Proyecto Docente

Mercedes Marqués Andrés

13 de julio de 2000

Índice General

Prefacio

Esta memoria ha sido elaborada para concursar a la plaza de Profesor Titular de Escuela Universitaria en el área de Ciencias de la Computación e Inteligencia Artificial (plaza número 15/1999), convocada por resolución de la Universitat Jaume I de Castelló de fecha 5 de julio de 1999 (BOE de 30 de julio de 1999).

Según dicha resolución, el perfil de la plaza es *Ficheros y bases de datos* y *Diseño de sistemas de bases de datos*, ambas asignaturas de Ingeniería Técnica en Informática de Gestión de la Universitat Jaume I. En función de este perfil se ha redactado la presente memoria, que contiene el proyecto docente propuesto para cada una de las asignaturas y se divide en cinco capítulos.

En el primer capítulo se introduce el ámbito conceptual en el que se enmarcan las asignaturas del perfil, la materia de bases de datos. En el segundo capítulo se analiza el contexto de este proyecto docente: contexto institucional, contexto curricular, contexto profesional y contexto personal del alumnado. El tercer capítulo presenta los objetivos, en cuanto a destrezas y conocimientos, que se pretende que consiga el alumnado al cursar las asignaturas que componen el perfil de la plaza, y se explica la metodología docente que se seguirá en su impartición. El cuarto capítulo presenta una estructuración de los contenidos de la materia de bases de datos y cómo éstos se organizan en las asignaturas del perfil. Los capítulos quinto y sexto contienen los programas propuestos para las asignaturas *Ficheros y bases de datos* y *Diseño de sistemas de bases de datos*, respectivamente. Ambos capítulos tienen la misma estructura: desarrollo del programa de teoría, desarrollo del programa de prácticas y metodología docente. Cada parte del programa (teoría y prácticas) va acompañada de una bibliografía.

Capítulo 1

Ámbito conceptual

Las asignaturas que componen el perfil de la plaza a concurso, *Ficheros y bases de datos*, y *Diseño de sistemas de bases de datos*, engloban la docencia sobre la materia de bases de datos en el plan de estudios de la Ingeniería Técnica en Informática de Gestión de la Universitat Jaume I. Es por ello que en este primer capítulo se explora el ámbito conceptual de las bases de datos.

En primer lugar, se estudian las etapas de la evolución experimentada por la tecnología de gestión de datos, analizándose brevemente las características de los sistemas de cada etapa y las causas de su evolución a la etapa siguiente. A continuación, se analizan las características de la generación actual de los sistemas de gestión de bases de datos y se comentan las nuevas tendencias en esta tecnología.

1.1 Reseña histórica

En este apartado se hace un recorrido por la evolución que ha experimentado el tratamiento de la información, desde los comienzos del procesamiento automático de la misma, hasta los modernos sistemas de gestión de bases de datos[1].

[1]Este apartado es un resumen de la reseña histórica que aparece en [17]

En un principio, el tratamiento de la información se realizaba mediante programas muy simples que se almacenaban junto a los datos en dispositivos que el computador era capaz de procesar: cintas de papel y tarjetas perforadas. Los programas incluían los datos que debían procesar, o bien, los leían desde los dispositivos de almacenamiento externo, cargándolos en memoria principal. Las cintas magnéticas comenzaron a utilizarse hacia 1945 y su impacto en la gestión de la información fue decisivo. Las cintas eran ligeras y robustas, y su capacidad de almacenamiento y su velocidad, comparadas con el almacenamiento en tarjetas, eran enormes. Sin embargo, las cintas magnéticas seguían siendo dispositivos de acceso secuencial.

A principios de los años sesenta se produjeron cambios importantes que afectaron al tratamiento de la información. En primer lugar, se pasó a disponer de computadores que permitían el acceso en línea frente al clásico acceso diferido (*batch*). En segundo lugar, la aparición de los discos magnéticos como dispositivos de almacenamiento externo favoreció el desarrollo de estructuras de ficheros más complejas que permitían el acceso directo a los registros.

La situación en esta primera etapa, formada por las décadas de los cincuenta y sesenta, se caracterizó por disponer de una gestión de la información apoyada básicamente en aplicaciones muy sencillas, diseñadas de forma aislada y con independencia de otras aplicaciones. Estas aplicaciones disponían de sus propios ficheros (o copias temporales) que contenían sólo aquella parte de la información que les era necesaria. Los ficheros eran totalmente dependientes de la aplicación a la que servían y de la máquina que los soportaba, situación que provocaba la aparición de redundancias de datos e inconsistencias, además de los problemas de mantenimiento de los programas que llevaba asociados.

La situación empezó a cambiar cuando en el intento de construir aplicaciones más complejas, surgió el concepto de fichero integrado: un fichero que pudiera ser compartido por distintas aplicaciones. A partir de ese momento, se inició una sucesión de cambios en el campo de las bases de datos, guiados por el objetivo común de conseguir un almacenamiento integrado de la información común no sólo de una aplicación, sino de toda una empresa u organización.

A mediados de los años sesenta apareció el concepto de sistema de gestión de información (SGI). En su origen, el trabajo con estos sistemas consistía, básicamente, en la ejecución de los programas del paquete SGI sobre los ficheros de salida de cada una de las aplicaciones. Pero pronto se puso de manifiesto que, para una gran empresa, el número de ficheros de entrada para el SGI era excesivamente grande, con los problemas de ordenación que ello implicaba. Otros problemas que planteaba este modo de trabajo eran la interrupción del proceso global si alguna aplicación fallaba y las inconsistencias provocadas por la duplicidad de datos en los ficheros.

Por tanto, los sistemas SGI basados en los ficheros integrados no eran tan efectivos como se deseaba, lo que provocó de nuevo la necesidad de una mayor integración de los datos. El objetivo que se perseguía era disponer de un conjunto integrado de datos (la base de datos) que fuese capaz de dar servicio a cada una de las aplicaciones de la empresa. Esta base de datos debía ser creada independientemente de las aplicaciones y de forma que éstas fuesen independientes de los datos, es decir, cualquier cambio en la representación de los datos no debía requerir cambios en las aplicaciones.

Siguiendo esta idea, en el mercado de finales de los sesenta y primeros de los setenta aparecieron sistemas basados en distintos modelos de datos. Hacia 1969, con precedentes desde 1965, aparece un sistema de gestión de bases de datos histórico, el IMS/1 de IBM, con una estructuración jerárquica de los datos. Aunque fue diseñado para el proyecto espacial Apolo, acabó siendo de propósito general. Desde entonces, distintas mejoras marcaron su evolución. El sistema IMS/1 fue el sistema más representativo de la primera familia de sistemas de gestión de bases de datos: los sistemas jerárquicos. En 1967, el grupo CODASYL se interesó por las bases de datos y creó el grupo de trabajo *Data Base Task Group* (DBTG), liderado por Bachman. Las primeras recomendaciones públicas del grupo DBTG se realizaron en 1969 [21] [22] y contenían la sintaxis y la semántica de un lenguaje de definición de datos para describir bases de datos estructuradas en red, así como el lenguaje de manipulación de datos. Siguiendo estas recomendaciones, distintas casas comerciales comenzaron a construir sistemas de gestión de bases de datos que constituyeron una segunda familia, los sistemas de red de CODASYL. Una diferencia esencial separa los sistemas jerárquicos y los sistemas de red, y es que los sistemas de red nacieron a partir de un modelo previamente definido (con pretensiones de generalización y estandarización) que posteriormente se implementó en varios sistemas; sin embargo, los sistemas jerárquicos surgieron de una implementación de la que posteriormente se obtuvo la abstracción del modelo.

Cuando en 1972, el organismo de normalización estadounidense ANSI decidió abordar el tema de las bases de datos, no tomó en consideración las especificaciones CODASYL/DBTG como punto de partida. Su subcomisión X3, que cuenta con una comisión de planificación y requisitos de normalización denominada SPARC, elaboró y publicó en 1975 [4] la arquitectura ANSI/SPARC de tres niveles para sistemas de gestión de bases de datos. Estos tres niveles son el nivel externo, el nivel conceptual y el nivel interno, correspondiendo a cada nivel una descripción o esquema de toda la base de datos o de parte de ella.

En 1978, siguiendo las directrices de ANSI, las recomendaciones de CODASYL evolucionaron hacia una arquitectura de tres niveles: esquema, subesquemas y esquema de almacenamiento. Algunos de los sistemas que siguen las propuestas de CODASYL son el sistema IDS/2 de Honeywell, el sistema DMS-1100 de Univac o el sistema IDMS de BF Goodrich.

Otro suceso importante en la época fue la aparición de los índices dinámicos con estructura en árbol B [10]. Estos índices permitieron la implementación de ficheros secuenciales indexados por varias claves. La aparición de los árboles B representó el final de una carrera que, a finales de los años sesenta, se había desarrollado con el objetivo de descubrir un método general para el almacenamiento y la extracción de datos en grandes sistemas de ficheros, que proporcionara un acceso rápido a los datos con un coste mínimo. Hoy en día, se puede afirmar que los árboles B y sus variantes son la organización estándar para construir índices en los sistemas de gestión de bases de datos.

Como resultado de todos los avances producidos durante la segunda mitad de los años sesenta y a principios de los setenta, se extendió el uso de los sistemas de gestión de bases de datos y se usaron sistemas de ficheros cada vez más sofisticados. En esta etapa, el tratamiento de la información experimentó un cambio cualitativo, representado, principalmente, por la introducción en el área de los dos principios básicos de la nueva filosofía de las bases de datos: la integración de la información y la independencia de datos (entre programas y ficheros físicos).

Entre los años 1969 y 1970, Codd definió un nuevo modelo de datos, el modelo relacional [24], que dio lugar a la tercera familia de sistemas de gestión de bases de datos: los sistemas relacionales. Es el primer modelo propiamente dicho, sencillo, elegante y bien fundamentado. El modelo propone una representación uniforme y única de la información en forma de relaciones, un conjunto de operadores para manipular las relaciones, y unas reglas de integridad. Codd propuso un lenguaje de tipo algebraico (álgebra relacional) y uno de tipo lógico (cálculo relacional [25]). Además, la teoría del modelo relacional dio lugar a la teoría de las formas normales [26], que proporciona un método más riguroso de diseño de bases de datos, con el que se pueden evitar determinadas anomalías. Si Bachman es considerado el padre de las bases de datos, se puede afirmar que Codd es el padre de la investigación en este área. Ambos investigadores recibieron el premio Turing en los años 1973 [8] y 1981 [28], respectivamente. La aparición del modelo relacional abrió una larga polémica en los años setenta entre los defensores de los modelos prerrelacionales (jerárquico y de red) y los defensores del nuevo modelo. Esta polémica quedó reflejada en todas las reuniones de la asociación ACM SIGFIDET (actualmente ACM SIGMOD), teniendo su punto culminante en un gran debate que tuvo lugar en la conferencia de ACM SIGMOD del año 1975 [78].

En esencia, las posturas de los dos campos eran las que se describen a continuación. Los defensores de CODASYL argumentaban que el modelo relacional era demasiado matemático, pensaban que ningún programador iba a ser capaz de entender sus lenguajes. Incluso si se consiguiera que los programadores aprendieran sus lenguajes, sería preciso conseguir una eficiente implementación de ellos. Y un último argumento era que las transacciones en línea exigían un procesamiento orientado a registros, y no a conjuntos como se hacía en el modelo relacional. Por su parte, los defensores de las ideas de Codd argumentaban que nada era tan complicado como la propuesta de CODASYL. Cualquier consulta orientada a conjuntos era demasiado difícil de programar utilizando el lenguaje de manipulación del DBTG. Además, el modelo CODASYL no tenía una semántica bien definida.

En la segunda mitad de los años setenta, el interés por el modelo CODASYL empezó a decaer debido, principalmente, a dos hechos: la aparición de lenguajes como QUEL y SQL, que superaban el carácter excesivamente matemático de los lenguajes originalmente propuestos por Codd, y la aparición de algunos prototipos que demostraban que se podía realizar una implementación eficiente del modelo relacional. Entre estos prototipos destacan INGRES, de la Universidad de Berkeley, y System R de IBM. Más tarde, aparecieron sistemas relacionales comerciales tales como Oracle y DB2.

Los primeros problemas de implementación asociados al modelo relacional ya se han solucionado y parece también cerrada la larga polémica sobre la valoración del modelo relacional con respecto a los modelos prerrelacionales. De hecho, el interés comercial en el enfoque relacional ha sido tan intenso en los últimos años, que los sistemas relacionales se han convertido en un estándar en el mercado, pudiéndose afirmar que la década de los ochenta ha representado la extensión generalizada del uso de los sistemas de bases de datos basados en el modelo relacional.

Un último comentario sobre el modelo relacional hace referencia al hecho de que ninguna organización internacional ha intentado promover una definición estándar del mismo, pareciendo que es Codd el único depositario de esta atribución. Su propuesta original del modelo relacional [24] ha ido evolucionando pudiéndose seguir esta evolución en sus trabajos posteriores ([27] – [33]).

1.2 Generación actual

La evolución reciente de la tecnología de bases de datos viene marcada por el afianzamiento de las bases de datos orientadas a objetos, la extensión de las bases de datos relacionales y el procesamiento distribuido.

En los últimos años, han aparecido toda una serie de aplicaciones para las que los sistemas de gestión de bases de datos tradicionales se muestran inadecuados. Algunas de estas aplicaciones son el diseño y la fabricación asistidos por computador (CAD, CAM), los sistemas expertos y las herramientas CASE. En este contexto surgen las bases de datos orientadas a objetos. Su objetivo final es integrar en un único sistema la tecnología de bases de datos, con las ideas de la orientación a objetos. El ODMG (*Object Database Management Group*) [71] ha desarrollado un modelo de datos orientado a objetos que ha dado lugar al actual estándar ODMG 2.0 [16], con su lenguaje OQL.

Por su parte, los sistemas de gestión de bases de datos relacionales han ido evolucionando estos últimos años para soportar objetos y reglas, y para ampliar el lenguaje SQL y hacerlo más extensible y computacionalmente completo, dando lugar a lo que se conocen como sistemas objeto–relacionales. El desarrollo del nuevo estándar de SQL [48] ha sido más largo de lo previsto ya que se ha intentado la convergencia con el modelo ODMG y su lenguaje OQL, lo que ha obligado a profundizar en el entendimiento del propio modelo relacional.

Durante la última década, el impacto de los avances en la tecnología de las comunicaciones ha sido muy importante. Esto ha contribuido a que en las empresas se haya producido una mayor distribución de la gestión automática de la información, en contraste con la filosofía centralizadora predominante en la tecnología inicial de bases de datos. Las bases de datos distribuidas posibilitan el proceso de datos pertenecientes a distintas bases de datos conectadas entre sí. El emplazamiento lógico de cada una de las bases de datos se denomina nodo, conteniendo cada uno su sistema de gestión de bases de datos, junto con las utilidades y facilidades propias del soporte distribuido. Los nodos, por lo general, están ubicados en emplazamientos físicos distantes geográficamente, y se encuentran conectados por una red de comunicación de datos. Se puede considerar que el objetivo fundamental de las bases de datos distribuidas es la transparencia en la localización, en las copias y en la fragmentación de los datos.

Ante la creciente necesidad de datos y objetos distribuidos, la utilización de la arquitectura CORBA (*Common Object Request Broker Architecture*) [37] resulta, en muchos casos, la más adecuada, y lo será más cuando los sistemas comerciales implementen todos los servicios especificados en esta arquitectura.

1.3 Perspectivas

Los sistemas de bases de datos activas [92] han sido propuestos como un nuevo paradigma de gestión de datos que satisface las necesidades de aquellas aplicaciones que requieren una respuesta puntual a situaciones críticas. Como ejemplos se pueden citar el control del tráfico aéreo o las aplicaciones de control de plantas industriales. Este paradigma también puede ser utilizado para soportar varias de las funciones del propio sistema de gestión de bases de datos, como son el control de accesos, el control de la integridad, el mantenimiento de vistas o el mantenimiento de atributos derivados. El factor común en todas estas aplicaciones es la necesidad de responder a sucesos, tanto externos como internos al propio sistema. A diferencia de los sistemas pasivos, un sistema de gestión de bases de datos activas responde automáticamente ante determinadas circunstancias descritas por el diseñador. La mayoría de los sistemas de gestión de bases de datos comerciales incorporan la posibilidad de definir reglas, por lo que son, en cierto modo, sistemas activos.

Las investigaciones sobre la relación entre la teoría de las bases de datos y la lógica se remontan a finales de la década de los setenta. Estas investigaciones han dado lugar a las bases de datos deductivas, que permiten derivar nuevas informaciones a partir de las introducidas explícitamente por el usuario. Esta función deductiva se realiza mediante la adecuada explotación de ciertas reglas de conocimiento relativas al dominio de la aplicación, utilizando para ello técnicas de programación lógica y de inteligencia artificial [84] [85]. Sin embargo, el campo de las bases de datos deductivas, tan prometedor hace pocos años, no ha eclosionado todavía.

Los sistemas de múltiples bases de datos permiten realizar operaciones que implican a varios sistemas de bases de datos, cada uno de los cuales puede ser centralizado o distribuido. Cada sistema de bases de datos que participa es denominado componente. Si todos los sistemas de gestión de bases de datos de los diferentes componentes son iguales, el sistema de múltiples bases de datos es homogéneo; en caso contrario, es heterogéneo. Un sistema de múltiples bases de datos es un sistema federado de bases de datos si permite una doble gestión: una de carácter global, realizada por el sistema de gestión de bases de datos federadas y otra en modo autónomo e independiente del sistema federado, por parte de los sistemas componentes.

La influencia de la Web (*World Wide Web*) lo abarca todo. En su desarrollo se han ignorado las técnicas de bases de datos, por lo que se han repetido los errores cometidos en las primeras generaciones de los sistemas de gestión de bases de datos. La Web se puede ver como una nueva interface de acceso a bases de datos y muchos sistemas de gestión de bases de datos ya proporcionan herramientas para generar dinámicamente, a partir de sus datos, páginas en lenguaje HTML. Pero la Web puede también ser considerada como una inmensa base de datos, siendo éste un tema de investigación en pleno auge [2] [11].

Aunque los grandes almacenes de datos (*data warehouses*) [55] [60] son un desarrollo reciente, ya han demostrado que si son implementados convenientemente, pueden ser de gran ayuda en la toma de decisiones y en el procesamiento analítico en tiempo real OLAP (*On-Line Analytical Processing*) [34]. Los datos son extraídos periódicamente de otras fuentes y son integrados en el almacén. Estos datos, relevantes para la empresa, son no volátiles y se agrupan según diversas granularidades en el tiempo y en otras dimensiones. En la actualidad, existe una gran competencia entre las extensiones de los sistemas de gestión de bases de datos comerciales para incorporar las características de este tipo de sistemas, y la creación de productos específicos.

La explotación de datos (*data mining* o *knowledge discovery in databases*) trata de descubrir conocimientos útiles y previamente no conocidos a partir de grandes volúmenes de datos, por lo que no sólo integra técnicas de bases de datos, sino también de la estadística y de la inteligencia artificial [51]. Las investigaciones se han plasmado rápidamente en productos comerciales, con un desarrollo reciente muy espectacular.

Existen también muchos trabajos de investigación en temas tales como las bases de datos temporales y las bases de datos multimedia. Las bases de datos temporales intentan, en primer lugar, definir un modelo de datos que capture la semántica del tiempo en el mundo real, y, en segundo lugar, realizar una implementación eficiente de tal modelo [77] [83]. Los recientes avances en el almacenamiento de distintos tipos de información, como voz, imágenes o sonido, han tenido su influencia en las bases de datos dando lugar a las bases de datos multimedia [81].

La rápida evolución que la tecnología de bases de datos ha experimentado en la última década, así como la variedad de nuevos caminos abiertos, han conducido a investigadores y asociaciones interesadas, a reflexionar sobre el futuro de esta tecnología. Estas reflexiones quedan recogidas en numerosos debates y manifiestos que intentan poner orden en un campo en continua expansión.

El informe *Third–Generation Database System Manifesto* fue publicado por el *Committee for Advanced DBMS Function*, en 1990 [35]. Este comité estaba compuesto por investigadores de universidades y casas comerciales (Oracle, Digital, etc.) y fue coordinado por Stonebraker, de la Universidad de Berkeley. En el informe se exponen los puntos en común y las discrepancias del equipo, con la comunidad de las bases de datos orientadas a objetos, defendiendo una evolución desde los actuales sistemas relacionales hacia sistemas que incorporen nuevas características (sistemas de definición de tipos, herencia, etc.), que al mismo tiempo incluyan a la generación anterior y que estén abiertos a otros sistemas. En el informe *The Object–Oriented Database System Manifesto*, publicado también en 1990 [6] por un grupo de investigadores coordinado por Atkinson, de la Universidad de Glasgow, se pretende definir un sistema de base de datos orientadas a objetos. El informe define los rasgos y las características principales que debe tener un sistema para calificarse como tal, clasificándolas en obligatorias, opcionales y abiertas.

El informe *The Third Manifesto* fue publicado por Darwen y Date en 1995 [39]. El título hace referencia a los anteriores manifiestos y en él se hace una reivindicación del modelo relacional, apostando por la eliminación del lenguaje SQL y la definición de un nuevo lenguaje que respete la esencia del modelo, aunque pueda incorporar algunas características de la orientación a objetos.

En el informe *The Active Database Management System Manifesto: a Ru-*

lebase of ADBMS Features [1], se pretende clarificar cuáles deberían ser las funciones de un sistema de gestión de bases de datos activas.

Respecto a los debates sobre el futuro y las nuevas direcciones de la tecnología de bases de datos, se suceden con frecuencia. Es interesante destacar la reunión que, en mayo de 1995, mantuvieron Silberschatz, Stonebraker y Ullman [74], como continuación a una anterior del año 1990. El objetivo era revisar la evolución de la investigación en el campo de las bases de datos. En el resumen apuntan a las siguientes áreas: problemas asociados con la incorporación de objetos multimedia en las bases de datos, problemas relacionados con nuevos paradigmas para la distribución de la información, nuevos modelos de procesamiento de transacciones, facilidad en el uso y la gestión de las bases de datos, y nuevos usos de las bases de datos: explotación de datos, grandes almacenes de datos, repositorios y gestión del flujo de trabajo (*workflow management*).

El taller *Database Systems — Breaking Out of the Box*, celebrado en 1996, reunió a 19 investigadores en sistemas de bases de datos. En el informe de sus resultados [75] se detallan las barreras que se encuentran las nuevas aplicaciones que necesitan la funcionalidad de las bases de datos más allá de los confines de un sistema de gestión de bases de datos. Dos de estas aplicaciones, que se están extendiendo hoy en día, son las empresas virtuales y los sistemas de información personales. Para superar estas barreras, se proponen las siguientes líneas de investigación: extensibilidad de tipos de datos y facilidad de integración de componentes, resultados imprecisos o inciertos, bases de datos sin esquema, facilidad de uso, nuevos modelos de trasacciones y optimización de consultas.

El informe *The Asilomar Report on Database Research* [11] es el resultado de un encuentro de 16 investigadores en la materia de bases de datos (algunos de ellos estuvieron en el taller *Database Systems — Breaking Out of the Box*), celebrado en agosto de 1998. El objetivo era el de elaborar una agenda sobre investigación en sistemas de bases de datos para la década que ahora comienza. Siguiendo la misma línea que el informe del citado taller, este grupo de investigadores propone dejar para la industria los temas de investigación cuyo objetivo es el de mejorar las áreas para las que ya existen productos comerciales y que la comunidad investigadora se ocupe de ampliar la definición de gestión de bases de datos para abarcar todo el contenido de la Web y otros almacenes de datos, y también se encargue de replantear las suposiciones más fundamentales que se realizan en la materia de bases de datos, a la luz de los cambios en la tecnología. Para acelerar esta transición, recomiendan cambiar el modo en que se evalúan y se presentan los resultados de investigación. Concretamente, abogan por fomentar los trabajos a largo plazo y más innovadores, aunque sean especulativos, que el proceso de revisión sea público y más participativo, pasar las conferencias al formato poster (al menos la mayoría de ellas, ya que suelen tratar temas muy específicos) y publicar todos los trabajos de investigación en la Web. Parte de esta propuesta (revisiones públicas y trabajos accesibles en la Web) ya se está llevando a cabo por la ACM (*Assiciation for Computing Machinery*): *ACM SIGMOD Digital Review*[2] y *ACM SIGMOD Anthology*[3].

[2] <http://www.eecs.umich.edu/digital-review/> [Consulta: 30 de junio de 2000]
[3] <http://www.acm.org/sigmod/dblp/db/anthology.html> [Consulta: 30 de junio de 2000]

Capítulo 2

Contexto

En este capítulo se describen las características del contexto en el que se imparten la asignaturas que forman el perfil de la plaza a concurso, *Ficheros y bases de datos*, y *Diseño de sistemas de bases de datos*. En primer lugar se analiza el marco institucional. A continuación, en el contexto curricular, se presenta el plan de estudios de Ingeniería Técnica en Informática de Gestión vigente en la Universitat Jaume I, relacionando las asignaturas del perfil con otras asignaturas del plan. En tercer lugar se analiza qué tipo de profesionales está demandando la sociedad en el campo de la informática y, por último, se consideran los aspectos de los alumnos que se deben tener en cuenta antes de diseñar los programas de las asignaturas.

2.1 Contexto institucional

La Universitat Jaume I

Hasta el año 1991, la ciudad de Castellón dependía, en lo referente a enseñanza universitaria, de la Universidad de Valencia. Existía un Colegio Universitario donde se cursaban los primeros ciclos de Matemáticas, Psicología, Pedagogía, Historia, Ciencias Químicas y Derecho, y las titulaciones medias de Graduado

Social y Magisterio. Con el tiempo, se implantó el segundo ciclo de Ciencias Químicas debido a la fuerte demanda de estos titulados en el sector cerámico[1].

La Universitat Jaume I (UJI), que comenzó su andadura en 1991, está estructurada en tres centros: la Facultad de Ciencias Jurídicas y Económicas (FCJE), la Facultad de Ciencias Humanas y Sociales (FCHS) y la Escuela Superior de Tecnología y Ciencias Experimentales (ESTCE). Cada una de las titulaciones que imparte la universidad está adscrita a uno de estos centros.

En el curso 1991/1992, además de las titulaciones heredadas del antiguo Colegio Universitario de Castellón (que se han extinguido de forma progresiva), se pusieron en marcha varios nuevos planes de estudio. Concretamente, los nuevos planes adscritos a la ESTCE fueron los de Ingeniería Informática, Ingeniería Técnica en Informática de Gestión e Ingeniería Técnica en Diseño Industrial. En cursos posteriores se han implantado nuevas titulaciones, siendo el mapa de titulaciones actual de la ESTCE el que se muestra en la tabla 2.1.

Titulación	**Año incio**
Ingeniería en Informática	1991
Ingeniería Técnica en Informática de Gestión	1991
Ingeniería Técnica en Diseño Industrial	1991
Licenciatura en Ciencias Químicas	1993
Ingeniería Química	1993
Ingeniería Industrial	1994
Ingeniería Técnica Agrícola (Hortofruticultura y Jardinería)	1998
Ingeniería Técnica Industrial (Mecánica)	1998

Tabla 2.1 Titulaciones adscritas a la ESTCE y su año de implantación.

La UJI cuenta con una plantilla de 808 profesores dedicados a tiempo completo y a tiempo parcial, cuya capacidad docente es evaluada cuatrimestralmente por el alumnado. Igualmente, su actividad investigadora es examinada anualmente por la universidad. En el curso actual, el número de alumnos matriculados es de 12.565.

[1]El 93% de la producción nacional de este sector se genera en la provincia de Castellón.

La Escuela Superior de Tecnología y Ciencias Experimentales

La ESTCE es la responsable de la organización de las ocho titulaciones adscritas que actualmente se encuentran implantadas (tabla 1). Los departamentos o unidades predepartamentales con mayor peso docente en la ESTCE son: Ciencias Experimentales, Informática, Ingeniería Química, Matemáticas, Química Orgánica e Inorgánica y Tecnología.

Al frente de la ESTCE están la directora, el secretario y el vicedirector. Para llevar a cabo la correcta organización de las titulaciones se crearon las denominadas Comisiones Interdepartamentales de Titulación (CIT), que en los nuevos estatutos, en vigor desde 1998, han dado lugar a las Comisiones de Titulación (CT). Existe una CT por cada una de las titulaciones, que depende del centro al cual está adscrita la titulación. Cada CT está formada por representantes de todos los departamentos que imparten docencia en la titulación, una representación del alumnado y es presidida por un director, que es elegido entre el profesorado con dedicación a tiempo completo y especializado en los estudios que se imparten en la titulación.

