

---

# Enterprise Data Warehouse Benchmarks

**How to Make an Informed Decision, Minimize Risk, and  
Measure More Than Performance**

A White paper by Sid Adelman

---

## Table of Contents

Executive Summary .....	3
Why Pursue a Benchmark .....	4
Making the Most of the Benchmark Investment.....	6
Technical Games Some Vendors Play .....	9
Benchmark Objectives .....	12
Preparing for a Benchmark.....	12
Vendor Evaluation Criteria .....	14
Executing the Benchmark.....	15
Configuration.....	17
Evaluating and Measuring Results.....	18
Verifying and Reconciling Results .....	19
Qualitative Metrics.....	20
Considerations for Expansion .....	20
Communicating Results Effectively.....	21
Case Study – Cardinal Health.....	22
Comparison of the Benchmark to Actual Results.....	23
Conclusion .....	23
Appendices.....	24
Appendix 1 – Best Practices .....	24
Appendix 2 – Project Plan .....	25
Appendix 3 – Sample Report ‘Table of Contents’ .....	26
Appendix 4 - Checklist for Success.....	27
Current (Looking out 12 months) .....	27
Future (Looking out 12 – 36 months) .....	27

---

## Executive Summary

The historical and continuing high rate of IT project failure and budget and schedule overruns, coupled with senior management's increasing insistence on quantifiable investment justification, leave IT in a position today of needing to control cost and mitigate risk like never before. Many organizations are facing the need to replace their data warehouse with a second- or even a third-generation implementation. Many are considering an enterprise data warehouse (EDW) that goes beyond their earlier data marts – an EDW that incorporates all or most of the areas that run the business and from which critical strategic, tactical, and sometimes operational decisions are made. They have several options in terms of vendors, technology, and architecture – and they can't afford to make a bad decision. One very effective way of 'test-driving' the options to see how well they'll perform in one's actual environment is through benchmarking. With first generation data warehouse implementations, many IT shops did not bother to use any benchmarking, instead opting to adapt their existing OLTP database for their data warehouse. What they have learned is that not only are no two databases alike, but also that no two implementations of the same database are alike. Others settled for using industry-standard benchmarks as a guide instead of conducting their own benchmarks. If you've seen one benchmark, you've only seen one benchmark. 'Standard benchmarks' bear little resemblance to the real world.

Benchmarks may be expensive and time-consuming, but when you use *your* data, *your* applications, and *your* service level requirements, benchmarks provide a degree of insurance that the system you're buying will be able to scale to your intended volumes of data, complexity of workload, and number of concurrent users. Benchmarks can also validate vendors' proposals and provide you with the quantifiable justification for your recommendation to management, thereby mitigating risk and allowing you to properly set expectations and predict overall cost. Benchmark results should reveal the performance characteristics of your intended system as well as some insight into the effort and difficulty of implementing the system. They should also give a strong indication of the capabilities and responsiveness of the vendors.

A number of organizations have effectively used realistic and challenging customer-specific benchmarks to determine relative performance and capabilities of proposed systems, to keep vendors honest, to accurately assess development time and cost, to properly set user expectations, and to minimize their risk and maximize their overall success. This paper documents their approaches and experiences and covers their recommendations of what works, what doesn't, lessons learned, traps to avoid, and how to ensure that you use benchmarking effectively to get the maximum benefit for your effort.

---

## Why Pursue a Benchmark?

Given that the effort and activity required by a benchmark could be profitably spent on other endeavors, the obvious question is why should you pursue a benchmark at all? Conducting a benchmark will allow you to compare the scalability, performance and manageability of two or more platforms. There are three key reasons to run a benchmark:

1. To significantly reduce the risk of selecting the wrong platform and to provide hard data to support the final decision.
2. To test using anticipated growth in data volume, number of concurrent users, complexity and size of the workload, or situations where the ETL process could have problems, such as short windows for load or update.
3. To gain additional information about the complexity and effort involved in developing an EDW.

There are standard benchmarks that claim to provide an indication of performance or value, but standard benchmarks should not be relied upon when making a platform decision. Standard benchmarks don't:

- Measure how much effort it takes to implement these systems.
- Tell you how much effort it takes to administer your proposed configuration.
- Help you determine what configuration will meet your workload requirements.
- Provide information about recovery.
- Provide information about the software and how it actually functions.

More importantly, standard benchmarks tell you nothing about the impact of your specific workload, problems that may occur in your environment, whether your availability and response-time service level agreements (SLAs) can be met, or about the quality of a vendor's support. Standard benchmarks can provide a preliminary indication of relative price/performance, although this is easily manipulated by vendors who sometimes play games with pricing. Standard benchmarks would have you believe that the results are an accurate indication of the system capabilities, but there is little likelihood that these results will be exactly or even closely replicated in the world in which you live.

The Transaction Processing Performance Council (TPC) is a group of vendors that provides rules and workloads for benchmarking. TPC benchmarks are frequently used by database vendors, and their results are often touted as an indicator of their products' capabilities. While the TPC might appear to be an independent group that is objective, the reality is that all of the TPC rules are governed by the vendors themselves, not by customers or other objective parties.

---

The TPC-H benchmark has been designated by the TPC as an ‘ad-hoc’ benchmark. The flaw in the benchmark is that it’s actually a static, predefined benchmark and so, by definition, it isn’t ad hoc. The queries and the data are known to those running the benchmark, which allows benchmark experts to configure the system, design the database, and optimize in other ways to get extraordinarily good (and unrealistic) results for that and only that very specific workload. Yet another flaw in the TPC-H benchmark is that the rules and data were released years ago, with some portions of the benchmark actually dating back to the early 1990s. This means that the database vendors have had years to perfect their benchmark executions, further reducing the likelihood that TPC-H results could ever be replicated in a customer environment.

### **Real-World Scenario**

Cardinal Health in Dublin, Ohio, needed to re-architect its data warehouse and needed to be sure the new platform would scale to handle its anticipated query workload with good performance and good manageability. They knew that the choice of the hardware configuration and DBMS was central to achieving their objectives, so they decided to benchmark their workloads with two different vendors. They conducted the benchmark in such a way as to ensure an objective comparison and a rigorous enough test to accurately project their costs and results. They have shared their approach and many of their guidelines which serve as a very effective ‘how-to’ in planning and executing effective benchmarks. The Cardinal Health approach is described throughout this paper.

---

## Making the Most of the Benchmark Investment

It costs about the same to do a benchmark right as it does to do it wrong. Remember, the goals of the benchmark are to generate information to help determine the best possible decision and to reduce the risk associated with that decision. It's critical to establish fair and objective requirements and measurement criteria that don't favor one vendor over another. If the benchmark is 'rigged,' you won't measure or understand what each vendor's system can really provide; this increases the risk associated with making a good decision rather than decreasing the risk. Having an outside, independent, and vendor-neutral consultant/auditor goes a long way to ensure fairness, reduce internal questions of fairness, and helps assure that vendors won't attempt to manipulate and misrepresent the benchmark results.

