Microsoft

Choosing Microsoft SQL Server 2008 for Data Warehousing

White Paper

Published: December 2006

Updated: July 2008

Summary: The warehousing of corporate data into an architecture from which it can all be accessed is no longer an option for midmarket and large companies that want to be successful. Consolidating information so that it is readily accessible, and making it visible for reporting, analytics, spot queries, and predictive capabilities is now necessary. This requirement is met in the field of data warehousing and business intelligence (BI).

Companies in every industry use data warehouses to improve customer service, guide product development, reduce prices, shorten cycle times, and improve quality. Companies not yet using data warehousing to make better decisions are already behind. The more quickly they can implement a data warehouse, the more quickly they can overcome a competitive disadvantage.

A Microsoft-based data warehouse can be implemented quickly and supports the analysis of data from many operational systems. The marketplace is aligning into complete frameworks and Microsoft brings all the necessary components to build, manage, and deliver data warehousing. In SQL Server 2008, Microsoft has the fastest-growing data warehouse platform, with a comprehensive data movement platform, a manageable/scalable DBMS, and close integration with the 2007 Microsoft Office system.

For the latest information, see Microsoft SQL Server 2008.

Contents

The State of Data Warehousing

The first published work on the data warehouse topic was by Barry Devlin and Paul Murphy of IBM Ireland. It took a comprehensive architecture point of view, integrating data warehousing into information systems development as a whole. It introduced the term *Information Warehouse* as: “A structured environment supporting end users in managing the complete business and supporting IS in ensuring data quality.”

Why a Data Warehouse?

The most widely published definition of data warehousing comes from Bill Inmon: “A data warehouse is a subject-oriented, integrated, time-variant, nonvolatile collection of data in support of management decisions.” These terms are used as follows:

- **Subject-oriented:** The focus is on natural data groups, not on application orientation.
- **Integrated:** It provides consistent formats and data values.
- **Time-variant:** Data is organized by time and its temporal nature is directly captured.
- **Nonvolatile:** No updates are allowed—only load and select operations.

Inmon’s definition provided concrete guidelines for building a data warehouse. Implicitly, this definition supports one of the most fundamental principles of data warehouse development—the principle that the data origination and the data access environments are physically separated onto different databases and different platforms.

A data warehouse is an information architecture deliverable that is dedicated to managing the availability, integration, and consistency of data defined by technical architecture guidelines, which define the explicit functional components needed for development and operation and governed by a set of real-world principles.

A well-built warehouse resolves inconsistencies in enterprise data and improves the data quality to produce a clean, integrated base of information.

The Functions of Data Warehousing

Just as a factory fashions raw materials into finished goods to meet consumer needs, a data warehouse fashions information for its consumers. The product mix must continually be realigned to the market. Though the raw materials do not change much, the product set evolves.

A data warehouse must develop a process to collect the pure raw materials and then continually repackage them to serve evolving business initiatives. Data warehouses must be built, managed, and delivered. We do not want to change the technology so it’s important to get it right. A long feature set is not

beneficial if development is onerous and cycle turnaround times are long and costly.

Each new initiative that a data warehouse serves should be treated as a project within a program. Each has a lifecycle that is more like a consumer product lifecycle than that of a typical technology project.

At its essence, the data warehouse process is an information product process. We must establish the guiding principles and champion, architect, deliver, and support iterations. The data warehouse engine is the company's information factory and it should have high reliability.

The Realities of Data Warehousing Today

Data warehousing has proven its value over and over again by providing the data to help companies compete in the relevant area of competitive advantage today—business intelligence (BI).

The characteristics of data warehousing today are many:

- Multiple, complex applications serving a variety of users
- Exploding data size that will continue to explode with RFID, POS, CDR, and all manner of transactional data extending back years into history
- Data latency becoming intolerable as business needs demand real-time data
- A varied set of data access tools, serving a variety of purposes, for each data warehouse
- Multiple workloads streaming into the data warehouse from varied corners of the company as well as from outside the company
- A progression towards more frequent, even continuous, loading
- Data types running the gamut beyond traditional alphanumeric types

Data warehousing is important to a business to the degree that it can provide either short- or long-term value to the bottom line.