Las responsabilidades de la dirección de titulación, según los estatutos, son: velar por la calidad docente de la titulación, procurar la actualización de los planes de estudios para garantizar su adecuación a las necesidades sociales, promover la orientación profesional del estudiantado y coordinar la realización de las prácticas externas.

La Unidad Predepartamental de Informática

La Unidad Predepartamental de Informática (UPI) se formó a partir de la separación de la antigua Unidad Predepartamental de Matemáticas e Informática, en febrero de 1993. La UPI está compuesta por tres áreas de conocimiento: Arquitectura y Tecnología de Computadores, Ciencias de la Computación e Inteligencia Artificial, y Lenguajes y Sistemas Informáticos.

En la actualidad, la UPI posee una plantilla de profesores de distintas categorías: catedráticos de universidad, titulares de universidad, titulares de escuela universitaria, ayudantes de facultad, ayudantes de escuela universitaria, asociados a tiempo completo y asociados a tiempo parcial. La UPI constituye uno de los departamentos de mayor peso en la UJI, tanto en cuanto al volumen de créditos impartidos como en cuanto al profesorado.

En el presente curso 1999/2000, la UPI está impartiendo un total de 1.735 créditos. De éstos, 469 créditos corresponden a asignaturas adscritas al área de Ciencias de la Computación e Inteligencia Artificial (CCIA), asignaturas que se reflejan en las tablas 2.2 y 2.3.

Titulación	**Asignatura**
Ingeniería en Informática	Algorítmica
	Estructura de datos I
	Automática
	Diseño de sistemas de bases de datos
	Estructura de datos II
	Robótica
	Teoría de sistemas
	Ficheros y bases de datos
	Teoría de autómatas y lenguajes formales
	Estancias en prácticas I
	Inteligencia artificial e ingeniería del conocimiento
	Estancias en prácticas II
	Lenguajes de programación III
	Nuevas tecnologías para la gestión
	Proyectos inteligentes
	Sistemas basados en el conocimiento
	Sistemas distribuidos
	Tratamiento de imágenes
	Proyectos informáticos
Ingeniería Técnica en Informática de Gestión	Algorítmica
	Estructura de datos I
	Automática
	Estructura de datos II
	Inteligencia artificial
	Robótica
	Sistemas distribuidos
	Teoría de sistemas
	Ficheros y bases de datos
	Estancias en prácticas
	Diseño de sistemas de bases de datos

Tabla 2.2 Asignaturas en las que imparte docencia el área de CCIA.

Titulación	**Asignatura**
Ingeniería Industrial	Fundamentos de la informática Control y ordenación del tráfico urbano
Maestro en Educación Musical	Informática
Derecho	Informática I
Licenciatura en Traducción e Interpretación	Informática II
Diplomatura en Gestión y Administración Pública	Informática
Licenciatura en Humanidades	Informática I Informática II
Facultad de Ciencas Jurídicas y Económicas	Herramientas informáticas para la documentación

Tabla 2.3 Asignaturas en las que imparte docencia el área de CCIA (continuación).

Tras las últimas incorporaciones, el área de CCIA está compuesta por 28 profesores, que ocupan las siguientes figuras:

- 2 catedráticos de universidad.
- 6 profesores titulares de universidad.
- 3 profesores titulares de escuela universitaria interinos.
- 5 profesores ayudantes de facultad.
- 2 profesores asociados a tiempo completo de tipo III.
- 6 profesores asociados a tiempo parcial de 6 horas.
- 4 profesores asociados a tiempo parcial de 3 horas.
- 3 becarios FPI.

Las líneas de investigación que se desarrollan en la UPI son las siguientes:

1. Bases de conocimiento temporales y sus aplicaciones.
2. Computación paralela y científica.
3. Control inteligente.
4. Investigación y desarrollo multimedia.

5. Robótica inteligente.
6. Visión por ordenador.
7. Arquitecturas avanzadas y tolerancia a fallos.
8. Sistemas y comunicaciones.
9. Gráficos por ordenador.
10. Integración de sistemas y re-ingeniería.
11. Reconocimiento automático y traducción del habla.
12. Redes neuronales.

Las líneas 1, 2, 3, 4 y 5 están prácticamente integradas en el área de CCIA. Algunos profesores del área forman parte de los grupos 6, 7 y 8.

2.2 Contexto curricular

El plan de estudios

La aparición de los estudios de informática en la UJI está ligada a la demanda, ya instaurada en la sociedad española, de técnicos cualificados en tecnologías de la información. En 1991 se ponen en marcha en la UJI las dos titulaciones de informática aprobadas por el Consejo de Universidades: Ingeniería en Informática (II) e Ingeniería Técnica en Informática de Gestión (ITIG). Ambas titulaciones son independientes la una de la otra.

De acuerdo a la definición que aparece en el BOE de 20 de noviembre de 1990, la estructura de las enseñanzas conducentes a la obtención del título de Ingeniero Técnico en Informática de Gestión es de primer ciclo y título terminal. La duración estimada de las enseñanzas es de tres años y el perfil se debe orientar a la formación y capacitación del titulado para la gestión y explotación de sistemas informáticos, pero otorgándole al mismo tiempo una formación básica que le permita proseguir los estudios de segundo ciclo conducentes al título de Ingeniero en Informática. El número mínimo de créditos que deben impartirse de cada una de las materias troncales se muestra en la tabla 2.4.

Materia troncal	Créditos
Estadística	9
Estructura de datos y de la información	12
Estructura y tecnología de computadores	9
Fundamentos matemáticos de la informática	18
Ingeniería del software de gestión	12
Metodología y tecnología de la programación	15
Sistemas operativos	6
Técnicas de organización y gestión empresarial	12

Tabla 2.4 Materias troncales de ITIG.

Teniendo en cuenta estas directrices generales, se confeccionó el plan de estudios de ITIG de la UJI, con una carga lectiva global de 250 créditos, distribuidos tal y como se muestra en la tabla 2.5.

Curso	Troncales	Obligatorios	Optativos	Lib. Config.	Total
Primero	52,5	20	0	7,5	80
Segundo	40	10	20	10	80
Tercero	12,5	40,5	27	10	90
Total	105	70,5	47	27,5	250

Tabla 2.5 Distribución de créditos por curso de ITIG en la UJI.

En las tablas que aparecen a continuación se muestra el plan de estudios de ITIG en la UJI. La tabla 2.6 muestra la distribución de las asignaturas troncales por cada materia, detallando el curso y número de créditos (teóricos, prácticos y totales). En la tabla 2.7 se indica el total de créditos de cada materia troncal del plan de estudios y en la columna *Exceso* se ha calculado el porcentaje que se ha aumentado sobre el mínimo establecido. En la tabla 2.8 aparecen las asignaturas obligatorias de cada curso, especificando sus créditos. Por último, se muestran en la tabla 2.9 las asignaturas optativas.

Materia troncal	**Asignaturas**	**Curso**	**CT**	**CP**	**Total**
Estadística	Estadística	1	3	2	5
	Investigación operativa	2	2,5	2,5	5
Estructura de datos y de la información	Estructura de datos I	1	1,5	3,5	5
	Ficheros y bases de datos	2	4	3,5	7,5
Estructura y tecnología de computadores	Introducción a la informática	1	2,5	2,5	5
	Tecnología de ordenadores	2	2,5	2,5	5
Fundamentos matemáticos de la informática	Matemática discreta	1	3	2	5
	Álgebra	1	4	3,5	7,5
	Cálculo	1	5	5	10
Ingeniería del software de gestión	Análisis y diseño de sistemas informáticos I	2	4	3,5	7,5
	Análisis y diseño de sistemas informáticos II	3	2,5	2,5	5
Metodología y tecnología de la programación	Algorítmica	1	1	4	5
	Lenguajes de programación I	1	3	7	10
Sistemas operativos	Sistemas operativos I	2	4	3,5	7,5
Técnicas de organización y gestión empresarial	Economía de la empresa	2	4	3,5	7,5
	Organización de empresas	3	4	3,5	7,5

Tabla 2.6 Distribución de asignaturas por materia troncal de ITIG en la UJI (CT: créditos de teoría, CP: créditos de prácticas).

Materia troncal	**Créditos**	**Exceso**
Estadística	10	11,11%
Estructura de datos y de la información	12,5	4,17%
Estructura y tecnología de computadores	10	11,11%
Fundamentos matemáticos de la informática	22,5	25%
Ingeniería del software de gestión	12,5	4,17%
Metodología y tecnología de la programación	15	0%
Sistemas operativos	7,5	25%
Técnicas de organización y gestión empresarial	15	25%

Tabla 2.7 Distribución de créditos de las materias troncales de ITIG en la UJI.

Asignatura	**Curso**	**CT**	**CP**	**Total**
Física	1	4	3,5	7,5
Informática y sociedad	1	4	1	5
Estructura de sistemas	1	2,5	2,5	5
Derecho de la informática	1	2	0,5	2,5
Ampliación de matemáticas	2	6	4	10
Compiladores e intérpretes	3	1,5	1,5	3
Inglés avanzado	3	5	2,5	7,5
Estancias en prácticas	3	0	30	30

Tabla 2.8 Asignaturas obligatorias de ITIG en la UJI (CT: créditos de teoría, CP: créditos de prácticas).

Asignatura	CT	CP	Total
Electrónica básica	3	2	5
Métodos numéricos	2,5	2,5	5
Lenguajes de programación II	2,5	5	7,5
Informática gráfica	2,5	2,5	5
Entornos de usuario	2,5	2,5	5
Diseño y fabricación asistida por ordenador	4	3,5	7,5
Robótica	2,5	2,5	5
Derecho empresarial	4	1	5
Seguridad y protección de la información	2,5	2,5	5
Inteligencia artificial	2,5	2,5	5
Automática	2,5	2,5	5
Teoría de sistemas	2,5	2,5	5
Teoría de autómatas y lenguajes formales	7,5	2,5	10
Metodologías y técnicas ofimáticas en las organizaciones	2,5	2,5	5
Redes informáticas	2,5	2,5	5
Ingeniería del software	4	3,5	7,5
Sistemas distribuidos	2,5	2,5	5
Auditoría informática	3	2	5
Marketing informático	2,5	2,5	5
Diseño de sistemas de bases de datos	4	3,5	7,5
Estructura de datos II	2,5	2,5	5
Gestión financiera	3	2	5
Contabilidad	3	2	5
Modelización matemática	4	3,5	7,5
Trabajo académicamente dirigido	0	10	10
Sistemas operativos II	2,5	2,5	5
Microinformática	2,5	2,5	5
Nuevas tecnologías para la gestión	2,5	2,5	5
Historia de Europa	2	1	3
Historia del pensamiento europeo	2	1	3
Historia de las instituciones jurídicas europeas	2	1	3
Historia de las instituciones económicas europeas	2	1	3

Tabla 2.9 Asignaturas optativas de ITIG en la UJI
(CT: créditos de teoría, CP: créditos de prácticas).

La UJI propone un modelo de formación que aporte visiones humanísticas para las titulaciones con perfil tecnológico. Los alumnos de las titulaciones adscritas a la ESTCE están obligados a cursar y aprobar al menos tres créditos del grupo de asignaturas optativas sobre Europa, su historia, sus instituciones y su economía (tabla 2.9).

Un aspecto común a todas las titulaciones de la UJI es la docencia de idiomas. Además de la obligatoriedad del idioma inglés en casi todas ellas, el alumnado puede optar por cursar un segundo idioma gracias a la oferta de asignaturas de libre configuración. Otro aspecto común es la obligatoriedad de realizar estancias en prácticas en instituciones y empresas. Estas estancias se deben ajustar a un mínimo de 12 créditos de carácter práctico, que en el caso de ITIG se han establecido en 30 créditos (300 horas).

La materia de bases de datos

La materia de bases de datos es fundamental en la formación de los ingenieros técnicos en informática de gestión. Estos titulados serán quienes gestionen y exploten los sistemas informáticos de empresas y organizaciones, siendo la base de datos una parte clave de estos sistemas. Cada vez más, los sistemas de gestión de bases de datos proporcionan herramientas que dan soporte a todo el ciclo de vida de los sistemas de información, por lo que su estudio resulta imprescindible.

La docencia sobre bases de datos en el plan de estudios de ITIG de la UJI está formada por la asignatura troncal *Ficheros y bases de datos* y la asignatura optativa *Diseño de sistemas de bases de datos*. La asignatura troncal forma parte del bloque *Estructura de datos y de la información*, junto con la asignatura *Estructura de datos I*. Los descripción de *Ficheros y bases de datos* en el plan de estudios es la siguiente:

> "Organización de ficheros. Acceso. Indexación y cálculo de direcciones. Agrupación y manipulación de ficheros. Protecciones. Bases de datos: tipos, organización y aplicaciones."

y la descripción de *Diseño de sistemas de bases de datos* es la siguiente:

> "Representación de la información. Métodos de acceso. Esquemas físicos y lógicos de las bases de datos. Subsistemas de integridad e independencia. Sistemas de gestión de bases de datos. Sistemas de consulta e interfase. Sistemas de quinta generación y bases de datos expertas."

La asignatura troncal se imparte en el segundo cuatrimestre de segundo curso y, hasta el curso pasado, era prerrequisito de la asignatura optativa, por lo que esta última se debía cursar en tercero. En el presente curso 1999/2000 se han eliminado todas las incompatibilidades del plan de estudios dada su inminente reforma, pero sigue siendo aconsejable cursar la asignatura optativa en tercer curso, tras haber superado la asignatura troncal.

Una asignatura directamente relacionada con las asignaturas de bases de datos es la troncal *Análisis y diseño de sistemas informáticos I*. Su objetivo fundamental es presentar al alumnado las principales técnicas y herramientas de análisis y diseño de sistemas de información. Una de las tareas a realizar en el diseño de un sistema de información es el diseño de la base de datos. La primera etapa de este diseño es el modelado conceptual de datos, que se estudia en esta asignatura y que es independiente del modelo de base de datos que se vaya a utilizar para la implementación. Esta asignatura está bien secuenciada con la asignatura troncal *Ficheros y bases de datos* ya que se encuentra en el primer cuatrimestre de segundo curso. Esto hace que en esta última asignatura se pueda partir de los conocimientos adquiridos en cuanto al modelado conceptual de datos, pudiéndose centrar en las etapas del diseño lógico y del diseño físico. Además, el paso por la asignatura de *Análisis y diseño de sistemas informáticos I* permite que el alumnado se familiarice con el ciclo de vida de los sistemas de información, al que se hace referencia en muchas ocasiones en la asignatura de *Ficheros y bases de datos*.

La asignatura troncal *Análisis y diseño de sistemas informáticos II* es continuación de *Análisis y diseño de sistemas informáticos I* y en ella se estudia la gestión de proyectos informáticos: planificación, estimación de costes, plazos, seguimiento y control del proyecto, y gestión de riesgos. La asignatura optativa *Ingeniería del software* complementa a las dos asignaturas de análisis y diseño de sistemas informáticos, profundizando en el análisis y diseño orientado a objetos, y el análisis y diseño de sistemas en tiempo real. Una última asignatura que forma parte de este bloque es la optativa *Auditoría informática*, en la que se realiza una introducción a la auditoría de sistemas de información.

Otras dos asignaturas relacionadas con las asignaturas de la materia de bases de datos y con los sistemas de información en general, son las optativas de *Entornos de usuario*, en donde se estudia el diseño y el desarrollo de entornos de usuario para las aplicaciones informáticas, y *Metodologías y técnicas ofimáticas en las organizaciones*, en la que se pretende que el alumnado adquiera conocimientos básicos sobre cómo la informática se estructura y se utiliza como herramienta de trabajo en las organizaciones y aprenda a manejar un conjunto de herramientas, presentes en la mayor parte de ellas, desde una perspectiva distinta a la habitual, recalcando que pueden ser parte fundamental de un sistema de información.

Ya que las aplicaciones informáticas que se van a diseñar y gestionar pertenecen al mundo de la gestión, es necesaria una formación básica en economía y administración de empresas que se consigue mediante las asignaturas troncales *Economía de la empresa* y *Organización de empresas* y las optativas *Contabilidad*, *Gestión financiera* y *Marketing informático.*

Reforma del plan de estudios

En el BOE de 11 de junio de 1994 se editó un Real Decreto por el que se establecían una serie de modificaciones generales sobre los planes de estudios de la LRU, algunas de las cuales se resumen a continuación:

(a) La carga lectiva en créditos fijada por las directrices generales propias para el conjunto de las materias troncales será, como mínimo, del 30% de la carga lectiva total del plan de estudios, si se trata del primer ciclo, y del 25% si se trata del segundo ciclo. Estos porcentajes no podrán ser incrementados por las universidades al elaborar los planes de estudios por encima del 15% de la carga lectiva troncal por ciclo o del 25% de la carga lectiva de cada materia.

(b) La carga lectiva global no deberá exceder más de un 15% de la carga lectiva mínima fijada en las directrices propias de la titulación. Se exceptúan de este límite las titulaciones de enseñanzas técnicas y las enseñanzas de sólo segundo ciclo, en las cuales podrá alcanzar un máximo de 75 créditos por año académico.

En el curso 1994/1995 se constituyó un comité para la revisión de los planes de estudios de las dos titulaciones de informática de la UJI. Esta iniciativa fue impulsada desde las antiguas CIT de ambas titulaciones debido a ciertas carencias que se detectaron en los planes de estudios y a las modificaciones promovidas por el Consejo de Universidades. Inicialmente, se pretendía elaborar un nuevo plan de estudios manteniendo las directrices generales de la LRU y de la misma universidad, pero adecuando algunos aspectos que, de momento, resultaban insatisfactorios. Teniendo en cuenta que el curso siguiente saldría la primera promoción de titulados en II y que ya habían finalizado dos promociones de ITIG, la modificación se podía comenzar a elaborar para ser puesta en marcha en cuanto fuera aprobada.

El comité de revisión de los planes de estudios se constituyó para trabajar de forma conjunta en ambas titulaciones. Estaba formado por profesores de los departamentos de Informática, Matemáticas, Economía y Ciencias Experimentales, además de las vicepresidentas de las CIT de cada titulación. Aunque, posteriormente, el trabajo de este comité se paralizó a la espera de las nuevas directrices del Consejo de Universidades y las nuevas directrices de la propia UJI, algunas de sus conclusiones han sido llevadas a la práctica en el presente curso 1999/2000.

En primer lugar, se han eliminado del plan de estudios todas las incompatibilidades entre las asignaturas. Estas constituían un problema burocrático y legal, y, en ocasiones, un problema para el profesorado. El alumnado podía matricularse sin tener en cuenta las incompatibilidades, pero en el momento de examinarse de una asignatura, debía tener aprobadas todas las asignaturas que eran requisito y, además, haberlo hecho en una convocatoria previa. Al eliminar las incompatibilidades, es el alumnado el que debe considerar si sus conocimientos son suficientes para enfrentarse a una determinada asignatura. Toda la información relativa a las asignaturas (objetivos, conocimientos previos, contenidos, evaluación, bibliografía, etc.) se encuentra, desde este curso, en los denominados *Libros de titulación*, que se entregan en soporte magnético al alumnado cuando realiza la matrícula y que también están accesibles a través de la Web [64].

Otra de las conclusiones del comité de revisión que ha sido llevada a la práctica desde este curso es la de establecer un mecanismo de control sobre la asignatura de *Estancias en prácticas*. Esta asignatura, obligatoria de tercer curso, no tenía prerrequisitos, por lo que, en ocasiones, no se realizaba un buen aprovechamiento de la misma al no haber cursado el alumnado ciertas asignaturas fundamentales de segundo curso, como *Ficheros y bases de datos* o *Análisis y diseño de sistemas informáticos I*. Para evitar este tipo de situaciones, se ha impuesto el requisito de que el alumnado tenga aprobado el 60% del total de créditos troncales y obligatorios para poder realizar la estancia en prácticas.

Al iniciarse, por parte de la UJI, la elaboración de una nueva normativa sobre modificación de planes de estudio, el comité de revisión se disolvió a la espera de que esta normativa se pusiera en marcha. Esta normativa, aprobada en julio de 1999, establece que la modificación de un plan de estudios se ha de realizar teniendo en cuenta los resultados de la evaluación institucional de la titulación y después del análisis de su conveniencia. Quien analiza esta conveniencia es la comisión de titulación y quien elabora la modificación son dos comisiones, una mixta universidad–sociedad, y otra académica interna.

Las titulaciones de II e ITIG fueron las primeras titulaciones de la UJI en ser evaluadas dentro de la primera convocatoria del plan nacional de evaluación de la calidad de las universidades, promovido por el Consejo de Universidades. Este proceso se inició en el curso 1996/1997, finalizando a mediados del curso 1997/1998 con la visita de la comisión de evaluadores externos, que elaboraron un informe de evaluación externo [89]. Una conclusión a destacar en este informe es la siguiente:

> "Los planes de estudio de ambas carreras tienen bastantes deficiencias en cuanto a que se requieren un número excesivo de créditos para obtener los títulos, contienen una distribución de los créditos entre asignaturas que no es común en otras unviersidades y contienen algunas lagunas en temas importantes tales como los de redes de computadores. Así pues, necesitan una redefinición."

En el período que transcurrió desde que se disolvió el comité de revisión hasta que se aprobó la nueva normativa de reforma de planes de estudio de la UJI, la Unidad Predepartamental de Informática (UPI) constituyó un grupo de trabajo que elaboró una propuesta para cada uno de los planes de estudios de informática de la UJI. Este grupo de trabajo partió de las conclusiones del comité de revisión y tuvo en cuenta las nuevas directrices del Consejo de Universidades (BOE de 16 de mayo de 1997), por las que se permite a las universidades, al elaborar y aprobar los planes de estudios, incrementar los actuales porcentajes de la carga lectiva troncal y de cada materia, siempre que con ello se consiga una mejor utilización de sus recursos y una integración de materias en un número menor de asignaturas. Por otra parte, se fija un número de créditos mínimo por cada materia o asignatura, según se trate de cuatrimestral (4,5 créditos) o anual (9 créditos), salvo que en las correspondientes directrices generales propias de los planes de estudios de que se trate se haya fijado, para las mismas, una menor carga lectiva de créditos.

En febrero de 1999, las comisiones de titulación de II e ITIG solicitaron la reforma de los planes de estudios de ambas titulaciones argumentando los siguientes motivos:

– Actualmente no hay un perfil explícito de los objetivos de las titulaciones y perfil de los titulados.

– Los planes presentan deficiencias de diseño en las incompatibilidades.

– Los planes de estudios son deficitarios en materias como redes de ordenadores y sistemas de información. No hay suficiente obligatoriedad en programación de ordenadores y es preciso aumentar el peso de los conocimientos sobre sistemas operativos.

– La optatividad es baja en proporción al número total de créditos. Las asignaturas optativas no están agrupadas en materias relacionadas o por intensificaciones, sino que son muy heterogéneas y, en algunos casos, ajenas a la informática. Las optativas que ofrece el plan superan lo que se puede financiar, por lo que hay asignaturas que no se activan.

– La organización docente se ve afectada negativamente por el exceso de créditos que debe cursar el estudiantado, el exceso de asignaturas por curso, la desestructuración de la optatividad y la dificultad de integrar las estancias en prácticas dentro del horario de un curso. Los problemas son más graves en el caso de primer curso, que en ITIG tiene 80 créditos.

– Los planes de estudios incumplen la legislación sobre carga docente. ITIG tiene 250 créditos cuando el máximo es de 225 créditos.

Las comisiones universidad–sociedad e interna ya se han constituido y su trabajo ha comenzado en el mes de mayo de este año 2000. Se espera que estas comisiones tengan en consideración la propuesta de modificación de la UPI. En esta propuesta, los contenidos de bases de datos en la ITIG son los que se muestran en la tabla 2.10.

Asignatura	**Curso**	**CT**	**CP**	**CL**	**Total**
Bases de datos	2	4,5	1,5	3	9
Sistemas de bases de datos	optativa	3	0	1,5	4,5
Bases de datos distribuidas y heterogéneas	optativa	3	0	1,5	4,5

Tabla 2.10 Asignaturas de bases de datos en la propuesta de la UPI para ITIG (CT: créditos de teoría, CP: créditos de problemas, CL: créditos de laboratorio).

Los contenidos de cada una de las asignaturas propuestas son los siguientes:

- Bases de datos: Modelo de datos relacional. Álgebra y cálculo relacional. Bases de datos relacionales. Diseño lógico y físico de bases de datos relacionales. Lenguajes de consulta estándar.
- Sistemas de bases de datos: Sistemas de gestión de bases de datos. Diseño lógico y físico avanzado. Procesamiento de consultas. Seguridad y recuperación de bases de datos.
- Bases de datos distribuidas y heterogéneas: Características de las bases de datos distribuidas y heterogéneas. Diseño lógico de bases de datos distribuidas. Integración de varios esquemas. Procesamiento de consultas distribuidas.

Recomendaciones nacionales e internacionales

En las Jornadas de Investigación y Docencia en Bases de Datos (JIDBD) de 1997 se creó un grupo de trabajo con el objetivo de definir los contenidos de bases de datos en los estudios de informática. Dicho grupo de trabajo elaboró una primera propuesta que se presentó en las Jornadas de 1998 (JIDBD'98) donde fue ampliamente discutida. Los resultados de tal discusión permitieron enriquecer la propuesta inicial dando lugar a un documento que se publicó en enero de 1999 [13]. Las asignaturas se dividen en esenciales y optativas, y su duración y contenidos para ITIG se muestran en la tabla 2.11.

En la tabla 2.12 se muestra el número de créditos troncales/obligatorios y optativos correspondientes a la materia de bases de datos que se imparten en las titulaciones de ITIG de las universidades españolas. Como se puede observar, ninguna universidad alcanza la propuesta. Aunque algunas se acercan a la propuesta de troncalidad (alrededor de un 35% tienen entre 12 y 14 créditos), la optatividad y por lo tanto, el volumen total de créditos, quedan muy alejados. Quizá esto es así porque al formular la propuesta es posible que se haya considerado aisladamente esta materia y no en el ámbito de una titulación en el que debe haber cabida para otras materias también fundamentales. Probablemente, si se unieran todas las propuestas formuladas para cada materia troncal de ITIG, se obtendría una titulación de ciclo corto con una carga lectiva equivalente a una titulación de ciclo largo. Es por todo esto que considero que la propuesta realizada por la UPI es bastante razonable para la titulación de ITIG en la UJI.

Asignatura	**Tipo**	**Créditos**	**Contenidos**
Bases de datos I	Esencial	9	Ficheros Introducción a las bases de datos Arquitectura de un SGBD Modelo relacional Lenguaje SQL Introducción al modelo entidad–relación Introducción a la normalización
Bases de datos II	Esencial	6	Evolución de los SGBD Introducción a concurrencia, recuperación y optimización Bases de datos distribuidas Introducción a la orientación a objetos Futuro de las bases de datos
Diseño de bases de datos	Optativa	4 a 6	Metodología de diseño de bases de datos Modelado conceptual Modelo entidad–relación extendido Diseño lógico Normalización y reestructuración de relaciones Diseño físico Diseño de bases de datos distribuidas
Administración de bases de datos	Optativa	4 a 6	Seguridad Integridad Recuperación y evolución Optimización Administración de bases de datos distribuidas
Aplicaciones de bases de datos	Optativa	6	Arquitecturas de las aplicaciones de BD Entornos de desarrollo de aplicaciones de BD

Tabla 2.11 Asignaturas de bases de datos propuestas para ITIG en las JIDBD'98.

Universidad	**OBL**	**OPT**	**Total**
Universidad de Almería	13,5	12	25,5
Universidad de Cádiz	12	6	18
Universidad de Córdoba	9	4,5	13,5
Universidad de Granada	12	12	24
Universidad de Huelva	6,75	6,75	13,5
Universidad de Jaén	13,5	0	13,5
Universidad de Málaga	12	0	12
Universidad de Sevilla	9	0	9
Universidad de Zaragoza	12	0	12
Universidad de Las Palmas de Gran Canaria	9	0	9
Universidad de Castilla – La Mancha	9	12	21
Universidad de Burgos	6	6	12
Universidad de Valladolid	7,5	0	7,5
Universitat Autònoma de Barcelona	6	0	6
Universitat de Girona	10,5	0	10,5
Universitat de Lleida	12	6	18
Universitat Politècnica de Catalunya	10,5	6	16,5
Universitat Rovira i Virgili	9	4,5	13,5
Universidad de Alcalá de Henares	9	0	9
Universidad Carlos III de Madrid	14	5	19
Universidad Complutense de Madrid	12	0	12
Universidad Politécnica de Madrid	9	14	23
Universidad Rey Juan Carlos	9	6	15
Universitat d'Alacant	12	0	12
Universitat Jaume I de Castelló	7,5	7,5	15
Universitat Politècnica de València	12	10,5	22,5
Universidad de Extremadura	6	6	12
Universidad da Coruña	7,5	9	16,5
Universidad de Vigo	7,5	6	13,5
Universitat de Les Illes Balears	9		
Universidad de Oviedo	9	0	9
Universidad de Murcia	6	6	12
Propuesta grupo trabajo JIDBD'98	15	18	33
Propuesta UPI	9	9	18

Tabla 2.12 Créditos de bases de datos en ITIG en las universidades españolas (OBL: créditos troncales u obligatorios, OPT: créditos optativos).