You absolutely need your benchmark to yield comparable and defensible results. You want to be in a position to comfortably answer questions from management or from your detractors who could ask embarrassing questions about how the benchmark was conducted. You want to know that the benchmark was conducted in such a way that it truly represents how your system will be running once it's in production. If you're going to recommend a particular platform, use this opportunity to test how it can really deliver.

Each vendor should be forced to play by the same rules. Despite the obviousness of this point, it's frequently ignored. To make a meaningful comparison, the machine power that each vendor brings to the benchmark must be comparable. The vendors should all have the same lead time, the same amount of time to run the benchmark, and be working with the same level of data quality. For example, if the first vendor is required to spend time cleaning data or correcting benchmark scripts, DDL and queries, vendors who run later will have a significant advantage. They will have more time

to rerun tests and to optimize their system based on the benchmark requirements. What may appear as better performing technology may in fact be a result of more time to do this optimization. (Note: every vendor performs optimization, which can take many forms.)

Just because you worked hard to establish a level playing field, don't expect the vendors to always accept your terms. We often see vendors attempt to change, skirt, interpret, or ignore established rules in an effort to make their system seem superior. If vendors want to skew the results in their favor, they may suggest the following:

**Don't waste your time. We'll take care of the benchmark and let you know the results. We'll even send you the PowerPoint slides for your presentation.**

You want to be involved in all phases of the benchmark, including the design of the databases, the initial database loading, the monitoring, administrative activities, and any tuning that may be required. You want to see how much effort is involved in these processes and how much your people will need to know to make the system work. Be aware of the administrative activities associated with each vendor's product. Some would suggest that you don't want to see how data warehouse sausage is made. In this case, you do.

**We just want to benchmark one capability of the data warehouse. We are only interested in, for example, load time, most intense query, or maximum number of concurrent users.**

This type of benchmark will not reflect true workload you will be experiencing, but will only provide information about this single activity. This should not be the basis for a vendor decision.

---

**You don't need to run with your projected volumes of data. You'll be able to extrapolate from the 10 percent we are suggesting.**

Some performance projections are linear; many are not. Don't assume linear performance. Run the benchmark with your entire projected data volumes. With new technology, with the additional size of memory and possible solid state disk, the performance of that small subset will be misleading and will indicate much better performance than can be achieved with a full set of data.

**We will tune the system to give you the best performance.**

A database that's tuned to perform well for known queries may be tuned terribly for other queries down the road. Vendors are experts at tuning systems for known workloads, and most customers don't have the expertise or the amount of time to devote to tune each poorly performing query. The vendors should fully disclose their tuning efforts including use of indexes, partitioning, caching, ordering of data, rewriting queries, creating summary tables, and managing workloads.

**The important part of the ETL process is the load, so that's what we will be measuring.**

The ETL performance could be the critical performance measure. The extract, sorting, splitting, pre-processing, as well as the index builds and summary table creation and anything else the vendor has to perform, will be a major portion of the ETL elapsed time. Be sure these are all included as you're determining how long your complete ETL process will take.

**You don't need to include all your data. We can generate data that will reflect your real environment.**

It's likely that your real data has some interesting and challenging outliers. These skews can cause performance problems for some systems. Use as much of your real production data as possible. Also, generated data can cause extremely unrealistic results if compression is employed, especially with repeating values and non-representative distributions of data.

**We will take care of the hardware configuration.**

Be sure that the configuration running the benchmark is the same as the one they're proposing and the one you're planning to purchase. Don't let a vendor extrapolate results for the proposed configuration from test results on a different configuration. If you're going through the effort of running the tests, run them on the configuration you're considering. The runs should always compare configurations that are equal in capacity and power.

**Let's run with a few of your queries; we don't need to run all of them.**

Even though your primary plan might be full-table scans, if there will be other activity, be sure to include those other queries in the benchmark. You will want to include as much of your projected workload as possible to not only determine which vendor is better but also to determine if the vendor's proposal will satisfy as many of your requirements as you can anticipate.

---

**The most effective way to measure query performance is to run a single query at a time.**

This isn't true. Systems behave very differently as concurrent workload increases. And the mechanisms for handling concurrent workloads vary considerably with significant implications for the variability and predictability of end-user response time. Unless you're acquiring a system that will be dedicated to a single query at a time – which is very unlikely – it's important to include tests with query (not user) concurrency levels equivalent to what you expect in the production environment. This should be a measure of 'queries in flight,' not queries that are queued waiting their turn and not simply a measure of concurrent sessions. It's also important to reflect the expected query arrival rates as some vendors will build 'sleep time' into their benchmark scripts to shelter their technology from concurrent workloads.

**We will provide you with throughput numbers to measure our results.**

Throughput is the amount of work completed in a given period of time – for example, the number of queries per hour. Throughput is an effective measure for batch systems but not for systems with end users. Throughput isn't a reflection of response time as it does not account for the time that submitted queries remain in queues. It's also

important to measure end-to-end user response time (not only the time that the query is initiated within the system) so that the queuing time is included in the measurement. Finally, end-user response time variability (variance or standard deviation) should be measured during concurrency tests to understand the end-user experience and the consistency and predictability of the vendors' technology.

**We won't give you any time to prepare.**

This is sometimes called a 'black box' benchmark. In this type of benchmark, the customer shows up day one with their data tapes, and expects the vendors to load the data, run the queries, and exercise at the same level they would have if given time to prepare in advance. This gives the vendors no time to deal with data quality problems, bad SQL, problems in integrating other ETL and BI tools, and working out bugs that always occur in benchmarks. This is a useless exercise in which the customer learns nothing about the vendors' capabilities, wastes time and money, and will provide no relevant or valuable recommendations to present to management. To paraphrase Mark Twain, 'Never try to teach a pig to sing. You'll waste your time and annoy the pig.' A black box benchmark will definitely waste your time and definitely annoy the vendors.



---

## Technical Games Some Vendors Play

Yes, it's true. Some vendors will play games and attempt to skew the results of their benchmark by interpreting your directives in their favor, by manipulating the way the benchmark is run, by misrepresenting their results, by not giving you the full information needed to audit the results, and by not fully complying with your specifications. This isn't putting their best foot forward; it's their attempt to misrepresent their capabilities. So, let's discuss some of the ways vendors might try to fool you.

