Information integration: A research agenda

by A. D. Jhingran
N. Mattos
H. Pirahesh

The theme for this special issue—information integration—reflects the growing importance of integration in general, and data integration in particular, as a driving force in information technology spending. This essay discusses information integration along three axes—data types, federation, and intelligence. Several important problem areas are emerging—storage and retrieval of XML (Extensible Markup Language) documents, federation and distribution across data sources, and holistic intelligence across different data modalities. This special issue is devoted to papers on many of these topics, and we expect this to be an active area of research for many years to come.

Integration is the driving force of this decade of IT (information technology) spending. As enterprises buy more and more packaged applications, it is estimated that the task of combining these application "silos" results in over 40 percent of the IT spending, even though the amount of code written for integration is significantly smaller than 40 percent. This is because integration projects tend to be one-of-a-kind, and complex to write. The question for software and services vendors is this: can the cost of integration be reduced to be more in line with that of packaged applications?

The essay is organized as follows. This section describes four integration models. The next section gives an overview of information integration. Following sections then explore some of the technical challenges along the three axes that are the basis for our model of information integration. Finally, we end with some conclusions.

There are four distinct forms of integration:

1. Portals (or "at-the-glass") integration is the shallowest form, bringing potentially disparate applications together in a (typically Web) single entry point.
2. Business-process integration orchestrates processes across application and possibly enterprise boundaries, such as those involved in a supply-chain relationship. Web services and their derivatives are becoming important here.
3. Application integration, in which applications that do similar or complementary things communicate with each other, is typically focused on data transformation and message queuing, increasingly in the XML (Extensible Markup Language) domain.
4. Information integration, wherein complementary data are either physically (through warehousing tools) or logically brought together, makes it possible for applications to be written to and make use of all the relevant data in the enterprise, even if the data are not directly under their control. A typical example of this would be a new customer relationship application that combines the relational call log with the speech-to-text translated call itself.

Fundamentally, integration revolves around people, processes, applications, and information. Different integration technologies are necessary for different classes of integration problems. For example, on-line customer orders must be accepted through an application, not a database API (application programming interface). Business rules embedded in application programming logic protect the database from inappropriate use. Alternatively, the application that responds with a projected delivery date might well access correlated information across manufacturing and shipping databases and depend on the data management system to handle the complexity of join operations and mask differences between the data sources. As in this example, the best solution will often utilize several technologies. This illustrates the need to move easily from one technology to another.

Although the four models of integration are complementary, this special issue deals with information integration. An important research issue is: "If the information is integrated, does it make the job of the other three integrations easier?" One of the papers in this issue[1] deals with the boundaries between information integration and process and application integration.

Information integration

There has been spectacular growth in quantity of information. Recent studies indicate that business-relevant information is growing at around 50 percent compound annual growth rate,[2] with about one to two exabytes (10^{18}) of information being generated each year. Management of a large amount of information, *per se*, is not a very difficult problem. Data warehouses tend to easily exceed one terabyte (10^{12}) in size and, with CPU and disks improving in performance and cost performance, we do not see the volume of data to be the issue, until the data begin to touch 10s of terabytes or more.[3]

At the same time, there have been three trends that have made the task of managing such data inherently more complex:

1. *The heterogeneity of data*. Data are no longer just records that sit in well-defined tables (typically referred to as "structured" data). Increasingly, an enterprise has to deal with unstructured content—such as text (in e-mails, Web pages, etc.), audio (call center logs), and video (employee broadcast). In addition, data are beginning to emerge in XML format, which in some ways is the bridge between the structured and unstructured worlds, though that is an oversimplification in the sense that a perfect solution for XML is often a less-than-perfect solution for the two extremes.