La Conferencia de Decanos y Directores de Titulación de Informática realizó en 1999 una propuesta de modificación de la troncalidad de las tres titulaciones universitarias. Concretamente, para ITIG proponen que el bloque troncal de *Estructura de datos y de la información*, en el que aparece la materia de bases de datos, aumente el número de créditos de 12 a 18. Estos aumento de créditos corresponde a esta materia, ya que, a los descriptores "ficheros, bases de datos" se han añadido "sus modelos y sus sistemas de gestión".

Sobre las recomendaciones internacionales, comentar que en los planes de estudios para *Information Systems*[2] (*undergraduate degree*) que propone conjuntamente ACM, AIS (*Association for Information Systems*) y AITP (*Association of Information Technology Professionals*) en un informe aparecido en 1997 [44], se considera que el curriculum está formado por los siguientes elementos:

- Fundamentos de sistemas de información: hardware, software, programación, datos y objetos.
- Fundamentos de organización de empresas: contabilidad, microeconomía y comportamiento de las organizaciones.
- Cuerpo central de sistemas de información: bases de datos, análisis, modelización y diseño, redes, gestión de proyectos y políticas y estrategias de tecnologías de la información.

Además, se pone énfasis en la integración de la perspectiva tecnológica con la organizativa y en la transmisión de las siguientes habilidades: pensamiento analítico y crítico, trabajo en equipo y comunicación interpersonal, y perspectiva sobre sobre el mundo profesional real.

Sobre la materia de bases de datos, se proponen dos cursos en los que se recorre todo el ciclo de vida de una base de datos, que forma parte de un sistema de información. Estos cursos son: *Analysis and Logical Design* y *Physical Design and Implementation with a DBMS*. Los conceptos básicos sobre bases de datos y sistemas de gestión de bases de datos están incluidos en una asignatura introductoria denominada *Fundamentals of Information Systems*. Los conceptos sobre organizaciones de ficheros y estructuras de acceso se encuentran en la asignatura denominada *Programming, Data, File and Object Structures*. Además, cada centro deberá ofrecer varias asignaturas de áreas emergentes, como ahora podrían ser los grandes almacenes de datos (*data warehouses*), la explotación de datos (*data mining*) o las aplicaciones de bases de datos en la Web.

Se puede concluir que las propuestas emanadas del grupo de trabajo de las JIDBD coinciden, en cuanto a contenidos, con las propuestas de ACM/AIS/AITP. Además, la propuesta española añade aspectos de administración de los sistemas de gestión de bases de datos (concurrencia, recuperación, etc.) que en las

[2]Se puede considerar como la titulación de Estados Unidos y Canadá que más se aproxima a ITIG.

recomendaciones internacionales no aparecen para los titulados en sistemas de información, sino para los titulados en ingeniería del software. Hay que tener en cuenta que en Estados Unidos y Canadá (países a los que se dirigen estas asociaciones) hay más titulaciones de informática, por lo que éstas se pueden hacer más específicas. En realidad, la ITIG en España es una mezcla de *Information Systems*, *Software Engineering* e, incluso, *Computer Science*. La cuarta titulación de informática es estos países es la de *Computer Engineering*, más similar a Ingeniería Técnica en Informática de Sistemas (ITIS).

2.3 Contexto profesional

En este apartado se analiza qué tipo de profesional informático está demandando la sociedad, para qué funciones y con qué conocimientos y habilidades [14] [15].

A continuación se citan algunos párrafos de un artículo aparecido en la sección de negocios del diario El País titulado *Escasez en la teconología* (Pablo García de Sola, 17 de octubre de 1999). Este artículo explica el restraso en multitud de empresas españolas en cuanto al uso de tecnologías informáticas.

> "El perfil de los profesionales informáticos demandados por las empresas es de especialistas con un alto nivel técnico que, además, aporten una gran visión y conocimiento del negocio. En la empresa española existe todavía un cierto retraso en la utilización de tecnologías informáticas respecto al resto de Europa. Este retraso lo muestran principalmente las pequeñas y medianas empresas que tienen unos bajos índices de implantación tecnológica y elevados plazos de renovación de soluciones y tecnología.
>
> A pesar de que gran parte del éxito empresarial depende del empleo eficiente y racional de estas tecnologías, los recursos destinados por las empresas a este capítulo siguen siendo escasos. Para que un proyecto informático tenga éxito, y las soluciones tecnológicas se aprovechen al máximo, la clave son los recursos humanos, puesto que no sólo se trata de implantar bien un sistema de información o tener buenos programadores, sino que la actitud de los usuarios y la formación que se les facilite son esenciales."

La mayoría de puestos de trabajo para informáticos no están en las grandes empresas usuarias o proveedoras de tecnología, sino en las empresas de servicios o subcontratación y en la enorme cantidad de pequeñas y medianas empresas con poca cultura informática. Todas esas empresas buscan un rendimiento inmediato y piden unos conocimientos de sistemas y herramientas que los titulados informáticos no suelen tener. De hecho, más del 95% de los conocimientos que se piden en las ofertas de trabajo informático que aparecen en la prensa, son totalmente desconocidos para los recién titulados de ITIG de la mayoría de universidades. Sin embargo, aunque los titulados desconozcan las herramientas y prácticas habituales, aprenden rápidamente y se adaptan a nuevas tecnologías con mayor facilidad que los profesionales informáticos tradicionales que sólo dispusieron de la vía de la experiencia.

Los sectores que más empleo cualificado han generado en España en los últimos años han sido el informático, la industria ligera, las telecomunicaciones y los servicios. El de las infocomunicaciones, que engloba informática y telecomunicaciones, es el sector que ha mostrado mayor crecimiento, generando casi el 20% de la oferta de empleo cualificado. Entre las ofertas de empleo que requieren experiencia ocupan las primeras posiciones ingeniería informática (7,3%), ingeniería técnica industrial (6,8%), ingeniería técnica informática (5,3%) e ingeniería en telecomunicaciones (5,2%). Más de la mitad de la oferta de empleo está orientada a titulaciones técnicas, cuya participación en la población universitaria apenas supera el 20%[3].

2.4 Contexto personal

La mayoría de los alumnos que cursan ITIG provienen directamente de COU, aunque hay un porcentaje importante de alumnos de FP (entre el 20% y el 30% de los de nuevo ingreso). Su motivación para escoger esta titulación frente a II es, fundamentalmente, su menor duración. Ven que en tres cursos pueden tener un título que les capacita para trabajar y que, además, tienen la posibilidad de pasar al segundo ciclo de II sin necesidad de un curso puente. También suelen influir sus expectativas profesionales, que son, en su mayoría, el desarrollo y mantenimiento de aplicaciones informáticas. También se matriculan en ITIG, antes que en II, profesionales interesados en la informática que ven en la titu-

[3]Datos obtenidos del artículo aparecido en la sección de negocios del diario El País de 26 de septiembre de 1999, *Ingenieros y economistas, los más solicitados*, de Pedro García de Sola.

lación una oportunidad de prosperar en el mundo laboral. Estos últimos suelen acudir a los grupos de la tarde.

La motivación por la materia de bases de datos es alta por parte de los estudiantes dada la gran aplicación de esta materia. Esto lo saben tanto por las enseñanzas teóricas que reciben en la universidad, como por la experiencia de sus compañeros que están realizando o han realizado la estancia en prácticas: alrededor del 60% de estas estancias conlleva el desarrollo de una aplicación de bases de datos.

La primera de las asignaturas de la materia de bases de datos en la titulación es troncal (*Ficheros y bases de datos*), por lo que todos los alumnos están obligados a cursarla. La segunda asignatura (*Diseño de sistemas de bases de datos*), de carácter optativo, es cursada por un 43% de los estudiantes que terminan la carrera. Esta asignatura no está activada en ITIG, pero sí en II, por lo que los alumnos de ITIG la cursan como libre configuración.

Capítulo 3

Objetivos y metodología docente

En este capítulo se presentan, en primer lugar, los objetivos que se pretenden conseguir como resultado del aprendizaje de los alumnos al cursar las asignaturas que comprenden el perfil de la plaza. Estos objetivos serán los que guiarán después la elección de los contenidos de cada una de las asignaturas. A continuación, se expone la metodología docente que se va a seguir al impartir estas asignaturas.

3.1 Formulación de objetivos

En este apartado se plantean como objetivos los conocimientos y destrezas en la materia de las bases de datos que se considera deben conocer los ingenieros técnicos en informática de gestión. Estos objetivos se han clasificado como esenciales y opcionales. Los objetivos esenciales son aquellos que se consideran de obligado conocimiento, mientras que los opcionales son aquellos que permiten ampliar conocimientos en la materia a modo de intensificación. En relación al perfil de la plaza a concurso, los objetivos esenciales serán lo que se utilizarán para elaborar el programa de la asignatura troncal *Ficheros y bases de datos*, mientras que los objetivos opcionales serán utilizados para elaborar el programa de la asignatura optativa *Diseño de sistemas de bases de datos*.

Objetivos esenciales

El primer objetivo de carácter esencial es que el alumnado conozca los distintos modos en que los registros se pueden organizar dentro de los ficheros físicos y los distintos métodos adicionales que se pueden utilizar en el acceso a los datos. Para ello, es preciso conocer, de modo muy básico, cómo funcionan los dispositivos de almacenamiento de datos, fundamentalmente los discos, llegando a ser conscientes del gran coste que supone un acceso continuado a los mismos. Este gran coste es el que ha movido a la comunidad investigadora a encontrar no sólo modos de organizar los ficheros, sino también estructuras de acceso adicionales que permitan un abaratamiento de este coste. El fruto de estas investigaciones, realizadas en la década de los setenta, fue muy postivo y prueba de ello es que hoy en día se siguen utilizando las mismas organizaciones de ficheros y estructuras de acceso de entonces. Sin embargo, dados los últimos avances en la tecnología, esta situación va a empezar a cambiar, y esto es algo que también se quiere transmitir al alumnado. Las CPU serán más rápidas, los discos serán más grandes y habrá grandes avances en las velocidades de comunicación. Es muy probable que dentro de una década sea común tener un terabyte de memoria principal, por lo que la mayor parte de los datos necesarios para las aplicaciones podrán residir en memoria principal. Estos cambios en la tecnología provocarán un replanteamiento de la organización de los datos y los métodos de acceso, ya que, por ejemplo, es bien conocido el hecho de que los mejores métodos de acceso a datos en disco no son los más apropiados para el acceso a datos en memoria principal.

El segundo objetivo esencial es introducir a los alumnos en la tecnología de bases de datos. Conocer sus predecesores, los sistemas de ficheros, y los inconvenientes que éstos plantean, para motivar así la aparición de los sistemas de bases de datos y la aplicación fundamental que los maneja: el sistema de gestión de bases de datos. En principio, sólo interesa conocer qué funciones realiza el sistema de gestión de bases de datos y no cómo las lleva a cabo ni cómo se puede influir sobre ellas. Se considera imprescindible que el alumnado llegue a ser usuario experto de estos sistemas, no el que llegue a administrarlos.

El tercer objetivo considerado de carácter fundamental es conocer a fondo el modelo de bases de datos relacional: la estructura de datos que ofrece, las reglas de integridad que son de obligado cumplimiento y los lenguajes que permiten manipular los datos de una base de datos relacional. Lo que se persigue mediante este objetivo es que el alumnado conozca en profundidad el lenguaje SQL y sea capaz de resolver cualquier consulta de datos mediante este lenguaje, por compleja que sea. Se pretende también transmitir la importancia de este lenguaje, que es un estándar para las bases de datos relacionales, las de uso más extendido en la actualidad.

El cuarto objetivo esencial consiste en introducir al alumnado en el diseño de bases de datos relacionales. En este punto, por los conocimientos adquiridos en otras asignaturas, el alumnado es capaz de obtener el modelo de datos conceptual que requiere un determinado sistema de información. Es por ello que el objetivo aquí es el de tomar este esquema conceptual, interpretarlo y ser capaces de transformarlo en un esquema lógico adecuado al modelo relacional, por lo que el dominio de este modelo es de extrema importancia (este dominio se consigue mediante el tercer objetivo esencial). Es muy importante transmitir al alumnado la gran importancia que tiene el seguir siempre una metodología en el diseño y qué problemas se pueden presentar por el hecho de no hacerlo.

El quinto y último objetivo esencial es de carácter introductorio y lo que persigue es que el alumnado conozca la existencia de herramientas que permiten el desarrollo de aplicaciones que acceden a bases de datos. Concretamente, se pretende experimentar con los precompiladores, herramientas que aceptan sentencias de SQL embebidas en programas escritos en lenguajes de programación de tercera generación, y también experimentar con generadores de formularios, herramientas de cuarta generación cuya filosofía y modo de funcionamiento difiere en gran medida de la filosofía procedural a la que el alumnado se encuentra acostumbrado.

Objetivos opcionales

El primer objetivo de carácter opcional es el de introducir al alumnado en nuevos sistemas de bases de datos, cuyo uso se está extendiendo hoy en día y que incorporan características que permiten ampliar el campo de aplicación de las bases de datos al romper con algunas limitaciones que los sistemas anteriores presentaban. En concreto, se han escogido las bases de datos activas, que permiten que la base de datos presente cierto comportamiento, las bases de datos orientadas a objetos, que incorporan muchas de las ventajas de la orientación a objetos, y las bases de datos distribuidas, que abren el acceso a datos remotos, imprescindible en multitud de aplicaciones. Es importante que el alumnado conozca cómo los productos comerciales basados en el modelo relacional tratan de ponerse al día, incorporando características de las bases de datos activas (mediante los denominados disparadores o *triggers*), de las bases de datos orientadas a objetos (han dado lugar a los sistemas objeto–relacionales, ampliamente aceptados) y de las bases de datos distribuidas. Las casas comerciales son conscientes de las ventajas que estos nuevos sistemas aportan y tratan de incorporar a sus productos aquellas que consideran más necesarias para sus clientes y el potencial mercado que éstas les pueden abrir.

El segundo objetivo opcional es el de conocer cómo el sistema de gestión de bases de datos lleva a cabo sus funciones. Se pretende llegar a conocer sobre cuáles de estas funciones puede ejercer alguna influencia el administrador del sistema, aunque no se pretende saber cómo, cuándo y de qué modo ejercerla. Uno de los aspectos que se consideran más importantes en este objetivo es el de transmitir la importancia del módulo del sistema que se encarga de la optimización de consultas y cómo puede influir sobre él tanto el esquema físico de la base de datos, como el modo en que se escriban las sentencias SQL que la consultan.

El tercer objetivo que se considera opcional se ocupa de las aplicaciones de bases de datos en la Web. La Web ha abierto nuevos modos de trabajo cuyo uso se está extendiendo muy rápidamente. Un ejemplo de ello es el comercio electrónico. Mediante este objetivo se pretende introducir al alumnado en este tipo de aplicaciones y en algunas de las herramientas que permiten generarlas, despertando su interés por esta nueva tecnología.

El cuarto y último objetivo de este bloque es conocer que existen otras nuevas tecnologías de bases de datos que abren aún más el campo de aplicación de esta materia, no sólo a otros ámbitos, sino dentro del ámbito de la gestión, a otros estamentos en empresas y organizaciones, como son, por ejemplo, aquellas personas que realizan la toma de decisiones. Mediante este objetivo también se pretende fomentar la curiosidad del alumnado por las nuevas tecnologías y constatar que la materia de las bases de datos está en continuo crecimiento.

3.2 Metodología docente

En este apartado se expone la metodología docente que planteo para impartir las asignaturas que componen el perfil de la plaza.

El método que se utiliza para las clases en las que se exponen los conceptos del temario por parte del profesor, es la explicación o clase magistral. El propósito de la explicación es transmitir la información, generar la comprensión por parte del alumnado y estimular su interés. Cuando se utiliza este método es muy importante tener en cuenta la respuesta de los alumnos. Las reacciones más inmediatas son normalmente signos no verbales que el profesor debe interpretar, aunque también son muy importantes las respuestas manifestadas a más largo plazo, como son los cambios de actitud y comprensión de los estudiantes. El hecho de que en estas clases se utilice la explicación no significa que sólo se realice transmisión de información, sino que también se promueve la participación activa del alumnado, dando lugar a la discusión.

En las clases de explicación se utilizan transparencias que contienen los conceptos teóricos y también ejemplos. La pizarra se utiliza constantemente como apoyo para explicar los conceptos y los ejemplos, así como para plantear pequeños ejercicios, invitando a que el alumnado razone la respuesta utilizando los conceptos que se acaban de explicar. De este modo, los alumnos participan en la clase proponiendo respuestas o planteando dudas sobre el nuevo concepto. La estructura de las clases de explicación es la siguiente:

- Enlazar con la sesión anterior (alrededor de 3 minutos).
- Informar del contenido de la clase (de 3 a 7 minutos, dependiendo de si se empieza un nuevo tema).
- Explicación del tema, intercalando actividades cortas: ejemplos, ejercicios, curiosidades (45 minutos).
- Síntesis de la clase y enlazar brevemente con la sesión siguiente (alrededor de 5 minutos).

Cuando los temas explicados requieren la realización de problemas, se dispone de un boletín de ejercicios de los cuales algunos se realizan en clase. Estas clases se planifican con anterioridad, de modo que el alumnado sabe cuándo van a tener lugar y cuáles van a ser los ejercicios concretos que se van a resolver en cada una de ellas. Para favorecer la participación, se resuelven los ejercicios en pequeños grupos y a dos de estos grupos, escogidos al azar, se les facilita una transparencia en la que deben escribir la resolución del problema. Después, la transparencia se proyecta para toda la clase y entre todos, alumnos y profesor, se discute si el ejercicio ha sido bien resuelto o si se puede mejorar. Es la etapa de corrección pública. Este método, una alternativa al método clásico de salir a la pizarra a realizar el ejercicio, lo considero más interesante por el hecho de que al resolver el ejercicio se tiene la posibilidad de discutirlo en grupo, favoreciendo este modo de trabajo. Además, el alumnado se siente menos presionado y observado al no tener que moverse de su sitio, lo que le permite pensar con más libertad e incluso arriesgarse más, cuando no tiene la seguridad de estar realizando correctamente el ejercicio. Al permanecer en su pupitre, el alumno dispone de toda la información de consulta que pueda necesitar, como son las copias de las transparencias de clase y sus propios apuntes.

Mientras los alumnos realizan el ejercicio, el profesor pasea por el aula resolviendo dudas a los grupos de trabajo que no tienen la transparencia. A los grupos que hacen el ejercicio en la transparencia no se les ayuda, ya que la intención es la de encontrar, más tarde y entre todos, los posibles fallos cometidos. Estos fallos suelen ser los mismos que han provocado las dudas en el resto de compañeros, con lo que estos últimos tienen la seguridad de estar dando la respuesta correcta y se atreven a partipar en la etapa de corrección. El motivo de escoger exactamente dos grupos de trabajo es que con dos hay más posibilidades de detectar errores que con uno. Con más de dos grupos los errores tienden a repetirse y el proceso puede llegar a hacerse aburrido. En esta etapa de corrección el profesor también puede proponer otros modos de resolver o mejorar el ejercicio.

Cuando en la asignatura se da la posibilidad al alumnado de presentar trabajos en clase, éstos pueden versar sobre temas del temario, o sobre otros temas que quedan fuera de él, pero que también son de interés. En cualquier caso, el profesor siempre debe dar el visto bueno a los temas de los trabajos y proporciona la información para el desarrollo de los mismos. Cuando el tema del trabajo forma parte del temario de la asignatura, la clase correspondiente a dicho tema se utiliza para que el grupo que ha desarrollado el trabajo, lo presente al resto de la clase. Si el tema tratado en el trabajo no formaba parte del temario, entra a formar parte del mismo y también se expone en clase.

Las clases de presentación de trabajos tendrán siempre la misma estructura: una vez expuesto el tema en clase, habrá un debate con preguntas del alumnado y del profesor. Todos los componentes del grupo deberán intervenir en la presentación, que tendrá una duración de 20 minutos, más 10 minutos de debate.

Es muy importante fomentar el uso de las tutorías, no sólo para consultar dudas sino también para realizar un seguimiento del desarrollo de los trabajos.

Las clases de prácticas se realizan en las aulas informáticas. Los grupos son de 30 alumnos, disponiendo cada uno de ellos de un ordenador. Para las prácticas los alumnos deben llevar el trabajo previamente preparado, por lo que la estructura de las clases es, en general, la siguiente:

- Explicación breve del trabajo a desarrollar en la clase, resolviendo las posibles dudas surgidas durante la preparación de la práctica (alrededor de 15 minutos).
- Desarrollo de la práctica por parte de los alumnos, consultando dudas con el profesor (2 horas y 30 minutos).
- Explicación breve del trabajo a desarrollar en la próxima clase (alrededor de 15 minutos).

Algunas clases deben seguir una estructura diferente, bien por ser la primera clase, o bien por ser una clase en la que se introduce el uso de una nueva herramienta. En este caso, después de realizar una breve introducción, la clase es dirigida por el profesor con la ayuda de un guión que siguen los alumnos.

3.3 Evaluación

Cuando en la evaluación se realiza una prueba objetiva para valorar los conocimientos teóricos, se tienen en cuenta las siguientes reglas en su elaboración:

- La redacción de los enunciados debe ser clara y sencilla, lo más breve posible y no ambigua.

- Cada ítem debe tratar aspectos importantes del contenido y no aspectos triviales. No se debe fomentar la memorización.
- Cada ítem ha de ser independiente, de manera que la respuesta de un ítem no se debe requerir para contestar a otro ítem.
- Proporcionar información relevante y necesaria para contestar el ítem, sin añadir información adicional que no sea relevante para su resolución.
- Evitar los enunciados negativos.

En la correción se elimina la influencia del azar en las respuestas correctas de la forma habitual:

$$N - \frac{n}{i-1}$$

donde N es el número de respuestas correctas, n es el número de respuestas erróneas e i es el número de alternativas en cada ítem.

Cuando en la evaluación se realiza una pueba de resolución de problemas, se permite consultar libros y apuntes.

Además, para superar las asignaturas se exige la asistencia a ciertas sesiones de prácticas y se controla su aprovechamiento. Para esto último se propone un pequeño ejercicio durante la clase, que debe ser presentado al profesor al finalizar la misma.

Capítulo 4

Estructuración de los contenidos de la materia

En este capítulo se realiza una enumeración de los contenidos de la materia de bases de datos. Estos contenidos se organizan en bloques temáticos, para los que se ofrece una breve justificación. A continuación, se especifica qué contenidos de la materia de bases de datos se han incluido en cada una de las asignaturas del perfil de la plaza: *Ficheros y bases de datos*, y *Diseño de sistemas de bases de datos*, para cumplir los objetivos propuestos en el capítulo anterior. Por último, se presentan distintas fuentes de información sobre la materia que nos ocupa.

4.1 Justificación de los bloques temáticos

Los contenidos de la materia de bases de datos se han dividido en ocho bloques temáticos. El primer bloque, *Conceptos fundamentales*, presenta los aspectos básicos de los sistemas de bases de datos, haciendo un repaso de sus predecesores, los sistemas de ficheros. El objetivo es el de conocer las características y componentes de un sistema de bases de datos y comprender su modo general de funcionamiento. En este bloque se ha incluido también un estudio de las principales organizaciones de ficheros y estructuras de acceso, ya que más tarde se utilizarán para almacenar físicamente la base de datos. El segundo bloque, *Bases de datos relacionales*, incluye los temas relativos al modelo relacional, puesto que es el modelo de bases de datos más extendido actualmente. Además de conocer el modelo, se pretende dominar el manejo de bases de datos relacionales mediante el lenguaje SQL. En el tercer bloque, *Planificación y diseño de bases de datos*, se estudia el ciclo de vida de los sistemas de información desde un enfoque orientado a los datos para tratar en profundidad todas las etapas del diseño de una base de datos relacional. Se pretende llegar a realizar correctamente el diseño de bases de datos relacionales, diseño que será verificado mediante la normalización. El cuarto bloque presenta las *Técnicas de implementación y administración de los sistemas de gestión de bases de datos*. El objetivo es conocer las funciones, características, estructura e implementación de los sistemas de gestión de bases de datos relacionales. Para ello, se estudian todos los aspectos del procesamiento de consultas y transacciones (optimización, concurrencia, recuperación, integridad) y el control de la seguridad. En el quinto bloque se presentan las *Bases de datos orientadas a objetos*, un modelo de datos que se está extendiendo para cubrir los requerimientos de otro tipo de aplicaciones, distinto de las típicas aplicaciones de gestión. Muchas de estas aplicaciones son de diseño asistido por computador. El sexto bloque se ocupa de las *Bases de datos distribuidas* y los problemas fundamentales de su diseño e implementacion. El séptimo bloque está dedicado a las distintas técnicas utilizadas en la implementación de *Aplicaciones de bases de datos* y el último bloque, *Bases de datos avanzadas*, presenta tecnologías emergentes de bases de datos, algunas de ellas aún poco maduras, pero todas ellas con un gran futuro.

4.2 Relación de temas

1. Conceptos fundamentales.
   - 1.1. Organizaciones de ficheros y estructuras de acceso.
   - 1.2. Sistemas de bases de datos. Arquitectura de los sistemas de gestión de bases de datos.
   - 1.3. Modelos de bases de datos.
2. Bases de datos relacionales.
   - 2.1. Modelo relacional.

2.2. Álgebra relacional y cálculo relacional.

2.3. Lenguaje SQL.

2.4. Vistas.

3. Planificación y diseño de bases de datos.

   3.1. Ciclo de vida de los sistemas de información.

   3.2. Diseño conceptual. Modelo entidad–relación.

   3.3. Diseño lógico de bases de datos relacionales. Normalización.

   3.4. Diseño físico de bases de datos relacionales.

   3.5. Diseño de vistas.

   3.6. Herramientas CASE en el diseño de bases de datos.

4. Aplicaciones de bases de datos.

   4.1. SQL embebido.

   4.2. Lenguajes de cuarta generación (4GL).

   4.3. Disparadores (*triggers*).

   4.4. Interfaces gráficos mediante librerías de clases visuales.

   4.5. Interfaces para la Web.

   4.6. Integración en entornos ofimáticos.

5. Técnicas de implementación y administración de sistemas de gestión de bases de datos.

   5.1. Diccionario de datos.

   5.2. Procesamiento de consultas y optimización.

   5.3. Procesamiento de transacciones.

   5.4. Control de concurrencia.

   5.5. Recuperación.

   5.6. Seguridad.

   5.7. Integridad.

6. Bases de datos orientadas a objetos.

   6.1. Modelización semántica y orientación a objetos.

6.2. Diseño de bases de datos orientadas a objetos.
6.3. Sistemas de gestión de bases de datos objeto–relacionales.
6.4. SQL:1999 (SQL3).

7. Bases de datos distribuidas.

7.1. Arquitectura de un sistema distribuido.
7.2. Diseño de bases de datos distribuidas.
7.3. Arquitectura cliente–servidor.
7.4. Administración de bases de datos distribuidas.

8. Bases de datos avanzadas.

8.1. Bases de datos activas.
8.2. Bases de datos y la Web.
8.3. Bases de datos temporales y espaciales.
8.4. Bases de datos multimedia.
8.5. Bases de datos deductivas.
8.6. Grandes almacenes de datos (*data warehousing*) y explotación de datos (*data mining*).
8.7. Bases de datos documentales.
8.8. Bases de datos móviles.
8.9. Flujos de trabajo (*workflow management*).

4.3 Distribución de temas por asignaturas

En el proyecto docente que se propone en esta memoria, las asignaturas que forman el perfil de plaza abordarán los temas del contenido de presentado para la materia de bases de datos, tal y como se muestra en la tabla 4.1.

Asignatura	Tema	Apartados
Ficheros y bases de datos	1. Conceptos fundamentales	1.1, 1.2, 1.3
	2. Bases de datos relacionales	2.1, 2.2, 2.3, 2.4
	3. Planificación y diseño de bases de datos	3.1, 3.2, 3.3, 3.4
	4. Aplicaciones de bases de datos	4.1, 4.2
Diseño de sistemas de bases de datos	4. Aplicaciones de bases de datos	4.3, 4.4, 4.5
	5. Técnicas de implementación y administración de sistemas de gestión de bases de datos	5.1, 5.2, 5.3, 5.4, 5.5, 5.6
	6. Bases de datos orientadas a objetos	6.1, 6.2, 6.3, 6.4
	7. Bases de datos distribuidas	7.1, 7.2, 7.3, 7.4
	8. Bases de datos avanzadas	8.1, 8.2, 8.3, 8.4, 8.5, 8.6

Tabla 4.1 Distribución de temas por asignaturas.