- It's important to understand how the vendor is going to run the tests, and understand how those tests match your expectations. If there are differences between your benchmark plans and the vendor's plan, there is work to be done to decide if the vendor's approach is what you want, is comparable to other database vendor execution plans, and if it will be relevant to your testing goals.
- Vendors may suggest that queries be altered in small ways to eliminate or reduce the amount of data returned to the client system that initiates the queries, thus eliminating some portion of the time it takes for these queries to transfer their result sets and write their results to client file systems. It can be debated that the client and network time needed to manage the query result sets isn't part of the database workload and thus should not be used in metrics to compare systems. It's important that any technique used to reduce or eliminate network and client time be completely understood and that it provides consistent, comparable results between vendor benchmarks. The complete query result sets must be materialized within the database, otherwise workload performance and resource requirements will be misrepresented. Thoroughly understand any modifications vendors may suggest for queries and ensure that these modifications provide comparable results across all benchmark executions and represent the work your system will be performing in production.
- It's important for you to state that the benchmark database design/structures will be included in the audit along with the SQL and results. If not, the benchmark system may be redeployed, and your database may be removed so that no audit of the system or its database can be accomplished.
- The vendor may choose to add indexes or other performance enhancing structures for the tests, and delete them before the system is audited. The vendors should be specifically notified that this is unacceptable.
- Vendors will often attempt to justify why some portions of the benchmark should be deferred, should not be run concurrently as the benchmark states, or should be dropped entirely from the benchmark. These are usually activities that are resource intensive and would have a measurable effect on key results. These activities were included because you knew they were part of your environment and needed to be considered in comparing vendors and in better understanding the performance you will be experiencing in your production environment. Deferring or dropping these activities will not allow you to properly compare the vendors and will not help you understand the hardware you will need.
- The vendor may alter the parameters of individual sessions so that, for each session, the behavior of the database optimizer is altered. They do this so that the costs for accessing different objects, such as tables, indexes and materialized views, are represented as more or less expensive than they would normally be, thus encouraging or discouraging the use of these objects on a session basis. The vendors should be told that altering anything that would affect the functioning of their own optimizer will not be allowed in the benchmark. The vendor may try to use predetermined query access paths which are stored in the database. These stored plans may override the default behavior of the optimizer and give

---

that vendor an unfair advantage. The vendor may rewrite queries to present SQL to the database that delivers desired optimizer plans over default plans that are created by customer supplied SQL. By overriding the optimizer and customizing the access paths, you won't see the types of performance you will actually achieve in your production environment.

- The vendor may alter system-level database parameters between benchmark test executions so that the system executes more optimally for each test. Often these changes are very specific to the test being executed and would not be optimal for normal SQL execution. An example of this might be enlarging the database sort area to support index creation on large tables after loads. This helps the load process, but does so at the expense of query performance. If the targeted, future database requirements can't tolerate this level of specialized tuning, and it rarely does, it should be disallowed.
- A benchmark workload usually includes a set of queries that is to be run at the same time. Sometimes a query may acquire system resources that block other queries from running until the first query finishes and releases the resources. The vendor may know that their execution time for the complete workload will be too long unless they terminate the blocking query. They may request that they be allowed to terminate the query so the other queries may finish. The problem is that the query is part of the workload. Terminating the query will result in a partial timing result for that query and, in reality, a partial result for the complete workload. However, from the vendor's perspective, eliminating the query has reduced the scope of the workload thus giving them a better chance of appearing to support the benchmark workload. This comes back to how the benchmark maps to your production environment. If your production

workload has the types of queries that may cause other queries to be temporarily blocked, then the benchmark should include those queries that do acquire a blocking system resource.

- The vendor may partition data to extremely small partitions and non-real-world partitions to support partition pruning/data reduction characteristics needed for query performance in the benchmark. They may partition the data so that the most used data is stored in more finely partitioned parts of the table and cold, rarely used data is stored in large, inactive portions of the table. In some cases, the unused data may also be stored on fewer larger/slower disks, perhaps with compression, so that data is stored most efficiently. But from a benchmark perspective, the data access performance isn't fairly measured. If all the data loaded isn't accessed, the performance of this design isn't realistic, cannot be understood, and cannot be properly evaluated. Only the performance of the data on the faster disk and in the most optimally designed parts of the table will be measured, and this will distort the results in favor of the vendor. The benchmark should map to the combination of types of disks you expect to run your production workload, including what data (for example, active or rarely accessed) you intend to place on which types of disks.
- The vendor may delete rows from tables not used in queries. This gives the vendor a less expensive configuration and better performance, but those deleted rows will be a part of your production configuration and should not be allowed to be deleted in the benchmark.
- Loading data into tables using compressed format will often allow queries to deliver faster I/O performance, but there is additional cost in the load, and that cost should be weighed against the improved query performance.

---

The cost of compression during the load will not be understood unless the same data is loaded with and without compression. Updates against compressed data have different performance characteristics than those against non-compressed data. Unless updates are run against the compressed data, only a partial view of the performance and usability of compression will be valuable in the benchmark results. Knowing the update requirement will help determine if compression is the optimal choice. If compression will be used in the benchmark, additional runs should include uncompressed data.

- When loading benchmark data, sometimes the data isn't clean; there may be abnormalities in the data that prevent it from being loaded. Some vendors drop the anomalous data records rather than report the problems and resolve the issues. This will result in less data loaded and, as a consequence, faster loads and faster running queries. The vendors should be told what to do when there are data quality problems so that you're able to fairly compare the vendors' results.
- If no updates are included in the benchmark, the data may be loaded so that there is no room for data expansion. This leads to the erroneous perception that the data does not need much storage once loaded into tables. The performance of updates and space management using this type of design will not be measured during the benchmark. The true costs of this design will

not be fully understood and will generally be understated. Systems being sized based on this design will surely be undersized. If you plan to expand the data – as you surely will – tell the vendors about your expansion plans, and ask them to size their configurations appropriately. You should also consider updates in the benchmark.

- You will be asking the vendors for a certain level of user-session concurrency. This can be interpreted in a number of ways. It can be the number of users logged on to the system, it can be the number of users actively running queries at the same time on the system, or it can be related to the number of concurrently active and running queries in the system. Vendors may choose the interpretation that best shows off their performance capabilities. A vendor can even interpret concurrency goals as logging on the number of user sessions requested, and then running queries from each user session one at a time or at some other level of concurrency. Sometimes it can be as extreme as running one query at a time. With some databases, system-level tuning parameters influence and control the actual number of concurrently active users with active executing queries. With others, it's a built-in feature of the system to limit user query concurrency. It's important that you clearly state your concurrency goals and, most importantly, how you're defining concurrency.