2. *The "federation" and "distribution" of data*. Data are no longer on one logical server (such as in a well-architected warehouse), but are distributed across multiple machines in different organizations (some within and some across enterprises). This is in the classic sense of distributed databases, except that the scale could be as large as billions of databases (whereas classic databases have handled distribution at the scale of around 10). In addition, federation (who owns and controls the data and access to the data) is a new problem that distributed database technology has typically not addressed. In federation scenarios, one typically cannot assume full SQL (Structured Query Language) or its equivalent access to distributed data sources. In addition, privacy and security issues need to be solved.

3. *Using data for competitive advantage*. The data need to be manipulated, aggregated, transformed, and analyzed in increasingly complex ways to produce business intelligence. And the speed of access and analysis is becoming closer to real time. A large fraction of the growth in relational databases in the early to mid-1990s was fueled by "business intelligence"—a term for a collection of tasks ranging from decision support through complex SQL queries, to on-line analytical processing (OLAP), and all the way to data mining—wherein the system automatically discovered and told the users about what it had found. With the increase in data, the ability for the decision makers to sift through the data is falling ever behind, and therefore data analysis that works across all the modalities of data is becoming increasingly important.

We refer to these three dimensions as heterogeneity, federation, and intelligence. Information integration, then, refers to the ability to analyze data across data types and over a span of control (Figure 1).

An example of this overall vision is IBM's work on information integration (Figure 2). Data of different forms go through federation and can be analyzed or accessed through SQL or XQuery (an XML query language).[4] See Reference 5 for a detailed description of IBM's vision.

Heterogeneity of data

Relational databases have typically dealt with fixed schema—that is, there is a set of tables, each consisting of an arbitrarily large number of rows; however, each row in a table has an identical structure with all other rows in the table. This has been very useful for SQL expressibility and optimization. In contrast, many newer forms of data (such as documents, images, videos, etc.) do not follow the same rigid pattern. Even if a database is a collection of books, and each book has a set of chapters, it is rarely the case that each book has the same number of chapters. Consequently, a breakup of the schema for books as shown in Table 1 is typically not possible. One is either forced to convert it into a vertical relation, such as Table 2, where operations to assemble the entire book then become fairly complex, or to leave the data in a more unstructured form and then derive some fixed-format attributes such as author or publisher.

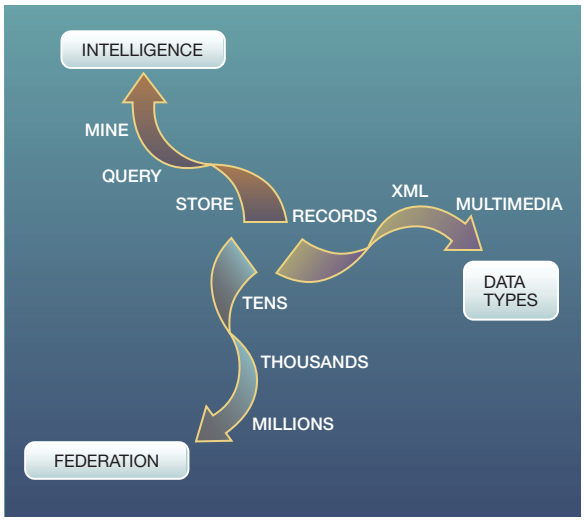
In the structure for Table 2, more unstructured queries, such as those typified by Web search engines, become easier to answer. This is the technique used by various content management solutions, such as IBM Content Manager, and various document management solutions such as Documentum**, and even pure text indexing solutions, such as Google** or Inktomi**.

Figure 3 describes the architecture of the IBM Content Manager. It uses a standard relational library server (LS) to store the meta-data for the content, but uses different resource managers (RMs) to actually manage the content.

Thus it is clear that there are two slightly different perspectives—well-formed structured schema and the relatively poorly structured world of documents. Bringing these two worldviews together is the "holy grail" of information integration, and Reference 6, in this issue, discusses several promising directions.

The world of XML, which sits between the two perspectives, can resemble either. A very structured document, such as an Electronic Data Interchange purchase order (EDI PO) could be very precise and could be modeled, with only a slight amount of discomfort, as a set of relational tables. However, a collection of books expressed as a set of XML documents does not have a rich enough schema (beyond meta-data such as authors, publishers, etc., and data that are often just a collection of chapters) to be expressible in a relational world in some interesting way.

