La Disciplina de los Sistemas de Bases de Datos. Historia, Situación Actual y Perspectivas.

José Hernández Orallo
(jorallo@dsic.upv.es)
Dep. de Sistemes Informàtics i Computació
Universitat Politècnica de València

mayo 2002

Aunque la materia Bases de Datos tiene un carácter propedéutico para la disciplina de los sistemas de bases de datos y el área más general de sistemas de información, es necesario conocer cuál ha sido la evolución y estado actual de la tecnología de bases de datos, con el objetivo de estar preparados para los cambios que, inevitablemente, se van a dar en el área de las bases de datos y los sistemas de información.

Para ello, en este informe se relata brevemente la evolución de los sistemas de bases de datos, centrándose en los fundamentos de la tecnología actual y su motivación. Haremos un repaso de las nociones y evolución básicas de los modelos pre-relacionales, relacional, objetual y objeto-relacional, las bases de datos paralelas y distribuidas, multimedia, los almacenes de datos, la relación entre las bases de datos y la web, así como otras áreas y aplicaciones. Esto nos lleva a evaluar la situación actual, especialmente las nuevas demandas sobre sistemas de información exigidas por el aumento de interconectividad, los nuevos imperativos de publicación e intercambio de información, los datos semiestructurados y el estándar XML, así como el análisis de datos para la toma de decisión y los avances y perspectivas en las "bases de conocimiento". Se comentan también las líneas de investigación abiertas más importantes en el área y una opinión personal sobre hacia donde parece dirigirse la disciplina. Finalmente, se estudia sucintamente la sociología de la disciplina, su interrelación con otras disciplinas del área de Lenguajes y Sistemas Informáticos y las organizaciones, congresos y publicaciones más importantes.

1.1. La Evolución de los Sistemas de Bases de Datos

Los sistemas de información existen desde las primeras civilizaciones. El concepto más esencial de sistema de información no ha variado desde los censos romanos, por poner un ejemplo. Los datos se recopilaban, se estructuraban, se centralizaban y se almacenaban convenientemente. El objetivo inmediato de este proceso era poder recuperar estos mismos datos u otros datos derivados de ellos en cualquier momento, sin necesidad de volverlos a recopilar, paso que solía ser el más costoso o incluso irrepetible. El objetivo ulterior de un sistema de información, no obstante, era proporcionar a los usuarios información fidedigna sobre el dominio que representaban, con el objetivo de tomar decisiones y realizar acciones más pertinentes que las que se realizarían sin dicha información.

Llamamos base de datos justamente a esta colección de datos recopilados y estructurados que existe durante un periodo de tiempo. Por ejemplo, un libro contable, debido a su estructura, se puede considerar una base de datos. Una novela, por el contrario, no tiene casi estructura, y no se suele considerar una base de datos. Generalmente, un sistema de información consta de una o más bases de datos, junto con los medios para almacenarlas y gestionarlas, sus usuarios y sus administradores.

Hoy en día, sin embargo, solemos asociar las bases de datos con los ordenadores, y su gestión no suele ser manual, sino altamente automatizada. Más concretamente, la tecnología actual insta a la delegación de la gestión de una base de datos a unos tipos de aplicaciones software específicas denominadas sistemas de gestión de bases de datos (SGBD) o, simplemente, sistemas de bases de datos. Por esta razón, hablar de la tecnología de bases de datos es prácticamente lo mismo que hablar de la tecnología de los sistemas de gestión de bases de datos.

Las funciones básicas de un sistema de gestión de base de datos son [Ullman & Widom 1997]:

1. Permitir a los usuarios crear nuevas bases de datos y especificar su estructura, utilizando un lenguaje o interfaz especializado, llamado lenguaje o interfaz de definición de datos.
2. Dar a los usuarios la posibilidad de consultar los datos (es decir, recuperarlos parcial o totalmente) y modificarlos, utilizando un lenguaje o interfaz apropiado, generalmente llamado lenguaje de consulta o lenguaje de manipulación de datos.
3. Permitir el almacenamiento de grandes cantidades de datos durante un largo periodo de tiempo, manteniéndolos seguros de accidentes o uso no autorizado y permitiendo un acceso eficiente a los datos para consultas y modificaciones.
4. Controlar el acceso a los datos de muchos usuarios a la vez, impidiendo que las acciones de un usuario puedan afectar a las acciones de otro sobre datos diferentes y que el acceso simultáneo no corrompa los datos.

1.1.1. Primeros Sistemas de Base de Datos

Los primeros sistemas de bases de datos aparecieron a finales de los cincuenta. En este periodo, muchas compañías se fueron dando cuenta de que los primeros sistemas informáticos brindaban la posibilidad de aplicar soluciones mecánicas más baratas y eficientes. Los primeros sistemas evolucionaron de los sistemas de ficheros que proporcionaban la función (3) comentada anteriormente: los sistemas de ficheros almacenan datos durante un largo periodo de tiempo y permiten el almacenamiento de grandes cantidades de datos. Sin embargo, los sistemas de ficheros no garantizaban generalmente que los datos no se perdían ante fallos bastante triviales, y se basaban casi exclusivamente en recuperación por copia de seguridad.

Además, los sistemas de ficheros proporcionaban de una manera limitada la función (2), es decir, un lenguaje de consultas para los datos en los ficheros. El soporte de estos sistemas para la función (1) —un esquema para los datos— también era limitada y de muy bajo nivel. Finalmente, los sistemas de ficheros no satisfacen la función (4). Cuando permiten acceso concurrente a ficheros por parte de varios usuarios o procesos, un sistema de ficheros no previene generalmente las situaciones en la que dos usuarios modifican el mismo fichero al mismo tiempo, con lo que los cambios realizados por uno de ellos no llegan aparecer definitivamente en el fichero.

Las primeras aplicaciones importantes de los sistemas de ficheros fueron aquellas en la que los datos estaban compuestos de partes bien diferenciadas y la interrelación entre ellas era reducida. Algunos ejemplos de estas aplicaciones eran los sistemas de reserva (p.ej. reserva e información de vuelos), los sistemas bancarios donde se almacenaban las operaciones secuencialmente y luego se procesaban, y los primeros sistemas de organización corporativos (ventas, facturación, nóminas, etc.).

Los primeros verdaderos SGBDs, evolucionados de los sistemas de ficheros, obligaban a que el usuario visualizara los datos de manera muy parecida a como se almacenaban. Los primeros sistemas de ficheros habían logrado pasar del código máquina a un lenguaje ensamblador con ciertas instrucciones de acceso a disco, nociones que se pueden ver en sistemas todavía en funcionamiento hoy en día, tales como la línea AS de IBM. No es de extrañar que con este nivel de abstracción la manera de recuperar los datos estaba estrechamente ligada al lenguaje de programación utilizado. Un avance importante lo constituyó el comité formado en la COnference on DAta SYstems and Languages, CODASYL, en 1960 estableciendo el COmmon Business-Oriented Language (COBOL) como un lenguaje estándar para interrelacionar con datos almacenados en ficheros. Aunque hoy en día puede parecer un lenguaje “muy físico”, en aquella época representó lo que se vinieron a llamar los lenguajes de programación de tercera generación. Las instrucciones específicas de un programa Cobol para tratamiento de ficheros eran las de abrir un fichero, leer un fichero y añadir un registro a un fichero. Lo típico en gestión de datos en esta época era un fichero ‘batch’ de transacciones que se aplicaba a un maestro viejo en cinta, produciendo como resultado un nuevo maestro también en cinta y la impresión para el siguiente día de trabajo. Pero pronto los discos magnéticos empezaron a sustituir a las cintas magnéticas, lo que supuso una reconcepción del almacenamiento, al pasarse **del acceso secuencial al acceso aleatorio** (este paso es el que se conoce como el paso de los sistemas de bases de datos de primera a segunda generación).

Durante los sesenta empezaron a aparecer distintos modelos de datos para describir la estructura de la información en una base de datos, con el objetivo de conseguir una independencia un poco mayor entre las aplicaciones y la organización física de los datos. Esto se consiguió inicialmente mediante la abstracción entre varios (sub)esquemas externos para las aplicaciones frente a la organización física de los mismos. Esta separación en dos niveles fue propuesta por el grupo Data Base Task Group (DBTG) del comité CODASYL.

Los modelos más popularizados fueron el modelo jerárquico o basado en árboles, y el modelo en red o basado en grafos. Los SGBD acordes con estos modelos se conocieron como SGBD de tercera generación. Pasemos a comentar brevemente estos dos modelos:

- El modelo jerárquico no tiene una historia demasiado bien documentada. Se deriva de los sistemas de gestión de información de los cincuenta y los sesenta. En 1968, IBM introdujo el sistema IMS, derivado del programa Apollo de la NASA sobre sus System/360, basado en el modelo jerárquico. Este modelo fue adoptado por muchos bancos y compañías de seguros que todavía los utilizan en algún caso hoy en día. Los sistemas de base de datos jerárquicos todavía se pueden encontrar en algunos departamentos de instituciones públicas y hospitales para gestionar el inventario y la contabilidad, aunque la renovación provocada por el efecto 2000 ha eliminado prácticamente su uso, así como el reciclaje de los expertos en estos sistemas a otros más modernos. El modelo jerárquico se basa en almacenar los datos en una serie de registros, los cuales tienen campos asociados a ellos. Para crear enlaces entre los tipos de registros, el modelo jerárquico utiliza las relaciones padre-hijo, correspondencias 1:N entre tipos de registro. Esto se realiza mediante el uso de árboles. A diferencia de otros modelos, como el modelo en red que veremos a continuación, el modelo jerárquico representa precisamente eso, todas las relaciones están jerarquizadas en un árbol, por lo que no es capaz de establecer enlaces entre hijos o entre capas, si no es padre-hijo. La ventaja del modelo jerárquico es su gran estructuración, que en aquel tiempo se veía como una gran ventaja para mejorar el rendimiento de las transacciones (inserción, modificación y borrado de registros), así como para simplificar la interfaz para los usuarios.
- El modelo en red se estandarizó a finales de los sesenta mediante un informe de CODASYL (COnference on DAta SYstems and Languages) Data Base Task Group (DBTG) [CODASYL 1968]. Por eso, a veces se le conoce como el modelo DBTG o el modelo CODASYL. El primer informe de CODASYL ya incluía la sintaxis y semántica de un lenguaje de definición de datos (DDL, Data Definition Language) y de un lenguaje de manipulación de datos (DML, Data Manipulation Language). Siguiendo muchos comentarios y revisiones de expertos y usuarios se realizó un nuevo informe [CODASYL 1971], en el que ya se incluía la posibilidad de definir vistas para los distintos grupos de usuarios. El término base de datos en red no se refería (al contrario de lo que se entendería hoy en día) a que la base de datos estuviera almacenada en una red de ordenadores, sino por la manera en la que los datos se enlazaban con otros datos. Se llama, por tanto, modelo en red porque representa los datos que contiene en la forma de una red de registros y conjuntos (en realidad listas circulares llamadas *sets*) que se relacionan entre sí, formando una red de enlaces. Para hacer esto utiliza registros, tipos de registro y tipos de conjunto. El modelo en red tampoco se utiliza casi hoy en día y si subsiste es como consecuencia del mantenimiento de un sistema todavía no reconvertido o no portado a modelos y SGBD más modernos. Aunque este modelo permite más flexibilidad que el modelo jerárquico, y en algunos casos se adapta muy bien a algunos tipos de transacciones, se considera superado por otros modelos, como el relacional, o subsumido en parte por modelos más modernos, como el objetual.

En resumen, los modelos jerárquico y red, con el paso de los años, se pueden considerar como modelos puente hacia el modelo relacional, ya que se incorporan en los primeros sistemas de gestión de bases de datos que introducen un mayor nivel de independencia, respecto a la estructura interna, pero siguen teniendo una estructura de cierto bajo nivel y de compleja manipulación.

Otro problema con estos modelos y sistemas iniciales era que no iban acompañados de lenguajes de consulta de alto nivel. Por ejemplo, el lenguaje de consulta CODASYL tenía sentencias que permitían al usuario saltar de un elemento de datos a otro a través de grafos de punteros entre estos elementos. Se requería un gran esfuerzo para escribir estos programas, incluso para consultas muy sencillas.

Antes de pasar a ver los sistemas de base de datos relacionales, hay que destacar el nacimiento y definición del concepto de transacción y sus propiedades asociadas, lo que se conoce como el “ACID test”. Aunque el concepto de transacción evoluciona en las primeras décadas del desarrollo de las bases de datos, se considera el trabajo de James Gray [Gray 1981] como el que le da su forma actual. Se dice que un SGBD cumple el “ACID test” si observa las propiedades de (A)tomicidad, (C)onsistencia, a(I)slamiento y (D)urabilidad. En concreto:

- Atomicidad: los resultados de una transacción o bien pasan a ser completados todos (commit) o bien pasan a ser todos deshechos (rollback). Es decir, o todos los cambios incluidos en una transacción tienen efecto o no lo tiene ninguno.

- Consistencia: las bases de datos se transforman de estados íntegros a estados íntegros, es decir, entre estados válidos. Una transacción sólo se puede completar si el estado final es íntegro.
- Aislamiento: los resultados de una transacción son invisibles para el resto de transacciones de otros procesos hasta que la transacción se ha completado.
- Durabilidad o permanencia: una vez una transacción ha sido completada, los resultados (cambios) de la transacción se hacen permanentes, incluso frente a fallos del sistema y de medios de almacenamiento.

Sólo si estas propiedades de las transacciones se cumplen, podemos considerar que un SGBD cumple las propiedades 3 y 4 comentadas al principio.

1.1.2. Sistemas de Base de Datos Relacionales

Al menos un investigador de IBM no estaba satisfecho ni con los productos Codasyl, ni con los sistemas jerárquicos de la propia IBM. Edgar F. (Ted) Codd, un matemático formado en Oxford, que había entrado en IBM en 1949, empezó a trabajar en una serie de informes técnicos acerca de una manera 'nueva' de organizar y acceder a los datos. A partir de estos trabajos publicó el artículo "A Relational Model of Data for Large Shared Data Banks" en 1970 [Codd 1970]. Codd propuso que los sistemas de bases de datos deberían presentarse a los usuarios con una visión de los datos organizados en estructuras llamadas *relaciones*. Las relaciones se definen como conjuntos de tuplas, no como series o secuencias de objetos, con lo que el orden no es importante. Por tanto, detrás de una relación puede haber cualquier estructura de datos compleja que permita una respuesta rápida a una variedad de consultas. Además, aunque no es un aspecto intrínseco del modelo relacional, según la propuesta de Codd, el usuario de un sistema relacional no tenía que preocuparse de la estructura de almacenamiento, sólo debía preocuparse por el qué consultar y no el cómo. Hoy en día es válido todavía el resumen de su artículo [Codd 1970], especialmente la siguiente parte:

> "Future users of large data banks must be protected from having to know how the data is organized in the machine (the internal representation). [...] Activities of users at terminals and most application programs should remain unaffected when the internal representation of data is changed and even when some aspects of the external representation are changed. Changes in data representation will often be needed as a result of changes in query, update, and report traffic and natural growth in the types of stored information."[1]

Además, las consultas podían expresarse en un lenguaje de muy alto nivel, lo que incrementaba en gran medida la eficiencia de los programadores sobre bases de datos. En resumen, Codd concibió un sistema donde el usuario sería capaz de acceder a la información con comandos parecidos al lenguaje natural y donde la información estuviera almacenada en 'tablas'.

Pese a sus virtudes, la aceptación del modelo relacional no fue inmediata, debido en parte a la naturaleza técnica del artículo y a su base matemática que, aunque muy simple, no era común para la industria de bases de datos de la época. Además, se dudaba de la eficiencia del modelo. Más aún, dentro de IBM, la reticencia fue muy grande, ya que IBM había invertido una gran cantidad de esfuerzo y dinero en el producto ya existente IMS y líder del mercado. La nueva tecnología relacional debía demostrar que era mucho mejor que la existente para cambiar la situación. De hecho, Codd publicó el artículo en una revista de ámbito científico y abierto porque nadie en IBM (ni siquiera él mismo) reconoció en su momento el impacto que tendría después. Todos se sorprendieron pronto de que la respuesta externa al artículo fuera muy positiva. Incluso se acogió la idea con todo su potencial *comercial*. La reacción inicial de IBM fue tajante: declaró IMS su producto estratégico con exclusividad y consideró el modelo relacional como contrario a sus intereses. Codd, por su parte, no cedió en su defensa pública de las ventajas de su propuesta e incluso mantuvo un debate público con Charles Bachman, el mayor defensor del estándar Codasyl, ignorando del debate al modelo jerárquico, en el cual se basaba el IMS. Esto dejaba al modelo jerárquico del IMS en una situación incómoda. Afortunadamente, y en parte debido a esta publicidad, desde fuera, la Universidad de California en Berkeley creyó en la idea del modelo relacional y obtuvo financiación militar y de la NSF (National Science Foundation) para desarrollar un sistema relacional, el Ingres.

[1] "Los usuarios futuros de grandes bancos de datos deben ser protegidos de tener que saber cómo están organizados los datos en la máquina (la representación interna). [...] Las actividades de los usarios en sus terminales y la mayoría de programas de aplicación no se deberían ver afectados cuando se cambia la representación interna de los datos o incluso cuando se cambian algunos de los aspectos de la representación externa. Se necesitará cambiar la representación de los datos a menudo como resultado de los cambios en el tráfico de las consultas, actualizaciones e informes y como consecuencia del crecimiento natural en los tipos de información almacenada."

La contra-reacción de IBM fue inmediata, puso en marcha el desarrollo de otro sistema relacional, el "System R"[2].

El grupo de investigación de IBM a cargo del proyecto esperaba crear un sistema de bases de datos relacional que pudiera convertirse eventualmente en un producto. Su primer prototipo, elaborado entre 1974 y 1975, se utilizó experimentalmente por diversas organizaciones, como p.ej. la "MIT Sloan School of Management" [Astrahan et al. 1976]. No obstante, este primer prototipo se abandonó en pro del rediseño de "System R" como un sistema multiusuario y con completa funcionalidad, con un lenguaje de consulta estructurado, el SEQUEL [Boyce & Chamberlin 1974], que luego pasaría a llamarse SQL (Structured Query Language).

Sin embargo, el primer SGBD Relacional (SGBDR) completamente funcional, Ingres, se desarrolló en la Universidad de California en Berkeley por un grupo liderado por Michael Stonebraker y Eugene Wong. Alrededor de 1974, consiguieron la primera versión completamente funcional del mismo Ingres [Stonebraker et al. 1976], aunque el producto se revisó continuamente durante toda la década con los comentarios y realimentación de muchos usuarios en otras Universidades y centros que lo adaptaban a sus cada día más baratas máquinas DEC (el código fuente del Ingres era inicialmente público) [Stonebraker 1980]. Ingres incluía su propio lenguaje de consultas, QUEL, que era similar en algunos aspectos al SQL. A finales de los setenta se comercializó por Relational Technology, Inc. Más tarde, Ingres se convirtió en un SGBDR comercial distribuido por Ingres, Inc., una subsidiaria de ASK, Inc., y, actualmente, lo distribuye Computer Associates. Pero no fue éste (ni el de IBM) el primer producto relacional comercial. Este hito le corresponde a Honeywell Information Systems Inc., que sacó su primer producto comercial relacional en junio de 1976. Se basaba en los mismos principios que el sistema de IBM, pero se diseñó de manera separada al trabajo de IBM.