---

## Benchmark Objectives

If you're diligent in your preparation, and you've closely managed vendor participation, you will see many benefits from your benchmark efforts. You should expect a successful benchmark to yield:

- Management and organizational confidence in the final decision.
- Hard data to support the choice of platform and a configuration that will support your future requirements.
- Confidence that your configuration will support your SLAs for availability and performance.
- An understanding of realistic hardware and software costs.
- Insight into the effort and resources needed to build and maintain your system.
- Reliable capacity planning estimates for additional hardware as data volumes and the number of concurrent users grows. Information and metrics about the scalability of each vendor's proposal, including scalability for data, number of queries, number of users, and complexity of the workload.
- Solid understanding of the vendor's capabilities and level of support.

Cardinal Health had a number of objectives for its benchmark. The first was to simulate the needs of the current as well as the future environments. They needed to understand the performance characteristics of each vendor and to know that whatever they chose would be able to scale to the volumes they were anticipating. Since Cardinal's data warehouse included multiple data marts and since some queries would access more than one data mart, they needed to understand how those queries would perform. Cardinal was expecting a mixed workload with complex ad-hoc queries, and they wanted to be sure the benchmarks actually ran those queries.

## Preparing for a Benchmark

The first step in preparing for a benchmark is to know where you are now and where you will be in three years in terms of requirements for scalability, complexity, and the ability to incorporate new and varied functions, including the possibility of supporting one or more master databases. The next step is to understand the costs involved. A benchmarking effort is going to cost both you and the vendors. Although it's not common, a vendor may even charge you for running the benchmark. You will also want to include travel and living costs. It can take from four to eight worker-weeks to prepare for the benchmark, and an additional three to five days on site to perform it. This all totals about two to five weeks elapsed time. The times involved depend on the complexity of the work, the data volumes, the number of users, the number and types of tests, and the ETL and BI tools to be used. You may also want to include a consultant/auditor on the project to be involved in all aspects of the benchmark. This can help to keep the vendors from misrepresenting performance. For example, the consultant/auditor would be aware of preload activities, such as sorting, that could give a vendor an unwarranted edge. The consultant/auditors would also be watchful in how the metrics are represented. While benchmarking costs are significant, the cost of purchasing the wrong system or making the wrong decision will certainly be more costly in the long run to your organization as well as to your career.

The benchmark team should include a data warehouse architect, one or more DBAs, key business users (those who will be creating many of the queries), operational managers responsible for the infrastructure (administration, facilities, computer room manager), technical personnel, and the data warehouse director. If the benchmark includes the use of a third-party ETL or BI tool, bring someone with an architectural understanding of the tool. Don't expect the vendor to have that expertise. Interested executives, perhaps the CIO,

---

might want to come for the last day or two of the benchmark. The team should include advocates for each platform to ensure that each is measured fairly, but not zealots who don't have an open mind.

The work involved includes preparing a test plan; writing scripts; writing SQL; and gathering, cleaning, and preparing the data to be used in the benchmark. The benchmark should be thoroughly tested before distributing it to the vendors to be sure that all the ETL processes and all the queries run successfully. All the SQL submitted should be in ANSI-standard SQL to eliminate any benefits from incumbent vendors who may've already tuned their queries. The preparation should include documenting specific SLAs for ETL run times, as well as SLAs for specific critical queries. Whether the conformance to these SLAs is to be mandatory (meaning that if they're not met, the vendor's proposal is to be rejected) or just "nice to have," should be determined before the benchmarks are executed.

The data load process isn't always fully measured in a benchmark. The time to prepare the data for loading, if multiple processing and data "landing" steps are executed, needs to be reported. This includes the preparation in order to load the data which might include multiple steps; reading, processing and writing data to disk multiple times even before the data gets to its final loadable state. It's important to understand

all these steps, the time, resources and the human effort, whether or not data loads are a part of the benchmark, and whether or not they will be a factor in evaluating the vendors. If specialized storage is required to load the data into the database, this should be reported so that the load performance can be repeated at the customer site.

It's particularly important that the customer is clear about what they are expecting the vendor to perform, and what results they are expecting from the benchmark. This should take the form of a checklist of criteria for success (see Appendix 4 – Criteria for Success). For example, the customer may have a requirement for a minimum number of queries running concurrently, for a workload with a certain level of complexity, how fast the queries are expected to run, and a requirement represented by a demanding availability SLA, such as 99 percent during scheduled hours. It's doubly important that those requirements are in the realm of reality. The customer should not be expecting results that no vendor can achieve.

Cardinal Health had a benchmark team composed of the manager of the data management team, a data analyst, a senior DBA, the director of data warehousing, a technical architect, a data modeler, the BI manager, and business partner liaisons. The same team members participated in the benchmarks with both vendors.

---

## Vendor Evaluation Criteria

Vendors should be measured on criteria that are important to you. These are some of the criteria you should consider:

- The vendor's ability to satisfy the organization's current and future business needs in the most expedient and cost-effective fashion. This also includes the vendor's ability to understand and provide meaningful capabilities in your industry, as well as the vendor's ability to support future requirements and to take you to the next step – even if you're not sure what that next step will be. That next step could be data mining, integration with the EDW, predictive analytics, or a real-time or near-real-time application that would be part of an operational BI implementation.
- Your applications may not involve complex queries, but if you expect any in the future, you will want to evaluate the ability of the vendor to handle complex ad-hoc queries.
- The ability to scale, to handle the projected growth rate including concurrency, arrival rates, much greater data volumes, and a larger footprint.
- The flexibility to handle diverse business models that included different volumes of data, different numbers of users, as well as the ability to handle ad-hoc complex queries.
- If the requirement arose to move to another platform, how difficult would the move be?
- Administrative ease and the cost of administration.
- As the data model is expanded and changed, what is the effort involved in implementing these changes? It should be noted that every organization has made modifications to its original data models.

- The intrinsic ability of the underlying database to support the organization's technical needs.
- Expansion capabilities that also raise the issue of what kind of outage would be required when an expansion is needed. The role of the vendor in the expansion should also be considered – who will be doing the work?
- High availability that supports stringent SLAs for performance and availability including the possible need to be running 24/7.

In the benchmark, you will want to verify and validate that:

- Queries and reports ran successfully (capture metrics about: when submitted, when started, when completed, number of rows returned, and CPU and I/O utilization).
- ETL process ran successfully (verify with row counts, and also verify that the database is available for access).
- ETL times are acceptable (you specify what is acceptable based on your windows).
- Query response times are acceptable (you specify what is acceptable based on your response time SLAs. The SLAs for certain critical queries should be met).
- Concurrent activities ran with the proposed configuration (these could include backups and other utilities).
- The effort to administer is understood and not out of the range of expectations or technical expertise of your staff.

The independent outside consultant/auditor is a key player in this verification and validation process.