Figure 1 Three dimensions of information integration



Precisely described XML could be split into constituent tables, or databases could be extended to support XML as a proper data type for documents. (In the latter model, storage, indexing, concurrency control and recovery, query language, and transaction processing of relational engines would need to be extended on this new data type.) While it is a subject of active debate in academia [7,8] as to which way to go, many commercial database vendors are making quick decisions. IBM Database 2* (DB2*), for example, currently supports XML natively through its extender technology.[9] However, it is further extending its relational engine with support for XML, all the way from storage to the query engine that supports the XQuery[6] language. In addition, for applications that require an SQL interface into XML stores, DB2's SQL query language has also been extended to SQLX, which provides support for XML extensions, such as path expressions.[10] XML documents conforming to schema-chaos[11] or to no schema at all can also be stored in such XML extensions, although the power of relational and XQuery engines against such ill-formed XML would be limited. Consequently, document collections conforming to these data types would more naturally be stored in content management systems that have been extended to support XML.

Beyond records, XML, and text, there are other data types that are in fact the primary drivers of the information growth—MP3 (Moving Picture Experts

Table 1 One possible relational schema for books

Book Name	Chapter 1 Text	Chapter 2 Text	Chapter 3 Text	...
...				

Table 2 A more plausible relational schema for books

Book Name	Chapter Number	Text
	1 2 3 ...	

Group 1, Audio Layer 3) files, digital photos, and call center recordings. The cost for storing these is becoming relatively inconsequential (one terabyte of disk space for home use will cost less than $500 by 2003). Two questions arise—first, will the storage for these be embedded in applications or will (at least logically) centralized content stores emerge (either at home or in enterprises); and second, what kind of "intelligence" can be derived from these new data types? We address the latter question in a later section; however, regarding a logical centralized store, we see the same pattern emerging as in the case of data management—while applications initially built their own data management solutions in the 1970s, once common functions in databases became available, the applications began to focus on the differentiated tasks and left data management to commercial systems. It is therefore expected that content management for the applications that make use of digital data of diverse forms will become a very important business. The Aberdeen Group estimates that new enterprise information integration technology will fuel a $7.5 billion market by 2003.[12]

Federation

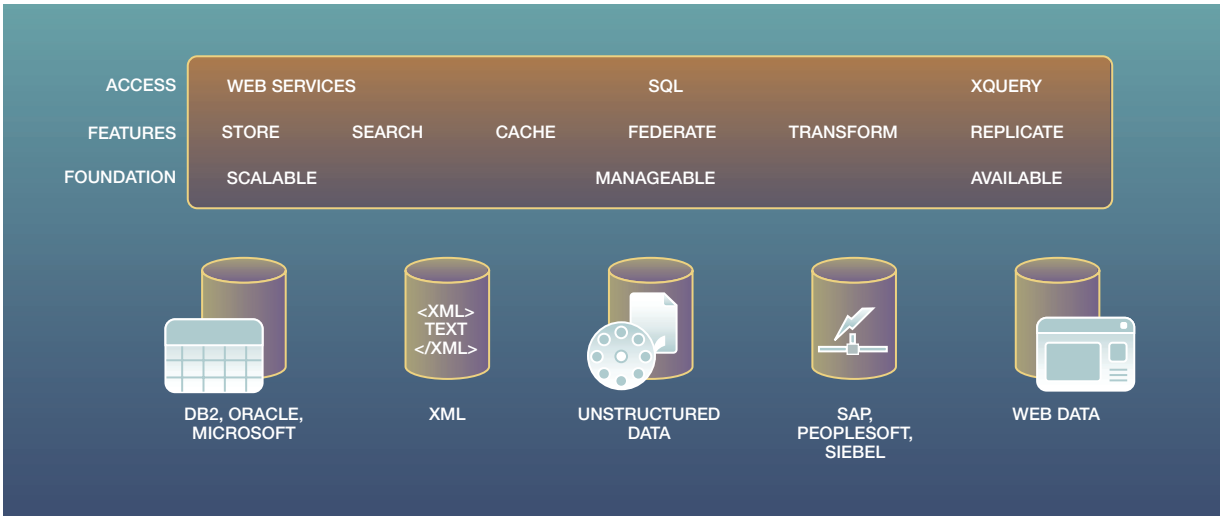
While centralization of data operations was a significant driving force of the growth in the database business (both for transaction processing and for decision support), it is clear that decentralized tendencies in the growth of data have accelerated in the recent past (the Internet is a good example of this). In addition, even within an enterprise, data typically cannot be shared freely between departments, or between different employees or different levels of employees. Consequently, centralizing the data (i.e., bringing the data together in one place) may not be possible in many environments. In these cases, the only choice is to leave the data where they are, and access the data through federation. Of course, there is no black-and-white world. The two models—centralization and federation—often have hybrids, such as data caching and replication.