Todos los temas no se presentan con el mismo nivel de profundidad, como se puede ver en el desarrollo de los programas de las asignaturas (capítulos 5 y 6). Algunos de los temas que no se tratan en las asignaturas del perfil quedan cubiertos por otras asignaturas del plan de estudios:

- *Análisis y diseño de sistemas de información I*

  3.6. Herramientas CASE en el diseño de bases de datos.

- *Metodologías y técnicas ofimáticas en las organizaciones*

  4.6. Integración en entornos ofimáticos.

4.4 Fuentes de información

Obras generales

En este apartado se presenta una lista de textos que sirven de referencia práctica al conjunto de la disciplina de bases de datos. Aquí se ha hecho una selección de los textos clásicos o los más completos, y en los programas de las asignaturas, que se presentan en los dos capítulos siguientes de esta memoria, se añaden otros textos alternativos.

– Sistemas de bases de datos.

[7] *Database Systems. Concepts, Languages and Architectures.* P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone (1999).

[36] *Database Systems. A Practical Approach to Design, Implementation and Management.* T. Connolly, C. Begg, A. Strachan (1998).

[40] *Introducción a los sistemas de bases de datos.* C. J. Date (1997).

[50] *Fundamentals of Database Systems.* R. Elmasri, S. B. Navathe (1999).

– Diseño de bases de datos relacionales.

[9] *Diseño Conceptual de Bases de Datos. Un Enfoque de Entidades–Interrelaciones.* C. Batini, S. Ceri, B. Navathe (1994).

[46] *Diseño de bases de datos relacionales.* A. de Miguel, M. Piattini, E. Marcos (1999).

– Bases de datos orientadas a objetos.

[12] *Sistemas de bases de datos orientadas a objetos. Conceptos y arquitecturas.* E. Bertino, L. Martino (1995).

[59] *Modern Database Systems. The Object Model, Interoperability and Beyond.* R. Kim (1995).

– Bases de datos distribuidas.

[18] *Distributed Databases. Principles and Systems.* S. Ceri, G. Pelagatti (1984).

[73] *Principles of Distributed Database Systems.* M. T. Özsu, P. Valduriez (1999).

– Bases de datos avanzadas.

[2] *Data on the Web: From Relations to Semistructured Data and XML.* S. Abiteboul, P. Bunemann, D. Suciu (1999).

[19] *Designing Database Applications with Objects and Rules. The IDEA Methodology.* S. Ceri, P. Fraternali (1997).

[60] *The Data Warehouse Toolkit.* R. Kimball (1996).

[84] *Principles of database and knowledge-base systems, Vol. I.* J. ULLMAN (1988).

[85] *Principles of database and knowledge-base systems, Vol. II.* J. ULLMAN (1989).

[92] *Active Database Systems.* J. WIDOM, S. CERI (1996).

[93] *Advanced Database Systems.* C. ZANIOLO, S. CERI, T. S. FALOUTSOS, V. S. SUBRAHMANIAN, R. ZICARI (1997).

Principales revistas y organizaciones

En este apartado se presentan las principales revistas donde se publican trabajos de la disciplina de bases de datos y las organizaciones que apoyan la investigación en bases de datos promoviendo congresos y publicando las actas de los mismos.

- *ACM SIGKDD (Special Interest Group on Knowledge Discovery in Data and Data Mining).* Este grupo de especial interés de la ACM se ocupa de la obtención de conocimiento y la explotación de datos. Patrocina un congreso internacional anual y publica una revista cada trimestre. Está presidido por Won Kim.
  <http://www.acm.org/sigkdd> [Consulta: 30 de junio de 2000].

- *ACM SIGMOD (Special Interest Group on Management of Data).* Este grupo de especial interés de la *ACM* es el que se dedica a la materia de bases de datos. Organiza un congreso anual y está presidido por Richard Snodgrass.
  <http://www.acm.org/sigmod> [Consulta: 30 de junio de 2000].

- *ACM SIGMOD Record.* Revista trimestral publicada por *ACM SIGMOD* de la que también existe una edición Web.
  <http://www.acm.org/sigmod/record/> [Consulta: 30 de junio de 2000].

- *ACM PODS (Symposium on Principles of Database Systems).* Este es un foro para la investigación teórica en bases de datos, que se celebra conjuntamente con los congresos de SIGMOD. Su presidente es Victor Vianu.
  <http://www.acm.org/sigmod/pods> [Consulta: 30 de junio de 2000].

- *ACM TODS (Transactions on Database Systems).* Esta revista publica artículos originals sobre los fundamentos lógicos y técnicos de la gestión de datos. Edita un volumen anual que aparece en cuatro números trimestrales y su editor es Won Kim.
  <http://www.acm.org/tods/> [Consulta: 30 de junio de 2000].

- *EDBT Endowment (Extending Database Technology).* Esta organización promueve y apoya la investigación en los campos de las bases de datos y de los sistemas de información. Cada dos años realiza un congreso internacional y está presidida por Paolo Atzeni.
  <http://www.edbt.org> [Consulta: 30 de junio de 2000].

- *ICDT–Council (Council of the International Conferences on Database Theory).* Esta organización promueve la investigación en teoría de bases de datos. Organiza un congreso internacional cada dos años y está presidida por Jan Paredaens.

- *IEEE TCDE (Technical Committee on Data Engineering).* Este comité se ocupa del papel de los datos en el diseño, desarrollo, gestión y utilización de sistemas de información. Está presidido por Betty J. Salzberg.
  <http://www.ccs.neu.edu/groups/IEEE/tcde> [Consulta: 30 de junio de 2000].
  Patrocina el congreso anual *International Conference on Data Engineering (ICDE)* y publica la revista trimestral *IEEE Database Engineering Bulletin*
  <http://www.research.microsoft.com/research/db/debull> [Consulta: 30 de junio de 2000].

- *Jornadas de Ingeniería del Software y Bases de Datos.* Estas jornadas nacionales, celebradas por primera vez en 1999, son el resultado de la unión de la parte de investigación de las Jornadas de Investigación y Docencia en Bases de Datos, y las Jornadas de Ingeniería del Software.

- *Journal of Dabatase Management.* Esta es la publicación oficial de la *Information Resources Management Association.* Está orientada tanto hacia investigadores, como hacia diseñadores, desarrolladores, educadores, consultores y administradores de sistemas de bases de datos.
  <http://www.idea-group.com/jdm.htm> [Consulta: 30 de junio de 2000].

- *VLDB (Very Large Data Bases Endowment).* Esta organización tiene como propósito promover e intercambiar trabajos eruditos en bases de

datos y disciplinas relacionadas, entre investigadores en bases de datos y profesionales de todo el mundo. Patrocina un congreso internacional anual (*VLDB Conference*) y publica una revista trimestral (*VLDB Journal*). Su actual presidente es John Mylopoulos.
<http://www.vldb.org> [Consulta: 30 de junio de 2000].

Revisiones

En este apartado se citan los artículos que sistematizan y actualizan los distintos temas de la materia de bases de datos. La mayor parte de estos artículos, que se han ordenado cronológicamente, han sido citados en el capítulo 1 que trata sobre el ámbito conceptual de este proyecto docente.

[35] *Third–Generation Database System Manifesto.* COMMITTEE FOR ADVANCED DBMS FUNCTION (1990).

[6] *The Object–Oriented Database System Manifesto.* M. ATKINSON *et al.* (1990).

[79] *DBMS Research at a Crossroad: The Viena Update.* M. STONEBRAKER *et al.* (1993).

[91] *Research Issues in Active Database Systems: Report from the Closing Panel at RIDE-ADS'94.* J. WIDOM (1994).

[39] *The Third Manifesto.* H. DARWEN, C. J. DATE (1995).

[74] *Database Research: Achievements and Oportunities Into the 21st Century.* A. SILBERSCHATZ, M. STONEBRAKER, J. ULLMAN (1996).

[75] *Database Systems — Breaking Out of the Box.* A. SILBERSCHATZ, S. ZDONIK (1996).

[1] *The Active Database Management System Manifesto: a Rulebase of Active DBMS Features.* ACT-NET CONSORTIUM (1996).

[20] *An Overview of Data Warehousing and OLAP Technology.* S. CHAUDHURI, E. DAYAL (1997).

[11] *The Asilomar Report on Database Research.* P. BERNSTEIN *et al.* (1998).

[48] *SQL:1999, formerly known as SQL3.* A. EISENBERG, J. MELTON (1999).

Servicios especiales

En este apartado se presentan servicios que proporcionan información sobre la comunidad científica: grupos de investigación, publicaciones, prototipos, proyectos en desarrollo, etc.

– *ACM SIGMOD Anthology.* Esta es una biblioteca digital para la comunidad investigadora en sistemas de bases de datos.
<http://www.acm.org/sigmod/dblp/db/anthology.html> [Consulta: 30 de junio de 2000].

– *ACM Digital Review.* Esta es una guía de toda la información disponible a través de la Web que puede ser de interés a la comunidad investigadora en la materia de bases de datos. Esta guía se basa en revisiones públicas realizadas por revisores no anónimos, entre los que se encuentran prestigiosos investigadores en la materia.
<http://www.eecs.umich.edu/digital-review> [Consulta: 30 de junio de 2000].

– *CoRR (Computing Research Repository).* Este es un repositorio en línea de documentos técnicos.
<http://xxx.lanl.gov/archive/cs/> [Consulta: 30 de junio de 2000].

– *Uni-Trier DBLP Bibliography.* El objetivo de esta herramienta en línea es el de proporcionar una bibliografía para los miembros de la comunidad de bases de datos. Hace poco tiempo que ha empezado a incoroporar versiones electrónicas de algunas publicaciones. En un futuro se incorporarán referencias cruzadas entre los documentos de esta bibliografía y las revisiones realizdas en *ACM SIGMOD Digital Review.*
<http://dblp.uni-trier.de> [Consulta: 30 de junio de 2000].

– *Universidad de Massachusetts*, laboratorio de sistemas de bases de datos. Además de toda la información sobre el grupo de investigación, sus proyectos y publicaciones, esta página web tiene un motor de búsqueda de bibliografía de bases de datos, enlaces a grupos de investigación en bases de datos en todo el mundo, sistemas de gestión de bases de datos de dominio público, productos comerciales y servicios clasificados por áreas temáticas, congresos y exposiciones sobre bases de datos.
<http://www-ccs.cs.umass.edu/db/maindb.html> [Consulta: 30 de junio de 2000].

Capítulo 5

Ficheros y bases de datos

En esta asignatura se introducen los sistemas de bases de datos como el modo fundamental de organizar los datos en los sistemas de información. Se centra el estudio en los sistemas de bases de datos relacionales, ya que son los más extendidos hoy en día y poseen una sólida base teórica que los sustenta. Además de aprender el uso y funcionamiento de los sistemas relacionales, se estudia una metodología de diseño que es también la más utilizada, tanto en el ámbito profesional como en el académico.

El objetivo general es, por lo tanto, el de ofrecer una base sólida en los aspectos fundamentales de la tecnología de bases de datos desde el enfoque relacional. Dichos aspectos comprenden las ideas, tanto teóricas como prácticas, que son la base de los sistemas relacionales, y el diseño de bases de datos.

El objetivo de las prácticas es aprender el lenguaje SQL y utilizarlo para interaccionar con bases de datos relacionales, tanto en el ámbito interactivo, como embebido en un lenguaje de programación de alto nivel. También se aprende el manejo de herramientas que permiten la generación de aplicaciones que interactúan con bases de datos.

5.1 Desarrollo del programa de teoría

Para cada tema que forma parte del programa de teoría, se incluyen cuatro puntos. En la *motivación* se justifica el estudio del tema, introduciendo los conceptos que trata. En los *contenidos* se enumeran los distintos apartados que componen el tema. En el *resumen* se expone brevemente la materia que se presenta en cada uno de los apartados. Por último, en la *bibliografía* se citan los capítulos de los textos recomendados que tratan la materia del tema presentado. Estas recomendaciones se dan alrededor de cuatro textos básicos: Elmasri y Navathe (1997) [49], Connolly, Begg y Strachan (1998) [36], Date (1997) [40] y Batini, Ceri y Navathe (1994) [9]. El estudiante no necesita los cuatro textos para desarrollar la asignatura, de hecho cualquiera de los tres primeros sería suficiente para cubrir todos los contenidos. Sin embargo, se ha decidido tratar los cuatro textos para que el alumno tenga la posibilidad de escoger el que mejor se le adapte (por el idioma en que se encuentra disponible, por la extensión o el modo en que se trata cada apartado, por el resto de temas que contiene y que no se tratan en la asignatura, etc.).

En el último apartado del desarrollo del programa de teoría se presenta una bibliografía. Además de los textos básicos ya citados, esta bibliografía contiene otros textos a los que se acompaña de un breve comentario.

Tema 1. Introducción a las bases de datos

Motivación

Los predecesores de los sistemas de bases de datos son los sistemas de ficheros. Un sistema de ficheros está formado por un conjunto de programas que dan servicio a los usuarios finales. Cada programa define y gestiona sus propios datos. Aunque los sistemas de ficheros supusieron un gran avance sobre los sistemas manuales, tienen inconvenientes bastante importantes, como la redundancia de datos y la dependencia entre programas y datos, por lo que se han quedado obsoletos. Sin embargo, hay dos buenas razones para estudiarlos. En primer lugar, el conocer los problemas de este tipo de sistemas nos previene de volver a cometerlos. En segundo lugar, si en algún momento fuera necesario convertir un sistema de ficheros en un sistema de bases de datos, comprender cómo trabaja este sistema puede ser una ayuda esencial.

Los sistemas de bases de datos surgieron con el objetivo de resolver los problemas que planteaban los sistemas de ficheros. Una base de datos es un conjunto de datos relacionados que recogen las necesidades de información de una empresa u organización y que se comparten por todos los usuarios. Las bases de datos se generan y mantienen automáticamente por los sistemas de gestión de bases de datos. Un sistema de gestión de bases de datos es un conjunto de programas que permiten a los usuarios definir, crear y mantener la base de datos, además de proporcionar un acceso controlado a la misma. Todos los accesos a la base de datos se realizan a través del sistema de gestión de bases de datos.

Los sistemas jerárquico y de red representan la primera generación de los sistemas de gestión de bases de datos, surgida hacia los años sesenta. El modelo relacional, propuesto por Codd en 1970, representa la segunda generación de los sistemas de gestión de bases de datos, siendo el modelo más extendido en la actualidad. La tercera generación de los sistemas de gestión de bases de datos se encuentra representada por el modelo relacional extendido y el modelo orientado a objetos.

Contenidos (2 horas)

1.1 Sistemas de ficheros.

1.2 Sistemas de bases de datos.

1.3 Papeles en el entorno de las bases de datos.

1.4 Historia de los sistemas de bases de datos.

1.5 Ventajas e inconvenientes de los sistemas de bases de datos.

Resumen

En este tema se presentan los sistemas de bases de datos, haciendo antes un repaso por sus predecesores, los sistemas de ficheros. También se presentan los sistemas de gestión de bases de datos, que son las aplicaciones que interaccionan con los usuarios de los programas de aplicación y la base de datos. En general, un sistema de gestión de bases de datos proporciona varios servicios. Por una parte, proporciona un lenguaje de definición de datos que permite a los usuarios definir la base de datos y un lenguaje de manejo de datos que permite a los usuarios la inserción, actualización, eliminación y consulta de datos de la base de datos. Además, el sistema de gestión de bases de datos proporciona un acceso controlado a la base de datos: proporciona seguridad, integridad, concurrencia y controla la recuperación ante fallos. También proporciona un mecanismo de vistas que permite mostrar a cada grupo de usuarios sólo aquellos datos que les interesan.

A continuación se presentan las personas involucradas en el entorno de una base de datos, que son: el administrador de la base de datos, los diseñadores de la base de datos, los programadores de aplicaciones y los usuarios finales, y se hace un breve repaso por la historia de los sistemas de gestión de bases de datos.

Por último, se comentan las ventajas e inconvenientes de los sistemas de bases de datos. Las ventajas surgen tanto por el hecho de compartir los datos, como por la existencia del sistema de gestión de bases de datos. Algunas de estas ventajas son el control de la redundancia, la consistencia de datos y la mejora en los aspectos de seguridad y la integridad. Algunos de sus inconvenientes son su elevada complejidad, su coste y su vulnerabilidad ante fallos.

Bibliografía

En la mayoría de los libros sobre sistemas de bases de datos se presentan estos sistemas haciendo un repaso de sus predecesores, los sistemas de ficheros. Se recomienda el capítulo 1 del texto de Elmasri y Navathe (1997). Se tratan los mismos conceptos en el capítulo 1 del texto de Connolly, Begg y Strachan (1998), y en los capítulos 1 y 2 del texto de Date (1997).

Tema 2. Organizaciones de ficheros y estructuras de acceso

Motivación

Las bases de datos se almacenan físicamente en ficheros de discos magnéticos, por lo que se hace necesario estudiar cómo se organizan las bases de datos en los dispositivos de almacenamiento y las técnicas que se pueden utilizar para acceder eficientemente a ellas.

Hay varias organizaciones de ficheros que determinan cómo se colocan los registros físicamente en el disco y, por lo tanto, cómo se puede acceder a los registros de datos. Un fichero desordenado pone los registros sin seguir un orden específico, añadiendo los registros nuevos al final del fichero, mientras que un fichero ordenado mantiene los registros ordenados según el valor de un campo concreto. Un fichero disperso utiliza una función, que aplicada sobre un campo en particular, determina la posición del registro en el disco. Cada una de estas organizaciones favorece un tipo de accesos, pero penaliza otros, por lo que no hay una que se pueda considerar la mejor. Dependiendo del tipo y la frecuencia de los accesos, será más conveniente una organización u otra.

Hay otras organizaciones de ficheros que utilizan estructuras en forma de árbol. Estas organizaciones no se suelen utilizar para almacenar datos, sino para crear estructuras de acceso auxiliares que permiten acceder a los datos eficientemente, a través de campos alternativos a aquellos que se han utilizado para organizar el fichero de datos. A estas estructuras de acceso se las denomina índices y también las hay de varios tipos.

Contenidos (7 horas)

2.1 Introducción.

2.2 Conceptos fundamentales de organizaciones de ficheros.

2.3 Dispositivos de almacenamiento secundario. Discos.

2.4 Ficheros desordenados.

2.5 Ficheros ordenados.

2.6 Ficheros dispersos.

2.7 Agrupamiento.

2.8 Índices.

Resumen

En este tema se presentan los conceptos fundamentales sobre el almacenamiento físico de la base de datos en dispositivos de almacenamiento secundario.

Se estudian los discos magnéticos por ser los dispositivos más utilizados para almacenar los ficheros de una base de datos. Los datos se almacenan en el disco en bloques. El acceso a un bloque es costoso debido al tiempo de búsqueda, el tiempo de rotación y el tiempo de transferencia del mismo.

Se presentan tres organizaciones primarias para los ficheros: desordenados, ordenados y dispersos. Los ficheros desordenados requieren una búsqueda lineal para localizar registros, pero la inserción es muy sencilla. La eliminación plantea problemas en el sentido de que se desperdicia espacio, recuperable mediante una reorganización del fichero. Los ficheros ordenados reducen el tiempo requerido para leer registros en orden a través del campo de ordenación. El tiempo necesario para buscar un registro cualquiera, dado el valor de su campo de ordenación, también se reduce, ya que se puede realizar una búsqueda binaria. Sin embargo, la necesidad de mantener los registros en orden hace muy costosa la inserción. La dispersión ofrece acceso muy rápido a un registro cualquiera dado el valor de su campo de dispersión. El acceso a un registro a través de un campo que no es el campo de dispersión, es lento, y lo mismo sucede con el acceso secuencial a los registros a través de cualquier campo. Hay varias técnicas que permiten al fichero expandirse y contraerse dinámicamente: la dispersión dinámica, la extensible y la lineal. El agrupamiento es otro modo de organizar los registros que se estudia en este tema, cuyo resultado es un fichero ordenado de un modo especial. Mediante el agrupamiento se implementan físicamente las relaciones entre los registros de varios ficheros, almacenándolos consecutivos en un mismo fichero.

Por último, se presentan los índices, que son estructuras de acceso adicionales con las que se mejora la eficiencia en la obtención de registros de un fichero de datos. Dichas estructuras de acceso pueden usarse junto con las organizaciones de ficheros vistas anteriormente. Hay tres tipos de índices ordenados de un solo nivel: primarios, secundarios y de agrupamiento. Los índices de un solo nivel son ficheros ordenados y se examinan mediante búsqueda binaria. Se pueden construir índices de varios niveles (multinivel) para mejorar la eficiencia en las búsquedas. Los árboles B y los árboles B+ son índices de varios niveles cuya estructura permite al índice expandirse y contraerse dinámicamente. También es posible construir índices mediante estructuras de datos que utilizan la técnica de la dispersión.

Bibliografía

Para este tema se recomiendan los capítulos 4 y 5 del texto de Elmasri y Navathe (1997), en donde se describen las principales organizaciones de ficheros para el almacenamiento de datos en disco. En el texto también se realiza un estudio de este dispositivo. Además, estos autores realizan un análisis de las distintas estructuras de índices para el acceso a estos ficheros. Parte de los contenidos de este tema también se analizan en el apéndice A del texto de Date (1997), aunque con menor profundidad. Un buen resumen del tema es el que se presenta en el apéndice B del texto de Connolly, Begg y Strachan (1998).

Tema 3. Sistemas de bases de datos

Motivación

Una característica fundamental de los sistemas de bases de datos es el hecho de que proporcionan cierto nivel de abstracción de datos, ocultando a los usuarios los detalles sobre cómo se almacenan los datos. Los modelos de datos proporcionan los medios necesarios para alcanzar tal nivel de abstracción. Un modelo de datos es un conjunto de conceptos que se pueden utilizar para describir la estructura de una base de datos. La estructura de la base de datos que se define mediante el modelo incluye los tipos de los datos, las relaciones entre ellos y las restricciones que deben cumplir. Muchos modelos de datos también poseen un conjunto de operaciones básicas para especificar los accesos a la base de datos.

Otras tres características importantes de los sistemas de bases de datos son: la independencia entre datos y programas, el soportar múltiples vistas de usuario y el disponer de un diccionario de datos en el que almacenar el esquema o estructura de la base de datos. Estas características se pueden conseguir si el sistema de gestión de bases de datos tiene la arquitectura de tres niveles ANSI–SPARC.

Los sistemas de gestión de bases de datos también deben proporcionar lenguajes e interfaces para cada categoría de usuarios (administrador de la base de datos, diseñadores de la base de datos, programadores de aplicaciones y usuarios finales). Ya que los sistemas de gestión de bases de datos son aplicaciones bastante complejas, están constituidos por distintos módulos que realizan funciones diferentes.

Contenidos (3 horas)

3.1 Introducción.

3.2 Modelos de datos.

3.3 Arquitectura de los sistemas de bases de datos.

3.4 Lenguajes de los sistemas de gestión de bases de datos.

3.5 Clasificación de los sistemas de gestión de bases de datos.

3.6 Funciones de los sistemas de gestión de bases de datos.

3.7 Componentes de un sistema de gestión de bases de datos.

Resumen

En este tema se presenta un análisis de los modelos de datos y se definen los conceptos de esquema y estado de una base de datos. Hay tres categorías principales de modelos de datos: modelos conceptuales, modelos lógicos y modelos físicos. Es importante distinguir entre el esquema (descripción de una base de datos) y la base de datos en sí misma. El esquema no cambia a menudo, en tanto que la base de datos cambia cada vez que se insertan, eliminan o modifican datos.

También se presenta la arquitectura de los sistemas de bases de datos y la independencia respecto a los datos. La arquitectura de los sistemas de bases de datos establecida por el comité ANSI–SPARC utiliza tres niveles de abstracción: externo, conceptual e interno. En el nivel externo, el esquema consta de las distintas visiones que tienen los usuarios de la base de datos. En el nivel conceptual, el esquema es la visión común de la base de datos: especifica el contenido de información de la base de datos independientemente de las consideraciones de almacenamiento. En el nivel interno, el esquema es la visión que el ordenador tiene de la base de datos: especifica cómo se representan los datos, en qué orden se almacenan los registros, qué índices y punteros se han creado y qué esquema de dispersión se ha utilizado, si es el caso. Todo sistema de gestión de bases de datos que separe los tres niveles deberá tener correspondencias entre los esquemas para transformar las peticiones de los usuarios y los resultados, de un nivel al siguiente. La mayoría de los sistemas de gestión de bases de datos no separan los tres niveles por completo. La independencia de datos hace que cada nivel de la arquitectura sea inmune a los cambios en los niveles de debajo. La independencia de datos lógica se refiere a la inmunidad de los esquemas externos frente a los cambios en el esquema conceptual. La independencia de datos física se refiere a la inmunidad del esquema conceptual frente a los cambios en el esquema interno.

También se definen los distintos lenguajes que proporcionan los sistemas de gestión de bases de datos y se estudian varias clasificaciones de estos sistemas. Un lenguaje de base de datos consta de dos partes: un lenguaje de definición de datos y un lenguaje de manejo de datos. El lenguaje de definición de datos se utiliza para especificar el esquema de la base de datos, las vistas de los usuarios y las estructuras de almacenamiento, mientras que el lenguaje de manejo de datos se utiliza para leer y actualizar los datos de la base de datos. Los sistemas de gestión de bases de datos se pueden clasificar según varios criterios: el modelo lógico que soportan, el número de usuarios, el número de puestos, el coste y la generalidad. La clasificación más importante es la que se basa en el modelo lógico, siendo los principales modelos que se utilizan en el mercado el modelo relacional, el de red, el jerárquico y el orientado a objetos.

Se comentan, por último, las funciones de los sistemas de gestión de bases de datos y los módulos que los componen. Los sistemas de gestión de bases de datos son sistemas informáticos muy complejos formados por una serie de componentes, cada uno de ellos con una función específica: gestión del diccionario de datos, gestión de las transacciones, control de la seguridad, control de la concurrencia, optimización de consultas, control de la integridad y gestión de la recuperación ante fallos. Además de estos módulos, los sistemas de gestión de bases de datos cuentan con una serie de herramientas que ayudan al administrador de la base de datos a manejar el sistema.

Bibliografía

La mayoría de los libros de texto sobre bases de datos analizan los diversos conceptos que se presentan en este tema. Se recomienda el capítulo 2 del texto de Elmasri y Navathe (1997). Los mismos contenidos se encuentran en el capítulo 2 del texto de Connolly, Begg y Strachan (1998) y también en los capítulos 1 y 2 del texto de Date (1997).

Tema 4. El modelo relacional

Motivación

Codd presentó el modelo relacional en 1970, atrayendo inmediatamente la atención dada su simplicidad y su fundamento matemático. El modelo utiliza el concepto de relación matemática (que es similar a una tabla) como estructura de datos, y tiene su base en la teoría de conjuntos y la lógica de predicados de primer orden. Como todo modelo de datos, incluye una estructura de datos, una serie de restricciones denominadas reglas de integridad y una serie de operadores que permiten manipular los datos almacenados en la estructura de datos para hacer consultas y actualizaciones.

El álgebra relacional consta de una serie de operadores que permiten al usuario especificar accesos a datos. El resultado de realizar una operación del álgebra es una nueva relación, que se habrá formado a partir de una o varias relaciones. Las operaciones del álgebra relacional se suelen dividir en dos grupos. Un grupo incluye las operaciones de conjuntos de la teoría matemática de conjuntos, mientras que el otro grupo consta de operaciones que se han desarrollado exclusivamente para las bases de datos relacionales. El cálculo relacional es un formalismo equivalente al álgebra relacional. Mientras que el álgebra relacional es un lenguaje procedural, el cálculo relacional es un lenguaje no procedural que utiliza predicados.

Contenidos (12 horas)

4.1 Introducción.

4.2 El modelo relacional.

4.3 Estructura de datos relacional. Claves.

4.4 Reglas de integridad.

4.5 Lenguajes relacionales: álgebra relacional y cálculo relacional.

4.6 Ejercicios de álgebra relacional y cálculo relacional.

4.7 Vistas.

Resumen

En este tema se presenta el modelo relacional, que es el modelo lógico en el que se basan la mayoría de los sistemas de gestión de bases de datos comerciales en uso hoy en día. En primer lugar, se trata la descripción de los principios básicos del modelo relacional: la estructura de datos relacional y las reglas de integridad. La relación es la estructura de datos del modelo relacional. Las relaciones se representan gráficamente como tablas, donde las filas corresponden a las tuplas y las columnas corresponden a los atributos, que se definen sobre dominios. También se describen los conceptos de clave primaria y clave ajena. Toda relación tiene una clave primaria, que es un atributo o un conjunto de atributos que se escogen para identificar las tuplas de la relación. La regla de integridad de entidades es una restricción que dice que ninguno de los atributos que forman la clave primaria puede ser nulo. Una clave ajena es un atributo o un conjunto de atributos que hacen referencia a la clave primaria de otra relación. La regla de integridad referencial dice que los valores de las claves ajenas deben coincidir con alguno de los valores de la clave primaria a la que hacen referencia, o bien ser completamente nulos.