---

## Executing the Benchmark

As with the preparation phase and development of success criteria, guidelines are needed to govern the process. Although it may seem like an obvious point, the benchmark should be run using the configuration the vendor is proposing. If given the opportunity, many vendors will run the benchmark on a larger configuration, providing results that you won't be able to verify or duplicate on the system configuration you actually buy. If the benchmark is only using half of the data volume expected in production, then the vendor should use half the configuration they are proposing. Oversized systems can have far more memory or cache or processing capabilities relative to the data volume than will be the case in production. This would result in very misleading results.

Cardinal Health requested proposals from vendors based on its projected one- to five-year growth patterns. Each vendor proposed a solution for those requirements, and that is what was tested in the benchmark.

It's also very important that the data used in the benchmark represent the true characteristics of your data, including the outliers – the skewed data and the aberrant data clusters. (Your data will not be evenly distributed). You will want to bring the representative volumes of data from your environment today, as well as what you expect to have in three years. Allow enough time for this task, and assign the task to people who are very familiar with the data, know how to extract it, and also know how to validate that the data represents your intended environment. Data warehouse installations typically grow from 20 to 50 percent (or more)

each year. The growth will be in additional historical data, more atomic data, more granular periodicity (loading daily versus monthly), new subject areas, additional external data and new columns to be added to existing databases.

The time to extract, transform, and load the data will increase with the size of the database. The vendor's process for executing this task will be indicative of performance, effort and difficulty of using their system. As in any production data warehouse, the transformation part of ETL will be labor intensive, and probably machine resource intensive. Whether you hand-code the conversion or use an ETL tool, the effort and the machine resource should be measured. It's very important to measure the whole ETL process, not just the load time. If you're planning to benchmark an operational BI environment, the benchmark will take longer, and the test plan will be more comprehensive and complex with far more concurrent users.

Once the hardware, data and loading issues are addressed, query performance must be evaluated at various user volumes including:

- Average number of users
- Peak number of concurrent users
- Peak periods, such as beginning of the day and end-of-month

The number of concurrent active user queries should be benchmarked for today's environment and for the environment anticipated in three years.

---

Above and beyond the queries that you will provide the vendors ahead of time, you should also introduce new queries not known during design in order to see how long it takes to execute these truly ad-hoc queries and with what degree of difficulty these queries can be introduced. These “surprise” queries should be different enough from the base set so as to provide you with a sense of what to expect when your users introduce unplanned queries – and they will.

In addition to surprise queries, you should also push the vendors to test a “surprise” disk, server and/or node failure, and also ask them to add a hardware module. The willingness or unwillingness of the vendor to demonstrate performance in the case of unplanned failures and system expansions will provide information about how the failure is actually handled. This should be done as part of a test of each system’s ability to meet your specific SLAs for load times, query response times, and availability.

Cardinal Health developed a set of guidelines to govern the execution of their benchmark. They conducted the benchmark based on current needs and what they intended to buy in two and in five years. They compared the vendors’ proposed configurations, and they used their own data on each vendor’s machine. This was full volume data from an existing data warehouse and from data marts. They did not allow summary or aggregate tables to be used, which would have significantly improved the vendor’s performance. They allotted time for Cardinal personnel to be in the benchmark center watching and monitoring the proceedings, and had the same benchmark team with each vendor. They specified that the benchmark should last only five days, and they captured log files of all the benchmark activities.



---

## Configuration

Since performance is a critical success factor for large data warehouse implementations, the benchmark should be exacting about how it will represent the performance once the system goes into production. The configuration will be based on power, price, and anticipated performance requirements. Configuration incorporates what most customers already do with capacity planning.

Using the configuration that matches what you plan to purchase is only the best approach if you're benchmarking with data and workload volumes that match what you will have in production. If you're only benchmarking 25 percent of the data volume you will have in production, then you should force the vendors to only use a system that is 25 percent of what they will sell. Otherwise, you will set expectations during the benchmark that will not be fulfilled when the system is used in production.

Configuration needs to include today's environment (one set of runs) and the future environment, as well. The last thing the team should be asked is, "Why didn't you anticipate our merging or acquiring an organization twice our size?" This means that, even though you're not buying a set of hardware and software to support an environment of three times what you have today, you still should be running a test with the larger volume of data and number of concurrent users, a more complex set of queries, and with an expanded portfolio of applications that is likely to be required in the future. It's important to assess the vendors' plans for use of new technologies in their planning horizon. A prime example is assessing how the use of high-speed

solid state disk (SSD) technology will be leveraged. SSD should establish new levels of query throughput and response time that will impact benchmark results.

No vendor should be allowed to benchmark with a bloated configuration that does not match the test workloads. For example, running a two terabyte workload on a 50 terabyte configuration will lead to results that will not be duplicated in production – unless you buy the whole 50 terabytes.

The configuration needs to include not just the hardware but the software you anticipate running, such as Informatica, Ab Initio, Cognos, and MicroStrategy. The configuration should include customer-specific availability requirements that might incorporate a dual active environment, and a disaster recovery stack of hardware and software and the mechanism to keep the disaster recovery databases current enough to support the never properly anticipated, catastrophic earthquake.

The people in your shop who support the infrastructure should have given you a few questions that can be answered by nosing around the vendor's machine room and asking specific questions about the electricity requirements, the cooling requirements, floor space (count the floor tiles), and anything extraordinary that might be needed to support the vendor's configuration such as any retrofitting of your machine room.

---

## Evaluating and Measuring Results

If you have provided the vendors with a well-defined benchmark, the results should be straightforward and easy to measure. Cardinal elected to measure these areas:

- Flexibility to be able to execute requirements heretofore unknown and be able to support multiple subject areas
- References – what the references reported for their performance, availability, and vendor support
- Initial cost
- Total Cost of Ownership (TCO) including staff and ongoing maintenance
- Scalability
- Manageability including the staff size, skills and administrative efforts
- Availability during scheduled hours
- DBMS features
- Portability
- Vendor industry comparison
- Vendor financial viability
- Vendor partnership

In addition, these technical points should also be evaluated and measured:

- Data storage scalability – How easy is it to grow your system's storage capacity?
- Processing source scalability – How easy is it to grow the processing capacity of your system?
- How do queries scale as more data is loaded? Do they take a predictably longer period of time to complete? Is the performance linear?

- If more hardware is added, what is the speed up of the queries? Do they speed up? Do they speed up linearly, and do they speed up predictably?
- Are there ways to have the database manage the work on the system so that the proper resource allocations are made to the proper users and applications?
- Is it possible to preserve response times for queries as more work is added to the system? Queries' response times will vary, but do they vary within a tolerable range?