As an example of federation, consider IBM's DiscoveryLink* offering.[13] DiscoveryLink extends DB2's Data Joiner technology (allows a relational engine to access other relational engines as if the data were local) and IBM Research's Garlic technology (allows federation across nonrelational data sources through "wrapper" technology) with specific wrappers and connectors to life sciences data sources, such as human genomic data. As a result, a user can connect to a DiscoveryLink "console" and express queries that join data from disparate data sources, some local, some not, some relational, some not. Another example of federation in DB2 is Microsoft Windows** OLE** DB support, which allows access to relational and nonrelational OLE DB-compliant data sources, such as Lotus Notes* and Microsoft's Excel**, Exchange Server, and SQL Server.[14]

There are several new trends in federation:

1. Web services technology is becoming an increasingly popular way of connecting distributed applications. It is an important development to put data management in this Web services framework.[15] Two aspects become interesting—databases as Web services providers, and databases as Web services initiators. In the latter, federation can be achieved by using more industry-standard Web services; however, one has to take into account the current state of the art in reliability and performance. Web services need to be extended (for example, through caching[16]) to achieve the better reliability and performance that is typically expected from more mature database technologies.
2. Grids make it possible to share computation. Recently, data sharing is becoming increasingly important in the grid environment. Shared databases are likely to play an important role, and federation and information integration technologies will expand to incorporate capabilities from, and

Figure 2 IBM's vision of information integration

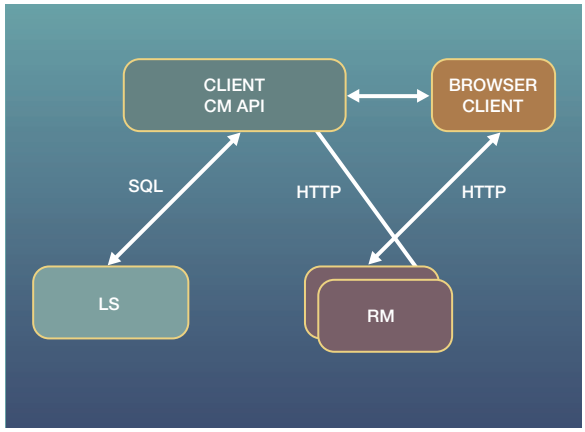


provide technologies to, grid standards such as Open Grid Services Architecture (OGSA).[17]

3. Privacy and security, in the data federation axis, are becoming very important. As supply chains become more integrated, and as national security applications rise in importance, the need for distributed computation across autonomous data sources is obvious. Recent works on watermarking, privacy-preserving data mining,[18] and distributed data mining are steps in that direction.
4. Tools for data integration (e.g., data analysis for automated data mining) are riding on the huge investments that the industry is making around XML. These tools are becoming more important, because the complexity of the schema that are brought together (often logically) is increasing—in numbers, as well as in scope. A good example of an emerging technology in this area is CLIO.[19]