A continuación, se presentan los lenguajes relacionales de manejo de datos. Estos lenguajes pueden clasificarse como procedurales, no procedurales, orientados a transformaciones, gráficos, de cuarta generación o de quinta generación. El álgebra relacional es un lenguaje procedural formal. Sus operaciones básicas son: restricción, proyección, producto cartesiano, unión, intersección, diferencia, división y varios tipos de concatenación (*join*). El cálculo relacional es un lenguaje no procedural formal que utiliza predicados. El álgebra relacional y el cálculo relacional son lenguajes equivalentes. El uso de estos lenguajes se practica resolviendo consultas de datos propuestas en un boletín de ejercicios

Por último se estudian las vistas. Una vista es una relación virtual. Las vistas proporcionan seguridad y permiten que el diseñador haga esquemas a medida de cada usuario. Además, se generan dinámicamente y no todas son actualizables.

Bibliografía

Todos los textos básicos sobre bases de datos tratan el modelo relacional, aunque no todos lo hacen con el mismo nivel de formalidad. Para la parte de la estructura de datos y las reglas de integridad se recomienda el capítulo 3 del texto de Connolly, Begg y Strachan (1998), ya que las definiciones formales vienen siempre precedidas de definiciones informales que ayudan a entenderlas. También es válido el capítulo 6 del texto de Elmasri y Navathe (1997), aunque es un tanto más formal, como ocurre con el texto de Date (1997) (capítulos 4 y 5). Sin embargo, para la parte del álgebra y del cálculo relacional se recomiendan los capítulos 6 y 7 del texto de Date (1997), ya que se ha considerado que su sintaxis es la más sencilla y fácil de recordar.

Tema 5. Planificación, diseño y administración de bases de datos

Motivación

Un sistema de información es el conjunto de recursos que se utilizan para recoger, gestionar, controlar y divulgar la información dentro de una empresa u organización. Desde los años setenta, los sistemas de bases de datos han ido reemplazando a los sistemas de ficheros en los sistemas de información. En la actualidad, un sistema de información consta de diversos componentes: la base de datos, el sistema de gestión de bases de datos, los programas de aplicación y los equipos informáticos.

Siendo la base de datos uno de los componentes principales de los sistemas de información, el ciclo de vida de un sistema de información estará inherentemente ligado al ciclo de vida de la base de datos sobre la que se apoya. El objetivo del diseño de la base de datos es el de desarrollar un esquema que refleje los datos de la organización. Este esquema ayuda a entender el significado de los datos y también facilita la comunicación con los usuarios en lo relativo a los requisitos de información.

El desarrollo de una base de datos se lleva a cabo en tres etapas sucesivas en las que siempre se representan los mismos datos (aquellos de interés para la empresa u organización), pero a diferentes niveles de abstracción. Estas etapas son: el diseño conceptual, el diseño lógico y el diseño físico.

Contenidos (3 horas)

5.1 Introducción.

5.2 Ciclo de vida de los sistemas de información.

5.3 Ciclo de vida de las aplicaciones de bases de datos.

5.4 Diseño de bases de datos: diseño conceptual, diseño lógico y diseño físico.

5.5 Diseño de aplicaciones.

5.6 Herramientas CASE.

5.7 Administración de datos y de la base de datos.

Resumen

En este tema se comentan las fases principales del ciclo de vida de un sistema de información y se ve cómo se relacionan con el desarrollo de aplicaciones de bases de datos. A continuación, se describen las tareas que se deben realizar en cada etapa del ciclo de vida de una aplicación de bases de datos para que se produzca un sistema que funcione correctamente. Las etapas de este ciclo de vida son: la planificación del proyecto, la definición del sistema, la recolección y el análisis de los requisitos, el diseño de la base de datos, la selección del sistema de gestión de bases de datos, el diseño de aplicaciones, la elaboración de prototipos, la implementación, la conversión y la carga de datos, y, por último, la prueba y el mantenimiento. Además también se presenta todo el personal responsable de la planificación, el diseño y la administración de una base de datos.

A su vez, el diseño de la base de datos se divide en tres etapas consecutivas. La primera etapa es el diseño conceptual, en donde se construye un esquema de la información de interés, independientemente de todas las consideraciones físicas. Después viene el diseño lógico, en el que el esquema producido en la etapa anterior se transforma según el modelo lógico de base de datos que se vaya a utilizar en la implementación (relacional, jerárquico, de red, etc.). Por último, en la etapa del diseño físico, se produce una descripción de la implementación de la base de datos en memoria secundaria (las organizaciones de ficheros y las estructuras de acceso que se van a utilizar).

Una fase que se debe llevar a cabo en paralelo con el diseño de la base de datos es el diseño de las aplicaciones. Esta fase está compuesta por dos actividades: el diseño de las transacciones y el diseño de las interfaces de usuario de informes y formularios. Las herramientas CASE ayudan a que el desarrollo de los sistemas de información se realice de modo eficiente y efectivo.

Por último, se distingue entre los papeles del administrador de datos y el administrador de la base de datos, señalando en qué etapas del ciclo de vida de las aplicaciones de bases de datos se encuentra involucrado cada uno de ellos. Fundamentalmente, la administración de datos consiste en la gestión de los datos como recurso, mientras que la administración de la base de datos es la gestión de la base de datos física.

Bibliografía

Siendo éste un tema introductorio, se encuentra reflejado en la mayoría de los textos sobre bases de datos. Se recomienda el capítulo 4 del texto de CONNOLLY, BEGG Y STRACHAN (1998) o el capítulo 1 del texto de BATINI, CERI Y NAVATHE (1994).

Tema 6. Diseño conceptual. Modelo entidad–relación

Motivación

El modelo entidad–relación es un modelo conceptual de datos de alto nivel que fue desarrollado por Chen en 1976 para ayudar en la tarea del diseño de bases de datos. Un modelo conceptual de datos es un conjunto de conceptos que describen la estructura de la base de datos y las transacciones de acceso y actualización sobre la misma. El objetivo principal de desarrollar un esquema conceptual de datos de alto nivel es el de reflejar la percepción de los datos desde el punto de vista de los usuarios, ocultando los aspectos más técnicos del diseño de bases de datos. Además, los modelos conceptuales de datos son independientes de los sistemas de gestión de bases de datos y de los equipos informáticos que los soportan.

El modelo entidad–relación original tiene algunas limitaciones en cuanto a la representación de la información, por lo que más tarde se añadieron algunos conceptos semánticos que han dado lugar al modelo entidad–relación extendido. Uno de estos conceptos es la especialización/generalización.

Contenidos (3 horas)

6.1 Introducción.

6.2 Metodología de diseño de bases de datos.

6.3 Modelos de datos.

6.4 El modelo entidad–relación.

6.5 Metodología de diseño conceptual.

Resumen

Este tema comienza haciendo un repaso de la metodología de diseño de bases de datos, que se descompone en tres etapas: diseño conceptual, diseño lógico y diseño físico. En la etapa del diseño conceptual se construye un esquema conceptual local para cada vista de cada usuario o grupo de usuarios. Cada vista de usuario comprende los datos que un usuario maneja para llevar a cabo una determinada tarea. Normalmente, estas vistas corresponden a las distintas áreas funcionales de la empresa, y se pueden identificar examinando los diagramas de flujo de datos o entrevistando a los usuarios, examinando los procedimientos, informes y formularios que se utilizan, y observando el funcionamiento de la empresa. En el diseño lógico se obtiene un esquema lógico local para cada esquema conceptual local. Estos esquemas lógicos se integran después para formar un esquema lógico global que represente todas las vistas de los distintos usuarios de la empresa. Por último, en el diseño físico, se construye la implementación de la base de datos sobre un sistema de gestión de bases de datos determinado. Ya que este diseño debe adaptarse a un sistema concreto, es posible que haya que introducir cambios en el esquema lógico para mejorar las prestaciones a nivel físico.

Concretamente, el diseño conceptual es el proceso por el cual se construye un modelo de la información que se utiliza en una empresa u organización, independientemente del sistema de gestión de bases de datos que se vaya a utilizar para implementar el sistema y de los equipos informáticos o cualquier otra consideración física. En esta etapa se utilizan los modelos de datos conceptuales. Un modelo de datos conceptual es un conjunto de conceptos que permiten describir la realidad mediante representaciones lingüísticas y gráficas. Los modelos conceptuales deben poseer una serie de propiedades: expresividad, simplicidad, minimalidad y formalidad. El modelo conceptual que se presenta en este tema es el más utilizado hoy en día: el modelo entidad–relación. Este modelo posee los siguientes conceptos: entidades, relaciones, atributos, dominios de atributos, identificadores y jerarquías de generalización.

Por último, se presenta la metodología del diseño conceptual, que consta de las siguientes etapas: identificar las entidades y las relaciones; identificar los atributos, asociarlos a entidades y relaciones y determinar sus dominios; determinar los identificadores; determinar las jerarquías de generalización, si las hay; dibujar el diagrama entidad–relación y revisar cada esquema conceptual local con el usuario correspondiente. Los esquemas conceptuales locales se completan documentándolos en el diccionario de datos.

Bibliografía

Se recomiendan los capítulos 2 y 3 del texto de Batini, Ceri y Navathe (1994), en donde se tratan muy bien los aspectos sobre el modelo entidad–relación. Este modelo también se trata en el capítulo 5 del texto de Connolly, Begg y Strachan (1998), en el capítulo 12 del texto de Date (1997) y en el capítulo 3 del texto de Elmasri y Navathe (1997).

También se recomienda el capítulo 7 del texto de Connolly, Begg y Strachan (1998), en el que se muestra la metodología del diseño conceptual proporcionando todo tipo de “trucos” para identificar cada uno de los componentes de un esquema conceptual.

Tema 7. Diseño lógico de bases de datos relacionales

Motivación

En el tema anterior se ha estudiado el diseño conceptual de datos mediante el modelo entidad–relación. En el tema 4 se ha presentado el modelo de bases de datos relacional. Estos aspectos se unen en el diseño lógico de bases de datos relacionales, en el que los esquemas conceptuales representados mediante diagramas entidad–relación se transforman en un conjunto de relaciones (tablas) que representan los datos de interés.

La normalización es una parte importante de este proceso de transformación. En ocasiones, las actualizaciones de datos sobre algunas relaciones de la base de datos pueden tener consecuencias anómalas. Mediante la normalización las relaciones que presentan estas anomalías se dividen dando lugar a nuevas relaciones que las evitan.

Contenidos (7 horas)

7.1 Introducción.

7.2 Metodología de diseño lógico en el modelo relacional.

7.3 Normalización.

7.4 Ejercicios de diseño lógico de bases de datos relacionales.

Resumen

El diseño de bases de datos consta de tres etapas: diseño conceptual, diseño lógico y diseño físico. En este tema se trata la etapa del diseño lógico, que es el proceso mediante el cual se construye un esquema que representa la información que maneja la empresa u organización. En esta etapa hay que basarse en un modelo de datos lógico determinado. Estos modelos son independientes del sistema de gestión de bases de datos concreto que se vaya a utilizar para implementar la base de datos y también son independientes de cualquier otra consideración física. El modelo lógico escogido es el modelo relacional, presentado en el tema 4.

A continuación, se estudian las dos fases de que consta el diseño lógico: la construcción y validación de los esquemas lógicos locales para cada vista de usuario, y la construcción y validación de un esquema lógico global. Cada una de estas fases consta de una serie de pasos. Uno de los pasos más importantes es la conversión del esquema conceptual a un esquema lógico adecuado al modelo relacional. Para ello, se deben hacer algunas transformaciones: eliminar las relaciones de muchos a muchos, eliminar las relaciones complejas, eliminar las relaciones recursivas, eliminar las relaciones con atributos, eliminar los atributos con múltiples valores, reconsiderar las relaciones de uno a uno y eliminar las relaciones redundantes.

Por último, se presenta la normalización, una técnica que se utiliza para mejorar el esquema lógico, de modo que éste satisfaga ciertas restricciones que eviten la duplicidad de datos. La normalización garantiza que el esquema resultante está más próximo al modelo de la empresa, es consistente, tiene la mínima redundancia y la máxima estabilidad.

El tema se completa realizando ejercicios de un boletín en el que para cada esquema de base de datos a diseñar, se proporcionan tanto las especificaciones de los requisitos como el esquema conceptual en el modelo entidad–relación.

Bibliografía

El diseño de bases de datos relacionales es un tema de consenso, coincidiendo la mayoría de autores en las tres etapas de diseño conceptual, diseño lógico y diseño físico. Sin embargo, los pasos de que consta cada una de estas etapas y la terminología utilizada no es muy uniforme. Para la metodología de diseño lógico se recomienda el capítulo 8 del texto de Connolly, Begg y Strachan (1998), combinando la terminología del capítulo 12 del texto de Batini, Ceri y Navathe (1994).

Para el apartado de normalización se recomienda el capítulo 6 del texto de Connolly, Begg y Strachan (1998). Este apartado también se trata en los capítulos 9 y 10 del texto de Date (1997) y en el capítulo 12 del texto de Elmasri y Navathe (1997).

Tema 8. Diseño físico de bases de datos relacionales

Motivación

En la etapa del diseño físico se parte del esquema lógico global obtenido durante la etapa del diseño lógico y se obtiene una descripción de la implementación de la base de datos en memoria secundaria. Esta descripción es completamente dependiente del sistema de gestión de bases de datos específico que se vaya a utilizar. Es más, el diseño de las relaciones base sólo se puede realizar cuando el diseñador conoce perfectamente toda la funcionalidad que presenta dicho sistema.

Una metodología de diseño físico para bases de datos relacionales ofrecerá una serie de directrices para escoger las estructuras de almacenamiento de las relaciones base, decidir cuándo crear índices y cuándo desnormalizar el esquema lógico e introducir redundancias.

Contenidos (2 horas)

8.1 Introducción.

8.2 Metodología de diseño físico en el modelo relacional.

Resumen

El diseño físico es el proceso de producir una descripción de la implementación de la base de datos en memoria secundaria. Describe las relaciones base y las estructuras de almacenamiento y métodos de acceso que se utilizarán para acceder a los datos de modo eficiente.

En este tema se pesenta una metodología de diseño físico de bases de datos relacionales. El primer paso consiste en traducir el esquema lógico global de modo que pueda ser fácilmente implementado por el sistema de gestión de bases de datos específico. A continuación, se escogen las organizaciones de ficheros más apropiadas para almacenar las relaciones base y los métodos de acceso, basándose en el análisis de las transacciones que se van a ejecutar sobre la base de datos. Se puede considerar la introducción de redundancias controladas para mejorar las prestaciones. Otra tarea a realizar en este paso es estimar el espacio en disco. La seguridad de la base de datos es fundamental, por lo que el siguiente paso consiste en diseñar las medidas de seguridad necesarias mediante la creación de vistas y el establecimiento de permisos para los usuarios. El último paso del diseño físico consiste en monitorizar y afinar el sistema para obtener las mejores prestaciones y satisfacer los cambios que se puedan producir en los requisitos.

Bibliografía

El diseño físico es un tema que no se suele tratar en los textos básicos sobre bases de datos. Se recomienda del capítulo 9 del texto de Connolly, Begg y Strachan (1998) en el que, además, se presenta un ejemplo ilustrativo muy completo (capítulo 12). Elmasri y Navathe (1997) ofrecen un resumen de la etapa del diseño físico en un apartado del capítulo 14.

Bibliografía del programa de teoría

En este apartado se comenta la bibliografía que se propone para la asignatura. Esta bibliografía contiene los cuatro textos que se han citado en el desarrollo del programa y otros textos que se consideran interesantes y que se pueden utilizar de forma alternativa o para ampliar conocimientos.

C. Batini, S. Ceri, B. Navathe (1994) [9]. En este libro se trata profundamente el diseño de bases de datos. Para el diseño conceptual se utiliza el modelo entidad–relación (extendido). El diseño lógico se presenta tanto para el modelo relacional, como para los modelos prerrelacionales: el modelo jerárquico y el modelo de red. En la etapa del diseño lógico se incluye el estudio de algunos aspectos orientados hacia el diseño físico, como son el tipo de transacciones y su frecuencia de realización, y el tamaño de las relaciones. En función de estos aspectos, se dan una serie de directrices para desnormalizar relaciones, realizar particiones horizontales o verticales de las mismas, o incluir atributos derivados. Este texto es muy completo para el diseño de bases de datos ya que incluye, además, una parte dedicada al análisis funcional.

T. Connolly, C. Begg, A. Strachan (1998) [36]. Este libro cubre ampliamente todos los temas del programa de teoría. Cabe destacar la parte dedicada al diseño de bases de datos, ilustrada con el desarrollo de un ejemplo completo, en la que se incluye la etapa del diseño físico, tan poco tratada en los textos básicos de bases de datos. Para esta etapa se utiliza el sistema de gestión de bases de datos Access. El libro tiene también una parte importante dedicada a lenguajes de bases de datos (SQL y QBE) y una parte dedicada a tratar aspectos de administración de bases de datos (integridad, seguridad, gestión de transacciones) y a otros modelos y aplicaciones de bases de datos.

C. J. Date (1997) [40]. Este libro no falta en ninguna bibliografía sobre bases de datos. Por muchos es considerado como "la biblia de las bases de datos". Contiene la mayor parte del programa de teoría, aunque se echa en falta un tratamiento más amplio del diseño de bases de datos. El libro presenta un panorama general de los sistemas de gestión de bases de datos relacionales DB2 e INGRES. Se trata también el lenguaje SQL, aspectos relacionados con la administración de bases de datos (recuperación, concurrencia, seguridad, integridad, optimización de consultas), así como otros modelos de bases de datos. Este mismo año ha aparecido una nueva edición de este texto [43].

A. de Miguel, M. Piattini (1999) [45]. Este libro presenta las bases de datos como una parte fundamental de los sistemas de información. Trata en profundidad el modelo relacional y el modelo de red (CODASYL). El modelo jerárquico se presenta como un caso particular del modelo de red. Además, dedica un capítulo completo al lenguaje SQL, presentando el estándar SQL3 (ahora SQL:1999), y estudia brevemente la fase del diseño lógico de bases de datos relacionales.

A. de Miguel, M. Piattini, E. Marcos (1999) [46]. Este texto trata en profundidad el diseño de bases de datos relacionales mediante el modelo entidad–relación. Hace un gran énfasis en la normalización y trata también los aspectos del diseño físico. Además, se presentan los lenguajes de cuarta generación como herramientas de desarrollo y se realiza un estudio de las características de las herramientas CASE enfocadas al diseño de bases de datos.

R. Elmasri, S. B. Navathe (1997) [49]. Este texto cubre el temario propuesto en el programa de la asignatura, destacando los capítulos dedicados a las organizaciones de ficheros y los métodos de acceso. Presenta con detalle el lenguaje SQL mediante el estándar SQL2 y también los lenguajes

QUEL y QBE. Además, incluye un recorrido completo por el sistema de gestión de bases de datos DB2. El libro presenta otros temas complementarios como técnicas de implementación de sistemas de gestión de bases de datos (recuperación, concurrencia, seguridad, optimización de consultas) y nuevas tendencias (bases de datos orientadas a objetos, bases de datos distribuidas y bases de datos deductivas).

R. Elmasri, S. B. Navathe (1999) [50]. En esta nueva edición se incluye un nuevo capítulo de diseño físico de bases de datos. Además, el libro contiene un resumen de los sistemas de gestión de bases de datos Access 97 y Oracle 8, y muestra cómo los sistemas relacionales Informix Univesal Server y Oracle 8 incorporan algunas de las características de la orientación a objetos. Además, se dedica una parte completa a los sistemas de bases de datos orientadas a objetos y a los sistemas objeto–relacionales, presentando el estándar SQL3. También se incluye una parte de aspectos avanzados (bases de datos activas, temporales, espaciales, multimedia, distribuidas, deductivas, grandes almacenes de datos, explotación de datos).

M. J. Folk, B. Zoellick (1992) [52]. En este libro se presentan con gran detalle todos los conceptos básicos sobre estructuras de ficheros y se estudian los dispositivos de almacenamiento secundario más utilizados: discos, cintas y CD-ROM. Además, se analiza de modo muy exhaustivo los distintos tipos de índices, los árboles B y los árboles B+, y la dispersión, haciendo un extenso estudio de la dispersión extensible.

M. J. Folk, B. Zoellick, G. Riccardi (1998) [53]. Este libro amplía la edición anterior presentando un enfoque orientado a objetos para la implementación de las distintas organizaciones de ficheros y los métodos de acceso.

G. W. Hansen, J. V. Hansen (1997) [57]. Este libro cubre la mayor parte del programa de teoría. Es de destacar cómo relaciona las bases de datos con los sistemas de información en los que éstas se encuentran integradas, tema poco tratado en los textos básicos de bases de datos, en los que se presta una mayor atención a los aspectos técnicos. Otros aspectos que se tratan en el libro son los sistemas cliente–servidor, la administración y el control de bases de datos, las bases de datos distribuidas, las bases de datos orientadas a objetos y los sistemas basados en el conocimiento.

D. M. Kroneke (1996) [61]. En este texto, el diseño conceptual se ve tanto con el modelo entidad–relación, como con el modelo semántico de objetos.

Un aspecto muy interesante del libro es el diseño de aplicaciones de bases de datos, otro tema poco tratado en los textos básicos de bases de datos, que se ilustra mediante un ejemplo completo desarrollado con Access y con DB2. El libro viene acompañado de SALSA, una aplicación para desarrollar y validar esquemas utilizando objetos semánticos. Además, se trata el procesamiento de bases de datos distribuidas. En 1997 se publicó la sexta edición del libro [62] y este año se ha publicado la séptima edición [63] (estas ediciones no han sido traducidas por el momento). En esta última edición, el ejemplo completo de desarrollo de una aplicación de bases de datos se lleva a cabo con Access 2000. El libro viene acompañado de una edición de evaluación del sistema de gestión de bases de datos relacionales SQL Server 7.0.

M. MARQUÉS (2000) [67]. Este libro de apuntes se ha desarrollado a partir de la bibliografía del programa de teoría y cubre todo el temario propuesto. El objetivo del texto es servir como apoyo a las clases de teoría y se complementa con las transparencias que se utilizan en ellas.

A. SILBERSCHATZ, H. KORTH, S. SUDARSHAN (1998) [76]. Este libro es la nueva edición de la obra con el mismo título en la que Korth figuraba como primer autor. En esta edición se amplía el modelo entidad–relación y se incluyen las bases de datos orientadas a objetos. Una novedad importante es que se tiene en cuenta la relación entre las bases de datos y las arquitecturas de computadores, tratando tanto los sistemas centralizados como los sistemas cliente–servidor. Además, se incluye un capítulo sobre las bases de datos de sistemas paralelos.

J. ULLMAN, J. WIDOM (1999) [86]. En este libro se trata el diseño y el manejo de sistemas de bases de datos relacionales que incluyen la tecnología de la orientación a objetos. El diseño conceptual se presenta tanto con el modelo entidad–relación, como con el modelo orientado a objetos. El diseño lógico se realiza en el modelo relacional y se presentan los disparadores para la implementación de las restricciones. En este texto también se estudia el estándar SQL3.

5.2 Desarrollo del programa de prácticas

El programa de prácticas se divide en cinco partes, cada una de ellas con un número de sesiones distinto. Para cada parte se incluyen dos puntos. En la *motivación* se justifica la inclusión de cada parte en el programa, introduciendo los conceptos que trata y en el *desarrollo* se describen las tareas a realizar en las sesiones correspondientes.

Para todas las sesiones de prácticas se utiliza una misma base de datos que recoge la información de un sistema típico de facturación. La descripción de esta base de datos, los conceptos necesarios para la realización de las prácticas y los ejercicios propuestos se encuentran en el texto de Aliaga et al. (2000) [3]. Este texto cubre completamente el programa de prácticas. En la bibliografía del programa de prácticas se citan otros textos que pueden ser utilizados de forma alternativa o para ampliar conceptos. Todas las sesiones de prácticas, excepto dos, se realizan bajo el sistema de gestión de bases de datos Oracle 8. Estas dos prácticas son la primera y la última, que se llevan a cabo con Access 97.

Parte 1. Introducción a las bases de datos relacionales

Motivación

Ya que las sesiones de prácticas se inician antes de que se estudie el modelo relacional en las clases de teoría, es necesario presentar el concepto de relación y de base de datos relacional antes de comenzar el aprendizaje del lenguaje de consultas.

Desarrollo (1 sesión de 2 horas)

Mediante el entorno gráfico del sistema de gestión de bases de datos relacionales Access 97 se presentan los distintos conceptos que aparecen en las bases de datos relacionales: relaciones (tablas), tuplas (filas), atributos (columnas), claves primarias y claves ajenas. También se introduce la base de datos de prácticas y se elaboran gráficamente algunas consultas sobre ella.

Parte 2. Consulta de datos en SQL: la sentencia SELECT

Motivación

SQL (*Structured Query Language*) es un lenguaje de bases de datos utilizado para trabajar con bases de datos relacionales. SQL es un estándar, existiendo un gran número de sistemas de gestión de bases de datos basados en él: Oracle, Informix, dBase IV, DB2, INGRES, Access, etc. Cuando se necesita obtener datos de la base de datos, se utiliza el lenguaje SQL para efectuar la petición. El sistema de gestión de bases de datos procesa la petición, obtiene los datos solicitados y los devuelve. Este proceso de solicitar datos de la base de datos y de recibir resultados se denomina consulta. Mediante el lenguaje SQL, el usuario especifica qué datos desea, sin tener que especificar el procedimiento que se debe seguir para extraerlos de la base de datos. Pero SQL es mucho más que una herramienta de consulta, ya que se utiliza para controlar todas las funciones que un sistema de gestión de bases de datos proporciona a sus usuarios (actualizaciones de datos, control de accesos, integridad, etc.).

Desarrollo (5 sesiones de 3 horas)

En las sesiones de prácticas de esta parte se presenta el lenguaje SQL mediante el sistema de gestión de bases de datos Oracle 8. El uso de SQL se realiza de modo interactivo utilizando la herramienta de Oracle SQL*Plus.

En las cuatro primeras sesiones se introducen las distintas cláusulas de la sentencia SELECT. Cada una de estas sesiones tiene diez ejercicios propuestos mediante los que se practican los conceptos introducidos. Para las dos últimas sesiones de esta parte se propone una serie de ejercicios variados (alrededor de treinta ejercicios), mediante los que se pueden practicar de forma combinada los conceptos introducidos en las sesiones anteriores.

Las sentencias SQL para crear y cargar la base de datos de prácticas están disponibles en la página Web de la asignatura ¡http://www4.uji.es/∼dba¿. También se puede encontrar en esta misma página los resultados que se obtienen al ejecutar las sentencias de los ejercicios de SQL propuestos en el libro de prácticas (Aliaga et al. (2000)).

Parte 3. Precompiladores (Pro*C) y programación con SQL embebido en C

Motivación

SQL es, al mismo tiempo, un lenguaje de consultas interactivo y un lenguaje utilizado por programas de aplicación para acceder a las bases de datos. La segunda parte del programa de prácticas se ha centrado en el uso interactivo de SQL. En esta tercera parte se muestra el uso de SQL dentro de los programas de aplicación.

SQL es un lenguaje y puede ser utilizado para programación, pero sería incorrecto decir que SQL es un lenguaje de programación. SQL carece de las características habituales de los lenguajes de programaión reales: no dispone de las sentencias para construir bucles, sentencias condicionales, etc. En realidad, SQL es un sublenguaje de datos que realiza tareas de gestión de bases de datos de propósito especial. Para escribir un programa que acceda a una base de datos debe comenzarse con un lenguaje de programación convencional, como por ejemplo C o Pascal, y luego añadir o embeber sentencias de SQL en el programa. Cualquier sentencia de SQL que se puede utilizar de modo interactivo también se puede emplear en un programa de aplicación. Esto, entre otras cosas, permite comprobar el funcionamiento de las sentencias de forma interactiva, para luego codificarlas en el programa.