Durante estos desarrollos se produjo la publicación de la separación en tres niveles de los SGBD descrita por el informe del comité ANSI/SPARC de 1975 [ANSI/SPARC 1975] [Fry & Sibley 1976]: externo, conceptual e interno. Ésta se hizo independientemente de la propuesta anterior de dos niveles (externo e interno) del CODAYSL/DBTG. Posteriormente, el CODASYL reformuló su propuesta a partir de la del comité ANSI/SPARC, presentando la arquitectura de tres niveles más popular: subesquemas externos, esquema lógico y esquema físico (también llamado almacenado o interno). Este comité propuso un lenguaje para definir este último esquema, el DSDL (Data Storage Definition Language). También comenzaba a perfilarse el uso diferenciado del valor nulo y la extensión del álgebra relacional a una lógica trivaluada.

Independientemente de la lentitud e indecisión para lanzar sus productos al mercado, el esfuerzo de desarrollo mayor se realizó en IBM en el San José Research Center (hoy llamado Almaden Research Center). Algunos de sus grandes méritos, como el SQL, tardó en ser reconocido por la compañía. De nuevo, fue la presión de otra compañía, en este caso Oracle, creada por Larry Ellison, la que, al desarrollar y vender un producto compatible con SQL, hizo reaccionar a IBM. Es importante destacar que Ellison había conocido el SQL a partir de las publicaciones del System R. Del mismo modo, la amenaza de otros productos, como los desarrollados por Software AG, motivaron a IBM a continuar investigando en la línea de System R. Esta investigación condujo al anuncio por parte de IBM de dos sistemas de gestión de bases de datos relacionales (SGBDR) en los ochenta: en 1981 se introdujo SQL/DS para los entornos DOS/VSE (disk operating system/virtual storage extended) y VM/CMS (virtual machine/conversational monitoring system); en 1983 se introdujo DB2 para el sistema operativo MVDS. En el desarrollo de estos productos, IBM introdujo ideas pioneras en la optimización de consultas, en la independencia de datos del esquema externo (vistas), en las transacciones (ficheros log y bloqueos) y en la seguridad (el modelo grant-revoke).

Otros SGBDR comerciales muy populares de esta época son, Oracle de Oracle Inc., como hemos visto; Sybase de Sybase Inc.; RDB de Digital Equipment Corp, ahora en propiedad de Compaq; Informix de Informix Inc.; y Unify de Unify Inc. Ésta es la época de los llamados SGBD de cuarta generación.

Aparte de los SGBDR mencionados anteriormente, muchas implementaciones del modelo relacional aparecieron en el ámbito de los ordenadores personales en los ochenta. Éstos son RIM, RBASE 5000, Paradox, OS/2 Database Manager, DBase IV, XDB, Watcom SQL, SQL Server de Sybase, Inc., SQL Server y Access de Microsoft, y MySQL. Inicialmente eran sistemas mono-usuario, pero muchos de ellos han comenzado a incorporar arquitecturas cliente/servidor e interconectividad con otras bases de datos.

Respecto a los lenguajes de consulta, Codd introdujo el álgebra relacional en [Codd 1970] y [Kuhns 1967] consideró anteriormente el uso de la lógica para realizar consultas, que llevaría a lenguajes lógicos como el Cálculo Relacional de Tuplas o Dominios. De hecho fue el propio Codd quien presentó el Cálculo Relacional como alternativa al álgebra relacional en [Codd 1971]. Con el tiempo se realizaron diversas generalizaciones de

[2] Una historia del System R se puede encontrar en [Chamberlin 1998] o en (http://www.mcjones.org/System_R/).

ambos lenguajes para incluir operaciones agregadas [Klug 1982] o cuantificadores del estilo “el número de” [Merrett 1978] [Badia et al. 1995].

Algunos resultados importantes teóricos son la demostración de la equivalencia expresiva entre el álgebra relacional y el cálculo relacional completo, demostrada por el mismo Codd [Codd 1972b], así como [Chandra & Merlin 1977] mostraron que determinar si dos consultas conjuntivas (es decir, que sólo contienen selecciones, proyecciones y productos cartesianos) son equivalentes es NP completo (sobre conjuntos de tuplas). La equivalencia entre consultas incluyendo unión se demostró decidible en [Sagiv & Yannakakis 1980]. La noción de optimización semántica de consultas se basa en transformaciones que preservan la equivalencia sólo cuando algunas restricciones de integridad se cumplen. Esta idea la introdujo [King 1981].

Sin embargo, en la práctica, aunque muy útiles para presentar el modelo relacional, o demostrar propiedades como las anteriores, ninguno de los lenguajes algebraicos o lógicos se llega a utilizar en los SGBDR comerciales. Es el SQL, una mezcla de ambos con sintaxis inspirada en el lenguaje natural (inglés), que se convierte en el lenguaje estándar de consultas. Como hemos dicho, la versión original de SQL se desarrolló como el lenguaje de consulta del proyecto System R de IBM [Boyce & Chamberlin 1974], [Chamberlin et al. 1976]. Gracias a sus virtudes (y a pesar de sus defectos [Date 1984]), el SQL comenzó una difusión y una estandarización (especialmente entre 1982 y 1986). El estándar SQL pasó de IBM a ANSI (American National Standard Institute) en 1982, formándose el comité X3H2. que junto a la ISO (International Standard Organization) publica el llamado SQL/ANS en 1986, siendo norma ISO en 1987, versión 1 del estándar. En 1989 se revisa en la versión 1 addendum. En 1992 aparece SQL2 o SQL92, la versión hoy en día más difundida [ISO/IEC 1992] [ANSI 1992] [ISO/IEC 1994]. Con la aparición de la segunda versión del estándar (SQL2) en 1992, prácticamente todos los SGBR, incluso los no relacionales, incluían soporte a SQL. Hoy en día, SQL se ha convertido en el lenguaje de consulta más utilizado. En 1999 apareció SQL3 [ANSI/ISO/IEC 1999] y, desde entonces, se trabaja sobre la estandarización del SQL/MM y se espera la versión SQL4 en los próximos años.

Existen opiniones, personificadas por C.J. Date, que el estándar ha cedido mucho a los SGBD comerciales, desvirtuándose el modelo relacional, especialmente en el hecho de permitir duplicados en las tablas.

El proyecto QBE (Query By Example) lo lideró Zloof [Zloof 1975], resultando en el primer lenguaje de consulta a base de datos “visual”, que influyó de manera importante a otros productos del mismo estilo como Paradox de Borland o Access de Microsoft.

Las dependencias funcionales también se introdujeron en [Codd 1970], junto con el concepto de tercera formal normal. Los axiomas para inferir dependencias funcionales y que caracterizan a las mismas se presentaron en [Armstrong 1974]. La BCNF (Boyce Codd Normal Form) se introdujo en [Codd 1972a]. El concepto de cobertura mínima fue introducido por [Paredaens 1977]. Los algoritmos para el cálculo de coberturas no redundantes se introdujo en [Beeri & Bernstein 1979]. En [Maier 1983] se puede encontrar una recopilación de todos estos resultados.

La razón y uso de las formas normales es evitar la repetición innecesaria de datos (redundancia). Una solución a este problema es repartirlos en varias tablas y utilizar referencias por valor entre ellas. Este es el ejemplo típico de que la tupla de cada empleado no debe repetir toda la información de su departamento, sino que debe utilizar una referencia por valor a la tupla de la relación departamento donde estén todos estos datos. Este procedimiento ahorra espacio de almacenamiento y al evitar la redundancia evita modificaciones parciales o incompletas que podrían dar lugar a inconsistencias.

Este proceso se conoce como normalización. Generalmente se habla de seis formas normales, de la forma normal 1 a la 3, luego la BNF, seguidas de las formas normales 4 y 5. Al nivel más bajo (primera forma normal) se tiene todo en una tabla con todos los atributos requeridos en ella. Al mayor nivel (quinta forma normal) se tienen un gran número de tablas pequeñas a las que sólo se hace referencia si se requiere la información que contienen. Parece ser que esto reduce el espacio, al no tener información repetida. Esto empieza a ser cierto hasta la tercera forma normal. A partir de ahí se utiliza mucho espacio para referencias por valor. Del mismo modo puede parecer que una normalización extrema podría reducir el tiempo para procesar las consultas, ya que los datos no necesarios no se recuperan y se necesita menos espacio de almacenamiento temporal. Sin embargo, a medida que nos movemos a formas normales mayores, se experimenta el problema de que se incrementa el tiempo necesario para concatenar las tablas. Por esta razón, en la práctica, la mayoría de organizaciones llegan hasta la tercera normal (ya introducias en [Codd 1970] como hemos dicho) a no ser que exista alguna otra razón para normalizarlas más.

Por todo lo anterior, en 1980, la ACM (Association for Computer Machinery) otorgaba a Codd el “Turing Award”, uno de los premios más prestigiosos en el campo de la informática. El modelo relacional era imparable.

Aún así, todavía a principios de los ochenta existían numerosas voces dispares al modelo relacional. Esta reacción al cambio venía fundamentada en ideas erróneas sobre los sistemas de bases de datos relacionales. Se decía que se necesitaban conocimientos teóricos para utilizar bases de datos relacionales, que la teoría de la normalización limitaba el uso de los mismos, que el rendimiento de los SGBDR era malo, que el modelo relacional era muy pobre y también era muy rígido. Estos mitos (recogidos con ironía en [Gardarin & Valduriez 1987]), fueron cayendo con el tiempo.

Teniendo en cuenta todo lo anterior, la aceptación del modelo relacional se puede considerar hoy en día como generalizada. En libros de hace poco más de una década, el modelo relacional aparecía a la par con los otros dos modelos tradicionales (también llamados navegacionales): el modelo en red o el modelo jerárquico [Ullman 1980] [Date 1981][Everest 1986] [Gardarin 1988]. Sólo los libros explícitamente denominados "bases de datos relacionales" (p.ej. [Delobel & Adiba 1985] [Gardarin & Valduriez 1987]) se atrevían a centrarse en este modelo. En los libros genéricos sobre bases de datos, los modelos navegacionales han sido desplazados a apéndices o han desaparecido completamente de los mismos. Bien es cierto que su lugar ha sido ocupado por el modelo objetual u objeto-relacional, aunque generalmente con menos profusión que el relacional [Date 1999] [Elsmasri & Navathe 2000]. También hay que destacar que estos dos últimos modelos tienen ciertas cosas en común con los navegacionales, cosa que justifica en gran medida el concepto de "modelo de datos" como abstracción que recoge todos estos cambios tecnológicos.

La siguiente gráfica muestra que hasta 1992 las ventas de SGBD relacionales no superaron las ventas de SGBD tradicionales (red, jerárquicos y otros pre-relacionales):

VENTAS MUNDIALES DE SGBD 1991-1999										
	1991	1992	1993	1994	1995	1996	1997	1998	1999	Crec. Medio 1994-1999
Pre-relacional	2.000	2.090	2.109	2.050	1.866	1.721	1.701	1.689	1.638	
Crecimiento	-	4,5	0,9	-2,8	-9,0	-7,8	-1,2	-0,7	-3,0	-4,4%
Cuota de mercado	52,0	45,5	38,8	31,6	24,0	18,4	15,2	12,6	10,3	
Relacional	1.844	2.502	3.328	4.435	5.925	7.652	9.513	11.685	14.254	
Crecimiento	-	35,7	33,0	33,3	33,6	29,1	24,3	22,8	22,0	26,3%
Cuota de mercado	48,0	54,5	61,2	68,4	76,0	81,6	84,8	87,4	89,7	
SGBD total	3.844	4.592	5.437	6.485	7.791	9.373	11.214	13.374	15.892	
Crecimiento	-	19,5	18,4	19,3	20,1	20,3	19,6	19,3	18,8	19,6%

Fuente: IDC (tomado de [de Miguel & Piattini 1997])

No obstante es de destacar que, en entornos empresariales, hay mucha cantidad de datos hoy en día que se almacenan en hojas de cálculo en vez de bases de datos, y que un porcentaje importante de los datos de producción todavía se encuentran en sistemas heredados (*legacy systems*), es decir, sistemas tecnológicamente obsoletos pero que se mantienen por el coste de realizar la migración.

Toda esta historia muestra que ciertas ideas requieren un contexto y una competencia para hacerse efectivas. Existen muchos otros casos de mejoras tecnológicas que no han llegado a imponerse por falta de apoyo por la industria o por la defensa a ultranza de soluciones peores por gigantes de la informática. En este caso, si no se hubiera producido el desarrollo por parte de la Universidad de California en Berkeley del sistema Ingres, IBM no habría visto ninguna amenaza en el modelo relacional, y no habría invertido esfuerzos en el System R. También fue importante la difusión del código de las primeras versiones del Ingres, lo que permitió a la comunidad científica experimentar con las ideas relacionales [NAP 1999]. Esto refuerza el papel de instituciones independientes, como las Universidades, para no sólo generar nuevas tecnologías sino promoverlas (aunque sean ajenas, como en el caso del modelo relacional), cuando las empresas grandes no tienen interés en perder sus cuotas de mercado.

Respecto al modelado de bases de datos, aparece la noción de modelo semántico, es decir, modelos dedicados específicamente a representar la realidad sobre la cual versa la base de datos. Al separar este modelo semántico del modelo lógico, el diseñador no tiene que preocuparse en cuestiones del modelo concreto de trabajo y de ciertas particularidades. El modelo entidad relación (ER) fue propuesto por [Chen 1976]. La generalización y la agregación se propusieron en [Smith & Smith 1977]. El modelo ER con estas dos extensiones se conoce como modelo entidad relación extendido (ERE). Debido a la simplicidad de este modelo, su expresividad y la relativamente sencilla transformación de este modelo a un esquema lógico relacional se popularizó rápidamente como herramienta para representar el modelo conceptual de un sistema de información, creando la clásica separación en las etapas del desarrollo de un sistema de información: diseño conceptual, diseño lógico, diseño

físico e implantación. El modelo ER era exclusivamente estático, expresaba las entidades de la realidad y sus relaciones. El diseño conceptual basado en los Diagramas ER (ERD) adolecía, por tanto, de dinámica. Para paliar esta carencia, inicialmente se solía combinar el ER con los populares diagramas de flujo de datos (DFD) [Constantine & Yourdon 1978], hablándose de análisis de datos (ERD) y análisis funcional o de procesos (DFD). Hoy en día se suele combinar los ERDs con lenguajes lógico-conceptuales o lenguajes de especificación de transaccionales. El modelo entidad relación ha tenido una influencia muy importante en otros lenguajes de modelado utilizados en ingeniería del software, incluso en los orientados a objetos. Posteriormente al modelo ER, aparecieron otros modelos, como el SDM de [Hammer & MacLeod 1981], el FDM (DAPLEX) de [Shipman 1981] y el modelo semántico general de [Hull & King 1987].

El software de los SGBDR se fue refinando continuamente durante los ochenta. Esto se debió en parte por la realimentación de los clientes, el desarrollo de sistemas para nuevas industrias que no solían utilizar SGBDs y el uso creciente de ordenadores personales y sistemas distribuidos. Por ejemplo, el proyecto Ingres continuó investigando las bases de datos distribuidas, la inferencia en bases de datos y las bases de datos activas.

La mayoría de sistemas empezaban a ir acompañados de un lenguaje de programación llamado de cuarta generación (los 4GLs). Dentro de estos lenguajes se podía utilizar el SQL embebido (embedded SQL). Por SQL embebido se entiende el uso de comandos SQL dentro de un lenguaje de programación, p.ej. lenguajes genéricos como el Pascal o el C, o lenguajes específicos como el PL/SQL de Oracle. Una de las doce reglas fundamentales de un SGBDR (enunciadas por Codd en 1985 [Codd 1985a, 1985b]) es que el lenguaje 4GL no puede saltarse las restricciones que se impongan sobre la base de datos. Otro de los usos de rápido crecimiento de los 4GL fue la de expresar la actividad en bases de datos, especialmente reglas de actividad (triggers) combinando el trío evento-condición-acción, donde el evento y la condición estaban expresados en términos similares al lenguaje de manipulación (adaptando SQL) y la acción se expresaba en 4GL (que a su vez podía contener sentencias de lenguaje de manipulación). Estas reglas de actividad pueden responder tanto a eventos internos como externos. Éste es el principio de las bases de datos activas [Widom & Ceri 1996], imprescindible para aplicaciones de control: plantas de fabricación, tráfico, sistemas de urgencias, reactores y motores, etc.

La aparición de diversos sistemas relacionales, aunque estuvieran la mayoría de ellos basados en SQL, planteaba problemas de compatibilidad. Existían muchas corporaciones que utilizaban distintos SGBDs para diferentes partes de su sistema de información, debido a razones históricas y evolución de la propias firmas, además de organizaciones departamentales muy estancas. Era muy usual que la organización contará con un sistema central con su SGBD además de otros SGBD sobre ordenadores personales. Que estos SGBD se interrelacionaran y pudieran compartir sus datos era fundamental. Ya no sólo el poder importarlos o exportarlos entre los distintos SGBD, sino poder acceder a ellos, es decir utilizar un SGBD para acceder a los datos que gestionaba otro SGBD. Como esta problemática era mayor cuando se combinaban sistemas centrales con personales, aparecieron protocolos para la conectividad entre bases de datos. El estándar más utilizado es el ODBC (Open Database Connectivity) de Microsoft que sirve tanto para conectar bases de datos como para que las aplicaciones puedan acceder a diferentes bases de datos. Aparece el concepto de “fuente de datos”, en el que incluso el SGBD que lo gestiona puede pasar desapercibido.

El ODBC (http://www.microsoft.com/data/odbc/) es una API (*Applications Programming Interface*) para enlazar aplicaciones con una base de datos. Se diseñó por Microsoft como un modo de que los programas se conectaran a una base de datos sin tener que usar los comandos y características específicos del SGBD. Los comandos ODBC se utilizan en los programas y luego se traducen en los comandos específicos por la interfaz ODBC que hay sobre el SGBD. Esto permite además que los programas se puedan portar de SGBD a SGBD con un mínimo de cambios de código, ya que permite al usuario indicar qué fuente de datos ODBC está utilizando, lo que permite un cambio y adaptación a otros SGBD, especialmente los nuevos SGBD y versiones que van apareciendo.

Existen muchas otras características desarrolladas durante los ochenta y los noventa que se asocian con el modelo relacional, auqneu son tecnologías en su mayor parte independientes del modlo. Quizás desde un punto de vista más teórico, el modelo relacional sí que ha generado muchos avances propios a él, en general ligados a la visión de una base de datos relacional como una teoría lógica, lo que ha permitido portar los avances de la programación lógica, como comentaremos.

Para una referencia actualizada y rigurosa sobre el modelo relacional se recomienda [Levene & Loizou 1999].

1.1.3. Bases de Datos Orientadas a Objetos

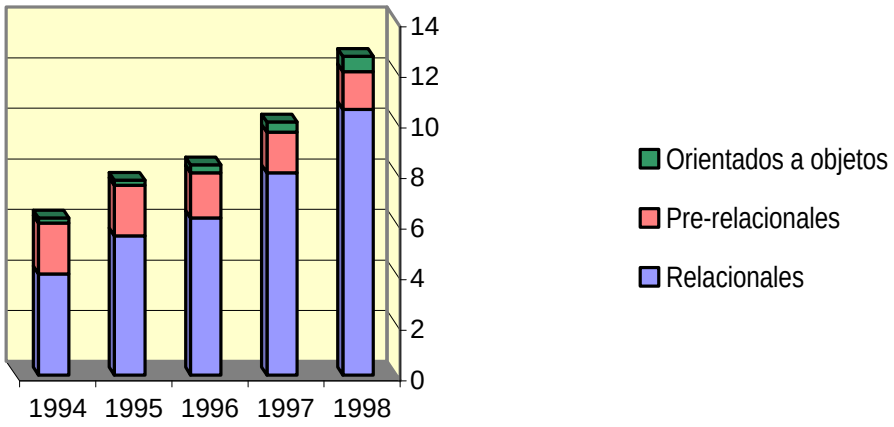
Alrededor de la mitad de los ochenta, algunas aplicaciones exigían mayor expresividad en los datos con los que trabajaban. Por ejemplo, las bases de datos médicas, las bases de datos multimedia y algunas bases de datos científicas requerían mayor flexibilidad en la manera en la que los datos se representaban y eran accedidos.