It is not necessarily the case that as data distribution and federation increase, the amount of data to be handled by the application increases. In fact, we have observed a strong correlation between the number of data sources and the amount of data at each data source. It is our hypothesis that over the course of the next five years, one petabyte (1024 terabytes) of data would become the focus of many applications. Some would require that amount to be kept in one or two large centralized warehouses. Other applications, such as content sharing on wide-area networks, might require a million databases, each having (in potentially redundant copies) one gigabyte of data. Research into distributions and sizes along this one-petabyte constant is just beginning and is likely to accelerate as the federation trend increases.

Figure 3 IBM's content management architecture



Intelligence

As data become heterogeneous and federated, how does one integrate these data into the businesses pro-

cesses? One of the primary data integration challenges is to integrate into applications that seek to derive intelligence from these data sources. An example of this intelligence might be in the context of a call-center application, where customers' calls are recorded and the call-center representative (CSR) also records, in a structured form, the time of the call, who called, etc. An integrated analysis across the two forms of data (structured and speech) might provide actionable results such as "when the customer calls, and is angry, if the company does not respond within five business days, there is a 45 percent chance of losing the customer." It is clear that the concept of "customer being angry" is not derivable from the structured data that the CSR has recorded. At the same time, just the speech recording cannot tell us about what actions followed the customer calls. It is only the holistic analysis that can lead to this kind of intelligence.

Even without these holistic analyses (which are just beginning to emerge), we already see a trend toward structured and unstructured data coming together in query systems. The two types of data have very different characteristics. Structured data are typically very precise (the answers always have 100 percent precision at any recall), whereas unstructured systems are fuzzier in both query specifications and in execution. The failure models of the systems also tend to be different—in databases, failure of any part of the system leads to failure of the entire system (to maintain very precise semantics), whereas in many text systems, unavailability of some part of the system does not stop the system.

Recent work in this area has come from many directions. Combination of ranked results has been dealt with comprehensively by Fagin,[20] and an interesting approach to imprecise specification of attributes is presented in Reference 21. We expect this to be a very fertile area of research. This special issue contains a perspective presented in Reference 22 on expanding the concept of OLAP cubes with unstructured data, and another in Reference 6 on combining content management systems with database systems.

The intelligence dimension of our three axes (see Figure 1) is associated with data analysis, such as detecting trends in a business and providing a closed feedback loop for business operation. Usually, analysis is based on a large amount of current and historical data stored in warehouses and datamarts. A popular model for analysis is the multidimensional OLAP cube data model, with the associated navigational API. Reference 23 describes an example of a system in which the multidimensional OLAP cube model is integrated with relational databases. OLAP Web services allow users to discover and explore analytic information across the Web through the XML protocol. This model is particularly attractive for integrating information from service providers with rich terabyte- or petabyte-class warehouses in real time.

Summary

This essay lays out the framework for the research agenda in information integration. As we view the problem of information integration along the three axes of data types, federation, and intelligence, many interesting problems emerge. Some of the active areas of research are emerging in XML—storage, querying, and mining; in distributed data analysis across hundreds or thousands of data sources; and in new data analysis techniques for combining structured and unstructured data. Cutting across all the dimensions are issues related to tools for information integration, and privacy and security around data. This special issue deals with many of these topics, and we expect this to be an important area of research for many years to come.

Acknowledgments

The authors wish to thank Kevin Beyer, Tobias Mayr, and Holly Hayes for their comments on various drafts of this essay, and to a very large number of people who helped formulate our thoughts as presented here.