There are many ways to express throughput. System throughput can be reported in raw bytes per second measurements or in more ways by expressing the throughput in terms of the amount of data that, if expanded, would be flowing through the system. Sometimes, throughput measurements are reported at a component level which, unless the system's configuration is balanced, it will not reflect the system's throughput. Unless the throughput numbers are explained, wrong conclusions are usually the result. It's important that the vendor thoroughly explains how the throughput numbers are calculated and represented.

It's not always clear if the data is being loaded in compressed or uncompressed formats, and thus it's important to understand the load speeds presented with an indication of the level of compression.

Cardinal also applied weights to the different categories. For example, vendor partnership and vendor financial viability were not as important to Cardinal as were flexibility, references and TCO. You should also determine whether or not each of the criteria is equal in importance – the answer is probably no, so it should be carefully considered.

---

## Verifying and Reconciling Results

You will want to be there when the benchmark is run. You will want to participate in reviewing the measurements, such as load times, response time and resource usage. If you choose to have an independent auditor/consultant participate in the benchmark, have that person or team verify the key measurements and validate any conclusions from the measurements.

Validating results also means the number of rows returned is the correct number compared to your expected number. You will not be able to validate all the results, but sample and verify the most critical queries.

In some situations where queries don't finish, no instructions are given to the vendor about how to report these queries. They may be reported as the elapsed execution

time before they were terminated. In other cases, no time may be reported for these queries at all. On graphs, any of these types of timings may be reported and can provide false impressions of the vendor's performance if the timings are not explained. When presenting overall workload performance, these incomplete measurements can skew the final reported results. Graphs can be dramatic and misleading, and in the situations where numerous queries or workloads are presented, it can be difficult to see the "underreported" results.

As part of the benchmark guidelines, Cardinal Health indicated that they would capture log files of all actions. This is an excellent way to verify exactly what transpires during the benchmark process.

---

## Qualitative Metrics

In some cases, talking to vendor references can be an alternative to running a benchmark. References are an excellent way to augment a benchmark by helping to evaluate criteria that can't be quantified. From references, you can get very important information, such as whether the vendor's claims match the reality of the system once implemented and the level and quality of vendor support.

You may also want to consider each vendor's relationship with the other vendors whose tools you intend to use. Look for technical relationships that include testing with each other's products and the R&D departments talking to each other.

Cardinal Health assessed a number of qualitative characteristics, such as the vendor's willingness to do the unexpected, the vendor's demonstration of data warehousing knowledge, and third-party tool support. A key metric was whether or not the vendor would allow them to speak directly with references without the vendor being present. Again, the vendor's response to this request will tell as much or more than what you will be learning from the reference. All of these items are important factors that should play a key role in the decision-making process.

## Considerations for Expansion

Will your application be the end-of-the-line for your data warehouse implementation, or will you be growing or expanding in some other way? You need to know the answers to these questions:

- Will you be growing the data volumes?
- Will you be expanding the user community with more and varied types of users?
- Will you be writing more and different types of queries?
- Will your use expand into other areas such as data mining or text mining?
- Will you have more demanding SLAs for scheduled hours and availability?
- Will your needs expand into a real-time application?
- Will your investment be protected as technology changes? Will you be able to co-mingle generations of hardware in your system?
- What about concurrent session scaling? Will you be able to grow the system's capacity to manage more concurrent work on the system?
- Does the database become more complex to manage as the system grows?
- As your data warehouse matures, will your data warehouse be able to accommodate those changes? How easy will it be to change your database design?

If you do plan to expand, or even if expansion is a possibility, you don't want to lock your organization into a situation where you will have to discard the configuration, and install a new infrastructure. If you're planning any of these expansions, ask the vendor to tell you and to document how that expansion would take place, including what efforts would be involved, what would have to be thrown away and what new equipment would be needed.

## Communicating Results Effectively

You will want to fully document the results of the benchmark, but this can be an enormous amount of information by the time your work is completed. The key to communicating the results will be to highlight those that are most meaningful. We all know that a picture is worth a thousand words, so in addition to the raw numbers and your conclusions, be sure to use charts and graphs that highlight the key results.

Figure 1 is an example of a chart that makes the comparisons that were most relevant to Cardinal. This chart is similar to one that Cardinal Health used when presenting the results of their benchmarks to management. The data is contrived, but the methodology is Cardinal's. The chart clearly shows which vendor performed better, and it's obvious how both vendors compared to the "ideal" that Cardinal had put forth.

The chronology of the evaluation process Cardinal used was:

1. They selected their key criteria (12 in this case).
2. They defined their "ideal" level for each criterion. The closer to 100, the heavier the weighting.
3. They measured each vendor on each criterion and plotted the scores, along with their "ideal."

From the chart, you can quickly determine that:

- Vendor B scores higher than Vendor A, except in the areas of portability and vendor financials.
- Selection of Vendor B would constitute a compromise in the areas of portability and vendor financials. By weighting the scores by importance, it's easy to determine the impact of selecting one vendor over the other.

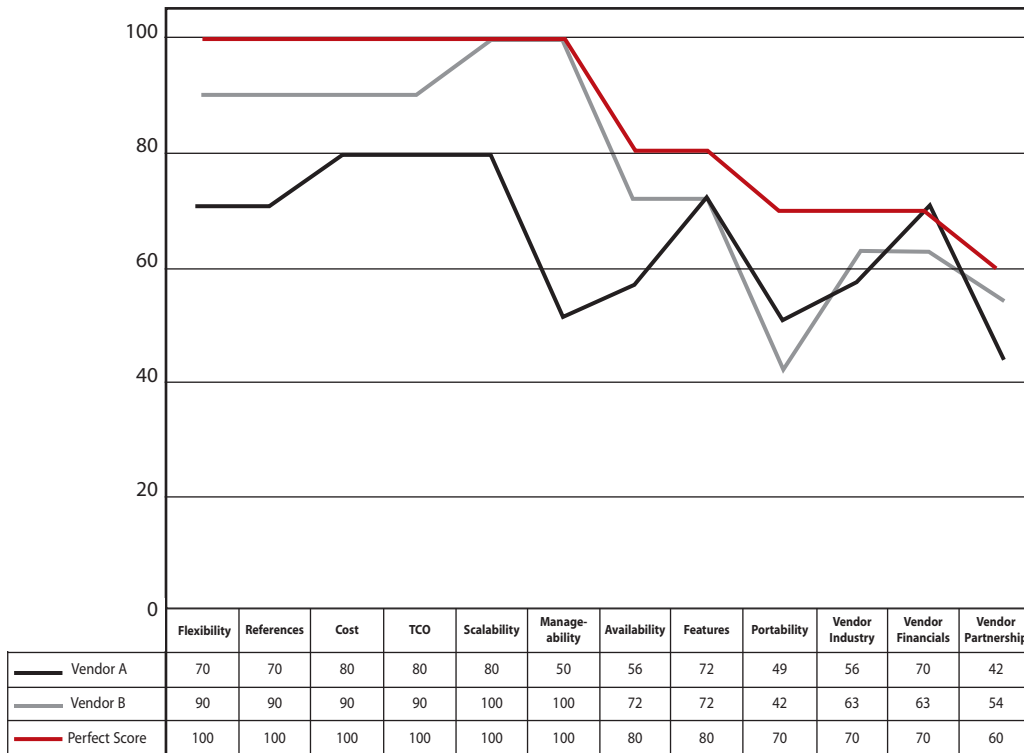


Figure 1. Cardinal's Relevant Comparisons.