Data warehousing is applicable to all facets of a company. There is scarcely a strategic or tactical company objective that cannot be supported with the information generated from a data warehouse.

Data warehouse size is booming due to a variety of factors, but mainly because success begets success with data warehousing. As the initial data generates profitable use and the platform proves able to handle the workload, it is a matter of time until new uses leverage the data and add their different data requirements to the warehouse. And, using detailed data in conjunction with summary data is important for effective decision making, further contributing to data overload.

Seldom is it feasible to delete or otherwise render inaccessible older data. Plan on data simply accumulating ad infinitum in the warehouse. Plan on loading all the historical data you have to seed the warehouse as well.

With all this investment and value, you'll want to leverage your data warehouse for customers, supply chain partners, and possibly selectively to the broader Internet. Make sure you choose a proven database management system (DBMS) not just for the initial, known requirements but also for future, to-be-determined requirements.

Companies are realizing the usefulness of data that is generated outside of their confines—so called third-party data. It is no longer difficult or untested to "subscribe" to external data feeds to augment internally generated data. Marketing departments in particular have grown in their sophistication to deal with all kinds of data and the more, the better.

These trends must be taken into consideration when choosing the toolset to build, manage, and deliver a data warehouse.

Putting the Technology Set Together

When making product decisions for a data warehousing environment, the DBMS is the most important. It is the foundational technical component from which all other product decisions naturally follow. Microsoft® SQL Server® 2008 is a solid choice when selecting a data warehouse DBMS for organizations with highly divergent data warehouse requirements. The latest IDC report shows Microsoft data warehousing growing at 22 percent.

Criteria for a Data Warehouse DBMS Selection

Given the state of the marketplace, in order to build, manage, and deliver a data warehouse, the technical architecture should be:

- **Manageable.** Through minimal support tasks requiring database administrator (DBA)/System Administrator intervention, the data warehouse should provide a single point of control to simplify system administration. You should be able to create and implement new tables and indexes at will.
- **Complete and integrated.** The toolset should be comprehensive across the spectrum of eventual requirements for data and its access.
- **Interoperable.** The toolset should provide integrated access to the Web, the Microsoft Office system, internal networks, and corporate mainframes.
- **Scalable.** The solution should have the ability to handle increasing data sizes and workloads with simple additions to the architecture, as opposed to the increases requiring a rearchitecture.
- **Affordable.** The solution (hardware, software, services, and required customer support) needs to provide a low total cost of ownership (TCO) over a multi-year period.
- **Proven and supported.** Don't risk the fundamental underpinning of the data warehouse environment on an unproven solution.
- **Flexible.** The solution should provide optimal performance across the full range of models with large numbers of tables. Look for a proven ability to support multiple applications from

different business units, leveraging data that is integrated across business functions and subject areas.

- **User accessible.** Compatibilities and interoperability with data access tools that provide a wide range of user-friendly access options.

The data warehouse DBMS selection is critical and drives all other technology decisions. The technology must support both immediate requirements as well as future unspecified and unknown requirements. Ideally, the DBMS selection should be the first technology decision made for a data warehouse project.

You may also see the affordable value proposition of SQL Server 2008 to data warehousing and consider initiating a rearchitecture project to continue the benefits of the data warehouse you have, but with a lower TCO platform. The relative simplicity of ongoing development or a single-vendor solution with SQL Server 2008 may also be a factor that precipitates a rearchitecture. A solid vision with an integration of all the necessary components for data warehousing may also hasten a platform project. In my many years architecting and leading data warehouse efforts for a variety of clients in multiple industries, no client has reached the end of data warehouse development. It is an ongoing process, delivering continual and increasing business value over time. It is never too late to consider value propositions.