Cited references and notes

1. F. Leymann and D. Roller, "Using Flows in Information Integration," *IBM Systems Journal* **41**, No. 4, 732–742 (2002, this issue).
2. H. Varian and P. Lyman, "How Much Information?" See http://www.sims.berkeley.edu/research/projects/how-much-info/.
3. The one challenge that remains for large databases, though, is the "manageability" of such a warehouse—efficient backup/restores, for example.
4. D. Chamberlin, "XQuery: An XML Query Language," *IBM Systems Journal* **41**, No. 4, 597–615 (2002, this issue).
5. M. A. Roth, D. C. Wolfson, J. C. Kleewein, and C. J. Nelin, "Information Integration: A New Generation of Information

Technology," *IBM Systems Journal* **41**, No. 4, 563–577 (2002, this issue).
6. A. Somani, D. Choy, and J. C. Kleewein, "Bringing Together Content and Data Management Systems: Challenges and Opportunities," *IBM Systems Journal* **41**, No. 4, 686–696 (2002, this issue).
7. J. E. Funderburk, G. Kiernan, J. Shanmugasundaram, E. Shekita, and C. Wei, "XTABLES: Bridging Relational Technology and XML," *IBM Systems Journal* **41**, No. 4, 616–641 (2002, this issue).
8. M. Fernandez, D. Suciu, and W.-C. Tan, "Silkroute: Trading Between Relations and XML," *Proceedings, 9th International World Wide Web Conference*, Amsterdam, Netherlands (May 15–19, 2000), pp. 723–746.
9. J. Xu and J. Cheng, "XML and DB2," *Proceedings*, *Sixteenth IEEE Conference on Data Engineering*, San Diego, CA (February 28–March 3, 2000).
10. J. E. Funderburk, S. Malaika, and B. Reinwald, "XML Programming with SQL/XML and XQuery," *IBM Systems Journal* **41**, No. 4, 642–665 (2002, this issue).
11. This refers to scenarios where the documents conform to a bounded, but large number (hundreds or thousands) of schemas.
12. W. T. Kernochan, *Enterprise Information Integration: The New Way to Leverage e-Information*, Aberdeen Group Report (May 2002).
13. L. M. Haas, E. T. Lin, and M. A. Roth, "Data Integration Through Database Federation," *IBM Systems Journal* **41**, No. 4, 578–596 (2002, this issue).
14. B. Reinwald, H. Pirahesh, G. Krishnamoorthy, G. Lapis, B. Tran, and S. Vora, "Heterogeneous Query Processing Through SQL Table Functions," *Proceedings*, *15th International Conference on Data Engineering*, Sydney, Australia (March 23–26, 1999), pp. 366–373.
15. S. Malaika, C. J. Nelin, R. Qu, B. Reinwald, and D. C. Wolfson, "DB2 and Web Services," *IBM Systems Journal* **41**, No. 4, 666–685 (2002, this issue).
16. Q. Luo, S. Krishnamurthy, C. Mohan, H. Pirahesh, H. Woo, B. Lindsay, and J. Naughton, "Middle-Tier Database Caching for e-Business," *Proceedings*, *ACM SIGMOD International Conference on Management of Data*, Madison, WI (June 3–6, 2002).
17. V. Raman, I. Narang, C. Crone, L. Haas, S. Malaika, T. Mukai, D. Wolfson, and C. Baru, "Data Access and Management Services on Grid," Informational Document, Global Grid Forum 5, Edinburgh, Scotland (July 21–24, 2002). Available at http://www.gridforum.org/Meetings/ggf5/pdf/dais/document2.pdf.
18. R. Agrawal and S. Ramakrishnan, "Privacy-Preserving Data Mining," *Proceedings*, *ACM SIGMOD Conference 2000*, Dallas, TX (May 16–18, 2000).
19. L. Popa, Y. Velegrakis, M. Hernandez, R. Miller, and R. Fagin, "Translating Web Data," *Proceedings*, *28th Conference for Very Large Databases*, Hong Kong, China (August 20–23, 2002).
20. R. Fagin, "Combining Fuzzy Information: An Overview," *ACM SIGMOD Record* **31**, No. 2, 109–118 (June 2002).
21. R. Agrawal and R. Srikant, "Searching with Numbers," *Proceedings*, *Eleventh International World Wide Web Conference*, Honolulu, Hawaii (May 7–11, 2002).
22. W. F. Cody, J. T. Kreulen, V. Krishna, and W. S. Spangler, "The Integration of Business Intelligence and Knowledge Management," *IBM Systems Journal* **41**, No. 4, 697–713 (2002, this issue).
23. N. Colossi, W. Malloy, and B. Reinwald, "Relational Extensions for OLAP," *IBM Systems Journal* **41**, No. 4, 714–731 (2002, this issue).