Desarrollo (2 sesiones de 3 horas)

En las sesiones de prácticas dedicadas a esta tercera parte se muestra el uso de SQL embebido en el lenguaje de programación C. Para ello se utiliza el precompilador para C que proporciona el sistema de gestión de bases de datos Oracle 8, Pro*C. Un precompilador es una herramienta de programación que permite incluir sentencias de SQL en un programa escrito en un lenguaje de alto nivel. Oracle tiene precompiladores para Ada, C, C++, COBOL, FORTRAN, Pascal y PL/I. El precompilador acepta como entrada un programa fuente y traduce las sentencias de SQL embebidas en él, en llamadas a rutinas de Oracle, generando un programa que se compila, enlaza y ejecuta de forma habitual.

En la primera sesión de esta parte se proporciona un programa escrito en C que incorpora sentencias de SQL para realizar una consulta simple sobre la base de datos de prácticas, generando un informe. El programa, que posee algunos errores intencionados de distinta índole, debe ser depurado hasta su puesta en funcionamiento. Para la segunda sesión se propone el desarrollo de un programa en C que realice una consulta compleja en la que se hace necesario el uso de cursores.

Parte 4. Generador de formularios de Developer 2000 de Oracle

Motivación

Para facilitar el desarrollo de aplicaciones interactivas, los sistemas de gestión de bases de datos disponen de generadores de aplicaciones, conocidos también como lenguajes de cuarta generación. Forms, el generador de aplicaciones de Oracle 8, permite generar formularios gráficamente, basándose en las definiciones del esquema de la base de datos. Una vez diseñados los formularios, se pueden usar para la entrada de datos y para realizar operaciones de consulta, modificación y borrado de los mismos. Además, mediante Forms se pueden construir prototipos de modo muy rápido.

PL/SQL es una extensión del lenguaje SQL que incluye capacidad procedural, el uso de bucles y de sentencias de control de flujo, el encapsulado de datos, ocultación de información y el manejo de errores. En un bloque de PL/SQL pueden aparecer varias sentencias SQL como si fueran una instrucción cualquiera dentro de la sintaxis del lenguaje. PL/SQL puede ser utilizado desde diferentes herramientas de Oracle, tales como SQL*Plus o Forms, y también se puede embeber dentro de lenguajes de programación tradicionales.

Desarrollo (3 sesiones de 3 horas)

En la primera sesión de esta parte se desarrolla la interface gráfica de un formulario que presenta toda la información relativa a las facturas de la base de datos de prácticas. En las dos sesiones restantes se debe completar el formulario con los disparadores correspondientes para que presente toda la funcionalidad requerida, tanto para consulta como para entrada y actualización de datos.

Además, el formulario deberá observar ciertas reglas de negocio.

Parte 5. Desarrollo de aplicaciones de bases de datos con Access 97

Motivación

Una vez familiarizado el alumnado con el sistema de gestión de bases de datos Oracle 8, sus precompiladores y la herramienta de desarrollo de formularios, se pretende, mediante esta última práctica, que los alumnos conozcan los puntos en común y las diferencias que presenta la herramienta Access 97, cuyo uso está casi generalizado en pequeñas y medianas empresas. Incluso las grandes empresas también utilizan este sistema para implementar pequeñas aplicaciones.

Desarrollo (1 sesión de 3 horas)

En esta última sesión se crean las tablas de la base de datos de prácticas en Access, se crean las relaciones que establecen las claves ajenas y se desarrolla un formulario, similar al desarrollado con Forms, para consultar, insertar, borrar y modificar facturas.

Bibliografía del programa de prácticas

A continuación se presenta la bibliografía para el programa de prácticas. Hay un texto básico que se ha desarrollado para las prácticas y una bibliografía complementaria. La bibliografía complementaria se ha clasificado por temas.

J. Aliaga, S. García, M. Marqués, G. Quintana (2000) [3]. Este texto muestra los conceptos necesarios para la realización de todas las sesiones de prácticas, excepto la primera. Se estudian las sentencias de consulta y actualización de datos del lenguaje SQL, ilustrándolas con multitud de ejemplos. Se presenta la herramienta SQL*Plus, el uso de SQL embebido en programas escritos en lenguaje C, el lenguaje PL/SQL y la herramienta

Forms de Developer 2000. Cada tema tiene una serie de ejercicios propuestos que son los que se tienen que desarrollar en las sesiones de prácticas.

Access

- J. L. Gordo, A. Gazo, J. Vega (1998) [56].
- T. M. O'Brien, S. J. Pogge, G. E. White (1997) [70].
- J. L. Viescas (1997) [90].

Oracle

- P. Corrigan, M. Gurry (1996) [38].
- P. Dorsey (1997) [47].
- R. J. Muller (1996) [69].
- S. Urman (1998) [88].

SQL

- C. J. Date, H. Darwen (1997) [41].
- R. Sunderraman (1998) [82].

5.3 Docencia y evaluación

Planificación temporal

La figura 5.1 muestra la distribución en el cuatrimestre de los temas de teoría y las sesiones de prácticas. Los cuatrimestres son de quince semanas y en cada semana se imparten tres horas de clase de teoría (cada hora en un día distinto) y se realiza una sesión de prácticas de tres horas.

La primera toma de contacto con un base de datos relacional se realiza en la primera sesión de prácticas, lo que ayuda a comprender mejor los conceptos teóricos que se tratan en el tema 4 sobre este modelo de bases de datos. El desarrollo de este tema se realiza en paralelo con las sesiones de prácticas correspondientes a consultas de datos, por lo que las nociones que aparecen en teoría se pueden utilizar de modo práctico y, a su vez, en las clases de teoría se pueden plantear ejemplos relacionados con la práctica. El resto de temas de teoría se dirigen hacia el diseño de bases de datos, mientras que las prácticas se orientan hacia las aplicaciones, por lo que no tan es necesario realizar una sincronización.

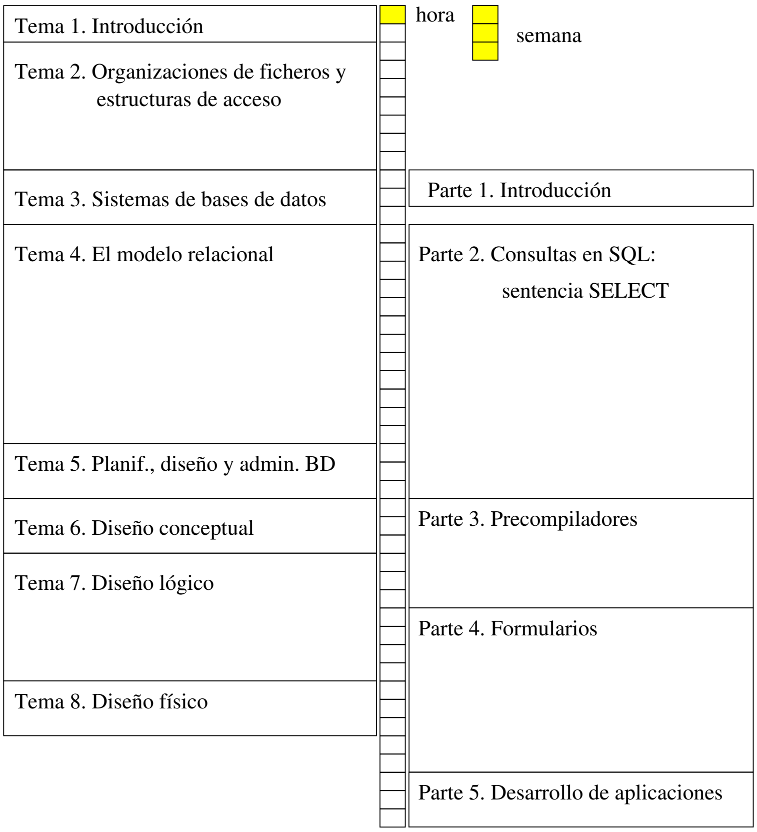

Figura 5.1 Distribución de los temarios por clases.

Desarrollo de las clases de teoría

El método utilizado para las clases de teoría es el presentado en el apartado 3.2 sobre metodología docente (capítulo 3). Durante la explicación se plantean preguntas para incitar al alumnado a razonar y a participar, evitando así que la clase se convierta en un monólogo del profesor. Todas las clases de explicación siguen una misma estructura y se apoyan en el uso de transparencias, que se encuentran disponibles para el alumnado en el servicio de reprografía y a través de la Web. En los temas 4 y 7 se dedican algunas clases a resolver ejercicios. Estos ejercicios se encuentran en un boletín que poseen los alumnos. Algunos de estos ejercicios se resuelven en clase tal y como se explica en el apartado 3.2.

Desarrollo de las sesiones de prácticas

La única sesión de la parte 1 está guiada por el profesorado, de modo que éste va indicando los pasos a realizar en cada momento y el objetivo que persigue cada uno de ellos. Hay que tener en cuenta que en esta sesión se tiene el primer contacto con una base de datos relacional y se ha de manejar una herramienta desconocida para el alumnado (Access).

Algo similar ocurre en la primera sesión de la parte 4, en la que se inicia el aprendizaje de la herramienta Forms de Oracle mediante el diseño de la interface gráfica de un formulario, por lo que esta sesión también es guiada.

Para el resto de las sesiones de prácticas, el alumnado debe llevar el trabajo a desarrollar previamente preparado. Para ello, dispone de los apuntes necesarios y los enunciados de los ejercicios. Estas sesiones comienzan con un breve comentario de los objetivos a cubrir, resolviéndose a continuación las posibles dudas surgidas en la preparación de los ejercicios. Durante la última sesión de las partes 3, 4 y 5 se propondrá un pequeño ejercicio que deberá ser resuelto en clase.

Método de evaluación

Para evaluar el aprovechamiento en la asignatura se realiza un examen. Este examen consta de una prueba objetiva y una serie de problemas. Para preparar la prueba objetiva se tienen en cuenta las reglas comentadas apartado 3.2 (capítulo 3).

Los problemas que se deben resolver en el examen son los siguientes:

- Realizar consultas de datos mediante el álgebra relacional.
- Realizar consultas de datos mediante el cálculo relacional.
- Realizar el diseño lógico de una base de datos relacional a partir de una especificación de requisitos y del esquema conceptual correspondiente.
- Realizar consultas de datos mediante el lenguaje SQL (resolver consultas y determinar qué consulta realiza una sentencia SQL dada).

Las consultas de álgebra relacional y de cálculo relacional se realizan sobre la base de datos que se ha utilizado en el boletín de ejercicios sobre el que se ha trabajado en el tema 4. Del mismo modo, las consultas de SQL se realizan sobre la base de datos de prácticas, sobre la que se han desarrollado todas las prácticas del curso. En esta parte del examen se pueden consultar libros y apuntes.

Además, para superar la asignatura se exige la asistencia a las sesiones prácticas correspondientes a las partes 3, 4 y 5 y se controla su aprovechamiento. Para esto último, se propone un pequeño ejercicio en la última sesión de cada parte que debe ser entregado al finalizar la sesión correspondiente.

Capítulo 6

Diseño de sistemas de bases de datos

Esta asignatura amplía los conocimientos adquiridos en la asignatura de *Ficheros y bases de datos*, profundizando en la estructura de los sistemas de gestión de bases de datos desde el punto de vista de sus componentes. Estos realizan tareas tales como la optimización de consultas, el control de concurrencia, la recuperación ante fallos y el control de la seguridad.

Un segundo objetivo es el de presentar las nuevas características que se incorporan a las bases de datos, como las reglas de comportamiento, y los nuevos modelos de bases de datos. Concretamente, se introducen las tendencias y características de los sistemas de gestión de bases de datos orientadas a objetos, analizando las distintas tendencias. Se introducen también algunos conceptos de bases de datos distribuidas y se presenta la arquitectura cliente–servidor.

El tercer y último objetivo es el de presentar algunas de las líneas de investigación en bases de datos, prestando especial atención a aquellas que parecen tener un mayor impacto, como son las aplicaciones de bases de datos en Internet.

Mediante las prácticas se pretende ejercitar gran parte de los conceptos del programa de teoría sobre un sistema de gestión de bases de datos real.

6.1 Desarrollo del programa de teoría

Para cada tema que forma parte del programa de teoría se incluyen cuatro puntos. En la *motivación* se justifica el estudio del tema, introduciendo los conceptos que trata. En los *contenidos* se enumeran los distintos apartados que componen el tema. En el *resumen* se expone brevemente la materia que se presenta en cada uno de los aparados anteriores. Por último, en la *bibliografía* se citan los capítulos de los textos recomendados que tratan la materia del tema presentado. Estas recomendaciones se dan alrededor de tres textos básicos: ELMASRI Y NAVATHE (1999) [50], CONNOLLY, BEGG Y STRACHAN (1998) [36] y ATZENI ET AL. (1999) [7]. El estudiante no necesita los tres textos para desarrollar la asignatura, de hecho cualquiera de ellos sería suficiente para cubrir la mayor parte de sus contenidos. Sin embargo, se ha decidido tratar los tres textos para que el alumno tenga la posibilidad de escoger el que mejor se le adapte (por la extensión o el modo en que se trata cada apartado, por otros temas que contiene y que no se tratan en la asignatura, etc.).

En el último apartado del desarrollo del programa de teoría se presenta una bibliografía. Además de los textos básicos ya citados, esta bibliografía contiene otros textos a los que se acompaña de un breve comentario.

Tema 1. Bases de datos activas

Motivación

En muchas aplicaciones, la base de datos debe evolucionar independientemente de la intervención del usuario como respuesta a un suceso o una determinada situación. En los sistemas de gestión de bases de datos tradicionales (pasivas), la evolución de la base de datos se programa en el código de las aplicaciones, mientras que en los sistemas de gestión de bases de datos activas esta evolución es autónoma y se define en el esquema de la base de datos.

El poder especificar reglas con una serie de acciones que se ejecutan automáticamente cuando se producen ciertos eventos, es una de las mejoras de los sistemas de gestión de bases de datos que se consideran de gran importancia desde hace algún tiempo. De hecho, la mayoría de los sistemas relacionales comerciales disponen de disparadores. Se ha hecho mucha investigación sobre lo que debería ser un modelo general de bases de datos activas desde que empezaron a aparecer los primeros disparadores. El modelo que se viene utilizando para especificar bases de datos activas es el modelo ECA (evento–condición–acción).

Mediante los sistemas de bases de datos activas se consigue un nuevo nivel de independencia de datos: la independencia de conocimiento. El conocimiento que provoca una reacción se elimina de los programas de aplicación y se codifica en forma de reglas activas. De este modo, al encontrarse las reglas definidas como parte del esquema de la base de datos, se comparten por todos los usuarios, en lugar de estar replicadas en todos los programas de aplicación. Cualquier cambio sobre el comportamiento reactivo se puede llevar a cabo cambiando solamente las reglas activas, sin necesidad de modificar las aplicaciones.

Además, mediante los sistemas de bases de datos activas se hace posible el integrar distintos subsistemas (control de accesos, gestión de vistas, etc.) y se extiende el ámbito de aplicación de la tecnología de bases de datos a otro tipo de aplicaciones.

Uno de los problemas que ha limitado el uso extensivo de reglas activas, a pesar de su potencial para simplificar el desarrollo de bases de datos y de aplicaciones, es el hecho de que no hay técnicas fáciles de usar para diseñar, escribir y verificar reglas. Por ejemplo, es bastante difícil verificar que un conjunto de reglas es consistente, es decir, que no se contradice. También es difícil garantizar la terminación de un conjunto de reglas bajo cualquier circunstancia. Para que las reglas activas alcancen todo su potencial, es necesario desarrollar herramientas para diseñar, depurar y monitorizar reglas activas que puedan ayudar a los usuarios en el diseño y depuración de sus reglas.

Contenidos (6 horas)

1.1 Introducción.

1.2 El modelo evento–condición–acción.

1.3 Propiedades de las reglas activas.

1.4 Aplicaciones de las bases de datos activas.

1.5 Sistemas de gestión de bases de datos activas.

Resumen

En este tema se presentan sistemas de bases de datos en los que se pueden especificar reglas activas. Se muestra el modelo evento–condición–acción, en el que las reglas son disparadas automáticamente por eventos, iniciando las acciones especificadas en la declaración de la regla si se cumple cierta condición. Los eventos ante los que se puede activar una regla son de diversos tipos: actualizaciones de la base de datos, consultas sobre la base de datos o el hecho de que se alcance un instante de tiempo determinado. Las acciones pueden realizar actualizaciones sobre la base de datos, consultas, operaciones de control de transacciones o procedimientos escritos en un lenguaje de programación.

Se estudian las propiedades que deben poseer las reglas activas cuando se definen: terminación, confluencia e idéntico comportamiento observable. La más importante de estas propiedades es la de terminación: ya que la acción de una regla puede provocar la activación de otra regla, es fundamental asegurarse de que todas las transacciones que provocan la activación de reglas finalizan en un número finito de pasos. Para verificar esta propiedad se utilizan los grafos de activación.

Mediante una serie de ejemplos se ilustran las distintas aplicaciones de las bases de datos activas: comprobación de la integridad, restauración de la consistencia, generación de datos derivados (materialización de vistas), control de la seguridad y definición de las reglas de funcionamiento o de negocio de la organización o empresa propietaria de los datos. Por último, se muestran las características de dos paquetes comerciales que ya tienen parte de la funcionalidad de las bases de datos activas en forma de disparadores: Ingres y Oracle 8.

Bibliografía

En el capítulo 23 del texto de Elmasri y Navathe (1999) se presentan los sistemas de bases de datos activas, con ejemplos basados en el sistema de gestión de bases de datos Oracle. Algunas de las aplicaciones de las bases de datos activas se estudian en otros capítulos, como las vistas materializadas o su uso en los grandes almacenes de datos (*data warehousing*). Atzeni et al. (1999) tratan las bases de datos activas en el capítulo 12, haciendo un estudio de los disparadores en Oracle y DB2.

Tema 2. Bases de datos orientadas a objetos

Motivación

Los modelos de bases de datos tradicionales (relacional, red y jerárquico) han sido capaces de satisfacer con éxito las necesidades, en cuanto a bases de datos, de las aplicaciones de gestión tradicionales. Sin embargo, presentan algunas deficiencias cuando se trata de aplicaciones más complejas o sofisticadas como, por ejemplo, el diseño y fabricación en ingeniería (CAD/CAM, CIM), los experimentos científicos, los sistemas de información geográfica o los sistemas multimedia. Los requerimientos y las características de estas nuevas aplicaciones difieren en gran medida de las típicas aplicaciones de gestión: la estructura de los objetos es más compleja, las transacciones son de larga duración, se necesitan nuevos tipos de datos para almacenar imágenes y textos, y hace falta definir operaciones no estándar, específicas para cada aplicación. Las bases de datos orientadas a objetos se crearon para tratar de satisfacer las necesidades de estas nuevas aplicaciones. La orientación a objetos ofrece flexibilidad para manejar algunos de estos requisitos y no está limitada por los tipos de datos y los lenguajes de consulta de los sistemas de bases de datos tradicionales.

Otro motivo para la creación de las bases de datos orientadas a objetos es el creciente uso de los lenguajes orientados a objetos para desarrollar aplicaciones. Las bases de datos se han convertido en piezas fundamentales de muchos sistemas de información y las bases de datos tradicionales son difíciles de utilizar cuando las aplicaciones que acceden a ellas están escritas en un lenguaje de programación orientado a objetos como C++, Smalltalk o Java.

Los fabricantes de los sistemas de gestión de bases de datos relacionales también se han dado cuenta de que existen nuevas necesidades en el modelado de datos, por lo que las nuevas versiones de sus sistemas incorporan muchos de los rasgos propuestos para las bases de datos orientadas a objetos, como ha ocurrido con Informix y Oracle. Esto ha dado lugar al modelo relacional extendido y a los sistemas que lo implementan se les denomina sistemas objeto–relacionales. La nueva versión del lenguaje SQL, SQL3 (SQL:1999), incluye características de la orientación a objetos.

Durante los últimos años, se han creado muchos prototipos experimentales de sistemas de bases de datos orientadas a objetos y también sistemas comerciales. Conforme éstos fueron apareciendo, surgió la necesidad de establecer un modelo estándar y un lenguaje para manejarlo. Para ello, los fabricantes de los sistemas de gestión de bases de datos orientadas a objetos formaron un grupo denominado ODMG (*Object Database Management Group*), que propuso el estándar ODMG–93 y que ha evolucionado en ODMG 2.0, su última versión. El uso de estándares proporciona portabilidad, permitiendo que una aplicación se pueda ejecutar sobre sistemas distintos con mínimas modificaciones. Los estándares también proporcionan interoperabilidad, permitiendo que una aplicación pueda acceder a varios sistemas diferentes. Y una tercera ventaja de los estándares es que permiten que los usuarios puedan comparar entre distintos sistemas comerciales, dependiendo de qué partes del estándar proporcionan.

Contenidos (6 horas)

2.1 Introducción.

2.2 Características de la orientación a objetos.

2.3 Constructores de tipos de objetos.

2.4 Jerarquías y herencia.

2.5 Polimorfismo.

2.6 El modelo de datos ODMG.

2.7 Lenguaje de definición de objetos.

2.8 Lenguaje de consultas de objetos (OQL).

2.9 Sistemas objeto–relacionales. SQL3 (SQL:1999).

Resumen

En este tema se presentan los conceptos de la orientación a objetos aplicada a los sistemas de bases de datos. Esta técnica fue propuesta para cubrir las necesidades de las nuevas aplicaciones de bases de datos y para añadir la funcionalidad que proporcionan las bases de datos a los lenguajes de programación orientados a objetos. En primer lugar, se estudian los conceptos principales utilizados en las bases de datos orientadas a objetos: identidad de objetos, constructores de tipos, encapsulamiento de operaciones, tipos de jerarquías y herencia, objetos complejos, polimorfismo y versiones.

En este tema se muestra cómo algunos de estos conceptos se implementan con el estándar ODMG. Este estándar tiene varios tipos de objetos incorporados que se describen mediante sus interfaces (las interfaces especifican las operaciones de cada tipo de objeto). Estos tipos son la base del lenguaje de definición de objetos y del lenguaje de consultas de objetos. Se muestra también cómo mediante el lenguaje de definición de objetos, los usuarios definen clases para sus aplicaciones que heredan las operaciones de las interfaces correspondientes. En estas clases, el usuario puede definir dos tipos de propiedades: atributos y relaciones, además de las operaciones que se pueden aplicar a los objetos de la clase. Se presenta después el lenguaje de consultas de objetos OQL. Este lenguaje sigue el principio de ortogonalidad en la construcción de consultas: una operación se puede aplicar al resultado de otra operación siempre que el tipo del resultado sea el adecuado para dicha operación. La sintaxis del lenguaje de consultas de objetos copia muchos de los constructores del lenguaje SQL, aunque incluye conceptos adicionales como trayectorias, herencia, métodos, relaciones y colecciones.

Por último, se hace un repaso por las tendencias actuales en el campo de las bases de datos, que han dado lugar a los sistemas de gestión de bases de datos objeto–relacionales. Estos son sistemas relacionales comerciales que han incorporado extensiones de tipos que permiten su aplicación a nuevos dominios, como las bases de datos espaciales, temporales o documentales. También se muestran las nuevas características que se han añadido al lenguaje estándar SQL, para dar soporte a las bases de datos orientadas a objetos y proporcionar tipos abstractos de datos. A esta nueva versión se la denomina SQL3 o SQL:1999.

Bibliografía

Los capítulos 11 y 12 del texto de ELMASRI Y NAVATHE (1999) tratan ampliamente la orientación a objetos en la tecnología de bases de datos, como también se hace en los capítulos 21 y 22 del texto de CONNOLLY, BEGG Y STRACHAN (1998) o en el capítulo 11 de ATZENI ET AL. (1999). Este es un tema que aparece en gran parte de los textos básicos sobre bases de datos, aunque sólo los más recientes tratan los sistemas objeto–relacionales. El texto de ELMASRI Y NAVATHE (1999) los analiza en el capítulo 13, mientras que el texto de CONNOLLY, BEGG Y STRACHAN (1998) lo hace en el capítulo 23 (ATZENI ET AL. (1999) los trata en el mismo capítulo 11).

Tema 3. Bases de datos distribuidas

Motivación

La tecnología de las bases de datos distribuidas surgió a partir de la mezcla de dos tecnologías: la tecnología de las bases de datos y la tecnología de las redes y las comunicaciones. Esta última ha hecho enormes progresos en tecnologías cableadas y no cableadas: desde las comunicaciones por satélite y celulares, a la estandarización de protocolos como Ethernet y TCP/IP, así como la explosión de Internet. Muchas de las bases de datos de los años setenta y de principios de los ochenta tendieron a la centralización, dando lugar a gigantes bases de datos monolíticas. Sin embargo, a finales de los ochenta, la tendencia ha sido hacia una mayor descentralización y autonomía de proceso. Tras los avances que se produjeron en computación distribuida en el campo de los sistemas operativos, los investigadores en bases de datos se esforzaron en orientar su trabajo hacia la distribución de datos, la distribución de consultas y de procesamiento de transacciones, y la gestión distribuida de metadatos, desarrollándose muchos prototipos de investigación. Sin embargo, todavía no ha surgido un sistema de gestión de bases de datos distribuidas que implemente la funcionalidad y las técnicas propuestas por los investigadores en bases de datos distribuidas. La mayor parte de los fabricantes no se han centrado en desarrollar sistemas de gestión de bases de datos distribuidas "puros", sino que se han dedicado a desarrollar sistemas basados en el modelo cliente–servidor, o sistemas de gestión de bases de datos heterogéneas activas.

Sin embargo, empresas y organizaciones están muy interesados en la descentralización de procesos (a nivel del sistema) y en la integración de sus recursos de información (a nivel lógico). Estos recursos de información los forman sus sistemas de bases de datos, distribuidas geográficamente, sus aplicaciones y sus usuarios.

Contenidos (6 horas)

3.1 Introducción.

3.2 Características de las bases de datos distribuidas.

3.3 Diseño de bases de datos distribuidas. Fragmentación y replicación de datos.

3.4 Tipos de sistemas de bases de datos distribuidas.

3.5 Procesamiento de consultas en bases de datos distribuidas.

3.6 Arquitectura cliente–servidor y bases de datos distribuidas.

3.7 Ejemplo: bases de datos distribuidas en Oracle.

3.8 Tendencias futuras en la tecnología cliente–servidor.

Resumen

En este tema se estudian las bases de datos distribuidas, los sistemas de gestión de bases de datos distribuidas y cómo se utiliza la arquitectura cliente–servidor como plataforma para el desarrollo de aplicaciones de bases de datos.

Las bases de datos distribuidas constituyen un tema muy amplio del que aquí se hace una introducción. Es primer lugar se discuten las ventajas que pueden ofrecer las bases de datos distribuidas frente a los sistemas centralizados. Se estudian los aspectos de diseño relativos a la fragmentación de datos (horizontal y vertical), la replicación de datos y su distribución. La replicación de datos mejora la fiabilidad del sistema y también su disponibilidad. Se clasifican los sistemas de gestión de bases de datos distribuidas utilizando criterios como el grado de homogeneidad de los módulos *software* y el grado de autonomía local. Se trata la gestión de los sistemas de bases de datos federadas, centrándose en la necesidad de soportar varios tipos de autonomías y de manejar la heterogeneidad semántica.

Se muestran algunas de las técnicas que se utilizan en el procesamiento de consultas distribuidas. En la optimización de consultas distribuidas, el factor que más influye es el coste de comunicación. Se comparan varias técnicas de ejecución de concatenaciones (*join*) y se presenta la técnica del *semijoin* para concatenar relaciones que se encuentran en distintas localizaciones.

En este tema también se presenta la arquitectura cliente–servidor, relacionándola con las bases de datos distribuidas, y se describen algunas características que incluye Oracle para gestionar bases de datos distribuidas. Por último, se hacen algunas observaciones sobre el futuro de la tecnología cliente–servidor.

Bibliografía

En el capítulo 24 del texto de Elmasri y Navathe (1999) se presentan las bases de datos distribuidas y se comenta el modo en que se están llevando a la práctica: mediante la arquitectura cliente–servidor. Connolly, Begg y Strachan (1998) presentan este mismo tema con más profundidad en los capítulos 19 y 20, y Atzeni et al. (1999) lo hacen en el capítulo 10.

Tema 4. Técnicas de implementación de sistemas de gestión de bases de datos

Motivación

En este tema se presentan las técnicas que se utilizan para implementar los sistemas de gestión de bases de datos. Una parte muy importante de todo sistema de gestión de bases de datos es el diccionario de datos o catálogo. El diccionario de datos es el corazón del sistema, es una "mini-base de datos" y su función principal es almacenar los esquemas o descripciones de las bases de datos que el sistema de gestión de bases de datos mantiene. Esta información es lo que se suele denominar metadatos y comprende la descripción del esquema conceptual de la base de datos, el esquema interno, los esquemas externos y las correspondencias entre los esquemas de diferentes niveles. Además, el diccionario de datos almacena otro tipo de información, necesaria para los distintos módulos del sistema de gestión de bases de datos como son el optimizador de consultas o el módulo que se encarga de la seguridad.