You will create a culture around your selected DBMS. You will hire and train your people to support it. It will become the primary driver for hardware and other software selections. Your people will attend user group meetings and interact with others using the DBMS for similar purposes. You will hire consultancy on the DBMS and research how most effectively to exploit the technology. You will need vendor support and you want the vendor to continue adding relevant features and capabilities to the DBMS that are needed for data warehousing in the future.

Some of the consequences of making inappropriate DBMS selection for data warehousing and BI include:

- Long development cycles
- High numbers of support staff required
- Cost expansion
- “Throwing hardware at problems” as a solution
- Users reverting to old means of data access with user interfaces that are not friendly
- A technology-focused culture rather than a user culture in IT
- Complex vendor relationships
- Hard to incorporate legacy systems and unstructured data
- Inability to keep pace with growing data volumes and user demands
- Inability to show profitability from data warehouse efforts, leading to slow program demise

The SQL Server 2008 Solution

SQL Server 2008 provides all the interoperable tools necessary to build, manage, and deliver a data warehouse/BI environment. While some organizations may need to supplement the Microsoft toolset, or add specialty tools, the major technology is provided and is reasonable. Should an alternative tool be preferred for any reason (data movement, data access, reporting), that is feasible.

The toolset contains tools for building, managing, and delivering a solid data warehouse/BI environment. In some areas, the tools are best practice. SQL Server 2008 Analysis Services, for example, is a best-of-breed tool for OLAP analysis engines. However, all tools are competitive in their respective spaces, even as standalone products.

There are some competitive advantages systemic in the Microsoft business intelligence toolset and they are flexibility, ease-of-use, and deployability. These are especially critical in informal cultures, fast-paced environments, and shops with budgetary considerations.

Microsoft is clearly committed to business intelligence. The tools in the Microsoft business intelligence framework described herein are being upgraded at a faster pace than most market competitors and any exposed seams in the integration of the tools is being addressed as a priority.

Building a Data Warehouse

SQL Server 2008 has an accessible set of business intelligence tools designed to build the data warehouse effectively in a rapid manner. The ability to develop rapidly and accurately is very important in building data warehouses. SQL Server 2008 provides an integrated development environment along with an enterprise data integration platform to accelerate data warehouse development. Business Intelligence Development Studio is a visual development environment that is built upon the productive framework of the Microsoft Visual Studio® development system, incorporating powerful debugging capabilities along with a consistent environment to build your cubes, reports, and extraction, transformation, and loading (ETL) packages. SQL Server 2008 Integration Services is the next generation data integration platform, which provides a scalable engine that can incorporate heterogeneous data sources, validate, and transform the data into your data warehouse.

SQL Server 2008 Integration Services

SQL Server 2008 Integration Services (SSIS) is the tool in the Microsoft framework for the extraction, transformation, and loading of data into both the SQL Server 2008 DBMS and into SQL Server 2008 Analysis Services structures. It facilitates the data movement required for data warehouse

success. Data movement jobs are scheduled and can be organized into complex nonlinear flows as required. Its GUI is very user-friendly.

Data flow and sequencing is controlled by *data flow tasks*. Data flow tasks access data from a source, process it, and write it to a target. A simple double-click on the data flow task switches you to a design interface where transformations can be added to the data flow.

Advanced transformations such as text mining transformations and fuzzy lookups, which involve the acceptance of close matches, are available. Fuzzy lookup creates similarity and confidence scores. A combination of this with text mining can be used to determine your acceptance systemically. Fuzzy grouping looks at a group of potential records for loading and determines the probability that two (i.e. customer names) are duplicates.

Some data sources have predefined source adapters, which greatly facilitate the acquisition of data from those sources. These include OLE DB, Microsoft ADO.NET, flat files and XML formats. The destination set is similar, Usually SQL Server 2008 Integration Services is used to serve data to SQL Server 2008, but you can move data into other DBMS products with it as well.