Accepted for publication August 20, 2002.

Anant Jhingran *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120 (electronic mail: anant@almaden.ibm.com).* Dr. Jhingran is the Director of Computer Science: Foundations, Software, and Services at IBM's Almaden Research Center. He manages a team of about 150 researchers working on data management, the Web, human-computer interaction, knowledge management, and computer science theory. Previously, he was Senior Manager of e-Commerce and data management at IBM's Thomas J. Watson Research Center. He has been with IBM since 1990. He received his Ph.D. degree in 1990, from the University of California at Berkeley, in the area of database systems, and his bachelor's degree in 1985, from the Indian Institute of Technology, Delhi, in electrical engineering. He is a member of the ACM and a senior member of the IEEE. He has published several papers in leading database conferences such as SIGMOD, VLDB, and Data Engineering, and he served on the program committees of many of these conferences. He has won several IBM awards, including a Corporate Award for "DB2 Common Database Servers." He also holds several patents and is a member of the IBM Academy of Technology.

Nelson Mattos *IBM Software Group, Silicon Valley Laboratory, 555 Bailey Avenue, San Jose, California 95141 (electronic mail: mattos@us.ibm.com).* Dr. Mattos, IBM Distinguished Engineer, is director of information integration at the IBM Silicon Valley Laboratory, where he is responsible for establishing IBM's leadership position in the emerging information integration market. Additionally, he is responsible for IBM's participation at different standards forums, including the ANSI SQL committee, the International Organization for Standardization (ISO) Committee for database, the World Wide Web Consortium (W3C), the Object Management Group (OMG), and Embedded SQL in Java® (SQLJ). In this capacity, he contributed extensively to the design of SQL99 through more than 300 accepted proposals. Before joining IBM, Dr. Mattos was an associate professor at the University of Kaiserslautern in Germany, where he was involved in research on object-oriented and knowledge base management systems and received a Ph.D. degree in computer science. He also holds bachelor of science and master of science degrees from the Federal University of Rio Grande do Sul in Brazil. Dr. Mattos has published over 75 papers on database management and related topics and is the author of the book, *An Approach to Knowledge Base Management*.

Hamid Pirahesh *IBM Research Division, Almaden Research Center, 650 Harry Road, San Jose, California 95120 (electric mail: pirahesh@almaden.ibm.com)*. Dr. Pirahesh is an IBM Fellow and a senior manager responsible for the exploratory database department at IBM Almaden Research Center in San Jose, California. He is also the manager of the DataBase Technology Institute (DBTI) at IBM Research. He has direct responsibilities for various aspects of the IBM DB2 product, including architecture, design, and development. He received his Ph.D. degree from the University of California at Los Angeles in the area of database systems. He is an IBM master inventor and a member of the IBM Academy of Technology. He is also an associate editor of *ACM Computing Surveys* and has served on the program committees of major computer conferences. He was a principal mem-

ber of the original team that designed the query processing architecture of the IBM DB2 Universal Database™ relational database management system and delivered the product to the marketplace. He has made major contributions to query language industry standards. His work optimization using aggregate data caching has resulted in dramatic performance improvement. This feature is now considered to be essential for processing of complex data analysis and OLAP queries in large databases. His research areas include OLAP and aggregate data management, query optimization, data warehousing, Web services, management of semi-structured and unstructured XML data, and information integration in Web-based federated and distributed systems. He also serves as a consultant to various IBM product divisions, including the software division and IBM Global Services.