Otro aspecto muy importante de los sistemas de gestión de bases de datos es el procesamiento de las consultas. Toda consulta expresada en un lenguaje de alto nivel como SQL debe ser reconocida, analizada y validada. El reconocedor identifica los símbolos del lenguaje en el texto de la consulta, y el analizador comprueba la sintaxis de la consulta para determinar si se ha expresado de acuerdo a las reglas de la gramática del lenguaje de consultas. Además, la consulta se valida comprobando que todos los nombres de los atributos y de las relaciones son válidos en el esquema de la base de datos que se está consultando. Entonces se crea una representación interna de la consulta, normalmente mediante una estructura en forma de árbol, denominada árbol de consulta. A partir de aquí, el sistema de gestión de bases de datos debe determinar una estrategia de ejecución para obtener los datos de la consulta de los ficheros de la base de datos. Lo típico es que una misma consulta tenga varias estrategias de ejecución posibles, por lo que los sistemas de gestión de bases de datos poseen un módulo que se encarga de escoger la más apropiada: el optimizador de consultas.

Para la implementación de la optimización de consultas hay dos técnicas. La primera de ellas se basa en reglas heurísticas para ordenar las operaciones en la estrategia de ejecución de la consulta. Una regla heurística es una regla que funciona bien en la mayoría de los casos, aunque no se garantiza que funcione bien en todos los casos posibles. La segunda técnica conlleva la estimación sistemática del coste de distintas estrategias de ejecución y la elección del plan de ejecución que tiene el menor coste estimado. Normalmente, los optimizadores de consultas combinan las dos técnicas.

En el procesamiento de bases de datos, el concepto de transacción proporciona un mecanismo para describir unidades lógicas de trabajo. Los sistemas de procesamiento de transacciones son sistemas con grandes bases de datos y cientos de usuarios concurrentes que ejecutan sus transacciones. En estos sistemas se requiere una alta disponibilidad y también respuesta rápida para todos los usuarios. El problema del control de la concurrencia sucede cuando varias transacciones lanzadas por varios usuarios interfieren unas con otras de modo que se producen resultados erróneos. Cuando una transacción se lanza al sistema de gestión de bases de datos para ser ejecutada, el sistema es responsable de que, o bien todas las operaciones de la transacción se ejecuten con éxito, con lo que su efecto quedará permanentemente en la base de datos, o bien la transacción no tenga ningún efecto sobre la base de datos ni sobre ninguna otra transacción. Esto último es necesario cuando la transacción falla tras la ejecución de alguna de sus operaciones.

Hay varias técnicas de control de concurrencia que garantizan que varias transacciones que se ejecutan concurrentemente, no interfieran. La mayoría de estas técnicas garantizan la serializabilidad de planes mediante el uso de protocolos. La mayoría de los sistemas de gestión de bases de datos comerciales utilizan protocolos de bloqueo de ítems de datos. También existen protocolos de control de concurrencia que utilizan marcas de tiempo para ordenar las transacciones, protocolos multiversión que mantienen varias versiones de cada dato, y protocolos basados en el concepto de validación o certificación de transacciones, a los que se suele llamar protocolos optimistas.

Otro aspecto a tener en cuenta en la implementación de los sistemas de gestión de bases de datos es la recuperación ante fallos, tanto fallos del sistema como fallos en las transacciones. El objetivo principal de la recuperación es garantizar la atomicidad de las transacciones. Si una transacción falla antes de finalizar, el mecanismo de recuperación se tiene que asegurar de que la transacción no ha tenido ningún efecto sobre la base de datos. Para hacer esto, el sistema debe mantener un diario con información sobre los cambios que las transacciones realizan sobre los ítems.

Los sistemas de gestión de bases de datos suelen tener un subsistema de seguridad que se encarga de garantizar la seguridad de la base de datos frente a accesos no autorizados. Esto es necesario en sistemas de bases de datos multiusuario en los que no todos los usuarios pueden acceder a cualquier porción de la base de datos. Hay dos tipos de mecanismos de seguridad. Los primeros, denominados mecanismos de seguridad discrecionales, se utilizan para conceder permisos a los usuarios de modo que puedan realizar operaciones sobre ciertos ficheros de datos, registros o campos. Los segundos son los mecanismos de seguridad obligatorios, que se utilizan para reforzar la seguridad multinivel, clasificando los datos y los usuarios en varias clases o niveles de seguridad e implementando la política de seguridad que establece la organización propietaria de la base de datos.

Contenidos (12 horas)

4.1 Introducción.

4.2 Diccionario de datos.

4.3 Procesamiento de consultas y optimización.

4.4 Procesamiento de transacciones.

4.5 Control de la concurrencia.

4.6 Recuperación.

4.7 Seguridad.

Resumen

En el primer apartado de este tema se muestra la estructura del diccionario de datos de un sistema de gestión de bases de datos relacionales y cómo éste almacena las construcciones del modelo relacional, incluyendo información sobre restricciones, índices y vistas. A modo de ejemplo, se ven algunos aspectos del diccionario de datos de Oracle. A continuación, se presenta cómo los distintos módulos del sistema de gestión de bases de datos acceden a la información del diccionario de datos y qué otros tipos de información se almacenan en éste.

En el segundo apartado se trata el procesamiento de consultas. Para ello, se muestra cómo las consultas SQL se traducen al álgebra relacional y diversos modos en que se pueden ejecutar las operaciones del álgebra relacional. Algunas operaciones, como la restricción y la concatenación, pueden ejecutarse de muchas formas diferentes. A continuación, se describen técnicas heurísticas de optimización de consultas que utilizan reglas heurísticas y técnicas algebraicas para mejorar la eficiencia de la ejecución de consultas. Se muestra cómo se puede optimizar heurísticamente un árbol de consulta que representa una expresión del álgebra relacional, reorganizando los nodos del árbol y tranformándolo en otro árbol equivalente que se ejecuta más eficientemente. También se muestran los planes de ejecución de las consultas SQL. Se presentan los métodos de optimización de consultas basados en costes y se muestra cómo se desarrollan funciones de costes para algunos algoritmos de acceso a la base de datos y cómo se usan estas funciones para estimar los costes de las distintas estrategias de ejecución. A modo de ejemplo, se ofrece un resumen del optimizador de consultas de Oracle.

En el tercer apartado se presenta el concepto de transacción y las operaciones principales del procesamiento de transacciones. Se comparan los sistemas monousuario con los sistemas multiusuario y se presentan ejemplos de cómo una ejecución incontrolada de transacciones concurrentes en un sistema multiusuario puede provocar datos erróneos en la base de datos. También se presentan varios tipos de fallos que se pueden dar durante la ejecución de las transacciones.

A continuación, se muestran los estados típicos que atraviesa una transacción durante su ejecución, y se discuten algunos de los conceptos que se utilizan en los métodos de recuperación y de control de concurrencia. El sistema mantiene un historial de los accesos realizados a la base de datos y utiliza esta información para recuperarse ante los posibles fallos. Una transacción puede terminar con éxito, alcanzando su punto de *commit*, o fallar, siendo entonces deshecha (*rollback*). Cuando una transacción termina con éxito, los cambios que ha realizado se hacen permanentes sobre la base de datos. Se presenta también un resumen de las propiedades que deberían tener las transacciones: atomicidad, preservación de la consistencia, aislamiento y perdurabilidad. Después se definen los planes de transacciones como secuencias de ejecución de las operaciones de varias transacciones que pueden intercalarse. Se clasifican los planes según su recuperabilidad. Los planes recuperables garantizan que, una vez se ha hecho el *commit* en una transacción, nunca hace falta deshacerla. Hay otro tipo de planes que añaden la condición de que se garantice que ninguna transacción abortada provoque el tener que abortar en cascada otras transacciones. Los planes estrictos proporcionan una condición más fuerte que permite que la recuperación sea más sencilla, de modo que tan solo hay que restablecer los antiguos valores de los datos que han sido modificados por una transacción abortada. Se define la equivalencia de planes y se ve que un plan serializable es un plan equivalente a algún plan serie. Un plan serializable es un plan correcto. Se presenta después un algoritmo para comprobar la serializabilidad de un plan. Aunque la serializabilidad no se comprueba en los sistemas reales, este algoritmo es útil para definir y verificar los protocolos de control de la concurrencia.

En el cuarto apartado del tema se presentan técnicas de recuperación ante fallos en las transacciones. En primer lugar, se da una idea de cómo es el proceso de recuperación y algunos conceptos que tienen que ver con ella. Se presentan dos tipos de algoritmos de recuperación distintos: los algoritmos de actualización diferida y los algoritmos de actualización inmediata. En los algoritmos de actualización diferida, la base de datos en disco sólo se actualiza cuando la transacción finaliza (*commit*). Esto simplifica la recuperación ya que nunca hay que deshacer transacciones, pero tiene el inconveniente de que se necesitan grandes *buffers* para mantener en memoria todas las actualizaciones que realiza la transacción. Por otra parte, en los algorimos de actualización inmediata, la base de datos se actualiza inmediatamente cuando una transacción hace una actualización, sin que haga falta que la transacción finalice. Cuando una transacción falla, hay que deshacer todas sus actualizaciones. En este apartado también se presenta el protocolo de dos fases que se utiliza en los sistemas de múltiples bases de datos y se comenta cómo se realiza la recuperación ante fallos catastróficos.

En el último apartado de este tema se presentan varias técnicas para controlar la seguridad en los sistemas de bases de datos. Estas técnicas controlan el acceso al sistema de la base de datos como un todo y controlan los permisos para acceder a porciones concretas de la base de datos. El control de acceso a la base de datos como un todo se realiza asignando cuentas con claves a los usuarios. Para controlar el acceso a porciones de la base de datos el sistema proporciona un mecanismo para conceder y revocar permisos. Se presentan los mecanismos que se utilizan para controlar la seguridad multinivel. Estos mecanismos necesitan que los datos y los usuarios se clasifiquen en los distintos niveles de seguridad y hacen que se respeten las reglas que prohiben el flujo de información desde niveles altos de seguridad hacia niveles bajos. Por último, se presenta brevemente el problema de controlar el acceso a las bases de datos estadísticas sobre la población para proteger la privacidad de la información individual, a la vez que se permite el acceso estadístico.

Bibliografía

Las técnicas y algoritmos que se utilizan para implementar los sistemas de gestión de bases de datos aparecen en la mayor parte de los textos básicos sobre bases de datos. Se recomiendan los capítulos del 17 al 22 del texto de ELMASRI Y NAVATHE (1999), o bien los capítulos del 16 al 18 del texto de CONNOLLY, BEGG Y STRACHAN (1998). ATZENI ET AL. (1999) tratan todos los aspectos de este tema de un modo más resumido en el capítulo 9.

Tema 5. Aplicaciones de bases de datos en Internet

Motivación

La evolución de Internet es uno de los fenómenos más importantes de la tecnología de la información. El número de usuarios crece y aumenta su variedad: desde grandes compañías con redes de alta velocidad, a usuarios individuales con conexiones lentas vía módem. Muchas organizaciones y empresas tienen redes privadas en las que utilizan los mismos protocolos y las mismas herramientas que se utilizan en Internet. Estas redes privadas, denominadas *intranets* suelen estar conectadas también a Internet. Una tecnología importante, tanto en Internet como en las *intranets* es la Web (*World Wide Web*). La Web ha introducido un nuevo paradigma, tanto para la diseminación como para la adquisición de información, que es fácil de usar, flexible y económico.

Originalmente, la Web era una interface para acceder a documentos distribuidos. Ahora es una plataforma para sistemas de información de todo tipo que lo único que tienen en común es el hecho de que están accesibles, directa o indirectamente, a través de herramientas genéricas denominadas navegadores. A los sistemas de este tipo se les suele denominar sistemas de información Web. Estos sistemas, en conjunto con un sistema de gestión de bases de datos, pueden proporcionar acceso a grandes cantidades de información.

Pero los sistemas de información Web también presentan ciertos problemas. En primer lugar, el tipo de información que soportan es muy heterogénea. Desde el principio la Web se ha utilizado para manejar tanto documentos de texto como información multimedia. Por contra, los sistemas de información tradicionales manejan datos que se pueden respresentar mediante estructuras de datos simples: conjuntos de tuplas con valores atómicos. En resumen, la estructura de la información que se maneja a través de la Web es más compleja y menos estructurada y regular que la que se utiliza en las bases de datos.

Otra característica que distingue a los sistemas de información Web es que ofrecen mayor flexibilidad al usuario en la interacción con el sistema: la interacción es conducida por los intereses del usuario y no sigue unos patrones previamente diseñados. Además, los usuarios tienen poca o ninguna posibilidad de modificar los datos.

Por otra parte, la interacción entre el navegador Web y la fuente de información utiliza herramientas distintas a las de los sistemas tradicionales. Aunque hay bastante estandarización en los navegadores (sus interfaces son bastante uniformes), sus características varían dependiendo de los dispositivos que se utilizan para acceder a la Web. Además, las aplicaciones se ofrecen a una gran comunidad: todos los usuarios de Internet. Por lo tanto, los objetivos a tener en cuenta son más amplios y más variados que en los sistemas de información tradicionales.

Muchos expertos opinan que dentro de algún tiempo, el acceso a todas las bases de datos será a través de la Web, incluso las bases de datos personales que se "publican" para un solo usuario.

Contenidos (4 horas)

5.1 Introducción.

5.2 Internet y World Wide Web.

5.3 Sistemas de información en la Web.

5.4 Diseño de sitios Web para suministrar datos.

5.5 Técnicas y herramientas para acceder a bases de datos desde la Web.

Resumen

En primer lugar se presenta una breve historia sobre el desarrollo de Internet y de la Web. La comunicación se realiza a través de una organización estructurada en capas mediante distintos protocolos y servicios (HTML, HTTP, TCP/IP, CGI) que también se presentan en el primer apartado. A continuación, se identifican los tipos de sistemas de información Web: publicación de información para su consulta, transacciones mediante la Web (actualizaciones de datos), comercio electrónico y otras nuevas aplicaciones.

Se presenta, brevemente, cómo se pueden adaptar las técnicas de diseño de bases de datos al diseño de sitios Web que se utilizan para realizar un acceso intensivo a datos. En uno de estos sitios Web se pueden distinguir tres aspectos distintos de los datos: el contenido de información (los datos), la estructura hipertexto (la organización de páginas y enlaces entre ellas) y la presentación (el aspecto gráfico de las páginas y su contenido). Estos tres aspectos presentan cierta similitud con los tres niveles de la arquitectura de una base de datos (interno, conceptual y externo). Cuando se diseña un sitio Web, se parte del esquema conceptual de la base de datos y de los requisitos de los usuarios. A continuación se genera un esquema conceptual hipertexto, después se genera el esquema lógico hipertexto y, a partir de él, se diseña la presentación. Por último se genera el sitio Web, conectándolo con la base de datos utilizando alguna de las tecnologías que se presentan más tarde. La mayor parte de las aplicaciones Web tienen una arquitectura de tres capas: el servidor de la base de datos, el servidor Web y los clientes. Cada una de estas capas puede utilizar distintos sistemas operativos y programas de distintos proveedores.

En el último apartado del tema se presentan diversas técnicas y herramientas que se pueden utilizar para acceder a bases de datos a través de la Web. Lo más habitual es utilizar programas basados en el protocolo CGI, aunque estos programas tienen algunas limitaciones como, por ejemplo, el hecho de que no es obvio el modo en que se pueden gestionar las transacciones que requieren varios accesos a la base de datos. Una de las soluciones que se utilizan son las populares *cookies*.

Hay otras alternativas distintas a los programas CGI, como la disponibilidad de librerías asociadas a los servidores Web, que permiten el acceso directo a la base de datos. Esta es una tecnología basada en los servidores, denominada servidores API (*Application Programming Interface*). Hay otras tecnologías basadas en el cliente que se suelen implementar mediante nuevos lenguajes cuyo desarrollo está muy ligado a Internet y la Web. El más popular es el lenguaje de programación orientado a objetos Java. También se utiliza Javascript, un lenguaje interpretado que permite la extensión y modificación dinámica del contenido de las páginas HTML. Otra solución es ActiveX, una arquitectura que extiende las funciones de los navegadores permitiéndoles ejecutar aplicaciones, aunque esta solución no es portable ni segura.

En el entorno de las soluciones Java, se ha creado el protocolo JDBC (*Java Data Base Connectivity*) que permite que las aplicaciones escritas en Java puedan acceder a bases de datos relacionales. Este protocolo es similar a ODBC (*Open Database Connectivity*), que se desarrolló a principios de los noventa para proporcionar un modo de procesar datos de bases de datos relacionales, independientemente del sistema de gestión de bases de datos.

Por último, se consideran dos estándares de la Web muy importantes: DHTML y XML, y se describe el uso y el papel de ASP (*Active Server Pages*). XML cubre las necesidades de las aplicaciones de bases de datos execpcionalmente bien, por lo que seguramente será la base de muchos nuevos productos y servicios de bases de datos.

Bibliografía

Sobre las aplicaciones de bases de datos en Internet, ELMASRI Y NAVATHE (1999) tan sólo incluyen un apartado en el capítulo 27, mientras que CONNOLLY, BEGG Y STRACHAN (1998) realizan un tratamiento más extenso de este tema en su capítulo 24, y al igual hacen ATZENI ET AL. (1999) en el capítulo 14.

Tema 6. Modelos avanzados de bases de datos

Motivación

Conforme el uso de los sistemas de bases de datos ha ido aumentando, los usuarios han ido demandando funciones adicionales a estos sistemas con el objeto de simplificar la implementación de aplicaciones más avanzadas y complejas. Las bases de datos orientadas a objetos y los sistemas objeto–relacionales permiten a los usuarios extender sus sistemas especificando tipos abstractos de datos adicionales para cada aplicación. Sin embargo, es muy útil identificar las características que tienen en común estas aplicaciones avanzadas, para crear modelos que puedan representarlas. Además, se pueden implementar estructuras de almacenamiento especiales, así como métodos de acceso, para mejorar las prestaciones de estas nuevas características.

Algunas de ellas son, además de las reglas activas que se utilizan en las aplicaciones de bases de datos activas, los conceptos temporales que se utilizan en aplicaciones de bases de datos temporales, así como varios aspectos relacionados con las bases de datos espaciales y las bases de datos multimedia.

Un área que se encuentra en la intersección de las bases de datos, la lógica y la inteligencia artificial, son las bases de datos deductivas. Un sistema de gestión de bases de datos deductivas es un sistema de gestión de bases de datos que posee la capacidad de definir reglas deductivas que pueden deducir o inferir información adicional de los hechos almacenados en la base de datos. Ya que parte del fundamento teórico de algunos sistemas de gestión de bases de datos deductivas es la lógica matemática, a estas bases de datos se las suele denominar bases de datos lógicas.

El incremento en la potencia de procesamiento y de la sofisticación de las herramientas y técnicas de análisis han producido el desarrollo de lo que se denominan grandes almacenes de datos (*data warehousing*). Estos sistemas proporcionan almacenamiento, funcionalidad y buena respuesta a consultas, más allá de las cualidades de las bases de datos orientadas a transacciones. Junto al crecimiento de la potencia, ha llegado una gran demanda de mejorar las prestaciones en el acceso a las bases de datos. Las bases de datos tradicionales equilibran el requisito del acceso a los datos con la necesidad de garantizar su integridad. Sin embargo, en las organizaciones más modernas, los usuarios se suelen encontrar alejados de las fuentes de datos. Muchos de estos usuarios sólo necesitan un acceso muy rápido a un gran volumen de datos, que puede ser cargado en su equipo. Normalmente, estos datos vienen de múltiples bases de datos. Ya que muchos de los análisis realizados son recurrentes y predecibles, se han empezado a diseñar sistemas que realicen estas funciones. Hoy en día, hay una gran necesidad de apoyo para la toma de decisiones, siendo necesario disponer de información al nivel de detalle adecuado. Los grandes almacenes de datos, el procesamiento de análisis en línea (OLAP) y la explotación de datos (*data mining*), proporcionan esta funcionalidad.

Contenidos (6 horas)

6.1 Introducción.

6.2 Bases de datos temporales.

6.3 Bases de datos espaciales.

6.4 Bases de datos multimedia.

6.5 Bases de datos deductivas.

6.6 Grandes almacenes de datos y explotación de datos.

Resumen

En este tema se presentan sistemas de bases de datos que satisfacen los requisitos que tienen en común algunas aplicaciones avanzadas. Estos modelos son las bases de datos temporales, las bases de datos espaciales y las bases de datos multimedia. También se presentan las bases de datos deductivas, en las que se pueden especificar reglas que permiten inferir nueva información basándose en ellas, y la tecnología de los grandes almacenes de datos y la explotación de datos, que permite extraer conocimiento a partir de los datos para la toma de decisiones.

En el primer apartado se presentan algunos conceptos de bases de datos temporales, que permiten que el sistema almacene una historia de los cambios sufridos por los datos, y que los usuarios puedan consultar tanto el estado actual de la base de datos, como los estados pasados. Se ve cómo se pueden implementar relaciones temporales en sistemas relacionales para guardar versiones de las tuplas, y cómo se pueden utilizar objetos complejos para implementar bases de datos temporales que mantienen versiones de cada uno de los atributos.

En el segundo apartado se presentan las bases de datos espaciales, que permiten almacenar objetos que tienen características espaciales, y que poseen operadores para compararlos y manipularlos. Las bases de datos multimedia permiten a los usuarios almacenar y consultar distintos tipos de información multimedia: imágenes, vídeos, sonido y documentos.

Los sistemas de bases de bases de datos deductivas son una rama bastante reciente de las bases de datos. Este campo se ha visto influido por los lenguajes de programación lógicos, concretamente por Prolog. Datalog, un subconjunto de Prolog, se utiliza principalmente como la base de los sistemas de bases de datos deductivas acuales. Las bases de datos deductivas se encuentran todavía en una etapa experimental. Se espera que su adopción por parte de la industria dará un empujón a su desarrollo. En el apartado dedicado a las bases de datos deductivas se mencionan algunas aplicaciones prácticas en las que se están utilizando algunos sistemas comerciales con muy buenos resultados. Últimamente se ha empezado a producir una unión entre bases de datos orientadas a objetos y bases de datos deductivas. Las bases de datos orientadas a objetos tratan de proporcionar un mecanismo para modelar objetos del mundo real encapsulando su estructura y su comportamiento. Las bases de datos deductivas tratan de derivar nuevo conocimiento a partir de los datos existentes, proporcionando relaciones adicionales del mundo real en forma de reglas.

En el último apartado de este tema se presentan dos ramas de la tecnología de bases de datos que van a jugar un papel importante en la década que ahora comienza: los grandes almacenes de datos y la explotación de datos. En los grandes almacenes de datos se tiene una visión multidimensional de los datos, lo que permite que las personas encargadas de tomar decisiones puedan ver los datos con el nivel de detalle que necesitan en cada momento. La explotación de datos es un proceso que permite extraer conocimiento a partir de un gran almacén de datos.

Bibliografía

De los textos básicos, es el de Elmasri y Navathe (1999) el que estudia todos los sistemas que se presentan en este tema (capítulos del 23 al 26). Connolly, Begg y Strachan (1998) incluyen un capítulo sobre grandes almacenes de datos y un capítulo sobre explotación de datos (capítulos 25 y 26, respectivamente), dejando sin tratar el resto de los contenidos de este tema. Atzeni et al. (1999) sólo tratan los grandes almacenes de datos y la explotación de datos en el capítulo 13.

Bibliografía del programa de teoría

S. Abiteboul, P. Bunemann, D. Suciu (1999) [2]. Este texto hace un repaso por la evolución de las estrategias de acceso y procesamiento de datos, que son de gran importancia en casi todas las aplicaciones Web. El libro presenta soluciones a una gran variedad de problemas prácticos, así como los aspectos fundamentales en cuanto a diseño, implementación y optimización. También se presenta en profundidad el lenguaje XML, que permite publicar información estructurada en la Web.

P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone (1999) [7]. Este es uno de los textos básicos de la asignatura. Además de cubrir la mayor parte del programa de teoría, también contiene todos los aspectos que se tratan en la asignatura de *Ficheros y bases de datos*.

E. Bertino, L. Martino (1995) [12]. Este libro trata los sistemas de bases de datos orientadas a objetos y analiza sus modelos de datos, sus lenguajes y sus arquitecturas. Para ilustrar varias características del modelo de datos orientado a objetos, el libro ofrece una descripción de los modelos y lenguajes de los sistemas GemStone, O2 e Iris.

S. Ceri, P. Fraternali (1997) [19]. En este libro se presenta una metodología de diseño orientada a objetos que también se centra en el modelado de sistemas de información que tienen que gestionar grandes cantidades

de datos estructurados. Ya que es una metodología orientada a los datos, en primer lugar se diseña la base de datos y después se diseñan las aplicaciones.

T. Connolly, C. Begg, A. Strachan (1998) [36]. Este es uno de los textos básicos de la asignatura. Además de cubrir la mayor parte del programa de teoría, contiene todos los aspectos que se tratan en la asignatura de *Ficheros y bases de datos*.

C. J. Date (1997) [40]. Este libro, además de contener la mayor parte del programa de teoría de la asignatura de *Ficheros y bases de datos*, incluye algunos capítulos que profundizan en la teoría del modelo relacional, como las relaciones, los dominios o la gestión de la información desconocida (nulos). También realiza un amplio estudio de las bases de datos distribuidas, relacionándolas con los sistemas cliente–servidor, y el estudio de los sistemas orientados a objetos, dedicando un capítulo completo a aproximar los sistemas orientados a objetos a los sistemas relacionales. En la nueva edición [43] se incorpora un capítulo sobre sistemas objeto–relacionales, con un nuevo apéndice sobre SQL3. También se incluye nuevo material sobre grandes almacenes de datos, explotación de datos y bases de datos temporales.

C. J. Date, H. Darwen (1998) [42]. En este texto se presenta una definición formal de un modelo de datos para los sistemas objeto–relacionales, construido sobre el modelo relacional de Codd. Se proporciona un método para definir tipos de datos que incluye un modelo completo de herencia de tipos.

R. Elmasri, S. B. Navathe (1999) [50]. Texto básico para la asignatura que cubre por completo el programa propuesto. Este libro cubre también completamente el programa propuesto para la asignatura de *Ficheros y bases de datos*. En el texto se muestra cómo los sistemas relacionales Informix Univesal Server y Oracle 8 incorporan algunas de las características de la orientación a objetos. Además, se dedica una parte completa a los sistemas de bases de datos orientadas a objetos y a los sistemas objeto–relacionales, presentando el estándar SQL3. También se incluye una parte de aspectos avanzados (bases de datos activas, temporales, espaciales, multimedia, distribuidas, deductivas, grandes almacenes de datos, explotación de datos).

H. García-Molina, J. Widom, J. D. Ullman (2000) [54]. Este libro trata en profundidad todos los aspectos de implementación de los sistemas de gestión de bases de datos: almacenamiento físico, índices, ejecución de consultas, compilación de consultas, fallos del sistema, control de concurrencia, gestión de transacciones e integración de la información (grandes almacenes de datos, OLAP).

R. Kim (1995) [59]. Este texto es un compendio de artículos que tratan tanto la transición de los sistemas relacionales a los postrrelacionales, como la necesidad de proporcionar una base para la coexistencia y la interoperabilidad de las tecnologías de bases de datos antiguas y las modernas. Una parte importante del texto se dedica a las bases de datos orientadas a objetos y también incorpora trabajos en bases de datos multimedia, espaciales, temporales, activas, distribuidas, paralelas y en el tratamiento de la incetidumbre.

R. Kimball (1996) [60]. En este libro se muestra, con varios ejemplos de diversa índole, cómo realizar modelos de datos y diseñar grandes almacenes de datos que soporten sistemas de apoyo a la toma de decisiones multidimensionales avanzados.

M. T. Özsu, P. Valduriez (1999) [73]. Este libro trata ampliamente los sistemas de bases de datos distribuidas, tanto relacionales como orientadas a objetos. Tiene en cuenta aspectos como el diseño de bases de datos distribuidas, la descomposición y localización de los datos, el control de la seguridad, la concurrencia, el procesamiento de consultas, etc. Además, estudia los sistemas paralelos y las nuevas tendencias (grandes almacenes de datos, aplicaciones en la Web y bases de datos móviles).

A. Silberschatz, H. Korth, S. Sudarshan (1998) [76]. Este libro es la nueva edición de la obra con el mismo título en la que Korth figuraba como primer autor. En esta edición se amplía el modelo entidad–relación y se incluyen las bases de datos orientadas a objetos. Una novedad importante es que se tiene en cuenta la relación entre las bases de datos y las arquitecturas de computadores, tratando tanto los sistemas centralizados como los sistemas cliente–servidor. Además, se incluye un capítulo sobre las bases de datos de sistemas paralelos.