Coincidiendo con la entrada de los lenguajes orientados a objetos como Smalltalk o C++ en el ámbito industrial, los investigadores se plantearon transportar estas ideas a las bases de datos y permitir que el tipo de datos marcara cómo se representaba y se manipulaba dependiendo de los métodos que se definían para dicho tipo o clase.

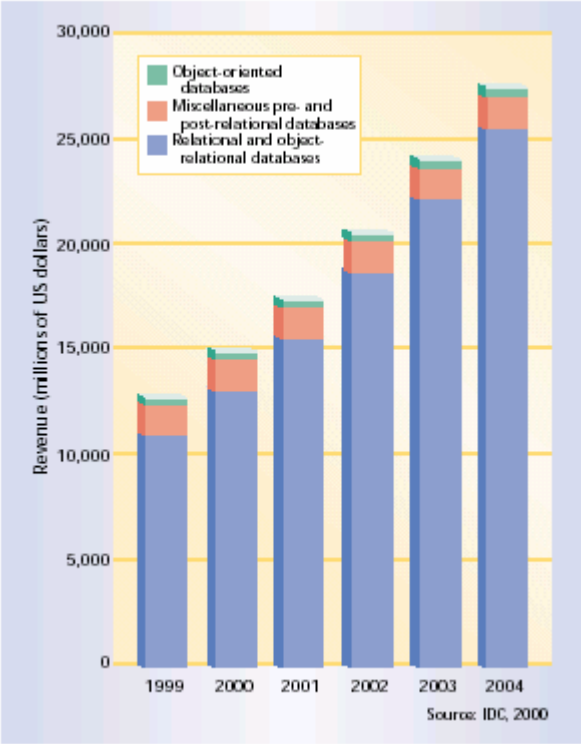
La idea de una base de datos orientada a objetos se articuló por primera vez por [Copeland & Maier 1984], con el sistema prototipo GemStone. Uno de los sistemas más famosos de los finales de los ochenta y principios de los noventa fue el sistema ObjectStore [Lamb et al. 1991]. Al principio de los noventa, los primeros Sistemas de Gestión de Bases de Datos Orientados a Objetos (SGBDOO, o simplemente, SGBDO) empezaron a aparecer en el mercado, a partir de compañías como Objectivity.

En este modelo la información sobre una entidad se almacena como un objeto persistente y no como una fila en una tabla. Esto, en principio, lo hace más eficiente en términos de requerimientos de espacio y asegura que los usuarios puedan manipular los datos sólo de las maneras en las que el programador haya especificado. También es más eficiente en el uso de espacio de disco requerido para las consultas, ya que en vez de almacenar la consulta, simplemente se construye una serie de índices (punteros) a los objetos seleccionados. A esto hay que sumar las ventajas derivadas del modelo orientado a objetos, ya explotadas en sus lenguajes de programación, la mayor expresividad y su adecuación para almacenar muchos tipos de datos diferentes.

Alguien podría pensar, por tanto, que las bases de datos orientadas a objetos deberían de haber superado en la práctica a las relacionales. De hecho, a veces se denominan postrelacionales. No obstante, después de más 15 años, el mercado de las bases de datos orientadas a objetos no supone más de un 5% del mercado de las relacionales, como se puede ver en las siguientes gráficas:



Ingresos de las ventas mundiales de SGBD (miles de millones de dólares) entre 1994 y 1998.
Fuente: Price Waterhouse (tomado de [de Miguel & Piattini 1997])



Ingresos de las ventas mundiales de SGBD (millones de dólares.) entre 1999 y 2000, y predicciones 2001-2004. Fuente: IDG 2000 (tomado de [Leavitt 2000])

La situación no ha variado significativamente en los últimos años, y, en el ejercicio 2001, de los 8.844 millones de dólares de ingresos por *nuevas* licencias, 7.107 millones de dólares correspondieron a sistemas relacionales. Esto supone un 80,4% de los *nuevos* sistemas, con lo que no parece haber ninguna señal clara de que la situación vaya a cambiar en el futuro [Graham 2002]. Más aún si tenemos en cuenta que los mercados de los SGBD pre-relacionales y objetuales tuvieron un incremento negativo en 2000, mientras que los relacionales crecieron un 15% [IDG 2001] [SMITT 2002].

Hay varias razones para explicar este hecho. En primer lugar, las bases de datos objetuales acarrean consigo algunas de las propiedades no deseables de los modelos pre-relacionales. El programador tiene que tener mucha información sobre la estructura de los datos. Si se conocen las propiedades de los objetos, la consulta es rápida y simple. Pero la realidad es que en muchos casos se desconocen las identidades de los objetos. Lo que preocupa o interesa es almacenar los atributos de los objetos y relacionar los valores de estos atributos, aspecto en el que el modelo relacional es más sencillo.

En segundo lugar, el hecho de que las organizaciones sean capaces de alterar los métodos de bajo nivel utilizados en los SGBDO, hace que sea más difícil para terceros el hacer productos añadidos. Mientras que las bases de datos relacionales se pueden beneficiar del software realizado por otras firmas, los usuarios de SGBDO tienen que producir el software en casa para adaptarse a sus propias particularidades, parte de ellas incorporadas al *comportamiento* de los objetos.

En tercer lugar y quizás más importante es el hecho que las organizaciones tiendan a ser conservadoras en relación con las bases de datos, uno de sus activos más valiosos. Muchas organizaciones aunque utilizan lenguajes orientados a objetos como el C++ para las aplicaciones microinformáticas o aplicaciones específicas, desconfían de los lenguajes orientados a objetos en general por no considerarlos suficientemente estables para trabajar con información crucial para la organización. Mito o realidad, el hecho es que no acaban de decidirse por embarcarse en un SGBDO y siguen aferrados al SQL para realizar sus informes y al SQL embebido para interrelacionar las aplicaciones con el SGBD, manteniendo una separación que consideran imprescindible.

Del mismo modo que evolucionaba el modelo objetual para el nivel lógico y el nivel físico, debían también evolucionar o desarrollarse nuevos modelos conceptuales y metodologías orientadas a objetos. El modelo entidad relación (extendido) se mostró insuficiente para la etapa de diseño conceptual de bases de datos orientadas a objetos. Para ello se importaron o adaptaron lenguajes de modelado e incluso gran parte de las metodologías utilizadas en ingeniería del software orientada a objetos. Por ejemplo, la metodología *Object Modeling Technique* (OMT) desarrollada a principios de los noventa en General Electric [Rumbaugh et al. 1996], la

metodología (análisis y diseño O-O) de Booch [Booch 1991, 1994] o posteriormente el lenguaje de modelado 'unificado' UML [Booch et al. 1997] [Booch et al. 1999], aunque pensadas para el desarrollo de software orientado a objetos, se han ido adaptando o importando al campo de las bases de datos como herramienta de modelado conceptual de bases de datos orientadas a objetos.

El estándar UML y su integración en metodologías de diseño y modelado de bases de datos ha supuesto quizás el mayor gran impulso a este modelo de datos desde su introducción en los ochenta. El cuerpo principal del lenguaje de modelado unificado UML se desarrolló principalmente a principio de los noventa, por un grupo de 'gurús' de la ingeniería de software orientada a objetos, especialmente los ahora llamados "tres amigos": Booch, Jacobson y Rumbaugh. Cuestiones comerciales aparte, el UML fue aceptado como estándar por el ODMG (del que hablaremos más abajo) y el hecho de haberse estandarizado un lenguaje de modelado orientado a objetos y que éste esté basado en lo "mejor" de otras metodologías anteriores, hace que la adaptación de profesionales y su movilidad entre distintas organizaciones sea más sencilla.

UML puede utilizarse para cualquier sistema orientado a objetos, lo que le hace también apropiado para el diseño de bases de datos orientadas a objetos. Especialmente lo hace apropiado para que los especialistas en bases de datos y analistas especifiquen lo que quieren que los programadores realicen, en términos del SGBDO y de los programas que interaccionan con él. Finalmente hay que recordar que UML es sólo un lenguaje y no una metodología, con lo que también es posible utilizar UML como notación para expresar el modelo entidad-relación (véase p.ej. [Muller 1999] o [Connolly & Begg 2000]).

El ODMG (*Object Database Management Group*) es un grupo de vendedores y usuarios de bases de datos que desarrollan estándares para los SGBDO (www.odmg.org). El primer documento fue el ODMG 1.0 in 1993, que incluía el OQL (*Object Query Language*), un lenguaje de consultas orientado a objetos muy inspirado en el SQL. Aunque durante los principios de los noventa tuvo bastante fuerza, hoy su importancia es menor debido a que SQL3 incluye muchas características orientadas a objetos, como veremos a continuación.

Además del OQL, el ODMG v.3.0 incluye un modelo de objetos estándar y enlaces para los tres lenguajes orientados a objetos más populares: C++, Smalltalk y Java. La especificación del ODMG 3.0 se puede encontrar en (www.odmg.org), en [ODMG 2000] o en [Cattell & Bary 2000]. Para una información más general sobre las bases de datos orientadas a objetos, se recomienda [Eaglestone 2000].

1.1.4. Bases de Datos Relacionales Orientadas a Objetos (u Objeto-Relacionales)

El modelo objeto-relacional es un desarrollo más reciente y parece haber tenido bastante efecto. No es una tecnología en sí, sino una aglutinación de los modelos relacional y orientado a objetos. De hecho, algunas extensiones objetuales a los sistemas relacionales se pueden datar en los principios de los ochenta [Zaniolo 1983].

Como hemos dicho antes, existía la necesidad imperiosa de la industria y de sus clientes de tratar con nuevos tipos de datos: audio, imágenes y vídeo, además de tipos definidos por el usuario con sus propias propiedades. Por otra parte, hemos visto que las organizaciones eran reticentes a migrar de un SGBDR a un SGBDO por diversos motivos.

Además, el mantenimiento de los SGBDR empezaba a crear *desajustes* con un cada día más generalizado uso de los lenguajes orientados a objetos. Los programadores en estos lenguajes tenían que realizar una serie de pasos de traducción de la estructura objetual del programa y de los datos en memoria principal a la estructura relacional de los datos.

Aquí es donde el modelo objeto-relacional puede demostrar su valor. El modelo objeto-relacional se define como una extensión objetual del modelo relacional, permitiendo la definición de nuevos tipos y de relaciones de herencia, entre otras cosas [Stonebraker et al. 1998] [Maro-Saracco 1998]. Permite a las organizaciones continuar usando sus sistemas existentes y sus datos, sin realizar prácticamente cambios y les permiten empezar a utilizar gradualmente características orientadas a objetos, especialmente si se hace en conjunción con aplicaciones desarrolladas en entornos también orientados a objetos.

Uno de los ejemplos de este mestizaje es, por ejemplo, JDBC (Java DataBase Connectivity). Aunque pensado para interrelacionar bases de datos relacionales con Java, va incorporando algunos detalles objeto-relacionales. De una manera más precisa, JDBC es una API (Applications Programming Interface) desarrollada por Sun Microsystems (http://java.sun.com/products/jdbc/index.html) para conectar Java con las bases de datos, con una filosofía inspirada en ODBC. De hecho es usual conectar un programa Java con una base de datos directamente o a través de ODBC, dependiendo de si el fabricante de la base de datos ha creado los *drivers* necesarios para Java.

Otro ejemplo de mestizaje a un nivel diferente es el SQL3 [ANSI/ISO/IEC 1999]. SQL3 es la nueva versión desarrollada por los comités ANSI X3H2 e ISO DBL para extender SQL2 con el tratamiento de datos orientados a objetos (entre otras cosas). Hablamos de mestizaje porque esta extensión se ha hecho de tal manera que el SQL3 es compatible hacia adelante con el SQL-2.

Las facilidades orientadas a objetos del SQL3 se centran en extensiones de los tipos y las tablas en SQL. Las partes del SQL3 que proporcionan la base para trabajar con estructuras orientadas a objetos son: tipos definidos por el usuario (user-defined types, UDTs), constructores de tipo para tipos de fila y tipos referencia, tipos constructores para colecciones (conjuntos, listas, ...) y funciones y procedimientos definidos por el usuario.

Una de las ideas básicas detrás de estas extensiones es que, además de los tipos predefinidos (built-in) de SQL, el usuario puede definir otros tipos que pueden ser utilizados después como los tipos predefinidos. Siguiendo el paradigma orientado a objetos, una definición de UDT encapsula los atributos y las operaciones en una única entidad o clase, que en SQL3 se llama *tipo*. En concreto en SQL3 se define un UDT declarando los atributos que almacenan el valor y estado del UDT, definiendo las operaciones de igualdad y de orden para el tipo, y, finalmente, definiendo las operaciones que determinan el comportamiento específico del UDT. Las operaciones (lo que en el paradigma orientado a objetos se suelen llamar métodos) se implementan mediante procedimientos llamados rutinas. Las operaciones pueden incluir comandos de manipulación SQL embebidas (SELECT, INSERT, UPDATE, DELETE).

El SQL3 también está provisto de herencia y delegación. Mediante la especificación de "UNDER <nombre del UDT >" en la parte de subtipo de una definición UDT, se puede definir un UDT como un subtipo (lo que se conoce normalmente como especialización, clase derivada o subclase) de un UDT existente. Un subtipo hereda todos los atributos y el comportamiento de sus supertipos y se pueden extender con más atributos y variar su comportamiento. Una instancia de un subtipo se puede utilizar en cualquier lugar donde una de las instancias de sus supertipos se pueda utilizar. De momento, SQ3 no soporta la herencia múltiple, con lo que un UDT puede tener como mucho un supertipo.

A pesar de todas estas extensiones, la estructura principal de representación lógica del SQL3 sigue siendo la tabla, con lo que el modelo subyacente del SQL3 se puede considerar objeto-relacional.

1.1.5. Bases de Datos Paralelas y Distribuidas

Las bases de datos *paralelas* se empezaron a desarrollar alrededor de 1980, especialmente con el proyecto Gamma [DeWitt et al. 1990], un sistema de base de datos sobre una serie de procesadores de propósito general funcionando en paralelo. Este sistema es en el que se inspiran la mayoría de sistemas paralelos de IBM, Tandem, Oracle, Informix, Sybase y AT&T. Además, el uso de sistemas paralelos para la minería de datos es uno de los campos de investigación más activos actualmente.

Los sistemas de bases de datos paralelos, como casi toda la tecnología paralela, fue acuñada como la tecnología del futuro en cuanto a altas prestaciones. Hoy en día la postura es más realista y se reconoce su uso en sistemas de *muy* altas prestaciones, aunque para sistemas de uso corriente, incluso grandes empresas, su uso es más limitado.

No obstante, por muy paralelo que fuera el sistema, todo ordenador tiene su límite. Aunque la capacidad de proceso aumentara, existían limitaciones en la cantidad de memoria que el sistema direccionaba, el número de discos duros que podían conectarse a un mismo procesador y el número de procesadores que podían correr en paralelo. En la práctica, esto significa que, a medida que la cantidad de información de una gran base de datos aumenta, un único sistema, aunque sea paralelo, deja de poder dar abasto con toda la información que tiene que almacenarse, ordenarse y consultarse.

Aunque es posible comprar sistemas cada día más grandes y rápidos, a veces no es económico sustituir el hardware cada pocos años o incluso meses, periodo en el que se suele duplicar la información de una organización. En cambio, es mucho más realista tener varios servidores de bases de datos e ir añadiendo a medida que la organización necesita más capacidad. Esto se debe hacer de manera que los usuarios crean que siguen trabajando con un único sistema, el sistema integrado de la organización. Mediante esta filosofía, la organización tiene más flexibilidad en sus ampliaciones, se realizan de una manera menos traumática y con ordenadores de talla media, que suelen ser mucho más baratos que uno grande equivalente en potencia.

Este concepto lleva al paradigma de las bases de datos distribuidas[3], y tienen las características comunes de que los datos se almacenan en dos o más ordenadores, llamados nodos, y que estos nodos están conectados en una red. Hoy en día, con el aumento de la descentralización, debido al abaratamiento, ancho de banda y flexibilidad de las redes de computadores, ha hecho que el uso de las bases de datos distribuidas haya aumentado considerablemente.

Si hablamos de un único sistema de gestión de bases de datos que actúa sobre distintos ordenadores distribuidos y gestionando la misma base de datos, hablamos propiamente de bases de datos distribuidas. En el caso que cada sistema esté formado por varios SGBD, se suele hablar de *Sistemas de Múltiples Bases de Datos*, que pueden estar fuertemente acoplados o débilmente acoplados, llamándose estos últimos sistemas interoperantes de bases de datos [Litwin & Chien 1994].

Sin entrar en demasiadas distinciones terminológicas entre los sistemas de bases de datos distribuidos y los sistemas de múltiples bases de datos, o si los datos pueden o no estar replicados, el punto fundamental de todos estos sistemas es que los usuarios no deben percatarse de esta dispersión espacial de los datos, es decir, los usuarios deben percibir lo mismo que si trabajaran con un único sistema centralizado.

En general, se suele realizar la siguiente clasificación habitual entre sistemas de bases de datos homogéneos y heterogéneos.

- Las bases de datos distribuidas homogéneas usan el mismo software de SGBD y tienen las mismas aplicaciones en cada nodo. Tienen un esquema común y pueden tener grados diversos de autonomía local. Pueden estar basadas en cualquier SGBD que soporte estas características, pero no puede haber más de un SGBD en el sistema. La autonomía local especifica cómo el sistema funciona desde la perspectiva de los usuarios y programadores. Por ejemplo, podemos tener un sistema con poca o sin autonomía local, donde todas las peticiones se envían a un nodo central, llamado *gateway*. Desde aquí se asigna al nodo que contiene esa información o aplicación requerida. Esto es lo típico que se ve con los *mirrors* de sitios web muy populares a los cuales una página central deriva las peticiones de sus usuarios dependiendo de su origen geográfico.
- En el otro lado de la escala, se encuentran las bases de datos heterogéneas con un alto grado de autonomía local. Cada nodo en el sistema tiene sus propios usuarios, aplicaciones y datos locales y es el sistema el que trata con ellos directamente y sólo conecta con otros nodos en busca de información que no tiene. Este tipo de base de datos se suele llamar *sistema federado* o *federación*. Se ha hecho cada día más popular en las organizaciones, tanto por su escalabilidad, su capacidad de mezclar distintos paquetes software y su reducido coste al añadir nuevos nodos cuando es necesario. A diferencia de los sistemas homogéneos, los sistemas heterogéneos pueden incluir diferentes SGBD en los nodos. Esto los hace atractivos en grandes corporaciones, ya que pueden mantener sus sistemas heredados antiguos (*legacy systems*) junto con los nuevos sistemas.