---

## Case Study – Cardinal Health

Cardinal Health needed to be sure that the data warehouse solution they selected would be able to scale and to handle an expected growth rate of 20 percent per year for the next five years, expecting multiple terabytes of raw data and more than 37 queries per minute with a large number of concurrent users. They established eleven criteria for evaluating the benchmark results in terms of how effectively and easily they'd be able to grow:

- Flexibility to run cross-subject queries and handle unknown requirements
- References
- Cost including total cost of ownership
- Scalability in data volume and volume of queries (or concurrent users)
- Manageability – ease of use including tuning effort and supporting the data warehouse
- Availability including downtime to manage and upgrade
- Features of the database necessary to support Cardinal requirements
- Portability – migration effort including data replication capability
- Ease of use to change the data model
- Flexibility of the data model
- Tuning required

Cardinal established the ground rules for the vendors including:

- Designating the hardware configuration to be what the vendors had proposed and that which Cardinal was intending to buy.
- No summary or aggregate tables.
- That Cardinal people would be present when the benchmarks were run.

- The log tapes from the benchmark would be made available to Cardinal.
- That the benchmark would last only five days.
- That testing the ability to add new hardware would be a major part of the benchmark.

The team conducting the benchmark was drawn from the business and technical sides of the organization, and included senior management, as well as very knowledgeable hands-on business and technical personnel. The benchmark began with single user queries run serially with no tuning allowed. This was followed by concurrent user testing using a mixed workload environment. This was followed by adding new columns and new tables and running the queries against the expanded database. To demonstrate how the vendor would be able to add new hardware, additional nodes were added in two steps, each time running the complex query workload. Everything was measured and recorded, and these measurements were presented to management in textual and graphical format.

The benchmark gave a fair comparison between the vendors, as each vendor showed strength with different criteria. Cardinal felt their time was well spent on the benchmark since it gave them a comfort level with the vendor selected that allowed them to scale well beyond their projections for five-year growth. It provided assurance to technical people who had previously been unfamiliar with the chosen technology. This is important because a decision of this magnitude requires full acceptance from management and from the technical community.

Between the time of the benchmark and the initial deployment, several factors, such as the number of queries, had changed, but benchmark performance has been a good indicator of the results that have been realized. The benchmark has also been used as input to capacity planning.

## Comparison of the Benchmark to Actual Results

Cardinal's actual results mapped reasonably closely to the benchmark. They were able to achieve their performance SLAs, and their staffing requirements were in line with the expectations coming from the benchmark. Cardinal reports that the benchmark effort was "...well worth it. It's also a small cost relative to the entire project expenditure. We were dealing with a whole new technology stack (introduced Informatica and MicroStrategy also), so it [the benchmark] is justifiable."

### 'The Cardinal Rules'

1. Establish clear objectives as to why you're running the benchmark. These objectives must be supported and approved by management.
2. Establish evaluation criteria along with the weightings before the benchmark begins so there will be no bias (actual or perceived) toward either vendor in evaluating the benchmark results.
3. Allow enough time for the preparation of the benchmark and expect to spend many sessions with the vendors setting your own guidelines or rules for them to follow.
4. If you're running the benchmark with more than one vendor, don't run them back-to-back. Allow at least two weeks between the sessions.
5. Have the same team involved in all vendor benchmarks. This should be a well-respected, cross-organizational team.
6. Results should be evaluated by those who did not have a predisposition toward either of the vendors.
7. Run real business queries.

## Conclusion

Benchmarks can serve to minimize data warehouse project failures as well as budget and schedule overruns. They allow you to test drive systems from more than one vendor before you buy. You should think of a benchmark as an insurance policy. You're insuring that the system you're buying will perform as expected, that you will be able to handle the volumes of data for your target environment, and that you will be able to support your specific workload and concurrent-use requirements with acceptable response times. You're insuring that you have minimized the risk in estimating the project delivery time. The time and effort expended on a benchmark should be well spent – be sure to do it right and to make sure the vendors play fair. The right way to execute a benchmark is with your data, your applications, and your SLAs. Even with these various benefits of a well-designed, well-executed and well-measured benchmark, this should only be one part of a technology evaluation process. It's impossible to accurately or adequately represent the multitude of challenges, query and workload scenarios, and data issues that occur in real-world EDW production systems. Therefore you should also assess benchmark results in the context of in-depth reference investigations, and other indicators of a vendor's capabilities and track record. Remember, you're about to make a set of very important decisions that will have a significant impact on your success and the success of your organization. The best practices represented in this paper – identified through customer experiences, such as Cardinal Health's – should streamline your own benchmark plans and activities.

---

## Appendices

### Appendix 1 – Best Practices

These best practices and lessons learned were gleaned from Cardinal Health and from other organizations with successful data warehouse implementations.

1. Don't let the vendor dictate how to run the benchmark or what to include or exclude.
2. Make sure all the vendors play by the same rules.
3. Make sure you have the right team mix in terms of expertise and objectivity.
4. Don't use a subset of your data or projected volumes and then try to extrapolate. You cannot assume that performance will be linear.
5. If your goal is an EDW, include multiple subject areas and have the benchmark demonstrate integration of these subject areas.
6. Use real and representative data.
7. Send sample data to the vendor with a small subset of data including all the layouts for each table. This should be exactly the way the vendor will be getting the data including the formats. Send this one month before the benchmark begins.
8. Send two copies of the data, as there often are data problems.
9. Know why you're performing the benchmark and what you hope to accomplish.
10. Establish success criteria including evaluation weighting before running the benchmark.
11. Involve stakeholders in preparing, setting goals, and presenting the results. Solicit input and concerns from stakeholders early in the process. Let them know your plans and methodology. Provide bi-weekly status reports to stakeholders.
12. Invite key stakeholders to visit the benchmark site.
13. Don't limit your communication solely to good news. Tell your management and stakeholders about the problems you encountered.
14. Don't misrepresent results to support your preconceived ideas of the benchmark expectations.
15. Create some business questions that will help you compete better or will allow you to make major changes in the way you do business.