Surrogate keys are commonly used in data warehouse fact tables to enable features such as slowly changing dimensions. The use of surrogate keys requires Integration Services to perform a lookup to transform the business key into its corresponding surrogate key. In SQL Server 2008 Integration Services, lookup performance is dramatically improved, and lookup operations scale to support the largest tables. Moreover, the performance of pipeline operations also has been substantially improved in SQL Server 2008.

Database Engine Enhancements

Various techniques can be employed to prevent the loading of duplicate records from source systems into the data warehouse, including TIMESTAMP columns, INNER JOIN statements, and lookup operations, but these techniques can be complex to develop and use. SQL Server 2008 introduces the MERGE statement, which you can use to insert or update data depending upon whether it exists in both the source and destination systems, significantly reducing the amount of custom logic required to load data into the data warehouse.

When changes are made to a source OLTP system, these changes need to be applied to the data warehouse system. When data is inserted or updated, this can be achieved using traditional Transact-SQL statements or the MERGE statement, but neither of these techniques provides a simple way of cascading a delete from the source system to the data warehouse. Change management through these techniques also requires a query of the entire contents of the source and destination tables to check for changes, which is time consuming and resource intensive and therefore leads to increased latency. Change data capture keeps a copy of any data modification and allows SQL Server 2008 to cascade inserts, updates, and deletes from source systems to the data

warehouse without the time and expense of creating a change management system. Because there is no longer any need to search the source system for changes, latency is reduced and real-time BI solutions can be achieved.

Data loading performance is also greatly improved by extensions to minimal logging in SQL Server 2008. This allows some common load operations to run two to three times faster.

Managing a Data Warehouse

SQL Server 2008 provides a manageable, scalable data warehouse platform. With several enhancements in SQL Server 2008, Microsoft enables Information Technology departments to productively manage their growing data volumes along with the rapid increase in usage of the data warehouse.

SQL Server 2008 Database Management System

The touchstone for the entire toolset is the underlying database engine—the SQL Server 2008 DBMS. This DBMS is the backbone of numerous applications from operational purposes to data warehousing. It handles enterprise data warehouse workloads. It supports high concurrency, high database sizes, and mixed workloads comprising reporting, OLAP, data mining, and operational purposes. Over the years, Microsoft has augmented its DBMS with the tools in this section and more, enabling the technical toolset for data warehouses to be entirely handled by the Microsoft framework, while maintaining an industry-leading ease-of-use.

The DBMS scales well at high data volumes, especially when the accessible techniques described later in this paper are deployed. Parallelization is key to SQL Server 2008 success and most operations are now “parallelized” to take full advantage of the parallel hardware that supports SQL Server 2008. Partitioning large sets of data makes it much easier to manage “sliding window” archival processes and makes it possible to reduce recovery time by enabling backup and restore operations for individual partitions based on their distribution across physical filegroups. In general, partitioning allows easier management by letting you work on manageable-sized chunks of data. Using partitioned tables and parallel load to independent partitions, fast loading is supported. Improvements to parallel query processing for partitioned tables and indexes in SQL Server 2008 can significantly improve performance in large databases.

As a system scales up and supports increasing numbers of concurrent queries, it becomes important that mission-critical operations are not impacted by specific queries or resource starvation caused by other unrelated processes. To achieve this, SQL Server 2008 includes Resource Governor. With Resource Governor, you can define the workloads to be performed by a database server based on users, applications, databases, and other differentiating factors. You can then assign limits on CPU and memory use for each workload, which make it possible to control resource utilization and

prevent a runaway query in one workload from affecting the performance of another workload. Furthermore, you can reserve resources for high-priority maintenance tasks to make sure they finish in the available time. This ability to proactively manage workload resource utilization results in predictable performance across the differentiated workloads, making management and troubleshooting of your data services much easier.