Uno de los primeros sistemas distribuidos fue R*, desarrollado en IBM [Williams et al. 1998] a principios de los ochenta. Últimamente el área de las bases de datos distribuidas está cada vez más en relación con Internet y la tecnología web, hablándose de bases de datos distribuidas de área global [Stonebraker et al. 1996]. Una fuente actual sobre sistemas de bases de datos distribuidas se puede encontrar en [Ozsu & Valduriez 1999]

1.1.6. Bases de Datos Multimedia

Las bases de datos multimedia almacenan una gran variedad de tipos de datos. Estos tipos incluyen texto, imágenes, audio y vídeo. Hasta hace unos años estas bases de datos eran difíciles de implementar, debido al tamaño de los objetos y la complejidad de los datos. El añadir metadatos a los ficheros multimedia (p.ej. el formato WAV) resolvía parcialmente el problema. Esta cabecera incluye datos del formato, el creador, el contenido, la longitud del *stream* de datos, etc. El problema es que está información no suele estar indizada conveniente y no se puede utilizar para realizar consultas.

Imaginemos una base de datos que contiene clases grabadas en vídeo. La metainformación que nos puede interesar de cada vídeo puede ser: compañía/universidad donde se dio la clase, quién la dio, cuándo se dio, de qué va, cuánto duró, etc. Esta información es la que se utilizará cuando los usuarios busquen clases en la base de datos.

[3] Debido al avance en las redes de computadores la división entre bases de datos paralelas y bases de datos distribuidas es cada día más sutil, y existen muchos sistemas híbridos.

Hay dos métodos principales para incluir metadatos en una base de datos multimedia: mediante análisis automático o mediante análisis manual [Chorafas 1994]:

- Aunque el análisis manual es más efectivo, porque permite anotar aquellas características del objeto multimedia que son importantes, requiere mucho tiempo y por tanto es muy costoso. Además es difícil conseguir homogeneidad en los criterios que se utilizan para agregar estos metadatos.
- El análisis automático es mucho más rápido, pero las técnicas todavía son limitadas en algunos casos (especialmente en imagen y sonido). Su mayor ventaja es que proporciona una descripción consistente de los datos y no se ve afectada por estilos individuales.

Si hace pocos años, las bases de datos multimedia no eran asequibles para los usuarios de ordenadores personales, el abaratamiento de discos con gran capacidad de almacenamiento, hace que cualquier ordenador personal pueda contener una biblioteca de imágenes y piezas musicales, y en breve, de películas. Otro problema es el uso de estas bases de datos para distribución por Internet (especialmente VoD, Video on Demand), para el cual el ancho de banda todavía tiene que crecer enormemente.

Una referencia más amplia y actualizada del área de las bases de datos multimedia se puede encontrar en [Subrahmanian 1998] o [Furht & Marques 2000].

1.1.7. Almacenes de Datos

La mayoría de *decisiones* de empresas, organizaciones e instituciones se basan en información de experiencias pasadas. Generalmente, la información que se quiere investigar sobre un cierto dominio de la organización se encuentra en bases de datos y otras fuentes muy diversas, tanto internas como externas.

Muchas de estas fuentes son las que se utilizan para el trabajo diario. Tradicionalmente el análisis para la toma de decisiones se realizaba sobre estas mismas bases de datos de trabajo o bases de datos transaccionales. La situación era la siguiente.

- Se mantiene el trabajo transaccional diario de los sistemas de información originales (conocido como OLTP, *On-Line Transactional Processing*).
- Se hace análisis de los datos en tiempo real sobre la misma base de datos (conocido como OLAP, *On-Line Analytical Processing*),

Esta perspectiva provoca algunos problemas. En primer lugar, disturba el trabajo transaccional diario de los sistemas de información originales, ya que se realizan consultas muy pesadas ("*killer queries*"). A veces, la perturbación es tal que estas consultas para generar informes se deben hacer por la noche o en fines de semana. En segundo lugar, la base de datos está diseñada para el trabajo transaccional, no para el análisis de los datos. Esto hace que el análisis sea lento, con lo que no podemos hablar de OLAP, sino simplemente de AP.

Para poder operar eficientemente con esos datos, y gracias a que los costes de almacenamiento masivo y conectividad se han reducido drásticamente en los últimos años, parece razonable recoger (copiar) los datos en un sistema unificado.

A partir de aquí nacen los almacenes de datos (*data-warehouses*) y toda su tecnología asociada (*data-warehousing*). Los almacenes de datos facilitan el análisis de los datos en tiempo real (OLAP) y no disturban el OLTP de las bases de datos originales. El hecho de separar los datos a analizar con respecto a sus fuentes transaccionales (se copia/almacena toda la información histórica) requiere una tecnología sobre cómo organizarlos y sobretodo cómo tenerlos actualizados (cargas periódicas) respecto a los datos originales.

Especialmente, un aspecto muy importante es la organización de la información copiada. El esquema del almacén de datos no suele coincidir con el esquema transaccional. De hecho, los esquemas de almacenes de datos suelen desnormalizarse, con el objetivo de acelerar ciertas consultas analíticas. En general, se distinguen dos tipos de almacenes de datos:

- ROLAP (Relational OLAP): el almacén de datos es relacional.
- MOLAP (Multidimensional OLAP): el almacén de datos es una matriz multidimensional.

El objetivo es que la información esté estructurada de manera que facilite la tarea de dos tipologías de usuarios:

- 'picapedreros' (o 'granjeros'): se dedican fundamentalmente a realizar informes periódicos, ver la evolución de determinados parámetros, controlar valores anómalos, etc.

- ‘exploradores’: encargados de encontrar nuevos patrones significativos utilizando técnicas de minería de datos, que comentaremos más adelante.

En definitiva, los almacenes de datos tienen un fin bien diferente de los sistemas transaccionales y por tanto tienen una tecnología y problemática diferentes. Las mayores diferencias entre los sistemas transaccionales y los almacenes de datos son las siguientes:

	SISTEMA TRANSACCIONAL	ALMACÉN DE DATOS
Propósito	Operaciones diarias	Recuperación de información histórica y análisis
SGBD más usuales	SGBDR	SGBDR o SGBDM
Estructura de los datos	Normalizado	Multi-dimensional
Tipo de datos	Datos para el funcionamiento de la organización	Datos que es interesante analizar
Condición de los datos	Cambiantes, incompletos	Históricos, descriptivos.

La organización matricial, también conocida como *Data Cubes* [Gray et al. 1997], está especialmente diseñada para los almacenes de datos con la ventaja de optimizar y unificar los operadores de agregación tradicionales: *group by* y subtotales. En esta organización, cada atributo relevante se establece en una dimensión, que se puede agregar o desagregar. La base de datos está completamente desnormalizada. Existe una terminología propia para estos operadores: *roll-up* (consolidación o sumarización), *drill-down* (detalle), *slicing and dicing* (combinaciones cruzadas).

En el caso de los almacenes de datos sobre sistemas relacionales, la estructura también está muy desnormalizada, centrándose alrededor de tablas de hechos, de las que derivan otras tablas detalle, según varias dimensiones. De ahí, el nombre que se le dan a estas estructuras: estrella simple o estrella jerárquica (copo de nieve). Esta estructura también permite la sumarización, la visualización y la navegación según las dimensiones de la estrella.

No es de extrañar que las consultas realizadas de tipo OLAP al almacén de datos sean complejas. Por ejemplo, elegir los cinco clientes con mayor volumen de compras no es una consulta trivial en SQL. Si además se requiere esta información por zonas geográficas y se quiere explotar la estructura del esquema del almacén de datos, la consulta puede complicarse bastante. Si además, se requiere que sea eficiente, es posible que se deban utilizar vistas parciales y otros mecanismos para obtener los informes más rápidamente.

Finalmente, si hace unos años los almacenes de datos se limitaban a recoger información interna de la organización, con el crecimiento de las redes y de la información disponible, los almacenes de datos también recogen información externa o del contexto de la organización, como por ejemplo:

- Demografías (censo), páginas amarillas, psicografías, gráficos web, información de otras organizaciones.
- Datos compartidos en una industria o área de negocio, organizaciones y colegios profesionales, catálogos, etc.
- Datos resumidos de áreas geográficas, distribución de la competencia, evolución de la economía, información de calendarios y climatológicas, programaciones televisivas-deportivas, catástrofes, etc.
- Bases de datos externas compradas a otras compañías.

En resumen, aunque los almacenes de datos es una área de gran interés actualmente, se puede considerar como una rama establecida de la tecnología de las bases de datos, especialmente vigente en grandes organizaciones e industrias de la distribución, y suele funcionar en relación con los sistemas de toma de decisión (DSS, Decision Support System) y los Sistemas de Información Ejecutiva (EIS, Executive Information System). Es relevante constatar que el uso de los almacenes de datos es el original de los sistemas de información: recopilar datos para analizarlos y tomar decisiones a partir de este análisis.

1.2. Situación Actual

Hoy en día se puede observar en perspectiva el cambio de los sistemas de gestión de bases de datos. Inicialmente se trataba de software muy caro, sobre grandes y costosos ordenadores. Actualmente existen sistemas de gestión de bases de datos para ordenadores personales, como hemos visto, siendo muchos de ellos económicos o incluso gratuitos. Esta tendencia al abaratamiento y disminución en tamaño físico de los sistemas contrasta con la cada

vez mayor capacidad, potencia y prestaciones de los SGBD. Sólo median menos de 30 años desde el primer sistema de gestión relacional, el "System R", cuyo primer prototipo podía almacenar 8MB de datos hasta los terabytes usuales hoy en día en cualquier organización discreta.

Algunos proyectos en desarrollo actualmente parecían impensables hace sólo unos años. Por ejemplo, el proyecto RD45 (http://wwwinfo.cern.ch/asd/cernlib/rd45/index.html), es uno de los proyectos más ambiciosos que se están desarrollando en el CERN con el objetivo de crear una base de datos distribuida capaz de almacenar en 2005 un Exabyte (1 Exabyte = 1.024 Petabytes $\cong 1 \times 10^{18}$ bytes).

1.2.1. Bases de Datos y la Web

Quizás uno de los aspectos que se han notado más recientemente en el campo de las bases de datos (como en casi cualquier otro campo de la informática) es el crecimiento vertiginoso de Internet y del WWW. La conexión de las bases de datos con la web ha ido progresando de una interrelación realizada a través de herramientas ad hoc hasta la situación actual, en la que prácticamente todo SGBD proporciona un módulo o toda una serie de herramientas para publicar la información de la base de datos en la red, siendo accesible desde cualquier punto, utilizando un navegador [Feiler 1999]. Con el uso de Internet y de las intranets, la disponibilidad de los datos de la organización se ha hecho prácticamente ubicua, sin necesidad para las operaciones más comunes y sencillas del desarrollo de ninguna aplicación cliente, exceptuando el navegador. Los catálogos, inventarios, stocks, indicadores, etc. de cualquier empresa o comercio están disponibles a cualquier usuario en cualquier momento, con sus respectivos permisos y de manera concurrente.

Es de destacar que la tecnología web ha hecho evolucionar la tecnología cliente/servidor de dos capas a una tecnología comúnmente estructura en tres capas (1. cliente / 2. servidor de aplicaciones / 3. servidor de datos), aunque la mayoría de los aspectos del paradigma cliente/servidor son aplicables a la web.

En definitiva, con el advenimiento de la web, la gestión de datos se ha ramificado para tratar con la variedad de información disponible en el WWW. La mayoría de accesos web de hoy en día disparan alguna forma de generación de contenido de una base de datos, mientras que el comercio electrónico está destinado a hacer un uso intensivo de las aplicaciones basadas en un SGBD.

En consecuencia de todo esto, el interés de la comunidad científica y de las empresas del sector se centra en la revisión o extensión de modelos de datos y de lenguajes de consulta, la integración de datos tan diversos, la reconcepción de los índices, las transacciones y el procesamiento de consultas, con el objetivo de adaptarse a las características y la escala de los datos en la web. Se han identificado nuevos problemas, como saber tratar con el solapamiento de información y la detección de copia, así como cuestiones específicas de la web como herramienta de publicación. También hay un gran interés por el lenguaje XML, del cual hablaremos más adelante, y por la extracción y recogida de información de la web, su almacenamiento en un almacén de datos y su prospección.

1.2.2. Situación del Mercado de los SGBD y Estandarización

La mayoría de sistemas de gestión de bases de datos de hoy en día son relacionales, como se ha visto anteriormente. Además, el mercado está muy concentrado por tres compañías: Oracle, IBM y Microsoft.

Según un estudio de Dataquest (www.dataquest.com), ahora llamado Gartner Group (www.gartner.com), IBM y Oracle han estado codo con codo durante los últimos años con cerca del 30% de ingresos por ventas de nuevas licencias en el mercado de SGBD cada uno [Graham 2002].

PORCENTAJE DEL MERCADO DE SGBD				
Compañía	**1998**	**1999**	**2000**	**2001**
Oracle	30.7	31.1	34.1	32.0
IBM	30.0	29.9	30.3	31.7
Microsoft	10.7	13.1	14.0	16.3
Informix	4.8	4.3	3.3	3.0
Sybase	3.6	3.3	3.2	2.6
Otros	20.2	18.3	15.0	14.4
TOTAL	100.0	100.0	100.0	100.0

No obstante, hay que destacar que la distribución de estos porcentajes no es uniforme en la gama de aplicaciones. Por ejemplo, IBM sigue siendo líder en el área de los mainframes, especialmente con su línea

OS/390 y AS/400. Por el contrario, si quitamos los mainframes, Oracle lidera el mercado con una gran ventaja. Según un último estudio del Gartner Group, la situación en mayo de 2002 sigue siendo hegemónica para Oracle, IBM y Microsoft [Nicolett 2002]. La gama de SGBD de Informix se ha venido a menos en cuota de mercado. De hecho en 2001, fue absorbida por IBM, para que ésta aumentara su cuota en las plataformas Unix y Windows [Burton 2001]. Además, si sumamos los SGBD de IBM actuales, incluyendo los absorbidos de Informix, tenemos la mayor cuota, un 34,7% en 2001. No obstante, a medio plazo ocurrirá que muchos de los usuarios de Informix migren a DB2. Más aún si tenemos en cuenta que IBM ha intentado aumentar su presencia mediante la potenciación de las versiones UNIX y NT de su DB2, que han crecido considerablemente en los últimos años.

Si atendemos al porcentaje de mercado de los sistemas de gestión de bases de datos relacionales, podemos observar todavía una mayor concentración:

PORCENTAJE DEL MERCADO DE SGBD RELACIONALES		
Compañía	**2000**	**2001**
Oracle	42.5	39.8
IBM	29.5	30.7
Microsoft	11.6	14.4
Informix	3.1	3.3
Sybase	4.0	3.3
Otros	9.3	8.5
TOTAL	100.0	100.0

Considerando sólo los sistemas relacionales, mientras sobre sistemas operativos Windows, Microsoft es el líder en 2001 con el 39.9% del mercado (respecto a un 34.0% de Oracle), en sistemas operativos UNIX, Oracle es claramente predominante, con un 63.3%.

La excesiva concentración del mercado plantea serias dudas sobre la implantación del nuevo SQL3, porque las compañías no tienen excesivo interés por hacer su SQL fácilmente portable a otros sistemas, con el riesgo de poder perder clientes. No obstante, parte del nuevo SQL3 ha ido recogiendo los estándares 'de facto' que estas mismas compañías han ido introduciendo, como el soporte para los objetos grandes incorporados (BLOB y LOB) o los triggers, ambas extensiones presentes en el sistema emblema de Oracle, por ejemplo.

La incorporación de estas y otras extensiones de SQL3 (especialmente las orientadas a objeto y las consultas recursivas) a la mayoría de sistemas será un proceso lento. De hecho, hoy en día, ningún sistema es todavía completamente compatible con todas las características de SQL2. El tema parece estar mucho peor para incorporar las instrucciones de control y de definición de rutinas del SQL3 (CASE, IF, THEN, ELSE, ELSEIF, LOOP, WHILE, REPEAT, FOR, ITERATE, LEAVE), ya que la mayoría de sistemas incorporan su propia sintaxis para los procedimientos, arrastrada de sus lenguajes 4GL.

1.2.3. Diseño y Desarrollo de Bases de Datos

No sólo la tecnología es suficiente para que los sistemas de información de hoy en día funcionen mejor que los de hace unos años. Asociadas a las tecnologías suelen asociarse unas metodologías, que intentan sacar provecho de las primeras. Utilizar un sistema de gestión de bases de datos relacional no es por sí solo una garantía de que el sistema de información que se construya utilizándolo vaya a funcionar bien. De hecho, dada la simplicidad del modelo relacional y de algunos SGBR para ordenadores personales, existen verdaderas plepas realizadas por no profesionales funcionando en pequeñas y grandes organizaciones, causando casi más problemas de los que resuelven.

A continuación se realiza un rápido repaso (y bastante simplista) de los pasos usuales que se suelen utilizar a la hora de diseñar una base de datos. Generalmente se habla de las siguientes etapas: planificación-definición del sistema, análisis de requerimientos, diseño conceptual, elección de SGBD/modelo, diseño lógico, diseño físico, implementación y ajuste de rendimiento. Para las primeras tres etapas los pasos suelen ser bastante coincidentes (aunque con herramientas diferentes) para las bases de datos relacionales y las objetuales, notándose más la diferencia en las cuatro últimas etapas.

1. Planificación y definición del sistema: aunque a veces estas fases se engloban en la siguiente fase de análisis de requerimientos, consisten en determinar cuáles van a ser las fases del diseño de la base de datos y dar una visión global del sistema.
2. Análisis de requerimientos: En esta fase de un proyecto, el mayor objetivo es proporcionar una imagen más clara del sistema de información cuyos datos se quiere informatizar. Para ello se deben definir cuáles

son los componentes concretos del sistema de información (usuarios, contexto, etc.), definir qué se espera que el sistema haga y qué datos en concreto se requerirán para su funcionamiento.

3. Diseño conceptual: Una vez que se tiene la especificación inicial del sistema, un profesional (o un grupo) experto en bases de datos o un analista puede empezar a realizar el esquema conceptual del sistema. Éste es una visión de alto nivel del sistema que registra qué información se va a almacenar, qué formato va a tener y cómo se relaciona con otra información. Este esquema conceptual también especifica los derechos de acceso de grupos y programas. Para las bases de datos relacionales se suele utilizar el modelo entidad relación para proporcionar una visión conjunta del sistema. En algunos casos, sobretodo si los datos son muy heterogéneos, se puede decidir realizar el modelo utilizando lenguajes de modelado orientados a objetos como UML.
4. Elección de SGBD: No es una fase realmente, porque este paso se suele pensar antes del desarrollo de una base de datos en concreto o se decide para ser utilizado con varios fines. No obstante, si la organización dispusiera de varios, o de ninguno, con lo que tendría que elegir, esta elección se limita por muchas razones: monetarias, conocimientos disponibles de los profesionales informáticos, decisiones de gestión y un número importante de otros factores, específicos a cada organización. De todos los sistemas disponibles, se intentará encontrar el sistema más apropiado para la base de datos que se desea diseñar. Las siguientes fases se podrán realizar ya acordes con las capacidades y limitaciones del sistema elegido.
5. Diseño lógico: Durante esta fase del proceso, se toma el esquema producido en el diseño conceptual y se convierte al modelo sobre el que trabaja el SGBD. Esta fase nos acerca ya al sistema final, ya que se tiene un modelo que se adapta al SGBD donde funcionará.
6. Diseño Físico: En esta fase, se centran los esfuerzos en la implementación práctica de la base de datos. En esta fase se incluyen las pruebas de hardware, el cálculo del nivel de estrés (carga) que el sistema puede aguantar. Esto es importante, especialmente si los datos van a accederse frecuentemente y pueden existir picos. Además los discos que albergan los datos más usados tienen un acceso más regular y por tanto aumenta su probabilidad de fallo. Este tipo de información, junto con otra información crítica, debe mantenerse distribuida y salvaguardada mediante el uso de *mirrors* o de organizaciones de pilas de discos. En esta fase también se estudian los índices y otras estructuras de organización física más apropiadas para optimizar el rendimiento.
7. Implementación: una vez realizados todos los pasos anteriores, debidamente documentados, se puede pasar a implementar el sistema. Los detalles de implementación dependen en gran medida del SGBD, de los lenguajes de programación de las aplicaciones, de una serie de estándares y protocolos que puedan estar utilizando tanto el SGBD como las aplicaciones y otros muchos factores. Cuanto más detallada y clara es la especificación inicial, más rápida será la implementación. Al contrario, si las primeras fases se han descuidado o han venido marcadas por las prisas, la fase de implementación se alargará y el sistema estará lleno de parches y de modificaciones sobre modificaciones, degradándose su eficacia y su seguridad. Además, las modificaciones, como en casi cualquier desarrollo de un proyecto, son más costosas cuanto más tarde se hagan.
8. Conversión y carga de datos y pruebas: en el caso de una migración se debe hacer una conversión de los datos existente en fuentes de información previas o anteriores a la base de datos que se acaba de implementar. Si no existe migración se deberá realizar una primera carga de datos ficticios o reales para realizar pruebas más completas del sistema.
9. Ajuste de rendimiento: Una vez el sistema comienza a estar operativo y se tienen datos (aunque en esta fase todavía pueden ser ficticios), se puede comenzar a ajustar el rendimiento, utilizando y simulando las cargas que se prevén en el funcionamiento normal del sistema.