J. Ullman (1988) [84]. Este texto integra los conceptos de los sistemas de bases de datos clásicos con la nueva tecnología que se estaba comenzando a desarrollar para poder almacenar conocimiento además de los datos. Este primer volumen está dedicado, fundamentalmente, a los sistemas clásicos, aunque se dedica un capítulo a la lógica como modelo de datos que sirve para explicar el cálculo relacional.

J. Ullman (1989) [85]. Tal y como dice el autor, este segundo volumen es una visión personal sobre lo que él cree que aporta la teoría de los sistemas de bases de datos, al desarrollo de sistemas de bases de datos y sistemas basados en el conocimiento. La mayor parte de este texto se dedica a la optimización de de programas lógicos, por lo que incluye el importante aspecto de la optimización de consultas.

J. Widom, S. Ceri (1996) [92]. Este texto es un compendio de artículos sobre bases de datos activas. Explican su motivación y su historia, las funciones de los sistemas de bases de datos activas y cómo éstas pueden mejorarse y extenderse, cómo los productos comerciales incorporan reglas activas mediante disparadores y, también, se presentan las futuras líneas de trabajo en este campo.

C. Zaniolo, S. Ceri, T. S. Faloutsos, V. S. Subrahmanian, R. Zicari (1997) [93]. Este texto estudia con profundidad los aspectos de algunos sistemas avanzados de bases de datos: bases de datos activas, bases de datos temporales, bases de datos y sistemas basados en el conocimiento, bases de datos espaciales, documentales y multimedia, y el tratamiento de la información incierta en las bases de datos.

Recursos en Internet

- *Object Data Management Group (ODMG).*
  <http://www.odmg.org> [Consulta: 30 de junio de 2000].

- *Extensible Markup Language (XML).*
  <http://www.w3.org/xml> [Consulta: 30 de junio de 2000].

- *Worldwide Web Consortium.*
  <http://www.w3.org> [Consulta: 30 de junio de 2000].

6.2 Desarrollo del programa de prácticas

El programa de prácticas se divide en cinco partes, cada una de ellas con un número de sesiones distinto. En cada parte se practican los conceptos de un tema distinto del programa teoría. En el último apartado del desarrollo del programa de prácticas se presenta la bibliografía.

Todas las sesiones de prácticas se realizan bajo el sistema de gestión de bases de datos Oracle 8. En una de las sesiones de prácticas también se trabaja con el sistema Access 97.

Parte 1. Bases de datos activas

Desarrollo (3 sesiones de 3 horas)

En las sesiones que componen esta primera parte se utiliza el mecanismo de disparadores de Oracle 8 para llevar a cabo distintas aplicaciones de las bases de datos activas. Para ello, se proporciona el esquema de una base de datos que será creado y para el que se deberá crear disparadores que observen ciertos requisitos propuestos y que tienen que ver con la comprobación de la integridad, la restauración de la consistencia, el mantenimiento de información derivada y el modelado de reglas de negocio de la organización propietaria de la base de datos.

También se proporcionará la definición de ciertas vistas, para las que se deberá definir disparadores que implementen las operaciones de actualización que se propongan para las mismas.

Los disparadores se escribirán en el lenguaje PL/SQL, que ya ha sido presentado y utilizado en la asignatura de *Ficheros y bases de datos*. Para comprobar que los disparadores se han implementado correctamente, se crearán formularios básicos con la herramienta Forms de Developer 2000, con la que también se ha trabajado en la asignatura de *Ficheros y bases de datos*.

Parte 2. Bases de datos objeto–relacionales

Desarrollo (3 sesiones de 3 horas)

En las sesiones que componen esta segunda parte del programa de prácticas se estudian las características de la orientación a objetos de Oracle 8, que es un sistema objeto–relacional. Para ello, se realizarán dos ejercicios. En el primer ejercicio se proporciona un diagrama entidad–relación a partir del que se deberá obtener un esquema de la base de datos con una visión orientada a objetos. En este esquema se deberán definir los tipos de objetos y las tablas de la base de datos, y se deberán realizar determinadas inserciones y consultas de datos. Para el segundo ejercicio también se proporcionará un diagrama entidad–relación sobre el que se harán ciertas consideraciones que harán necesario el uso de métodos en los tipos de objetos para su implementación.

Parte 3. Técnicas de implementación de sistemas de gestión de bases de datos: optimización de consultas

Desarrollo (2 sesiones de 3 horas)

En las sesiones que componen esta parte se pretende comprender el modo en que un sistema relacional y, concretamente Oracle 8, realiza la optimización de consultas. En ambas sesiones se facilita una consulta de datos resuelta de distintos modos en SQL y se examinan los planes de ejecución que genera Oracle para cada una a fin de determinar determinar cuál de las sentencias resulta más eficiente. En la primera sesión se tratará la aproximación por reglas y en la segunda sesión se tratará la aproximación por costes. Además de las sentencias propuestas, el alumno debe encontrar otras sentencias alternativas para resolver la misma consulta, analizando su eficiencia.

Parte 4. Técnicas de implementación de sistemas de gestión de bases de datos: diccionario de datos y seguridad

Desarrollo (1 sesión de 2 horas y una sesión de 3 horas)

Para la primera sesión de esta cuarta parte se proporciona el esquema de una base de datos (tablas, vistas y restricciones). Conforme se vayan ejecutando las sentencias de creación del esquema, se realizarán consultas sobre el diccionario de datos de Oracle 8 para comprobar la información que se registra.

En la segunda sesión se especifican ciertos requisitos de seguridad en cuanto a una serie de usuarios y sobre ciertos objetos del esquema creado en la sesión anterior, de modo que se deberá crear *roles* y usuarios que se encarguen de respetarlos. También se propone como ejercicio averiguar cómo se implementa la seguridad con Access, tratando de llevar a cabo la parte que sea posible del ejercicio realizado con Oracle.

Parte 5. Aplicaciones de bases de datos en Internet

Desarrollo (2 sesiones de 3 horas)

En esta tercera parte se desarrolla una aplicación para la Web que permite realizar consultas y actualizaciones de la base de datos de prácticas de forma remota. Para ello se utilizarán clases de Java de Oracle que permiten que el cliente acceda al sistema de gestión de bases de datos Oracle 8. Mediante el uso de clases de Java para construir este tipo de aplicaciones, cualquier máquina conectada a la red y en la que funcione un navegador puede ser un cliente, sin necesitar tener instalado ningún componente de Oracle.

Bibliografía del programa de prácticas

Access

– J. L. Gordo, A. Gazo, J. Vega (1998) [56].

– T. M. O'Brien, S. J. Pogge, G. E. White (1997) [70].

– J. L. Viescas (1997) [90].

Oracle

– P. Corrigan, M. Gurry (1996) [38].

– R. Sunderraman (1998) [82].

– S. Urman (1998) [88].

6.3 Metodología docente

Planificación temporal

La figura 6.1 muestra la distribución en el cuatrimestre de los temas de teoría y las sesiones de prácticas. Los cuatrimestres son de quince semanas y en cada semana se imparten tres horas de clase de teoría (cada hora en un día distinto) y se realiza una sesión de prácticas de tres horas.

Las prácticas se han organizado de modo que los conceptos teóricos necesarios para desarrollarlas, antecedan a sus sesiones. El objetivo es que los alumnos tengan tiempo suficiente para preparlas.

Desarrollo de las clases de teoría

El método utilizado para las clases de teoría es el presentado en el apartado 3.2 sobre metodología docente (capítulo 3). Durante la explicación se plantean preguntas para incitar al alumnado a razonar y a participar, evitando así que la clase se convierta en un monólogo del profesor. Todas las clases de explicación siguen una misma estructura y se apoyan en el uso de transparencias, que se encuentran disponibles para el alumnado en el servicio de reprografía y a través de la Web.

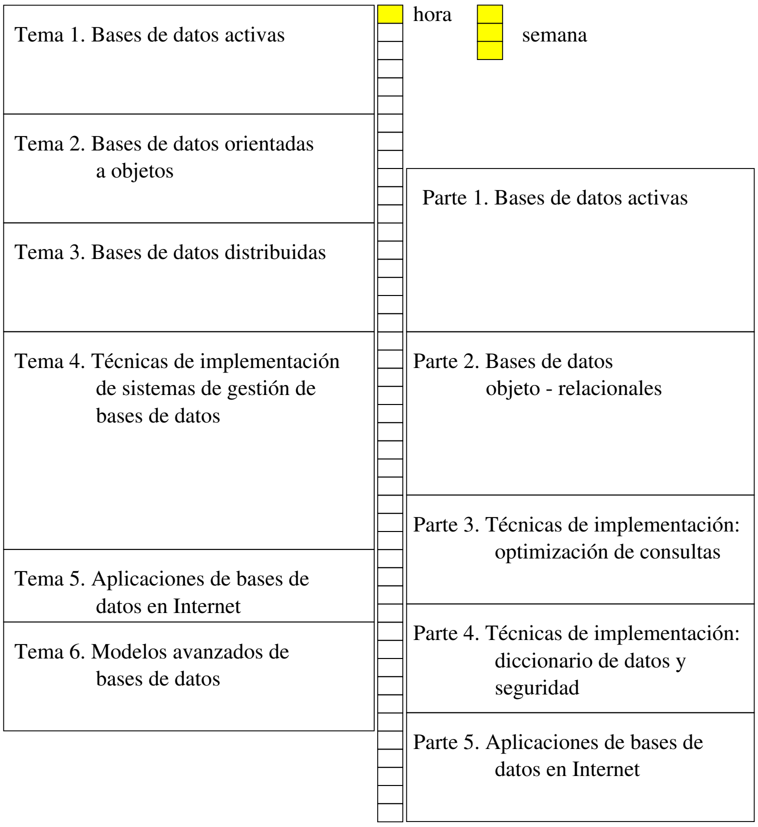

Figura 6.1 Distribución de los temarios por clases.

Desarrollo de las sesiones de prácticas

Para las sesiones de prácticas el alumnado debe llevar el trabajo a desarrollar previamente preparado. Para ello dispone de la documentación necesaria y los enunciados de los ejercicios. Cada sesión comienza con un breve comentario de los objetivos a cubrir, resolviéndose a continuación las posibles dudas surgidas en la preparación de los ejercicios.

Método de evaluación

Para evaluar el aprovechamiento en la asignatura se realiza un examen. El examen consta de una serie de cuestiones de respuesta corta y con él que se podrá obtener una calificación máxima de notable 7. Al ser ésta una asignatura optativa, en la que se introducen conceptos nuevos, algunos de ellos sin mucha profundidad, se considera que para obtener una calificación mayor, se deben desarrollar distintas actividades que permitan profundizar más en los temas del curso o incluso de la materia, aunque no se hayan cubierto por el temario.

Transcurridas las cinco primeras semanas del cuatrimestre en que se imparte la asignatura, el alumnado podrá escoger entre uno de los tres itinerarios siguientes. El primer itinerario constará solamente de la actividad (a), el segundo itinerario constará de las actividades (a) y (b), y el tercer itinerario constará de las actividades (a), (b) y (c).

(a) Realizar el examen. La nota de la asignatura vendrá dada por la siguiente tabla:

Examen	5	6	7	8	9	10
Nota	4,5	5	5,5	6	6,5	7

(b) Desarrollar un tema por escrito, entre varios propuestos por el profesorado. La extensión del trabajo será de unas diez páginas y tendrá un valor máximo de 1,5 puntos, por lo que puede subir la nota hasta sobresaliente 8,5.

(c) Realizar una presentación en clase de un tema, entre los temas propuestos por el profesorado. Este podrá ser un trabajo monográfico, una reseña sobre un artículo o bien una experiencia, y se llevará a cabo por un grupo de cuatro personas. La documentación para desarrollar el trabajo la facilitará el profesorado. Una vez presentado el tema en clase, habrá un debate con preguntas del alumnado y el profesor. Todos los componentes del grupo deberán intervenir en la presentación, que tendrá una duración de 20 minutos (más 10 minutos de debate). Cada componente del grupo podrá tener una nota diferente. Mediante la realización de este trabajo en grupo se podrá subir la nota hasta sobresaliente 10.

Los temas para el desarrollo de trabajos o exposiciones de clase pueden ser, además de los temas que aparecen en el temario, otros temas como las bases de datos móviles, los sistemas de información geográfica, las bases de datos biológicas (gestión de los datos del genoma humano), las bibliotecas digitales o cualquier otro tema propuesto por el alumnado.

Para la evaluación de las prácticas, se exige la asistencia al 70% de las sesiones prácticas y se controla el aprovechamiento durante su realización.

Bibliografía

[1] ACT-NET Consortium (1996). *The Active Database Management System Manifesto: a Rulebase of ADBMS Features.* SIGMOD Record, Vol. 25, N. 3.
<http://www.acm.org/sigmod/record/issues/9609/adbms.ps> [Consulta: 30 de junio de 2000].

[2] S. Abiteboul, P. Bunemann, D. Suciu (1999). *Data on the Web: From Relations to Semistructured Data and XML.* Morgan Kaufmann Publishers.

[3] J. Aliaga, S. García, M. Marqués, G. Quintana (2000). *Apuntes de prácticas de ficheros y bases de datos.* Publicacions de la Universitat Jaume I. Col-lecció Materials, N.21.

[4] American National Standards Institute (1975). *ANSI/X3/SPARC Study Group on Data Base Management Systems.* Interim Report, FDT. ACM SIGMOD Bulletin Vol. 7, N. 2.

[5] American National Standards Institute (1986). *Database Language – SQL (X3.135).* ANSI, Technical Committee X3H2.

[6] M. Atkinson, F. Bancilhon, D. J. DeWitt, K. R. Dittrich, D. Maier, S. B. Zdonik (1990). *The Object–Oriented Database System Manifesto.* SIGMOD Record, Vol. 19, N. 3.

[7] P. Atzeni, S. Ceri, S. Paraboschi, R. Torlone (1999). *Database Systems. Concepts, Languages and Architectures.* McGraw Hill.

[8] C. Bachman (1973). *The programmer as a navigator.* Comunications of the ACM, Vol. 16, N. 11. ACM Turing Award Lecture.

[9] C. Batini, S. Ceri, B. Navathe (1994). *Diseño Conceptual de Bases de Datos. Un Enfoque de Entidades–Interrelaciones.* Addison–Wesley/Díaz de Santos. Traducción de *Conceptual Database Design: An Entity–Relationship Approach (1992).*

[10] R. Bayer, E. McCreight (1972). *Organization and Maintenance of Large Ordered Indexes.* Acta Informatica, Vol. 1, N. 3.

[11] P. Bernstein, M. Brodie, S. Ceri, D. DeWitt, M. Franklin, H. Garcia-Molina, J. Gray, J. Held, J. Hellerstein, H. Jagadish, M. Lesk, D. Maier, J.Ñaughton, H. Pirahesh, M. Stonebraker, J. Ullman (1998). *The Asilomar Report on Database Research.* SIGMOD Record, Vol. 27, N. 4.
<http://www.acm.org/sigmod/record/issues/9812/asilomar.html> [Consulta: 30 de junio de 2000].

[12] E. Bertino, L. Martino (1995). *Sistemas de bases de datos orientadas a objetos. Conceptos y arquitecturas.* Addison–Wesley/Díaz de Santos. Traducción de *Object–Oriented Database Systems: Concepts and Architectures (1993).*

[13] P. Blesa, N. Brisaboa, V. Canivell, J. Garbajosa, J. Maudes, M. Piattini (1999). *Propuesta de contenidos de bases de datos de los planes de estudio de Informática.* NOVÁTICA, N. 137.

[14] R. Camps (1999). *¿Qué Informática se enseña en la Universidad? (Primera Parte).* NOVÁTICA, N. 141.

[15] R. Camps (1999). *¿Qué Informática se enseña en la Universidad? (Segunda Parte).* NOVÁTICA, N. 142.

[16] R. Cattell, D. Barry (1997). *The Object Database Standard: ODMG 2.0.* Mogran Kaufmann Publishers.

[17] M. Celma (1998). *Apuntes de bases de datos avanzadas.* Facultad de Informática. Universidad Politécnica de Valencia.
<http://www.dsic.upv.es/asignaturas/facultad/bdv/> [Consulta: 30 de junio de 2000].

[18] S. Ceri, G. Pelagatti (1984). *Distributed Databases. Principles and Systems.* McGraw–Hill.

[19] S. Ceri, P. Fraternali (1997). *Designing Database Applications with Objects and Rules. The IDEA Methodology.* Addison–Wesley.

[20] S. Chaudhuri, E. Dayal (1997). *An Overview of Data Warehousing and OLAP Technology.* SIGMOD Record, Vol. 26, N. 1. <ftp://ftp.research.microsoft.com/users/surajitc/sigrecord.pdf> [Consulta: 30 de junio de 2000].

[21] CODASYL Systems Committee (1969). *A Survey of Generalized Data Base Management Systems.* CODASYL ACM.

[22] CODASYL Systems Committee (1971). *Feature Analisys of Generalized Data Base Management Systems.* CODASYL ACM.

[23] CODASYL Systems Committee (1978). *Data Description Language Committee. Journal of Development.* CODASYL ACM.

[24] E. F. Codd (1970). *A Relational Model of Data for Large Shared Data Banks.* Communications of ACM, Vol. 13, N. 6.

[25] E. F. Codd (1971). *A Data Base Sublanguage Founded on the Relational Calculus.* Proceedings of the ACM SIGFIDET Conference on Data Description, Acces and Control.

[26] E. F. Codd (1972). *Further Normalization of the Data Base Relational Model.* Data Base Systems, Prentice–Hall.

[27] E. F. Codd (1972). *Extending the Data Base Relational Model to Capture more Meaning.* ACM Transactions on Database Systems, Vol. 4, N. 4.

[28] E. F. Codd (1982). *Relational Database: A Practical Foundation for Productivity.* Comunications of the ACM, Vol. 25, N. 2. ACM Turing Award Lecture.

[29] E. F. Codd (1985). *Is Your DBMS Really Relational?* Computerworld, N. 14.

[30] E. F. Codd (1985). *Does Your DBMS Run by the Rules?* Computerworld, N. 21.

[31] E. F. Codd (1986). *Missing Information (applicalble and inapplicable) in Relational Databases.* SIGMOD Record, Vol. 15, N. 4.

[32] E. F. Codd (1987). *More Commentary on Missing Information in Relational Databases.* SIGMOD Record, Vol. 16, N. 1.

[33] E. F. Codd (1988). *Domains, Keys and Referential Integrity in Relational Databases.* InfoDB, Vol. 3, N. 1.

[34] E. F. Codd, S. B. Codd, C. T. Salley. *Providing OLAP to User Analyst: An IT Mandate.*
<http://www.arborsoft.com/OLAP.html> [Consulta: 30 de junio de 2000].

[35] Committee for Advanced DBMS Function (1990). *Third-Generation Database System Manifesto.* SIGMOD Record, Vol. 19, N. 3.

[36] T. Connolly, C. Begg, A. Strachan (1998). *Database Systems. A Practical Approach to Design, Implementation and Management.* Segunda edición. Addison–Wesley.

[37] CORBA. Common Object Request Broker Architecture.
<http://www.omg.org/corba/> [Consulta: 30 de junio de 2000].

[38] P. Corrigan, M. Gurry (1996). *Oracle Performance Tuning.* O'Reilly & Associates.

[39] H. Darwen, C. J. Date (1995). *The Third Manifesto.* SIGMOD Record, Vol. 24, N. 1.
<http://www.acm.org/sigmod/record/issues/9503/manifesto.ps> [Consulta: 30 de junio de 2000].

[40] C. J. Date (1997). *Introducción a los sistemas de bases de datos.* Sexta edición. Addison–Wesley Iberoamericana. Traducción de *An Introduction to Database Systems, Sixth edition (1995).*

[41] C. J. Date, H. Darwen (1997). *A Guide to the SQL Standard.* Addison–Wesley.

[42] C. J. Date, H. Darwen (1998). *Foundation for Object/Relational Databases. The Third Manifesto.* Addison–Wesley.

[43] C. J. Date (2000). *An Introduction to Database Systems.* Séptima edición. Addison–Wesley.

[44] G. Davis, J. Gorgone, J. Couger, D. Feinstein, H. Longenecker (1997). *IS'97 Model Curriculum and Guidelines for Undergraduate Degree Programs in Information Systems.*
<http://webfoot.csom.umn.edu/faculty/gdavis/curcomre.pdf> [Consulta: 30 de junio de 2000].
<http://www.acm.org/about_acm/97rep/edu.html> [Consulta: 30 de junio de 2000].

[45] A. de Miguel, M. Piattini (1999). *Fundamentos y modelos de bases de datos.* Segunda edición. Ra–ma.

[46] A. de Miguel, M. Piattini, E. Marcos (1999). *Diseño de bases de datos relacionales.* Ra–ma.

[47] P. Dorsey (1997). *Oracle Designer/2000 Handbook.* Osborne McGraw–Hill.

[48] A. Eisenberg, J. Melton (1999). *SQL:1999, formerly known as SQL3.* SIGMOD Record, Vol. 28, N. 1.
<http://www.acm.org/sigmod/record/issues/9903/standards.ps.gz>
[Consulta: 30 de junio de 2000].

[49] R. Elmasri, S. B. Navathe (1997). *Sistemas de bases de datos. Conceptos fundamentales.* Segunda edición. Addison–Wesley Iberoamericana. Traducción de *Fundamentals of Database Systems, Second edition (1994).*

[50] R. Elmasri, S. B. Navathe (1999). *Fundamentals of Database Systems.* Tercera edición. Addison–Wesley.

[51] U. Fayyad, G. Piatetsky-Shapiro, P. Smyth, R. Uthurusamy (1996). *Advances in Knowledge Discovery and Data Mining.* AAAI Press/ MIT Press.

[52] M. J. Folk, B. Zoellick (1992). *File Structures.* Segunda edición. Addison–Wesley.

[53] M. J. Folk, B. Zoellick, G. Riccardi (1998). *File Structures. An Object-Oriented Approach with C++.* Tercera edición. Addison–Wesley.

[54] H. García-Molina, J. Widom, J. D. Ullman (2000). *Database System Implementation.* Prentice–Hall.

[55] S. Gatziu, M. A. Jeusfeld, M. Staudt, Y. Vassiliou (1999)(editores). *Design and Management of Data Warehouses.* Proceedings of the International Workshop DMDW'99 at CAiSE*99. También en SIGMOD Record, Vol. 28, N. 4.
<http://www.acm.org/sigmod/record/issues/9912/dmdw.pdf> [Consulta: 30 de junio de 2000].

[56] J. L. Gordo, A. Gazo, J. Vega (1998). *Manual avanzado de Access 97.* Anaya Multimedia.

[57] G. W. Hansen, J. V. Hansen (1997). *Diseño y administración de bases de datos.* Segunda edición. Prentice–Hall. Traducción de *Database Management and Design, Second edition (1996).*

[58] P. Hernández (1998). *Diseñar y enseñar.* Segunda edición. Editorial Narcea.

[59] R. Kim (1995). *Modern Database Systems. The Object Model, Interoperability and Beyond.* Addison–Wesley.

[60] R. Kimball (1996). *The Data Warehouse Toolkit.* John Wiley.

[61] D. M. Kroneke (1996). *Procesamiento de bases de datos. Fundamentos, diseño e instrumentación.* Quinta edición. Prentice–Hall. Traducción de *Database Processing. Fundamentals, Design & Implementation, Fifth edition (1995).*

[62] D. M. Kroneke (1997). *Database Processing. Fundamentals, Design & Implementation.* Sexta edición. Prentice–Hall.

[63] D. M. Kroneke (2000). *Database Processing. Fundamentals, Design & Implementation.* Séptima edición. Prentice–Hall.

[64] Libro de Titulación de la Ingeniería Técnica en Informática de Gestión.
<http://estce.act.uji.es/llibres/1999-2000/F/> [Consulta: 30 de junio de 2000].

[65] I. Luján (1997). *El Proyecto Docente del Profesor Universitario.* Universidad de Las Palmas de Gran Canaria.

[66] C. Marcelo, A. Estebaranz, C. Mayor, P. Mingorance, A. Parrilla, M. Moreno (1997). *El Proyecto Docente.* Materiales para la Enseñanza Universitaria, Nº2. Universidad de Extremadura. Instituto de Ciencias de la Educación.

[67] M. Marqués (2000). *Apuntes de ficheros y bases de datos.* Publicacions de la Universitat Jaume I. Col-lecció Materials, N.20.

[68] J. Melton, N. Mattos (1996). *An Overview of SQL3—The Emerging New Generation of the SQL Standard.* Tutorial número T5, Proceedings of the International Conference on Very Large Data Bases, Bombay.

[69] R. J. Muller (1996). *Oracle Developer/2000 Handbook.* Osborne McGraw–Hill.

[70] T. M. O'Brien, S. J. Pogge, G. E. White (1997). *Microsoft Access 97 desarrollo de soluciones.* McGraw–Hill.

[71] Object Data Management Group (ODMG). <http://www.odmg.org/> [Consulta: 30 de junio de 2000].

[72] On-Line Analytical Processing (OLAP) Council. <http://www.olapcouncil.org> [Consulta: 30 de junio de 2000].

[73] M. T. Özsu, P. Valduriez (1999). *Principles of Distributed Database Systems.* Segunda edición. Prentice–Hall.

[74] A. Silberschatz, M. Stonebraker, J. Ullman (1996). *Database Research: Achievements and Oportunities Into the 21st Century.* SIGMOD Record, Vol. 25, N. 1. <http://www.acm.org/sigmod/record/issues/9603/lagunita.ps> [Consulta: 30 de junio de 2000].

[75] A. Silberschatz, S. Zdonik (1996). *Database Systems — Breaking Out of the Box.* ACM Computing Surveys, Vol. 28, N. 4. Republicado en SIGMOD Record, Vol. 25, N. 3 (1997). <http://www.acm.org/sigmod/record/issues/9709/stratdir.ps> [Consulta: 30 de junio de 2000].

[76] A. Silberschatz, H. Korth, S. Sudarshan (1998). *Fundamentos de bases de datos.* Tercera edición. McGraw–Hill. Traducción de *Database System Concepts, Third edition* (1997).

[77] R. T. Snodgrass (1995) (editor). *The TSQL2 Temporal Query Language.* Kluwer.

[78] M. Stonebraker (1988). *Readings in Database Systems.* Morgan Kaufman.

[79] M. Stonebraker, R. Agrawal, U. Dayal, E. J. Neuhold, A. Reuter (1993). *DBMS Research at a Crossroad: The Viena Update.* Proceedings of the International Conference on Very Large Databases.

[80] M. Stonebraker (1994). *Readings in Database Systems*, Segunda edición. Morgan Kaufman.

[81] V. S. Subrahmanian, S. Jajodia (1996) (editores). *Multimedia Database System: Issues and Research Direction.* Spinger.

[82] R. Sunderraman (1998). *Programming with Oracle SQL.* Addison–Wesley.

[83] A. U. Tansel, J. Clifford, S. K. Gadia, S. Jajodia, A. Segev, R. T. Snodgrass (1993) (editores). *Temporal Databases: Theory, Design, and Implementation.* Benjamin/Cummings.

[84] J. Ullman (1988). *Principles of database and knowledge-base systems*, Volumen I. Computer Science Press.

[85] J. Ullman (1989). *Principles of database and knowledge-base systems*, Volumen II. Computer Science Press.

[86] J. Ullman, J. Widom (1999). *Introducción a los sistemas de bases de datos.* Prentice–Hall. Traducción de *A First Course in Database Systems (1997).*

[87] Unified Modeling Language (UML).
<http://www.rational.com/uml/> [Consulta: 30 de junio de 2000].

[88] S. Urman (1998). *Oracle 8 Programación PL/SQL.* Osborne McGraw–Hill.

[89] M. Valero, A. Lázaro, J. Alberto (1997). *Evaluación de las Titulaciones de Ingeniería en Informática de Gestión e Ingeniería Informática de la Universidad Jaume I de Castelló.* Universitat Jaume I.

[90] J. L. VIESCAS (1997). *Guía completa de Microsoft Access 97.* McGraw–Hill.

[91] J. WIDOM (1994). *Research Issues in Active Database Systems: Report from the Closing Panel at RIDE-ADS'94.* SIGMOD Record, Vol. 23, N. 3.

[92] J. WIDOM, S. CERI (1996). *Active Database Systems.* Morgan Kaufmann Publishers.

[93] C. ZANIOLO, S. CERI, T. S. FALOUTSOS, V. S. SUBRAHMANIAN, R. ZICARI (1997). *Advanced Database Systems.* Morgan Kaufmann Publishers.