Data warehouse systems typically have large data volumes that often reach terabytes in size. Although most organizations accept the increased storage as a natural and acceptable cost of data warehousing, storage cost is a significant portion of overall data warehouse installation cost, and the movement of large amounts of data in I/O intensive systems can cause a reduction in performance. SQL Server 2008 includes native data compression to reduce the physical size of large fact tables, thus reducing storage costs and improving the performance of I/O intensive operations. Data compression in SQL Server 2008 can be used to compress tables and indexes, and can be enabled at the partition level for partitioned tables. Data can be compressed by row (less compression, less CPU overhead) or by page (more compression, more CPU overhead). Extensions to the Transact-SQL table and index data definition language (DDL) statements that are used to control compression are provided. A new procedure stored on the system called **sp_estimate_data_compression_savings** is also included to enable data warehouse developers to implement and manage compression. Backup compression is also included to reduce the size and cost of backup devices, and speed up backup and restore.

As a relational DBMS, the SQL Server 2008 DBMS provides the ability to model relational, dimensional, and hybrid database schemas. It is manageable, integrated to the Microsoft Office system, affordable, proven, and supported by Microsoft.

Accessible Techniques to Manage Deployment

Any data warehouse, especially one that gets into data volumes over a terabyte, needs some level of management despite what a vendor may claim. Fortunately, SQL Server 2008 offers numerous accessible techniques to manage the deployment of a data warehouse rollout. A database professional would recognize the techniques and need only apply them in a Microsoft data warehouse environment. I name a few of them here, although there are many others.

Feature-function comparisons are interesting academic endeavors, but at the end of the day, it’s about the ability to build, manage, and deliver a data warehouse that achieves business objectives in both the short and long terms. One technique in SQL Server 2008 is to use indexed views. Views are simply stored SQL statements which provide transparency from table design. SQL Server 2008 allows for the indexing (materialization) of these views. The most common use of indexed views is as summary aggregates. The query optimizer can use indexed views to accelerate queries by orders of magnitude

in some cases. SQL Server 2008 also extends indexed views to allow them to be aligned with partitioned tables, making them far more useful for large partitioned tables.

Partitioning is a way of breaking up large tables into smaller, more manageable chunks to create smaller units for utility processing such as load, backup, and reindexing. SQL Server 2008 supports up to 1,000 partitions per table. Many data warehouses partition fact (i.e., transaction) tables by month or by date, depending on the load volume. This, and other techniques, should be used as needed for management convenience, and to control the cost of maintenance operations. In SQL Server 2008, partitioned table parallelism is specifically designed to improve the performance of a partitioned system. Now, multiple threads can work concurrently on a single partition. There is no need to tune the partition scheme to your hardware to get full utilization of the processing power. This simplifies management.

Finally, the join capabilities in the star schema query optimizations of SQL Server 2008 provide a high-performance means of querying a database that has been modeled dimensionally. This type of modeling is a common means of modeling a data warehouse. The query processing engine in SQL Server 2008 has been enhanced to provide improved recognition of star join patterns, better query plan generation for star join queries, improved costing of star join query plans, and a multiple bitmap filter query execution strategy. All of these enhancements result in a significant improvement in query performance against fact and dimension tables in a star schema–based data warehouse.

These are just some of the many leading-edge techniques built into SQL Server 2008. For a more in-depth look at data warehouse scalability features in SQL Server 2008, see the white paper, Introduction to New Data Warehouse Scalability Features in SQL Server 2008.

Delivering a Data Warehouse

Organizations need a data warehouse platform to support multiple different workloads such as reporting, ad-hoc querying, and analysis. They need a system that can provide predictable system response without succumbing to the runaway query. The Microsoft toolset has a variety of robust tools designed to cover the range of user access requirements. From basic to complex reporting to interactive access to data mining, the Microsoft business intelligence toolset is complete and integrated across the spectrum of user need.

SQL Server 2008 Analysis Services

SQL Server 2008 Analysis Services (SSAS) is an optional component of a data warehouse stack, but it is quickly becoming essential. SSAS is used to build a multidimensional OLAP cube structure.

The benefits of using SSAS cubes are numerous and usually it's the performance (of specific queries) that is cited as the main benefit. However, one of the lesser-cited benefits of using shared cubes is one that I've found to be among its biggest benefits.