Una vez que la base de datos adquiere un volumen de datos real importante (ya sea por migración de otra base de datos, la cual se debería diseñar cuidadosamente, o por inserción de nuevos datos), es cuando realmente se empieza a evaluar si las decisiones de diseño fueron correctas. Evidentemente, casi todos los sistemas se revisan y se amplían, pero estas modificaciones deben seguir los pasos anteriores a partir de aquél que haya motivado el cambio o extensión (requerimientos, conceptual, lógico, físico o de rendimiento). En estos cambios hay que tener en cuenta los costes a medio plazo, ya que pequeños retoques baratos a corto plazo pueden no resolver el problema a medio y largo plazo.

1.2.4. Aplicaciones

Hoy en día, la existencia de un sistema de información organizacional sin un SGBD detrás es un error, ya no sólo tecnológico sino económico. Incluso áreas que tenían unas particularidades muy especiales para las cuales hace unos años se dudaba del uso de SGBD, hoy han desarrollado una tecnología particular, o los SGBD generales son capaces de soportarlas sin demasiados problemas. De este modo, el rango de aplicaciones de las bases de datos a principios de este siglo XXI se ha ampliado de una manera importante. Así, se utilizan sistemas de gestión de bases de datos para los sistemas de información geográfica [Burrough et al. 1998], las bases de datos multimedia en [Furht & Marques 2000], las bases de datos médicas [Rondel et al. 1999], las bases de datos genéticas [Bishop 1999] y otras bases de datos científicas y estadísticas.

No obstante, siguen siendo las bases de datos de gestión de empresas y organizaciones las más importantes en volumen y número. Además, la perspectiva de las compañías suministradoras de SGBD tiende a proporcionar herramientas más globales e integradoras, en las que la organización se debe dedicar a personalizarlas para su área de negocio. Estamos hablando de los paquetes integrados de gestión, las ERP (*Enterprise Resource Planning*) y CRM (*Customer Relationship Management*). Aunque estos paquetes existen desde hace una o dos décadas y hay algunos muy populares (Baan, Navision, SAP), los paquetes globales perfectamente integrados con el SGBD, como el Oracle E-Business Suite, han sido acogidos de una manera espectacular por el mercado, con un 66% de crecimiento en el año 2000.

A medida que las empresas despliegan más sistemas web para la interacción con los clientes, los paquetes CRM se hacen más importantes. Según Dataquest (ahora Gartner Group, www.gartner.com), los paquetes CRM alcanzaron los 19,9 billones de dólares en 2000, un incremento del 28% sobre 1999. Además, se prevé unos crecimientos mayores en el futuro.

1.3. Líneas de Investigación Actuales y Futuro de las Bases de Datos

Muchas de las tecnologías que aparecen como futuras en libros clásicos de bases de datos las hemos incluido en los apartados anteriores. Las restricciones y las reglas de actividad, la tecnología orientada a objetos en las bases de datos (especialmente en la etapa de diseño), los datos multimedia, los almacenes de datos, la interrelación con la web o incluso las bases de datos distribuidas, son tecnologías maduras que se utilizan *hoy* en numerosas aplicaciones.

La tecnología de bases de datos ha ido automatizando los procesos que tienen lugar en los sistemas de información: recopilación, almacenamiento, consulta, reacción, análisis y toma de decisiones. El primer paso, el de recopilación de datos era antiguamente manual; hoy en día está altamente automatizado o semiautomatizado, el SGBD se encarga de manejar los datos hasta su lugar de almacenamiento, de manera masiva y eficiente. La recuperación de información básica o derivada simple (agregada, interrelacionada) experimentó su madurez a partir de los lenguajes de consulta declarativos, como SQL. El mantenimiento de la integridad y seguridad de la información ha sido uno de los aspectos donde el avance ha sido más significativo. Los sistemas impiden cualquier operación que contravenga esta integridad (actuación preventiva) y son capaces de tomar medidas compensatorias para mantenerla (actuación curativa). Esta última ha derivado hacia la posibilidad de que los sistemas sean reactivos, es decir, ciertos estados u operaciones de la base de datos hacen que el sistema tome unas medidas (alertas, informes, procesos, ...) que antes habían de dispararse manualmente. Esta función está muy automatizada gracias a la existencia de reglas de actividad ('disparadores'), con lo que, para muchas aplicaciones, no es necesario el uso de controladores humanos. Respecto al análisis, los sistemas de hoy en día permiten hacer consultas impensables hace unos años, gracias a la tecnología OLAP y los almacenes de datos, lo que permite tener información altamente sumarizada en tiempo real. Éste es el penúltimo paso para el uso final (y no transaccional) de la información, tomar decisiones acerca del contexto que esa información representa. Esta toma de decisiones empieza a semiautomatizarse, mediante la evolución de las herramientas de generación de modelos estadísticos a los sistemas de prospección de datos, minería de datos y simulación predictiva, que facilitan y automatizan gran parte del proceso de toma de decisiones, cambiando la filosofía de los sistemas de toma de decisión (DSS, *decision support systems*) tradicionales.

Toda esta automatización y la integración con otras tecnologías abre nuevas posibilidades y plantea nuevos retos. Se sigue investigando activamente en indización de datos, en el uso de inferencia para la recuperación de datos, en la compilación más eficiente de consultas, la ejecución de consultas en paralelo, en la integración de datos a partir de fuentes diversas, en el análisis del rendimiento, en la extensión del modelo transaccional para poder tratar transacciones largas y flujos de trabajo (transacciones que incluyen tanto pasos de un sistema

informático como de un humano), etc. La disponibilidad de almacenamiento masivo terciario ha motivado también el estudio de modelos de consulta para dispositivos de acceso muy lento.

Por poner un ejemplo de las áreas de interés en bases de datos, el "Call For Papers" de la conferencia internacional más importante en bases de datos, la conferencia del SIGMOD (Special Interest Group on the Management of Data), incluía aspectos tanto aspectos clásicos como más novedosos. Las áreas de la edición de 2001 (la edición de 2002 no incluía un listado de áreas) fueron las que se muestran en la siguiente tabla:

ÁREAS SOLICITADAS PARA LA CONFERENCIA SIGMOD 2001	
Access Methods and Data Structures	Imprecise and Uncertain Information
Active Databases	Information Retrieval and Databases
Application Issues (interfaces, models, architectures)	Industrial Challenges and Applications
Constraint Databases	Legacy Databases
Database Security	Object-Orientation and Database Systems
Data Models	Parallel Database Systems
Data Warehousing and On-Line Analytical Processing	Query Languages
Data Mining	Query Processing and Optimization
Database Performance and Benchmarking	Scientific and Statistical Databases
Database Programming Languages	Semi-structured Databases, World Wide Web and Databases
Distributed / Heterogeneous / Mobile Database Systems	Spatial and Temporal Databases
Image / Text / Multimedia Database Systems	User and Application Interfaces

De todas las áreas anteriores algunas reciben mayor importancia, ya sea por el número de ponencias enviadas como de las seleccionadas. Por ejemplo, respecto a las aplicaciones, la conferencia SIGMOD de 2001 animaba a realizar trabajos sobre lo que llamaban "aplicaciones de la nueva era, como p.ej. el comercio electrónico, las bibliotecas digitales y las gestión del conocimiento organizacional".

En otra de las conferencias más importantes en bases de datos, la PODS (*Principles of Database Systems*), ésta ligeramente más teórica, las áreas priorizadas en 2002 eran:

ÁREAS SOLICITADAS PARA LA CONFERENCIA PODS 2002		
Access Methods and Physical Design	Complexity and Performance Evaluation	Concurrency Control
Transaction Management	Integrity and Security	Data Models
Logic in Databases	Query Languages	Query Optimization
Database Programming Languages	Database Updates	Active Databases
Deductive Databases and Knowledge Bases	Object-oriented Databases	Multimedia Databases
Spatial and Temporal Databases	Constraint Databases	Real-time Databases
Distributed Databases	Data Integration and Interoperability	Views and Warehousing
Data Mining	Databases and Information Retrieval	Semistructured Data and XML
Information Processing on the Web	Databases in E-commerce	Databases and Workflows

Si nos fijamos en una conferencia más ingenieril, la ICDE (International Conference on Database Engineering) 2001 se concentra en aspectos más innovadores:

ÁREAS Y SUBÁREAS SOLICITADAS PARA LA CONFERENCIA ICDE 2001	
XML, METADATA, and SEMISTRUCTURED DATA - ontologies and tools for semantic data integration - querying and management of network directories - storing, processing, and warehousing XML data	ADVANCED INFO. SYSTEMS MIDDLEWARE - event processing and event-based infrastructure - push and pull technologies - transaction processing - parallel and distributed processing - integrity, security and fault tolerance - architects. and description langs. for large OLTP systems
DATABASE ENGINES & ENGINEERING - engine technology - system administration - usability - high performance and very large database engineering - performance evaluation of algorithms and systems	SCIENTIFIC AND ENGINEERING DATABASES - molecular and genomic databases - product data management - design process management - data exchange standards - fusion of heterogeneous data sources - federations of engineering and scientific DBs
QUERY PROCESSING - query processing with approximate results - query optimization for interactive queries - query optimization for first solution - incremental query processing - wide-area query processing	EXTREME DATABASES - small footprint databases - petabyte databases - real-time distributed processing - multimedia databases
DATA WAREHOUSES, DATA MINING, AND KDD - OLAP - information and knowledge discovery - loading and maintaining the Data Warehouse - Web warehousing - knowledge mining	E-COMMERCE and E-SERVICES - managing Web data and other semistructured data - database support for e-commerce - service creation, selection, and composition
WORKFLOW and PROCESS-ORIENTED SYSTEMS - workflow and process management systems - transactional workflow middleware - workflow-based virtual enterprises - scientific applications of process oriented systems - process-oriented infrastructure	EMERGING TRENDS - personalization - agent-based integration of heterogeneous data - database and agent support for ubiquitous computing
	SYSTEM APPLICATIONS AND EXPERIENCE

En el ICDE 2003, se han limitado a nombrar las áreas más importantes:

ÁREAS SOLICITADAS PARA LA CONFERENCIA ICDE 2003
Indexing, access methods, data structures
Query/trans. processing & optimization
Data Warehousing and OLAP
Mining Data, Text and Web
Semi-structured Data, Metadata & XML
WWW & Databases
Middleware, Workflow & Security
Database engines
Database applications & experiences
Distributed, Parallel, Mobile Databases
Temporal, Spatial, Scientific, Statistical, Biological Databases

Aunque ya no tan en boga como a mediados de los ochenta y en los noventa, existe también un gran interés en unificar, tanto en la teoría como en la *metodología* y en la *práctica*, los conceptos orientados a objetos con el modelo relacional. Los nuevos tipos de datos (imágenes, sonidos, documentos, gráficos, etc.) se tratan mejor como los métodos que los describen que los bytes que contienen. La aproximación orientada a objetos es todavía un área de investigación muy activa tanto en la industria como al nivel académico. En este sentido es de destacar el "tercer manifesto" [Date & Darwen 1998/2000], el cual actualiza los fundamentos de las bases de datos objeto-relacionales para el futuro.

El lento crecimiento de las bases de datos objetuales ha llevado a replantearse las nuevas características del paradigma orientado a objetos, en particular ciertos aspectos novedosos relacionados y su introducción en el mundo de bases de datos. Por ejemplo, han aparecido de nuevo conceptos prestados de la ingeniería del software como son las *bases de datos basados en componentes* [Dittrich & Geppert 1997] [Dittrich & Geppert 2001].

También está muy abierta la estandarización que seguirá el SQL. El ISO/IEC 13249 trabaja en SQL/MM para multimedia y paquetes de aplicación (incluyendo extensiones al SQL para minería de datos) y una serie de comités trabajan en extensiones independientes del SQL, que acaben posiblemente en un futuro SQL4, que de momento prefieren llamar SQL200*n*.

En resumen, es muy difícil estimar las tendencias a medio y largo plazo en un área tan cambiante como las bases de datos. Existen algunos trabajos sobre las perspectivas de las bases de datos para el principio del siglo XXI [Silberschatz et al. 1991] [Silberschatz & Zdnik 1996] [Silberschatz et al. 1996].

Desde mi opinión, los aspectos que pueden tener un impacto mayor a corto y medio plazo en el campo de las bases de datos son el tratamiento de información semi-estructurada (especialmente alrededor del XML y estándares relacionados), la extracción o descubrimiento de conocimiento en bases de datos (prospección/minería de datos) y el empuje definitivo de las bases de conocimiento, muy en relación con las dos áreas anteriores. Pasemos a ver con un poco más de detalle estas tres áreas.

1.3.1. Bases de Datos y el eXtended Mark-up Language (XML)

El Lenguaje de Marcas Generalizado Estandarizado (Standardized Generalized Markup Language, SGML) fue definido por ISO 8879 en 1986 mucho antes de que la web fuera una palabra familiar en todos los ámbitos. El SGML es el formato normalizado de documento estructurado más antiguo reconocido por el ISO. Aunque reconocido por el ISO en 1986, el SGML se inspira en los trabajos de Charles F. Goldfarb que desarrolló en IBM el lenguaje de marcas generalizado (GML, Generalized Markup Language) a partir de 1969. Por tanto, El SGML es un lenguaje de marcas generalizado y el HTML no es más que una instancia hipertexto del mismo, que se ha utilizado para publicar información a través de Internet, constituyendo hace unos años el World Wide Web (WWW).

Sin embargo el HTML es un lenguaje de *publicación* de información, donde el formato es tanto o más importante que el contenido. Aunque el HTML se hubiera podido considerar inicialmente como un lenguaje para el intercambio de información, el gran número de etiquetas, scripts y demás parafernalia con la que los fabricantes (especialmente los fabricantes de navegadores) han ido dotando al lenguaje ha hecho que la información contenida en una página web sea difícilmente disociable de su presentación y muy poco digerible con herramientas automáticas.

En los últimos años se ha visto necesaria la existencia de estándares de intercambio de información, con el objetivo de que las organizaciones puedan compartir su información de una manera más cómoda y, sobretodo, más automática y eficiente.

XML (eXtended Mark-up Language) es un lenguaje de marcas inspirado también en el SGML con el objetivo de permitir el intercambio de información de muy diversos tipos. Conjuntamente el XML permite tratar datos semi-estructurados de la web, organizar colecciones de datos de distintas fuentes y formatos, e intercambiar datos entre diferentes sitios/organizaciones.

En lo que concierne a las bases de datos, el XML permite integrar sistemas de información hasta ahora separados:

- Sistemas de información basados en documentos (o ficheros): tienen estructura irregular, anidados profundamente, utilizan tipos de datos relativamente simples y dan gran importancia al orden.
- Sistemas de información estructurados (p.ej. las bases de datos relacionales): tienen una estructura muy regular, son relativamente planos, utilizan tipos de datos relativamente complejos y dan poca importancia al orden.

Para todo esto, la sintaxis del XML es extremadamente simple. Consta exclusivamente de marcas (de apertura y cierre como HTML) y de atributos. Los datos son de tipo texto y se sitúan entre las marcas de apertura y las marcas de cierre. Aparte de los documentos XML existen DTD's (Document Type Definition) que opcionalmente pueden ir incluidos en el propio documento XML o en otro URL. Los documentos DTD tienen una sintaxis similar a la definición de una gramática regular. Por tanto, un documento XML puede estar bien formado o no, pero además, si está bien formado puede ser válido o no respecto a una DTD.

La posibilidad de definir DTDs, o más recientemente XML Schemas, para las más diversas aplicaciones es lo que hace al XML tan potente. En realidad no es más que un metalenguaje que se puede especializar para distintos usos utilizando DTDs apropiadas.

El hecho de que el XML permita expresar información de características y estructuras muy diversas no quiere decir que XML venga a sustituir a las bases de datos tradicionales. En primer lugar, XML no es una base de datos, no es tampoco un modelo de datos; XML es simplemente un lenguaje de marcas y un documento XML no es, en principio, nada más que un documento de texto.

Cuando por XML entendemos un documento XML, su DTD asociada, la interpretación del mismo respecto a una semántica, y todas las tecnologías que lo rodean, podemos llegar a comparar la tecnología XML con las bases de datos. Aparecen muchas cosas en común: la tecnología XML usa uno o más documentos (ficheros) para almacenar la información, define esquemas sobre la información (DTDs y otros lenguajes de esquema XML), tiene lenguajes de consulta específicos para recuperar la información requerida (XQL, XML-QL, QUILT, etc.), dispone de interfaces de programación (SAX, DOM), etc.

Pero aparecen muchas más cosas que lo diferencian. La tecnología XML carece, en parte por principio, de algunas de las siguientes características comunes en las bases de datos: almacenamiento y actualización eficientes, índices, seguridad, transacciones, integridad de datos, acceso concurrente, disparadores, etc. Por tanto es imposible pensar que XML se vaya a utilizar para las tareas transaccionales de una organización para las cuales sigue estando sobradamente más justificado utilizar una base de datos.

Aclarado ya este punto, pese a ser un lenguaje de marcas, XML permite representar la mayoría de modelos de datos existentes: de red, jerárquico, relacional, objetual, etc. Es decir, podemos representar la información contenida en una base de datos relacional en uno o más documentos XML.

Quizás, la interrelación entre XML y el modelo objetual es cada vez más fuerte [Chaudhri & Zicari 2000], especialmente en lenguajes de consulta. Éstos suelen inspirarse en el SQL y en otros lenguajes de consultas como el OQL, aunque también utilizan ciertas nociones de la programación funcional y de las técnicas de búsqueda de rutas de árboles de directorios, ya que un documento XML se puede ver como un árbol de términos, recorrible a través de caminos.

De momento existe el estándar XPath/XPointer para especificar rutas de búsqueda en un documento [W3C 1999]. Sin embargo, hasta hace poco no existía todavía un estándar de consulta; existen numerosas propuestas [Abiteboul et al. 1999]: XML-QL [Deutsch et al. 1998-1999], XQL [Robie et al. 1998], Quilt [Chamberlin et al. 2000]. Al igual que en SQL, la fuente y el resultado XML de las consultas son compatibles (documentos XML), permitiendo el encadenamiento (subconsultas).

