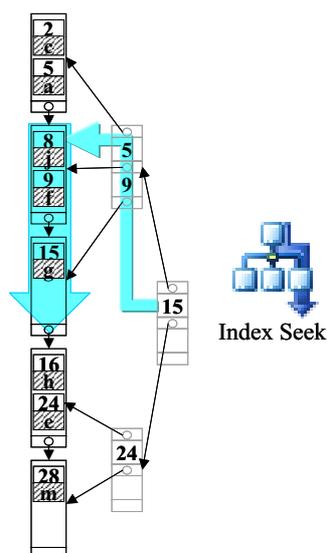


Figura 6. Representación del recorrido *Index Seek*.

Así mismo conviene mencionar que la representación de los algoritmos no está estandarizada. Esto supone que otra herramienta puede mostrar el mismo algoritmo de una forma completamente diferente. Por ejemplo, la última versión de la herramienta utilizada ha variado notablemente la representación de algunos algoritmos (afortunadamente pocos) respecto a la versión anterior.

Por último, es interesante evaluar de algún modo los conocimientos prácticos adquiridos en las sesiones de laboratorio para motivar tanto la asistencia como la comprensión de sus contenidos. En nuestro caso se realizan preguntas de tipo test dentro del examen final de la asignatura.

## 6. Trabajos que tratan temas similares

Hemos buscado propuestas de laboratorios sobre diseño físico sin éxito. Tan sólo en los libros específicos de diseño físico se pueden encontrar algunos ejemplos que sirvan de inicio para desarrollar un laboratorio (ver [16] para el caso de SQL Server 2005).

Existen trabajos para mostrar otros aspectos del modelo relacional o trabajos para la comprensión de estructuras de datos donde se llegan a construir herramientas elaboradas [9, 11]. En nuestro caso es difícil elaborar una herramienta útil para cualquier sistema ya que los temas tratados son dependientes de la implementación del SGBD e incluso de la versión del mismo.

## 7. Conclusiones

Los laboratorios son herramientas fundamentales para la enseñanza-aprendizaje práctica y activa por parte del alumno. En particular, en este trabajo hemos incluido una experiencia en el desarrollo de laboratorios para la enseñanza del diseño físico de BDs. A través de este laboratorio tratamos de evitar que este tema se quede en meros conceptos teóricos.

Las ideas básicas de nuestra propuesta son:

- Llevar al laboratorio una materia que sólo se trata a nivel teórico en los libros de consulta habituales sobre BDs.
- Superar la dificultad de comprensión de los planes de ejecución con esquemas y animaciones.
- Plantear ejercicios similares para su resolución en el laboratorio, evitando quedarse en la exposición de soluciones.
- Proporcionar un resumen de las pautas a seguir, encontradas en la literatura, para cada caso de estudio insistiendo en el interés de hacer pruebas.

Debemos destacar que el desarrollo del laboratorio pone de manifiesto las dificultades inherentes al desarrollo de índices en BDs y que esas dificultades pueden ser superadas a través de este tipo de prácticas. Esta afirmación se avala por la facilidad mostrada por los alumnos para resolver problemas similares a los desarrollados en el laboratorio.

Aunque en este trabajo no incluimos una evaluación objetiva de los resultados obtenidos con estos laboratorios, sí podemos comentar que tanto la valoración subjetiva de los mismos por parte de los profesores como el *feedback* recibido de los alumnos son positivos.

Como posibles mejoras, nos planteamos incorporar el uso de herramientas que automatizan este proceso de propuesta de índices y que son proporcionadas por los propios SGBD. También

podemos incluir otras actividades relacionadas con el diseño físico. Ambas mejoras están condicionadas por el tiempo disponible para desarrollar la asignatura.

Hace un tiempo nos pareció interesante hacer participe al resto de profesores de la titulación de la forma en la que se trabajan estos temas en la asignatura. Esto ha facilitado que los directores de proyectos de fin de carrera sean más conscientes del interés que puede tener el diseño físico de BDs en casos concretos.

### Agradecimientos

A Julio Rubio por animarnos a escribir este trabajo. A Tomás A. Pérez que nos ayudó en los primeros pasos de la puesta en marcha de estos laboratorios. A Kimberly Tripp por su magnífica presentación en TechEd03 de las posibilidades que ofrecía *SQL Server 2000* para el examen del proceso de consultas.

### Referencias

- [1] ACM Computing Curricula 2005: The Overview Report. 2005.  
<http://www.acm.org/education/curricula.html-CC2005>.
- [2] ANECA. Libro Blanco del título de grado en Ingeniería Informática. 2005.  
[http://www.aneca.es/activin/activin\\_conver\\_LLBB.asp](http://www.aneca.es/activin/activin_conver_LLBB.asp).
- [3] Burleson, D.K. *Physical Database Design using Oracle*. Auerbach Publications, 2005.
- [4] Connolly, T.M., Begg C.E. *Sistemas de bases de datos. Un enfoque práctico para diseño, implementación y gestión*. 4ª Edición. Addison Wesley, 2005.
- [5] Date, C.J. *Introducción a los sistemas de bases de datos*. 7ª Edición. Prentice Hall, 2001.
- [6] De Miguel, A., Piattini, M., Marcos, E. *Diseño de bases de datos relacionales*. Rama, 1999.
- [7] Elmasri, R. Navathe S.B. *Fundamentos de sistemas de bases de datos*. 5ª Edición. Addison Wesley, 2007.
- [8] England, K., Powell, G. *Microsoft SQL Server 2005 Performance, optimization and tuning handbook*. Elsevier, 2007.
- [9] Gutiérrez, J.J., Escalona, M.J. Villadiego, D., Mejías, M. *Comparativa de herramientas para la enseñanza de lenguajes relacionales*. XI Jornadas de Enseñanza Universitaria de la Informática. 2005.
- [10] Kroenke, D.M. *Procesamiento de bases de datos*. 8ª Edición. Prentice Hall, 2003.
- [11] Lacuesta, R., Urzelai, K. *Mejora de la comprensión de las estructuras de datos*. IX Jornadas de Enseñanza Universitaria de la Informática. 2003.
- [12] Lightstone, S., Teorey, T., Nadeau, T. *Physical database design*. Morgan Kaufmann Publishers, Elsevier, 2007.
- [13] McConnell, J.J. Active and cooperative learning: Tips and tricks (part I). The SIGCSE Bulletin, 37 (2), 2005, pp 27-38.
- [14] MSDN. Documentación on-line de SQL-Server 2005.
- [15] Pons, O., Marín, N., Medina, J M., Acid, S., Vila, M.A. *Introducción a las Bases de datos. El modelo relacional*. Thomson, 2005.
- [16] Riverio, E., Guardia C, Reig J.C. *Bases de datos relacionales: Diseño Físico (Orientado al DB2 para s/OS de IBM)*. Publicaciones de la Universidad Pontificia Comillas, 2004.
- [17] Rob, P. Coronel, C. *Sistemas de bases de datos. Diseño, implementación y administración*. Thomson, 2004.
- [18] Silberschatz, A., Korth H.F., Sudarshan, S. *Fundamentos de diseño de bases de datos*. 5ª Edición. Mc Graw Hill, 2007.
- [19] Tripp, K. *Designing for performance: Optimization with indexes*. Microsoft Tech.Ed 03 Conference. 2003.
- [20] Web con el material de los laboratorios.  
<http://www.unirioja.es/cu/arjaim/jenui08>