Cubes are not just the storage format. These physical cubes are accessed through specialized access layers that can make numerous assumptions about the way in which the cube data is accessed. This access layer is one of the biggest benefits of cubes. You can get your users up and running with fairly robust slice/dice and drill-down capabilities just by building a cube and pointing an OLAP Services cube interface tool such as Microsoft Office PerformancePoint® Server, described later, Cognos, Crystal Decisions, or other data access tools at the cube.

SSAS really builds "hyper dimensional" structures since the cubes are modeled dimensionally, yet stored with every possible combination of fact-dimension values precalculated so that random access anywhere within the cube is fast.

One of the important SQL Server 2008 features is the Unified Dimensional Model (UDM), which allows access to Analysis Services data in multiple cubes in one query. With UDM, a model can access numerous data sources while presenting the end user with a single view. SQL Server 2008 Analysis Services is also more flexible about the types of models that can be imported.

Data mining has long been a means to attain high business value from a warehouse. It can make you aware of situations that may represent new market opportunities or business problems that have yet to surface. With seven powerful out-of-the-box algorithms, Microsoft data mining with SSAS provides a rich set of data mining algorithms for use by all levels within an organization.

Whether through data mining or simpler analysis, the data warehouse is a good place to develop and manage organizational key performance indicators, or KPIs. By summarizing and focusing the detailed data you deal with every day into something coherent for management, you help to ensure synergy and attain corporate goals. By using data warehouse data to support real-time KPI metrics generation, you can very effectively score and support strategic business objectives, monitor progress in real time, provide drill-to-detail capability, generate cause-and-effect models and, most importantly, measurably improve the bottom line.

The analysis capabilities in Microsoft products provide a strong KPI delivery capability to any organization. SSAS with data mining is an example of the Microsoft commitment to a robust, well-rounded toolset for data warehousing.

Reporting Services

SQL Server 2008 Reporting Services provides developer and end user tools for reporting. Reporting tools carry features to format and structure query results in a visually pleasing and understandable way. With SQL Server 2008 Reporting Services, users are afforded much more flexibility.

Users do not have to know any Structured Query Language—the usual means of interfacing to a DBMS. Reporting Services is also able to generate more complex reports with complex nesting and subqueries and distribute those reports to multiple channels in a publish-subscribe mechanism.

Drill-down within reports, which expands the usability of any report, is enabled in Reporting Services and there is a “Find” interface so that lookups can be performed quickly within a report set—for example, to find a specific customer number or product description.

To provide a richer report design and browsing experience, SQL Server 2008 Reporting Services includes the Tablix control. This control enables the user to display data as a table, or matrix, or using elements of each.

Integration with the Microsoft Office System and SharePoint

Microsoft Office Excel® is the most used business intelligence tool. It is so ubiquitous that its relevance has extended well beyond its initial spreadsheet focus into capabilities for working with all manner of data. The unique ability to flow information between SQL Server 2008 Analysis Services and Excel enables companies to take advantage of their investment in Excel skills and gain new insight through analysis of data in a data warehouse. Additionally, Excel 2007 enables the use of data-mining capabilities in SQL Server 2008 Analysis Services to provide powerful analytical capabilities to spreadsheet data.

Microsoft Office SharePoint® Server 2007 provides a classification for all corporate data. It is the collaborative portal application on the Microsoft Office system that has online publishing, version control, document approval and a search facility for internal and external sources organized in hierarchies with personalization. Office SharePoint Server 2007 is increasing its utilization as a portal to corporate data.

SQL Server 2008 Reporting Services is deeply integrated with Office SharePoint Server 2007. Objects such as reports, data sources, models, and resources are stored in Office SharePoint Server 2007 along with “metadata” about subscriptions and a few other pieces of functionality. This integration provides document versioning, workflow, and collaboration for Reporting Services, enabling it truly to provide robust, dynamic enterprise reporting.