Recientemente, el W3C XML Query Working Group (http://www.w3.org/XML/Query) ha publicado la primera versión del borrador del lenguaje XQuery, destinado a ser el lenguaje de consulta estándar sobre XML y que recoge muchas de las características de sus predecesores [W3C 2002].

Además de la anterior, existen numerosas propuestas relacionadas con el XML y las bases de datos:

- El SDQL (SGML Document Query Language) definido por el comité ISO10179 es un lenguaje de consulta sobre documentos SGML y está basado en las facilidades de identificación de nodos de SGML definidas en el estándar ISO 10744. SDQL no permite, sin embargo, controlar el acceso o modificar un repositorio de documentos SGML. Estas facilidades, usuales en lenguajes como SQL y OQL, están motivando el desarrollo de nuevos estándares, como los que vemos a continuación:
- El ISO TC184/SC4 está actualmente definiendo métodos para almacenar datos SGML como parte de sus estándares de representación e intercambio de datos de productos (ISO 10303).
- El ISO/IEC JTC1/WG4 N1946 desarrolla estándares para representar metainformación, accederla y modificarla en sistemas de información basados en documentos SGML.
- En el ISO/IEC JTC1/SC32 están definiendo maneras de almacenar documentos estructurados en bases de datos relacionales, como parte del SQL/MM.
- SQL/JRT. Surge como una mezcla del SQLJ Part 1 *(SQL Routines Using the Java Programming Language*) y SQLJ Part 2 *(SQL Types Using the Java Programming Language*) del SQLJ Group. Ahora su desarrollo está bajo los subcomités 331.1 y 331.2 del NCITS (National Committee for Information Technology Standards).
- SQL/XML contiene las especificaciones para representar datos relacionales SQL (específicamente filas y tablas de filas, así como vistas y resultado de consultas) en formato XML, y *viceversa*. También se están desarrollando especificaciones para representar esquemas SQL en XML, así como acciones (insert, update, delete) y la especificación de protocolos relacionados con el transporte de XML cuando se utiliza con SQL.
- El World Wide Web Consortium (W3C) ha desarrollado el Resource Description Format (RDF) bajo el RDF Working Group, con el objetivo de definir fuentes de datos con mayor nivel de abstracción.

1.3.2. Descubrimiento de Conocimiento en Bases de Datos

El aumento del volumen y variedad de información que se encuentra informatizada en bases de datos digitales ha crecido espectacularmente en la última década. Gran parte de esta información es histórica, es decir, representa transacciones o situaciones que se han producido. Aparte de su función de "memoria de la organización", la información histórica es útil para predecir la información futura.

El área de la extracción (semi-)automática de conocimiento de bases de datos se basa en la construcción de modelos a partir de la información existente, con el objetivo de extrapolar la información todavía desconocida. No es extraño, por tanto, que esta área ha adquirido recientemente una importancia científica y económica inusual.

Según [Fayyad et al. 1996], el descubrimiento de conocimiento a partir de Bases de Datos (KDD), del inglés *Knowledge Discovery from Databases*, se puede definir como el "*proceso no trivial de identificar patrones válidos, novedosos, potencialmente útiles y en última instancia comprensibles a partir de los datos*".

Las fases del KDD según [Fayyad et al. 1996] son:

- Determinar las fuentes de información que pueden ser útiles y dónde conseguirlas.
- Diseñar el esquema de un almacén de datos (Data Warehouse) que consiga unificar de manera operativa toda la información recogida.
- Implantación del almacén de datos que permita la "navegación" y visualización previa de sus datos, para discernir qué aspectos puede interesar que sean estudiados.
- Selección, limpieza y transformación de los datos que se van a analizar. La selección incluye tanto una criba o fusión horizontal (filas) como vertical (atributos).
- Seleccionar y aplicar el método de *minería de datos* apropiado.
- Interpretación, transformación y representación de los patrones extraídos.
- Difusión y uso del nuevo conocimiento.

Por tanto, la minería de datos (o prospección de datos) no es más que una fase de todo el proceso, la parte que genera los nuevos patrones, que si bien es la más difícil computacionalmente, se ve muy afectada por todas las fases anteriores, en especial la de preparación de datos [Pyle 1999].

El KDD se nutre de diferentes disciplinas y nace como interfaz entre ellas: estadística, sistemas de información / bases de datos, aprendizaje automático / inteligencia artificial, visualización de datos, computación paralela / distribuida e interfaces de lenguaje natural a bases de datos.

Es preciso distinguir las diferencias del KDD con algunas de estas disciplinas. Por ejemplo, existe una diferencia clara con los métodos estadísticos: la estadística se utiliza para validar o parametrizar un *modelo sugerido y preexistente*, no para generarlo. Además, los sistemas clásicos de estadística son difíciles de usar, sus modelos a veces difíciles de interpretar y no escalan al número de datos típicos en bases de datos.

La diferencia con "Análisis Inteligente de Datos" (IDA, del inglés *Intelligent Data Análisis*, véase p.ej. [Berthold & Hand 1999] es más sutil, ya que éste correspondía con el uso de técnicas de inteligencia artificial en el análisis de los datos. Es un término cuyo uso está decayendo frente al más común de minería de datos, aunque muchos paquetes integrales de KDD o de OLAP se suelen llamar "*business intelligence software*".

Por último, la diferencia más importante con el aprendizaje automático es que los datos usuales en KDD son poco habituales para algoritmos clásicos de aprendizaje automático: el número de registros (ejemplos) es muy grande (10^8-10^{12} bytes) y los datos altamente dimensionales (nº de columnas/atributos): 10^2-10^4. Otras características especiales de los datos son que éstos residen en el disco y no se pueden escanear múltiples veces, como requieren algunos algoritmos clásicos. Otro problema es que las bases de datos contienen sólo información positiva, mientras que algunos algoritmos funcionan mejor si tienen evidencia positiva y negativa. También existe el problema que algunas técnicas de muestreo no son compatibles con algoritmos no incrementales. Finalmente, los datos de una bases de datos suelen ser imperfectos (erróneos o faltantes).

A todo lo anterior debemos añadir una serie de restricciones. El usuario final no es un experto en aprendizaje automático ni en estadística. El usuario no puede perder mucho tiempo analizando los datos; en la industria un retraso en las decisiones más efectivas haría perder ventajas competitivas, en las aplicaciones científicas se perdería la oportunidad de investigar datos nunca analizados, bancos no cruzados, etc. Y al nivel personal, los usuarios no verían útiles estas herramientas si no les ayudan a liberarse del "*information overload*".

Todo esto hace que haya incluido el KDD como tecnología futura. Aunque existen paquetes disponibles actualmente y aplicaciones específicas donde el KDD puede obtener modelos predictivos para mejorar mucho la toma de decisiones, los requisitos anteriores de conjugar la identificación de patrones válidos, novedosos, útiles y comprensibles no es posible hoy en día en general.

En el momento presente, las técnicas de minería de datos más usuales [Witten & Frank 1999] [Hand et al. 2000] [Han & Kamber 2001] son aquéllas de aprendizaje automático y estadística: clasificación, regresión y segmentación, modelos de dependencia (modelos gráficos, estimación de densidad), sumarización (relación entre campos, asociaciones, visualización), detección (y modelado) de cambios y desviaciones.

Quizás el ejemplo más paradigmático de adaptación o de algoritmo específico para la minería de datos es el descubrimiento de reglas de asociaciones, muy útil en los problemas del estilo "cesta de la compra", en los cuales se intenta detectar aquellos items que suelen aparecer conjuntamente con suma frecuencia. A diferencia de otros problemas más complejos, el establecimiento de reglas de asociaciones se puede realizar eficientemente [Agrawal & Srikant 1994] [Adamo 2000].

Las áreas de aplicación del KDD son todas aquellas relacionadas con la toma de decisiones (banca, finanzas, seguros, márketing, políticas sanitarias/demográficas, ...) y con la investigación científica (medicina, astronomía, meteorología, psicología, ...). Entre las aplicaciones específicas podemos citar algunas de las más famosas [Berry & Linoff 2000]: identificar patrones de compra de los clientes, buscar asociaciones entre clientes y características demográficas, predecir respuesta a campañas de *mailing*, análisis de cestas de la compra, detectar patrones de uso fraudulento de tarjetas de crédito, identificar clientes leales, determinar gasto en tarjeta de créditos por grupos y un largo etcétera. Algunas otras aplicaciones todavía se hallan en fases incipientes: soporte al diseño de bases de datos [Blockeel & De Raedt 1996, 1998], ingeniería inversa, calidad de datos y optimización de consultas [Hsu & Knoblock 1996].

Una de las áreas que ha irrumpido con fuerza es la minería web (*Web Mining*) [Chang et al. 2001]. Ésta se refiere al proceso global de descubrir información o conocimiento potencialmente útil y previamente desconocido a partir de datos de la Web [Etzioni 1996]. Es obvio que la minería web comparte muchas técnicas con la minería de datos y combina objetivos y técnicas de distintas áreas: recuperación de la información, procesamiento del lenguaje natural, minería de datos, bases de datos, tecnología web y tecnología de agentes. La minería web se puede clasificar (no disjuntamente) en tres tipos [Kosala & Blockeel 2000]: minería del contenido web: se trata de extraer información del contenido de los documentos en la web; minería de la estructura web: se intenta descubrir un modelo a partir de la topología de enlaces de la red; minería de uso de la web: se intenta extraer información (hábitos, preferencias, etc. de los usuarios o contenidos y relevancia de documentos) a partir de las sesiones y comportamientos de los usuarios y navegantes.

Finalmente, en relación con el área de las bases de datos hay dos conceptos recientemente aparecidos en el campo del KDD que merecen una atención especial, por su interés conceptual. Se trata de las consultas inductivas y de las bases de datos inductivas.

En las primeras, el descubrimiento en bases de datos se ve como un proceso de consulta a una base de datos [Imielinski and Manilla 1996]. La situación actual de estos lenguajes es muy incipiente y se parece al desarrollo de lenguajes de consulta en los sesenta y setenta.

Una consulta inductiva o de búsqueda de patrones debe permitir al usuario restringir la búsqueda inductiva en los siguientes aspectos [Han et al. 1999]: la parte de la base de datos a ser minada (también llamada la vista minable o vista relevante) [Ng et al. 1998], el tipo de patrón/reglas a ser minado (también llamado restricciones del conocimiento), debe permitir el uso de cuantificadores estadísticos: representatividad (support), precisión (confidence/accuracy) y debe permitir expresar otras propiedades que el patrón debería cumplir (número y forma de las reglas, interés, novedad, etc.).

Una base de datos inductiva es un concepto más controvertido. Se trata de que el sistema induzca reglas a partir de los datos que contiene y utilice estas reglas para poder responder a las consultas de los usuarios. Como la inducción es un proceso hipotético aumenta la posibilidad de que la información extraída mediante consultas sea errónea. Para disponer de una base de datos inductiva es necesario previamente tener una base de datos deductiva, y también es necesario que el sistema sea capaz de trabajar con distintos grados de certeza. En realidad, llegados a este punto, de lo que se está hablando es de una base de conocimiento.

1.3.3. Bases de Datos Deductivas y Bases de Conocimiento

Aunque situemos esta área dentro de las líneas actuales y futuras, hay que aclarar que las bases de datos deductivas nacen con el propio modelo relacional, con lo que el concepto no es, en absoluto, novedoso. De

hecho, como hemos dicho, antes de la introducción del modelo relacional por Codd en 1970, Kuhns consideró el uso de la lógica para realizar consultas [Kuhns 1967]. De hecho, una relación puede verse como la extensión de un predicado lógico de primer orden. Con la aparición del lenguaje Prolog [Robinson 1979] [Kowalski 1979] y del Cálculo Relacional de Tuplas [Codd 1971], se plantea la posibilidad de definir un subconjunto de la lógica de primer orden que sea suficiente para representar la extensión de la base de datos y las consultas sobre ella. Nace el Datalog (véase p.ej. [Maier & Warren 1988]), que, aunque se restringe (respecto a Prolog) a constantes y variables en los argumentos de las relaciones/predicados, mantiene la recursividad. Proliferan las reuniones y conferencias sobre la conexión entre "Lógica y Bases de Datos", comenzando por la reunión celebrada en Toulouse en 1977 [Gallaire & Minker 1978]. Se crea la equivalencia entre base de datos relacional y teoría lógica de primer orden (con datos básicos), con lo que se plantea la definición de relaciones en función de otras, es decir, con definiciones intensionales en vez de extensionales. Esta visión extiende el concepto de *vista como predicado derivado deductivamente*. Nacen las bases de datos deductivas [Das 1992] que permiten consultar información derivada de la información introducida extensionalmente con anterioridad.

Esta equivalencia ha permitido portar al campo de las bases de datos resultados y desarrollos fundamentales del campo de la lógica, la programación lógica y la inteligencia artificial, como ha sido la comprobación de la integridad de la deducción automática, la asimilación de conocimiento del área de actualización y revisión de programas lógicos, la optimización de consultas a partir de la optimización y transformación de programas lógicos, las bases de datos con restricciones y un largo etcétera. Una muestra de la situación de bases de datos deductivas se puede encontrar en [Ramakrishnan & Ullman 1995] [Liu 1999].

Los sistemas basados en el conocimiento (o knowledge-based systems) han sido un área de investigación importante durante al menos los últimos veinte años, aunque recientemente la perspectiva ha cambiado ligeramente y se llaman *bases de conocimiento* [Levesque & Lakemeyer 2001], o incluso, desde mi punto de vista más inapropiadamente, sistemas de *bases de datos inteligentes* [Bertino et al. 2001]. Las bases de conocimiento derivan de los sistemas expertos de los setenta y los ochenta, pero con una perspectiva más informacional, a veces llamadas bases de datos expertas [Jeffery 1992]. Se trata de sistemas que almacenan reglas, además de datos. Estos sistemas son capaces además de aplicar estas reglas consecuentemente a las situaciones que se les plantean, en cierto modo, recordando las bases de datos activas. Su uso más extendido se centra en las herramientas de diagnóstico, en las que se le introducen al sistema una serie de síntomas o hechos y éste aplica las reglas para responder las causas y/o las consecuencias. La aplicación más directa puede parecer la ayuda al diagnóstico médico, pero se usan también en análisis de fallos en la industria.

En realidad los modernos DSS (Decision Support Systems) empresariales no son más que sistemas expertos sobre un dominio bastante particular, el entorno del negocio, que además tienen una interfaz muy fluida con la fuente de información organizacional o el almacén de datos. Este tipo de herramientas pueden significar un paso intermedio hacia las bases de conocimiento del futuro [Leondes 2000].

El mayor inconveniente de las bases de conocimiento (al igual que los sistemas expertos) es que el conocimiento (el conjunto de reglas) se ha de incorporar manualmente. Esto hace la creación de una base de conocimiento un proceso lento y costoso.

El hecho de que esta área se esté revitalizando recientemente se debe a la combinación de las bases de conocimiento con las bases de datos deductivas, activas e inductivas, las bases de datos temporales [Tansel et al. 1993], junto con la importación y exportación de ontologías [Bouguettaya 1999] (posiblemente en XML) para potenciar las posibilidades de los sistemas y facilitar su desarrollo. También es muy importante la incorporación de metainformación en estos sistemas, es decir, reglas que versen sobre el grado de veracidad de otros datos, su aplicabilidad, su contexto, etc. Por tanto, veremos en el futuro un gran avance de las bases de conocimiento, incorporando diferentes procesos de adquisición, extracción, recuperación e intercambio de información extensional (factual) o intensional (en forma de reglas, conceptos o entidades).

Finalmente, el salto a la industria y comercialización masiva de estos sistemas vendrá cuando se puedan combinar con los sistemas de gestión de bases de datos. El objetivo es superar las limitaciones de los SGBD (dificultades para tratar reglas declarativas, metainformación e inconsistencias) y de los sistemas de conocimiento (problemas de actualización de la información, de la asimilación de conocimiento, mantenimiento de la integridad, robustez y consulta). El interés está por tanto en combinar la teoría de los modelos relacionales y objetuales en su versión más amplia (incluyendo reglas deductivas y reactivas) junto con los sistemas de conocimientos generales, capaces de tratar información difusa, temporal, no fiable e inconsistente [Wagner 1998].

1.4. La Situación y Sociología de la Disciplina en el Área de Conocimiento

Las bases de datos es un área clásica y bien establecida dentro del área de conocimiento de lenguajes y sistemas informáticos. La UNESCO, en su clasificación de la ciencia, la engloba en el área 1203 (Ciencia de los ordenadores), con una entrada (1203/12) para las bases de datos (bajo la terminología "bancos de datos") y otra entrada (1203/18) para los sistemas de información, su diseño y componentes, si bien estamos hablando de una clasificación de los años 1985-86. Al ser un área bastante central de la informática, las bases de datos se interrelacionan con prácticamente todas las otras áreas, especialmente las de lenguajes y sistemas informáticos.

Inicialmente el desarrollo de las bases de datos estaba muy ligado al desarrollo de las estructuras de datos y los sistemas operativos. Para la fundamentación teórica, las bases de datos se nutren de la lógica de primer orden, la teoría de grafos, el álgebra, las lógicas para la concurrencia y otras áreas de la matemática discreta.

El área se engloba dentro de un marco más general (y también más difuso) que corresponde a los sistemas de información, donde la importancia se centra en esta última (la información) y en su adecuación con la realidad (su calidad), para que esté al servicio a la organización o contexto que está modelando. En este sentido se diferencia del punto de vista del proceso, en el que el objetivo es automatizar o facilitar procesos, mediante el desarrollo de aplicaciones que se ajusten a unas ciertas especificaciones de funcionamiento. Evidentemente a los procesos siempre se les asocian datos y esta dualidad (tan bien representada, p.ej., en el paradigma orientado a objetos) es indisociable a la naturaleza de la informática, si ésta se ve como la ciencia del *procesamiento* de la *información*.

Como toda área científica, las bases de datos tienen su sociología, entendida ésta como una serie de organizaciones, autoridades, congresos y revistas respetadas y clásicas en la disciplina. Pasemos a ver algunas de las más importantes.

1.4.1. Organizaciones

Un extracto de las organizaciones más importantes en la disciplina de bases de datos se muestra a continuación:

- ACM/SIGMOD - Special Interest Group on the Management Of Data (http://www.acm.org/sigmod/): El grupo de interés especial de la ACM dedicado a la gestión de datos se preocupa de los principios, técnicas y aplicaciones de los sistemas de gestión de bases de datos y de la tecnología de gestión de datos. Sus miembros incluyen desarrolladores de software, investigadores de los mundos académico e industrial, practicantes, usuarios y estudiantes. SIGMOD patrocina la conferencia anual SIGMOD/PODS, publica revistas, colecciones de literatura y publicaciones, así como otros materiales en papel o digital.
- IEEE TCDE - Technical Committee on Data Engineering (http://www.ccs.neu.edu/groups/IEEE/tcde/index.html). El comité técnico de ingeniería de datos de la IEEE Computer Society se preocupa del papel de los datos en el diseño, desarrollo, gestión y utilización de sistemas de información. Los aspectos de mayor interés incluyen el diseño de bases de datos, el conocimiento de los datos y su procesamiento, los lenguajes para describir los datos, definir el acceso y la manipulación de bases de datos, las estrategias y mecanismos para el acceso a los datos, la seguridad y control de integridad, y los sistemas distribuidos. El TCDE patrocina la International Conference on Data Engineering (ICDE) y publica trimestralmetne el *Data Engineering Bulletin*.
- VLDB Endowment - Very Large Data Base Endowment Inc. (http://www.vldb.org/). Es una fundación estadounidense sin ánimo de lucro con el propósito de promover e intercambiar trabajos eruditos en bases de datos y áreas relacionadas en todo el mundo. Sus actividades principales son la organización de las conferencias VLDB y la publicación de la VLDB Journal, en colaboración con Springer-Verlag.
- EDBT - Extending Database Technology (www.edbt.org). La fundación EDBT es una organización apolítica y sin ánimo de lucro con el objetivo de promover y apoyar el progreso en los campos de las bases de datos y la tecnología y aplicaciones de los sistemas de información. Su mayor actividad es la promoción de la International Conference on Extending Database Technology (EDBT), que se celebra bienalmente desde 1988. La fundación también promueve escuelas de verano internacionales, desde 1991.