---

## Appendix 2 – Project Plan

1. Determine the need, cost and benefits of a benchmark.
2. Establish the benchmark goals and objectives, the critical success factors, and measures of success.
3. Determine what combination of tools will be included in the benchmark and how important the success of those tools is to your implementation. Note: You should already have decided which ETL and BI tools you're planning to use in the benchmark. The benchmark isn't the time or place to evaluate which BI tool to use.
4. Determine what it is you're trying to validate, for example, the maximum number of concurrent users, the impact of complex queries, the time to load.
5. Identify a short list of no more than three vendors.
6. Identify how the results are to be weighted.
7. Sell management on the need and benefit of a benchmark.
8. Secure a budget for the benchmark.
9. Assemble the right team that includes a business person, a DBA, an access and analysis person and a system person.
10. Establish the architecture you're targeting, possibly including an enterprise data warehouse, data marts, an ODS, and a staging area.
11. Gather requirements from multiple sources including your stakeholders as well as potential assassins.
12. Determine the workload that will be used in the benchmark. Information on the workload should come from the business (ask them to look ahead three years) and from IT – especially the DBAs.
13. Determine how to incorporate these requirements in your benchmark. For example, "Department A needs the system available by 6:00 a.m. every day, Department B needs to run 40 complex ad-hoc queries expecting an SLA of an average response time of one minute, and Department C has an availability SLA of 99 percent."
14. Identify your data sources.
15. Define what needs to be tested.
16. Establish multiple scenarios that need to be run.
17. Meet with the vendor on whose premises you will be conducting the benchmark and discuss how best to proceed. Tell the vendor what tools you would like to incorporate in the benchmark including ETL and BI tools.
18. Prepare a schedule and plan.
19. Compile data, test scripts, queries and reports.
20. Book the benchmark with the vendor.
21. Prepare any non-disclosure and security agreements.
22. Run the scenarios, and measure and document everything.
23. Evaluate the results and make any appropriate modifications in the runs. This could include a larger hardware configuration.
24. Rerun with the modified scenarios.
25. Prepare a report and presentation about the results of the benchmark.
26. Celebrate completion of the benchmark.
27. Write a thank-you note to the vendor benchmark team.
28. Present the results of the benchmark to your benchmark committee, and ask for their feedback.
29. If the benchmark did not meet your expectations and requirements for performance, cost, and availability, consider a benchmark with another vendor, reconsider the architecture, go back to the drawing board, or drop the project entirely.

Preparation for the benchmark should take approximately three to four weeks. This would include developing your benchmark plan, communicating with the vendor, data modeling, writing the DDL and the SQL, writing the queries, creating test data, and creating the real volume data extracts.

---

### **Appendix 3 – Sample Report ‘Table of Contents’**

1. Goals of the benchmark
2. Benchmark team – names and titles
3. Proposed hardware configuration(s). (If more than one configuration was tested, all should be included in the report.)
4. System architecture – ODS, staging areas, enterprise data warehouse, data marts, etc.
5. High-level database design – such as fact and dimension tables
6. Workload – such as database size, number of concurrent users, queries, and reports
7. Other tools (ETL and BI) included in the benchmark
8. System constraints – such as availability requirements, scheduling (for example, 24 X 7), and window constraints
9. Problems encountered in the benchmark and how the vendor responded to the problems
10. Benchmark results
  - a. Times to perform the ETL processes
  - b. Response times by type of query and report
  - c. Time to recover
  - d. Administrative effort and activities
  - e. Function and performance of other tools included in the benchmark
11. Issues and opportunities uncovered in the benchmark
12. Observations about the vendor
13. Responses from the vendors’ references
14. Recommendations

---

## Appendix 4 – Checklist for Success

How will you know the benchmark is successful and that you can sign off on the results to your management? The following is a sample set of questions that you will want to tailor for your current and future environments.

### Current (Looking out 12 months)

1. The benchmark must successfully support \_\_\_ terabytes of data.
2. The benchmark must support an availability SLA of \_\_\_.
3. The benchmark must be able to support \_\_\_ concurrent queries.
4. The nightly ETL must run within \_\_\_ hours.
5. The workload must be able to successfully run \_\_\_ complex queries, simultaneously with \_\_\_ queries of moderate complexity and \_\_\_ simple queries (a detailed description of what complex and moderate means must be well-defined).
6. If you have an active requirement, the data must be current as of \_\_\_ minutes/seconds.
7. The data warehouse tools we intend to use all function properly. (List them.)
8. The skills needed to support our anticipated workload and SLAs are \_\_\_? (For example, DBA, data administrator. List them.)
9. The price/performance and TCO we need to achieve is \_\_\_.
10. The vendor is able to handle a change in the physical model, add and delete columns, change partitions, add and change indexes, add summary tables and create materialized views.
11. The vendor is competent and willing to help us when we run into trouble.
12. The vendor is able to handle a “surprise” set of tasks, about which the vendor was given no advance notice, including (for example) long-running “monster” queries that may never have returned, all end-of-month jobs running along with the normal days’ activities, a very heavy ETL load, and even a vastly increased volume of data and queries, perhaps resulting from a major acquisition.

### Future (Looking out 12 – 36 months)

1. The benchmark has to be able to successfully support \_\_\_ terabytes of data.
2. The benchmark must support an availability SLA of \_\_\_.
3. The benchmark must be able to support \_\_\_ concurrent queries.
4. The nightly ETL must run within \_\_\_ hours.
5. The workload successfully runs \_\_\_ complex queries, simultaneously with \_\_\_ queries of moderate complexity and \_\_\_ simple queries (a detailed description of what complex and moderate means must be well-defined).
6. If you have an active requirement, the data must be current as of \_\_\_ minutes/seconds.
7. All of the data warehouse tools we intend to use function properly. (List them.)
8. The skills needed to support our anticipated workload and SLAs are \_\_\_? (For example, DBA, data administrator. List them.)
9. The price/performance and TCO we need to achieve is \_\_\_.
10. The vendor is able to handle a change in the physical model, add and delete columns, change partitions, add and change indexes, add summary tables and create materialized views.
11. The vendor is competent and willing to help us when we run into trouble.
12. The vendor is able to handle a “surprise” set of tasks, about which the vendor was given no advance notice, including (for example) long-running “monster” queries that may never have returned, all end-of-month jobs running along with the normal days’ activities, a very heavy ETL load, and even a vastly increased volume of data and queries, perhaps resulting from a major acquisition.
13. The benchmark is able to support \_\_\_ times the number of concurrent queries.
14. The benchmark is able to support \_\_\_ times the volume of our existing data.