Users are typically familiar with the Microsoft Office system products. Rather than forcing these users to use unfamiliar applications, SQL Server 2008 enables users to easily export reports to Microsoft Office Word and Office

Excel and to publish and share them within the organization with Office SharePoint Server 2007.

Previously, Microsoft did not directly compete with the market-leading BI front-end solutions. Instead, most organizations chose to implement a solution from ProClarity. Microsoft acquired ProClarity in 2006 and evolved it into Office PerformancePoint Server 2007, a single application that combines the functionality of the Microsoft Office system with ProClarity technologies to deliver scorecards, dashboards, management reporting, analytics, planning, budgeting, forecasting, and consolidation solutions.

Rounding It Out

The process of managing and delivering a data warehouse begins during its construction. Proper technology selection is important to success. It is equally important to overcome business challenges. If quality usability is absent from the data that is delivered with the technical architecture, the entire effort could be compromised. If business representatives are not active participants, the solution could be rejected. If the iterations are not planned to coincide with important business passions, corporate interest could wane. If training and support are not provided, the solution could be deemed too difficult to use. These factors are critical to success, regardless of the technology base used.

Non-technical Requirements for Data Warehousing Success

Users must be activated as co-conspirators in the success of data warehousing and BI. It actually should mean something to be a data warehouse user. Reasonable strategies to “programmize” the data warehouse always have positive outcomes. This includes providing training, access to support, feedback mechanisms, and automated descriptions of structure and data.

Data stewardship is a programmatic mechanism to ensure tangible business participation in data warehousing and BI. These extended team members are subject matter experts in the terminology and requirements for their subject area. Cultivating stewardship should be considered a requirement for Microsoft data warehousing /BI success.

Program governance is the mechanism for ensuring linkage between the projects and business priorities. Governance facilitates the continuance of budget, the communication of business direction to the build team and the support of upper management. It involves an Executive Sponsor, who can ultimately be responsible for strategic discernment and prioritization over the major additions of usage, subject areas, and data sources to the data warehouse.

Data warehousing/BI is a living, thriving entity within a business. Data, uses, and numbers of users will continually grow. Cultivating growth is achieved

through being responsive to direct requirements as well as through cultivating new requirements. A program that “gets the word out” will go a long way to soliciting users and goodwill.

The Intangibles

A great data warehouse/BI developer and end user experience is an ultimate requirement for success. Fortunately, the Microsoft toolset has provided the backbone of numerous careers, many of which are transferring skills to the Microsoft business intelligence platform. The interface to Microsoft business intelligence tools is intuitive, comparable to the familiar toolset in the Microsoft Office system.

Software partners for Microsoft abound. The ubiquity of Microsoft products and the accessibility of the tools to code development mean greater “buy” opportunities for the Microsoft business intelligence customer. It is a proven solution that is well supported. And you can be reasonably sure Microsoft will remain in business for the foreseeable future.

Conclusion

Ignoring or underperforming any business function due to lack of information is done at peril in today’s economy, which demands that companies compete based on their data and information analysis, not solely on product lines or operational efficiencies. It demands data warehousing.

Microsoft data warehousing and business intelligence delivers the value of information to its users.

Microsoft business intelligence components are built to modern standards of data warehousing and immediately open up previously inaccessible data to multiple interests within a business, allowing for trend and historical analysis of cleansed company data in areas such as customer analysis, product, pricing, organization, costs, geography, accounts receivable, promotions effectiveness, seasonal trends, returns, and so on.

For more information:
Microsoft SQL Server 2008
http://www.microsoft.com/sqlserver/2008/en/us/default.aspx

Please give us your feedback:

Did this paper help you? Tell us on a scale of 1 (poor) to 5 (excellent), how would you rate this paper and why have you given it this rating? For example:

- Are you giving it a high rating because it has good examples, excellent screenshots, clear writing, or another reason?
- Are you giving it a low rating because it has poor examples, fuzzy screenshots, unclear writing?

This feedback will help us improve the quality of white papers we release. Send feedback.