- ODMG - Object Database Management Group (www.odmg.org) Como hemos visto antes, es un grupo de fabricantes y usuarios de bases de datos que desarrolla estándares para los sistemas de gestión de bases de datos orientadas a objetos.
- ACM/SIGKDD - Special Interest Group on Knowledge Discovery in Data and Data Mining (http://www.acm.org/sigkdd/). La tarea principal del SIGKDD es proporcionar un foro para el avance y la adopción de la "ciencia" del descubrimiento de conocimiento y la minería de datos. Para ellos, el SIGKDD fomenta la investigación básica en KDD (a través de conferencias de investigación anuales, un boletín y otras actividades relacionadas), la adopción de "estándares" en el mercado sobre terminología, evaluación y metodología, así como la educación interdisciplinar entre investigadores, practicantes y usuarios del KDD. Las actividades concretas del SIGKDD incluyen la conferencia anual de Knowledge Discovery and Data Mining y el boletín SIGKDD Explorations.

En menor medida, existen otras organizaciones, como la ACM/SIGMIS - Special Interest Group on Management Information Systems (http://www.acm.org/sigmis/) o el Transaction Processing Performance Council (http://www.tpc.org/), la AIS (Association for Information Systems, http://www.aisnet.org/), la AITP (Association of Information Technology Professionals, http://www.aitp.org/, antes DPMA), la SIM (Society for Information Management, http://www.simnet.org/), la IACIS (International Association for Computer Information Systems, http://www.iacis.org/) y la IAIM (International Academy for Information Management, http://www.iaim.org/).

1.4.2. Congresos

El número de congresos relacionados con las bases de datos es ingente y continúa creciendo. Atendiendo a las organizaciones que figuran detrás de los congresos y de su publicación, podemos citar algunos pocos de los más importantes:

- SIGMOD/PODS Conferences (http://www.acm.org/sigmod/conferences/index.html) Las conferencias PODS (Principles Of Database Systems) han sido el foro principal para que investigadores, practicantes, desarrolladores y usuarios de bases de datos presenten su trabajo y discutan los aspectos críticos y sus visiones sobre la tecnología, aplicaciones y técnicas punteras en bases de datos.
- VLDB Conferences (http://www.vldb.org/dblp/db/conf/vldb/index.html). Realizadas por el grupo VLDB mencionado anteriormente, las conferencias VLDB constituyen una de las citas más importantes (y quizás la más internacional) para la diseminación periódica de los resultados de investigación y desarrollo en el campo de la gestión de bases de datos.
- EDBT Conferences (www.edbt.org). Organizadas por el grupo EDBT, se realizan cada dos años y tienen un marcado carácter europeo. La de 1998 se organizó en Valencia [Schek et al. 1998].
- ICDE (International Conference on Data Engineering). El TCDE (http://www.ccs.neu.edu/groups/IEEE/tcde/index.html) patrocina esta conferencia, que suele tener un contenido más tecnológico y actual que las anteriores.

Otras conferencias importantes son la "International Conference on Database Theory", la "ACM SIGMOD Conference on Management of Data" o la "International Conference on Data Mining And Knowledge Discovery". Además, cabe citar un congreso relevante al nivel español:

- JISBD (Jornadas de Ingeniería del Software y Bases de Datos). Antes llamadas "Jornadas de Investigación y Docencia en Bases de Datos" y desde 1999 conjuntas con las jornadas de ingeniería del software, permiten un intercambio de resultados, tanto docentes como investigadores, en el campo de las bases de datos en España.

1.4.3. Revistas

Al igual que los congresos, existen más de una centena de revistas internacionales sobre la disciplina. Mostramos exclusivamente las revistas incluidas en el ISI SCI (las más referenciadas).

- ACM Transactions on Database Systems (http://www.acm.org/tods/). Publica: Association for Computing Machinery. La revista Transactions On Database Systems (TODS) publica artículos de

carácter archivístico en el área de las bases de datos y disciplinas afines. La mayoría de los artículos que han aparecido en TODS abordan los fundamentos lógicos y técnicos de la gestión de datos.

- Data Mining And Knowledge Discovery, (http://www.wkap.nl/journalhome.htm/1384-5810). Publica: Kluwer Academic Publishers. Esta revista publica artículos sobre todos los aspectos del descubrimiento de conocimiento en bases de datos y en métodos de minería de datos para extraer representaciones de alto nivel (patrones y modelos) a partir de los datos.
- IEEE Transactions on Knowledge and Data Engineering (http://www.computer.org/tkde/). Publica: IEEE Computer. Está diseñada a informar a los investigadores, desarrolladores, gestores, analistas estratégicos, usuarios y otros interesados en actividades actuales en el área de la ingeniería del conocimiento y de los datos.
- ACM Transactions on Information Systems (http://www.acm.org/tois/). Publica: Association for Computing Machinery. Esta revista considera el diseño, rendimiento y evaluación de sistemas informáticos que facilitan la presentación de la información en una variedad de medios, así como las tecnologías subyacentes que soportan estos sistemas.
- Information Systems (http://www.elsevier.nl). Publica: Pergamon-Elsevier Science. *Information Systems* publica artículos relativos al diseño e implementación de lenguajes, modelos de datos, software y hardware para sistemas de información.

También mostramos algunas más orientadas a la gestión empresarial:

- IEEE IT Professional (http://www.computer.org/itpro/): Para desarrolladores y gestores de sistemas de información empresarial. IT Professional (ITPro) explica la tecnología que ayuda a construir y gestionar un sistema de información en la actualidad y proporciona consejos sobre las tendencias tecnológicas que pueden marcar el área empresarial en los próximos años.
- Datamation, versión española (http://www.mcediciones.es/datamation/). Es una revista más genérica, con orientación para el entorno empresarial (paquetes de gestión, comercio electrónico, etc.)

ANEXO 1. REFERENCIAS

[Abiteboul et al. 1999] Abiteboul, S.; Buneman, P.; Suciu, D. "Data on the Web: from Relations to Semistructured Data and XML" Morgan Kaufmann, 1999

[Adamo 2000] Adamo, J.M. “Data Mining for Association Rules and Sequential Patterns” Springer 2000.

[Agrawal & Srikant 1994] Agrawal and Srikant. "Fast Algorithms for Mining Association Rules." Very Large Databases (VLDB) 1994: 487-499. También reimpreso en [Stonebraker & Hellerstein 1998].

[ANSI 1992] ANSI X3.135-1992, "Database Language SQL"

[ANSI/ISO/IEC 1999] ANSI/ISO/IEC “A/I 9075-1:1999. Database Language SQL - Part 1: Framework (for SQL3)”, “A/I 9075-2:1999. Database Language SQL - Part 2 SQL Foundation (for SQL 3)”, “A/I 9075-3:1999. Database Language SQL - Part 3: SQL Call Level Interface (for SQL 3)”, “A/I 9075-4:1999 Database Language SQL - Part 4: Persistent Stored Modules (for SQL 3)”, “A/I 9075-5:1999 Database Language SQL - Part 5: SQL Language Bindings (for SQL 3)”.

[ANSI/ISO/IEC 2000] ANSI/ISO “A/I 9075-10:2000, Information Systems — Database Languages — SQL — Part 10: Object Language Bindings (SQL/OLB)”.

[ANSI/SPARC 1975] ANSI/SPARC “Study group on data base management systems: interim report”, FDT 7:2, ACM, New York, 1975.

[Astrahan et al. 1976] Astrahan, M., et al: “System R : Relational Approach to Database Management.” ACM Transactions on Database Systems (TODS) 1(2): 97-137, 1976 . También reimpreso en [Stonebraker & Hellerstein 1998].

[Badia et al. 1995] Badia, A.; Van Gucht, D.; Gyssens, M. “Querying with generalized quantifiers” Applications of Logic Databases, ed. R.Ramakrishman, Kluwer Academic, 1995.

[Beeri & Bernstein 1979] Beeri, C.; Bernstein, P.A. “Computational Problems Related to the design of Normal Form Relations Schemes”, ACM Transactions on Database Systems 4:1, March 1979.

[Berry & Linoff 2000] Berry M.J.A.; Linoff, G.S. “Mastering Data Mining” Wiley 2000.

[Berthold & Hand 1999] Berthold, M.; Hand, D.J. (ed.) “Intelligent Data Analysis. An Introduction” Springer 1999. (Nueva edición a aparecer en 2002).

[Bertino et al. 2001] Bertino, E.; Catania, B.; Zarri, G.P. “Intelligent Database Systems” Addison-Wesley, 2001.

[Bishop 1999] Bishop, M. J. (Editor) “Genetic Databases” Academic Press, 1999.

[Blockeel & De Raedt 1996] Blockeel, H. and De Raedt, L. “Inductive database design”, Proceedings of Foundations of Intelligent Systems, Proc. of the 9th International Symposium on Methodologies for Intelligent Systems, Lecture Notes in Artificial Intelligence, Vol. 1079, 1996, pp. 376-385.

[Blockeel & De Raedt 1998] Blockeel, H. and De Raedt, L. “IsIdd: An interactive system for inductive database design”, Appl. Artif. Intell. 12 (1998), no. 5, 385-421.

[Booch 1991] Booch G., Object-oriented Design with Applications, Benjamin/Cummings, Redwood City, CA, 1991

[Booch 1994] Booch, G.: Object-Oriented Analysis and Design with Applications, Benjamin/Cummings, 1994

[Booch et al. 1997] Booch G., Jacobson I., Rumbaugh J et. al., “The Unified Modeling Language for Object-Oriented Development Version 1.0, UML Notation Guide, UML Summary, UML Semantics”, Rational Software Corporation, January 1997 and the UML 1.1 update of Sept. 1997. http://www.rational.com/uml

[Booch et al. 1999] Booch G., Jacobson I., Rumbaugh J., “El Lenguaje Unificado de Modelado”, Addison Wesley 1999.

[Bouguettaya 1999] Bouguettaya, Athman (Editor), “Ontologies and Databases”, Kluwer Academic Publishers, 1999.

[Boyce & Chamberlin 1974] Boyce, R.; Chamberlin, D. “SEQUEL: a structured English query language” ACM SIGMOD Conf. On the Management of Data, 1974.

[Burrough et al. 1998] Burrough, Peter A.; McDonnell, Rachael; McDonnell, Rachel A. “Principles of Geographical Information Systems” 2nd edition, Oxford University Press, 1998.

[Burton 2001] Burton, B. “IBM Attempts to Buy DBMS Market Share” Gartner Group (www.gartner.com), 30 April 2001.

[Cattell & Barry 2000] Cattell, R.G.G. “The Object Data Standard: ODMG 3.0”, Morgan Kaufmann, 2000.

[Cattell 1994] Cattell, R. “Object Data Management. Object-Oriented and Extended Relational Database Systems”. Addison-Wesley, Readings, Massachusetts, 1994.

[Cattell 1997] Cattell, R. “Object Database Standard, ODMG 2.0”, 1997.

[Celma et al. 1997] Celma, M; Casamayor, J. C; Mota, L. “Bases de datos relacionales” Servicio de Publicaciones de la UPV, SPUPV nº 97.767, 1997

[Chamberlin 1998] Chamberlin, D.D.; et al: "A History and Evaluation of System R" en [Stonebraker & Hellerstein 1998].

[Chamberlin et al. 1976] Chamberlin, D.; Astrahan, M.; Eswaran, K.; Griffiths, P.; Lorie, R.; Mehl, J.; Reisner, P.; Wade, B. “Sequel 2: A unified approach to data definition, manipulation, and control” IBM Journal of Research and Development, 20(6): 560-575, 1976.

[Chamberlin et al. 2000] Don Chamberlin, Jonathan Robbie, Daniela Florescu, "Quilt; An XML Query Language for Heterogeneous Data Source" , Proc. of the workshop on Web and databases (WebDb), in conj. with SIGMOD'00 , Addison-Wesley ,Dallas, Texas , May , 2000.

[Chandra & Merlin 1977] Chandra, A.; Merlin, P. “Optimal implementaion of conjunctive queries in relational databases” Annual ACM SIGACT Symposium on Theory of Computing, 1977.

[Chang et al. 2001] Chang, G.; Healey, M.J.; McHugh, J.A.M.; Wang, J.T.L. “Mining the World Wide Web. An Information Search Approach” Kluwer Academic Publishers, 2001.

[Chaudhri & Zicari 2000] Chaudhri, Akmal B.; Zicari, Roberto “Succeeding with Object Databases: A Practical Look at Today's. Implementations with Java and XML” John Wiley & Sons 2000.

[Checkland 1990] Checkland, P. “Systems thinking, systems practice”. Wiley, 1990.

[Chen 1976] Chen, P.P. “The entity-relationshipo model —toward a unified view of data” ACM Transactions on Database Systems, 1(1): 9-36, 1976.

[Chorafas 1994] Chorafas, D.N. “Intelligent Multimedia Database” Prentice Hall, 1994.

[CODASYL 1968] COmmitteee on DAta SYstems and Languages (CODASYL) “A survey of generalized data base management systems”, CODASYL, ACM, 1968.

[CODASYL 1971] Codasyl Data Base Task Group April 1971 Report, ACM, New York, 1971.

[CODASYL 1978] COmmitteee on DAta SYstems and Languages (CODASYL) “Data Description Language Committee”, Journal of Development, CODASYL, 1978.

[Codd 1970] Codd, E.F. “A Relational Model for Large Shared Data Banks” Communications of the ACM, 13:6, pp. 377-387, 1970. También reimpreso en [Stonebraker & Hellerstein 1998].

[Codd 1971] Codd, E.F. “A Data Base sublanguage Founded on the Relational Calculus” Proc. of ACM SIGFIDET Workshop on Data Description, Access and Control, San Diego, California, 1971.

[Codd 1971] Codd, E.F. “A Data Base sublanguage Founded on the Relational Calculus” Proc. of ACM SIGFIDET Workshop on Data Description, Access and Control, San Diego, California, 1971.

[Codd 1972a] Codd, E. “Further Normalization of the data base relational model” Data Base Systems, R. Rustin (ed), Prentice Hall, 1972.

[Codd 1972b] Codd, E. “Relational completeness of data base sub-languages” in Data Base Systems, R. Rustin (ed.), Prentice Hall, 1972.

[Codd 1982] Codd, E.F. “Relational Database: A Practiacal Fioundation for Productivity” Communications of ACM, Vol. 25, no. 2, February, 1992.

[Codd 1985a] Codd, E.F. “Is Your DBMS Really Relational?”, Computerworld, 15 de octubre, 1985.

[Codd 1985b] Codd, E.F. “Does Your DMBS Run by the Rules?” Computerworld, 15 de octubre, 1985.

[Codd 1988a] Codd, E.F. “Why Choose a Relational DBMS” Relational Institute, 1988

[Codd 1990] Codd, E.F. “The Relational Model for Database Management. Version 2” Reading, Massachusets, Addison Wesley, 1990.

[Codd 1990] Codd, E.F. “The Relational Model for Database Management. Version 2” Reading, Massachusets, Addison Wesley, 1990.

[Connolly & Begg 2000] Connolly, T.; Begg, C. “Database Solutions. A step-by-step guide to building databases”, Addison-Wesley 2000.

[Connolly et al. 1998] Connolly, T.; Begg, C.; Strachan, A. “Database Systems: A Practical Approach to Design, Implementation, and Management” Addison-Wesley, 1998.

[Connolly & Begg 1999] Connolly, T.; Begg, C.; “Database Systems: A Practical Approach to Design, Implementation, and Management” 2nd Edition, Addison-Wesley, 1999.

[Connolly et al. 2001] Connolly, T.; Begg, C., Begg, C.E.; “Database Systems: A Practical Approach to Design, Implementation, and Management” 3rd Edition, Addison-Wesley, 2001.

[Constantine & Yourdon 1978] Larry L. Constantine & Edward Yourdon, “Structured Design: Fundamentals of a Discipline for Computer Program and Systems”, New York, Yourdon Press, 1978.

[Copeland & Maier 1984] Copeland, D.; Maier “Making SMALLTALK a database systme” ACM SIGMOD Conf. on the Management of Data, 1984.

[Darwen & Date 1997] by Hugh Darwen (Contributor), Chris J. Date "A Guide to the Sql Standard : A User's Guide to the Standard Database Language Sql", Addison Wesley, 1997

[Das 1992] Das, K. "Deductive Databases and Logic Programming" Addison Wesley, 1992.

[Date & Darwen 1998] Date, C. J.; Darwen, H. "Foundation for Object/Relational Databases : The Third Manifesto" Addison-Wesley-Longman 1998.

[Date & Darwen 2000] Date, C. J.; Darwen, H. "Foundation for Future Database Systems: The Third Manifesto" 2nd Edition of [Date & Darwen 1998], Addison-Wesley 2000.

[Date 1981] Date, C.J. "An Introduction to Database Systems", Addison-Wesley, 1981.

[Date 1982] Date, C.J. "Relational Databae: Some Topics for Investigation" IBM Sta. Teresa Programming Center, 17 may 1982.

[Date 1983] Date, C.J. "An Introduction to Database Systems (Vol. 2)", Addison-Wesley, 1983.

[Date 1984] Date, C.J. "A critique of the SQL database language" ACM SIGMOD Record, 14(3): 8-54, 1984.

[Date 1988] Date, C.J. "Why Relational" Codd and Date International, 1988.

[Date 1993a] Date, C.J. "Introducción a los Sistemas de Bases de Datos, Quinta Edición" Addison-Wesley Iberoamericana, 1993

[Date 1993b] Date, C.J. "A Guide to the SQL Standard" 3rd Edition, Addison-Wesley, 1993

[Date 1995] Date, C.J. "An Introduction to Database Systems, Sixth Edition" Addison-Wesley 1995

[Date 1999] Date, C.J. "An Introduction to Database Systems", 7th Edition (25th Anniversary Edition), Addison-Wesley, 1999.

[Date 2001a] Date, C. J. "The Database Relational Model : A Retrospective Review and Analysis: A Historical Account and Assessment of E. F. Codd's Contribution to the Field of", Addison-Wesley 2001.

[Date 2001b] Date, C.J. "Introducción a los Sistemas de Bases de Datos", 7ª Edición, Prentice Hall, 2001.

[de Miguel & Piattini 1997] de Miguel, A.; Piattini, M. "Fundamentos y modelos de bases de datos" Ra-Ma, Madrid, España, 1997.

[de Miguel & Piattini 1999] de Miguel, A.; Piattini, M. "Fundamentos y modelos de Bases de Datos", 2ª Ed., Ra-Ma, 1999.

[de Miguel et al. 1999] de Miguel, A.; Piattini, M.; Marcos, E. "Diseño de bases de datos" Ra-Ma, Madrid, España, 1999.

[de Miguel et al. 2000] de Miguel, A.; Martínez, P.; Castro, E.; Cavero, J.M.; Cuadra, D.; Iglesias, A.M.; Nieto, C. "Diseño de bases de datos. Problemas resueltos" Ra-Ma, Madrid, España, 2000.

[Delobel & Adiba 1985] Delobel, C.; Adiba, M. "Relational Database Systems" North-Holland 1985.

[Deutsch et al. 1998-1999] Deutsch, A.; Fernández, M.; Florescu, D.; Levy, A.; Suciu, D. "A Query Language for XML" en http://www.research.att.com/~mff/files/final.html.

[DeWitt & Gray 1992] DeWitt, D.J. and Gray, J. "Parallel Database Systems: The Future of High Performance Database Systems". Communications of the ACM 35(6): 85-98, 1992. También reimpreso en [Stonebraker & Hellerstein 1998].

[DeWitt et al. 1990] DeWitt, D.J.; Ghandeharizadeh, S.; Schneider, D.A.; Bricker, A.; Hsiao, H.I.; Rasmussen, R. "The Gamma Database Machine Project." TKDE 2(1): 44-62, 1990. También reimpreso en [Stonebraker & Hellerstein 1998].

[Dittrich & Geppert 1997] Dittrich, K.R.; Geppert, A. "Object-oriented DBMS and beyond" *in Procs. Of International Conference on Current Trends in Theory and Practice of Informatics*, pp. 275-294, 1997.

[Dittrich & Geppert 2001] Dittrich, K.R.; Geppert, A. *Component Database Systems*, Morgan Kaufmann Publishers, 2001.

[Eaglestone 2000] Eaglestone, Barry "Object Databases : An Introduction", McGraw-Hill 2000.

[Elmasri & Navathe 1994] Elmasri, R.; Navathe, S.B. "Fundamentals of Database Systems" 2nd Edition, Addison-Wesley, 1994.

[Elmasri & Navathe 1997] Elmasri, R.; Navathe, S.B. "Sistemas de Bases Datos. Conceptos Fundamentales" 2nd. Edition, Addison-Wesley Iberoamericana, 1997.

[Elmasri & Navathe 2000] Elmasri, R.; Navathe, S.B. "Fundamentals of Database Systems" 3rd Edition, Addison-Wesley, 2000.

[Etzioni 1996] Etzioni, O. "The World-Wide Web. Quagmire or Gold Mine" Communications of the ACM, November 1996, Vol. 39, no.11.

[Everest 1986] Everest, G.C. "Database Management. Objetives, System Functions, and Administration" McGraw-Hill Book Company, 1986.

[Fayyad et al. 1996] Fayyad, U.M.; Piatetskiy-Shapiro, G.; Smith, P.; Ramasasmy, U. "Advances in Knowledge Discovery and Data Mining", AAAI Press / MIT Press, 1996.

[Feiler 1999] Feiler, J. "Database-Driven Web Sites", Morgan Kaufmann, 1999.

[Fry & Sibley 1976] Fry, J.; Sibley, E. "Evolution of data-base management systems" ACM Computing Surveys, 8(1): 7-42, 1976.

[Furht & Marques 2000] Furht, Borko; Marques, Oge "Handbook of Video Databases : Design and Applications" CRC Press, December 2000.

[Gallaire & Minker 1978] Gallaire, H.; Minker, J. "Logic and Databases" Plenum Press 1978.

[Garcia-Molina et al. 2000] Garcia-Molina, H.; Ullman, J.D. Widom, J.; "Database System Implementation" Prentice Hall 2000.

[Gardarin & Valduriez 1987] Gardarin, G.; Valduriez P. "Bases de Données Relationnelles. Analyse et Comparaison de Systèmes" 2eme Édition, Eyrolles 1987.

[Gardarin 1987] Gardarin, G. "Bases de Datos" Paraninfo 1987 (ed. original "Bases de Données" Eyrolles 1983).

[Gardarin 1988] Gardarin, G. "Bases de Données. Les systèmes et leurs langages" 5eme Édition, Eyrolles 1988.

[Gardarin 1993] Gardarin, G. "Dominar las Bases de Datos: modelos y lenguajes", Ediciones Gestión, 2000.

[Graham 2002] Graham, C. "DBMS Software Market: Flat but Not Calm" Gartner Group (www.gartner.com), 6 May 2002

[Gray & Reuter 1992] Gray, J.; Reuter, A. "Transaction Processing: Concepts and Techniques" Morgan Kaufmann, 1992.

[Gray 1981] Gray, J.N. "The Transaction Concept: Virtues and Limitations" in *Very Large DataBases*, 1981.

[Gray 2001] Gray, James N. "Database systems: A Textbook Case of Research Paying Off" Microsoft Research, http://www.cs.washington.edu/homes/lazowska/cra/database.html.

[Gray et al. 1997] Gray, J.N. et al.: "Data Cube: A Relational Aggregation Operator Generalizing Group-by, Cross-Tab, and Sub Totals." Data Mining and Knowledge Discovery 1(1): 29-53 (1997) . También reimpreso en [Stonebraker & Hellerstein 1998].

[Groff & Weinberg 1998] Groff, J.R.; Weinberg, P.N. "Guía LAN TIMES de SQL2" McGraw-Hill 1998.

[Gulutzan & Pelzer 1999] Gulutzan, P.; Pelzer, T. "SQL-99 Complete, Really" CMP Books 1999.

[Hammer & MacLeod 1981] Hammer, M.; McLeod, D. "Database description with SDM: a semantic database model approach" ACM Transactions on Database Systems, vol. 6, nº3, 1981.

[Han et al. 1999] Han, J.; Lakshmanan, V.S.; Ng, R.T. "Constraint-Based, Multidimensional Data Mining" Computer, Vol. 32, nº8, 1999.

[Han & Kamber 2001] Han, J.; Kamber, M. "Data Mining: Concepts and Techniques" Morgan Kaufmann, 2001.

[Hand et al. 2000] Hand, D.J.; Mannila, H. and Smyth, P. "Principles of Data Mining", The MIT Press, 2000.

[Hansen & Hansen 1997] Hansen, G.W.; Hansen, J.V.; Diseño y Administración de Bases de Datos (2ª edición). Prentice Hall, 1997.

[Hernández et al. 2000] Hernández, M. J.; Viescas, J.L.; Celko, J. "SQL Queries for Mere Mortals: A Hands-On Guide to Data Manipulation in SQL", Addison-Wesley, 2000.

[Hsu & Knoblock 1996] Hsu, C-N.; Knoblock, C.A. "Using Inductive Learning to Generate Rules for Semantic Query Optimization" in Fayyad, U.M.; Piatetsky-Shapiro, G.; Smyth, P. Advances in Knowledge Discovery and Data Mining, AAAI Press/The MIT Press, 425-445, 1996

[Hull & King 1987] Hull, R.; King, R. "Semantics database modelling: survey, applications and research", ACM Computing Survey, vol. 19, nº3, 1987.

[IDG 2001] "Oracle extends lead in database sales" IDG News Service 05/23/01. http://www.itworld.com/AppDev/119/DataquestOracleexte413/

[Imielinski and Manilla 1996] Imielinski, T.; Mannila, H. "A database perspective on knowledge discovery" Communications of the ACM, 39(11):58-64, 1996.

[ISO/IEC 1992] ISO/IEC 9075:1992, "Information Technology --- Database Languages --- SQL"

[ISO/IEC 1994] ISO/IEC 9075:1992 "Technical Corrigendum 1:1994" to ISO/IEC 9075:1992

[King 1981] King, J. "Quist: a system for semantic query optimization in relational databases" in Proc. Of the Conf. on Very Large Databases, 1981.

[Klug 1982] Klug, A. "Equivalence of relational algebra and relational calculus query languages having aggregate functions" Journal of the ACM, 29(39):699-717, 1982.

[Kosala & Blockeel 2000] Kosala, R.; Blockeel, H. "Web Mining Research: A Survey" ACM SIGKDD Explorations, Newsletter of the ACM Special Interest Group on Knowledge Discovery and Data Mining, June 2000, Vol. 2, nº1, pp. 1-15.

[Kowalski 1979] Kowalski, R. "Logic for Problem Solving" North-Holland, Amsterdam, 1979.

[Kuhns 1967] Kuhns, J. "Logical aspects of question answering by computer" Technical report, Rand Corporation, RM-5428-Pr, 1967.

[Lamb et al. 1991] Lamb, C.; Landis, G.; Orenstein, J.; Weinreb, D. "The ObjectStore System." CACM 34(10): 50-63, 1991. También reimpreso en [Stonebraker & Hellerstein 1998].

[Leavitt 2000] Leavitt, N. "Whatever Happened to Object-Oriented Databases?" Computer, August 2000, Vol. 33, No.8, pp. 16-19.

[Leondes 2000] Leondes, C.T. (ed). "Knowledge-Based Systems. Vols. I-IV" Harcourt International, Academic Press 2000. http://www.apnet.com/knowledgesystems/

[Levene & Loizou 1999] Levene, M.; Loizou, G. "A Guided Tour of Relational Databases and Beyond" Springer 1999.

[Levesque & Lakemeyer 2001] Levesque, H.J.; Lakemeyer, G. "The Logic of Knowledge Bases" MIT Press, 2001.

[Litwin & Chien 1994] Litwin, W.; Chien, S.M. "Introduction to interoperable multidatabase systems" Prentice Hall, 1994.

[Liu 1999] Liu, M. "Deductive Database Languages: Problems and solutions" *ACM Computing Surveys*, 31(1): 27-62, 1999.

[Maier 1983] Maier, D. "The Theory of Relational Databases" Pitman, 1983

[Maier and Warren 1988] Maier, D.; Warren, D.S. "Computing with Logic", Benjamin Cummings, 1988.

[Maro-Saracco 1998] Maro-Saracco, C. "Universal Database Management: A Guide to Object/Relational Technology" Morgan Kaufmann, 1998.

[Merrett 1978] Merrett, T. "The extended relational algebra, a basis for query languages" in Databases, Shneiderman (ed.), Academic Press, 1978.

[Moldes 1995] Moldes, F.J. "Tecnología de los Sistemas de Información Geográfica" Ra-ma, 1995

[Mota & Celma 1994] Mota, L.; Celma, M.; Casamayor, J.C.; Bases de Datos Relacionales: Teoría y Diseño. Universidad Politécnica de Valencia, 1994.

[Muller 1999] Muller, R.J. "Database Design for Smarties. Using UML for Data Modeling" Morgan Kaufmann, 1999.

[NAP 1999] National Academy Press "The Rise of Relational Databases" Chapter 6 in "Funding a Revolution. Government Support fo Computing Research", Commitee on Innovations in Computing and Communications, Computer Science and telecommunications Board, Commission on Physical Sciences, Mathematics, and Applications, National Research Council, National Academy Press, Washington, D.C., 1999. http://books.nap.edu/books/0309062780/html/index.html

[Ng et al. 1998] Ng et al. "Exploratory Mining and Pruning Optimizations of Constrained Asociation Rules" Proc. ACM SIGMOD Int'l Conf. Management of Data, ACM Press, New York, 1998, pp. 13-24.

[Nicolett 2002] Nicolett, M. "The Market for Database Management Systems Heats Up" Gartner Group (www.gartner.com), 6 May 2002

[ODMG 2000] ODMG "Object Data Standard" Edited by R. G. G. Cattell, Douglas K. Barry, Mark Berler, Jeff Eastman, David Jordan, Craig Russell, Olaf Schadow, Torsten Stanienda, and Fernando Velez, Morgan Kaufmann Publishers, 2000.

[Ozsu & Valduriez 1999] Ozsu, M.T., Valduriez, P. "Principles of Distributed Database Systems" Prentice Hall 1999.

[Paredaens 1977] Paredaens, J. "About Functional Dependencies in a Database Structure and their Coverings" Phillips MBLE Lab. Report 342, 1977.

[Pyle 1999] Pyle, D. "Data Preparation for Data Mining" Morgan Kaufmann, Harcourt Intl., 1999.

[Ramakrishnan 1998] Ramakrishnan, R. "Database Management Systems" McGraw-Hill, 1998.

[Ramakrishnan & Gehrke 2000] Ramakrishnan, R.; Gehrke, J. "Database Management Systems" 2nd Edition, McGraw-Hill, 2000.

[Ramakrishnan & Ullman 1995] Ramakrishnan, R.; Ullman, J. "A Survey of Deductive Database Systems" *Journal of Logic Programming* 1995.

[Robie et al. 1998] Robie, J.; Lapp, J. Schach, D. "XML Query Language (XQL)" http://www.w3.org/TandS/QL/QL98/pp/xql.html

[Robinson 1979] Robinson, J.A. "Logic: Form and Function: The Mechanization of Deductive Reasoning", University Press, Edinburgh, 1978.

[Rondel et al. 1999] Rondel, Richard K., Varley, Sheila; Webb, Colin F. (eds.) "Clinical Data Management" 2nd edition, John Wiley & Sons, 1999.

[Rumbaugh et al. 1996] Rumbaugh, J.; Blaha, M.; Premerlani, W.; Eddy, F.; Lorensen., W. "Modelado y Diseño Orientado a Objetos", Prentice Hall, 1996.

[Sagiv & Yannakakis 1980] Sagiv, Y; Yannakakis, M. "Equivalence among expressions with the union and difference operators" Journal of the ACM, 27(49: 633-655, 1980.

[Saltor 1976] Saltor, F. "Dels fitxers clàssics a les bases de dades", 1976

[Schek et al. 1998] Hans-Jörg Schek, Fèlix Saltor, Isidro Ramos, Gustovo Alonso (Eds.): Advances in Database Technology - EDBT'98, 6th International Conference on Extending Database Technology, Valencia, Spain, March 23-27, 1998, Proceedings. Lecture Notes in Computer Science, Vol. 1377, Springer, 1998, ISBN 3-540-64264-1 http://www-dbs.inf.ethz.ch/dbs/edbt98.html

[Shipman 1981] Shipman, D.W. "The Functional Data Model and the data language DAPLEX" ACM Transactions on Database Systems, vol. 6, nº1, 1981.

[Silberchatz and Zdnik 1996] Silberschatz, A. & Zdonik, S. "Strategic Directions in Database Systems - Breaking Out of the Box". Computing Surveys 28(4): 764-778, 1996. También reimpreso en [Stonebraker & Hellerstein 1998].

[Silberschatz et al. 1991] Silberschatz, A. et al.: “Database Research: Achievements and Opportunities Into the 21st Century”. CACM 34(10): 110-120 (1991). También reimpreso en [Stonebraker & Hellerstein 1998].

[Silberschatz et al. 1996] Silberschatz, A.; Stonebraker, M.; Ullman, J.D. “Database research: achievements and opportunities into the 21th century”, SIGMOD RECORD, vol. 25, nº1, 1996.

[Silberschatz et al. 1998] Silberschatz, A.; Korth, H.F.; Sundarshan, S. “Fundamentos de Bases de Datos” 3ª Edición, McGraw Hill/International, 1998.

[Silberschatz et al. 2001] Silberschatz, A.; Korth, H.F.; Sundarshan, S. “Database System Concepts” *McGraw Hill College Div*; 4th edition, 2001.

[Smith & Smith 1977] Smith, J.; Smith, D. “Database abstractions: Aggregation and generalization” ACM Transactions on Database Systems, 1(1):105-133, 1977.

[SMITT 2002] “Slowed DBMS Market Means Tight Competition” http://www.advisor.com/Articles.nsf/aid/SMITT247

[Stonebraker & Hellerstein 1998] Stonebraker, M. and Hellerstein, J. (ed.) “Readings in Database Systems”, Third Edition, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, San Francisco, 1998. http://redbook.cs.berkeley.edu/

[Stonebraker 1980] Stonebraker, M. “Retrospection on a Database System”. ACM Transactions on Database Systems (TODS) 5(2): 225-240, 1980. También reimpreso en [Stonebraker & Hellerstein 1998].

[Stonebraker 1994] Stonebraker, M. (ed.) “Readings in Database Systems”, Second Edition, The Morgan Kaufmann Series in Data Management Systems, Morgan Kaufmann, San Francisco, 1994.

[Stonebraker 1996] Stonebraker, M. “Object-relational DBMSs. The next great wave”, 1996.

[Stonebraker et al. 1976] Stonebraker, M.; Wong, E.; Kreps, P.; Held, G. “The Design and Implementation of INGRES”. ACM Transactions on Database Systems (TODS) 1(3): 189-222, 1976. También reimpreso en [Stonebraker & Hellerstein 1998].

[Stonebraker et al. 1996] Stonebraker, M.; et al. "Mariposa: A Wide-Area Distributed Database”, Very Large Databases (VLDB) Journal 5(1): 48-63, 1996. También reimpreso en [Stonebraker & Hellerstein 1998].

[Stonebraker et al. 1998] Stonebraker, M.; Brown, P.; Moore, D. “Object-Relational DBMSs: Tracking the Next Great Wave” Second Edition, 1998.

[Subrahmanian 1998] Subrahmanian, V.S. “Principles of Multimedia Database Systems” Morgan Kaufmann, 1998.

[Tansel et al. 1993] Tansel, A.; Clifford, J.; Gadia, S. et al. (eds.) “Temporal Databases” Benjamin Cummings, 1993.

[Ullman & Widom 1997] Ullman, J.D.; Widom, J. “A First Course in Database Systems” Prentice-Hall International, 1997.

[Ullman & Widom 1999] Ullman, J.D.; Widom, J. “Introducción a las Bases de Datos” Prentice-Hall International, 1999.

[Ullman & Widom 2001] Ullman, J.D.; Widom, J. “A First Course in Database Systems” 2nd Edition, Prentice-Hall International, 2001.

[Ullman 1980] Ullman, J.D. “Principles of database systems”, Computer Science Press, 1980.

[Ullman 1988/1989] Ullman, J.D. “Principles of Database Systems” 2ª Edición, Vol. I y II, Computer Science Press, 1988, 1989.

[Ullman 1988] Ullman, J.D. “Principles of Database and Knowledge-Based Systems”, Vol. I, Computer Science Press, 1988.

[W3C 1999] World Wide Web Consortium “XML Path Language (XPath) Version 1.0. W3C Recommendation, Nov. 16, 1999. http://www.w3.org/TR/xpath.html

[Wagner 1998] Wagner, G. “Foundations of Knowledge Systems - with Applications to Databases and Agents”, Kluwer Academic Publishers, 1998.

[Widom & Ceri 1996] Widom, J.; Ceri, S. “Active Database Systems. Triggers and Rules for Advanced Database Processing” Morgan Kaufmann Publishers, Inc., 1996.

[Williams et al. 1998] Williams, R. et al. “R*: An Overview of the Architecture”, IBM Research Report RJ3325. Reimpreso en [Stonebraker & Hellerstein 1998].

[Witten & Frank 1999] Witten, I.H.; Frank, E. “Tools for Data Mining”, Morgan Kaufmann, 1999.

[W3C 2002] "XQuery 1.0: An XML Query Language" W3C Working Draft 30 April 2002, http://www.w3.org/TR/xquery/.

[Zaniolo 1983] Zaniolo, C. “The Database Language GEM.” SIGMOD Conference 1983: 207-218. También reimpreso en [Stonebraker & Hellerstein 1998].

[Zaniolo et al. 1997] Zaniolo C.; Ceri, S.; Faloutsos, T.S.; Subrahmanian, V.S.; Zicari, R “Advanced database systems”, 1997

[Zloof 1975] Zloof, M. “Query By Example” Proc. Of National Computer Conference, American Federation of Information Processing Societies, 44, 1975.

[Zloof 1982] Zloof, M.M. “Query-by-example: a database language” IBM Systems Journal, 16(4): 324-343, 1977.