DOWNLOAD PDF DATA WAREHOUSE PERFORMANCE

Chapter 1 : SQL Data Warehouse | Microsoft Azure

DW Sentry accelerates Azure SQL Data Warehouse performance. DW Sentry gives you detailed visibility into the queries, loads, backups, and restores of all your data. With the event calendar and intelligent movement dashboard, you always know what factors are impacting workload.

Technology No Comments One could almost call it tunnel vision. The way that we consistently hold on to one way in which to fulfil each and every need for information: We consider it almost a given that we will have to make concessions in terms of flexibility and speed at which data is delivered. After all, it is the only way in which decent query performance can be obtained with the enormous volumes of data that we work with. Unfortunately, I must disappoint you, because nowadays it is truly possible to do things differently. I would love to show you how in this blog post! Everything with Just One Approach? In fact, the applications and source systems from which we retrieve data were not created for data analysis. The only downside is that we continue to hit the boundaries of this approach in terms of supply speed and data volumes. This is why I have advocated, several times, that a hybrid approach â€" using data virtualization technology â€" works much better. Where this approach is truly not feasible, because of the need to build history, for example, we then rely on the traditional approach. In all other cases we use data virtualization to virtually unlock and integrate data sources. This results in the logical data warehouse, which is the best of both worlds. An explanation and arguments in favour of this approach can be found in my previous blog post. Data Virtualization In this post I would like to elaborate on the question regarding speed performance , which is so often directed to me: After all, there is a reason why we replicate so many times? Data virtualization platforms have developed tremendously and include extensive functionality for performance optimization. At Kadenza we use the Denodo Platform for data virtualization. The examples in this post were derived from Denodo practices, but naturally other tools will offer similar functionality. Smart Retrieval Firstly, these types of platforms ensure that only the data that is needed is retrieved from the source systems, and records are therefore a thing of the past. After all, the less data needs to be processed, the faster the process will be. Each of these sub-selections is sent through to the data source responsible for providing the data for that sub-selection. In the Denodo Platform this is called full aggregation pushdown. The emphasis here is placed on limiting and filtering when selecting data from the source. The data is only retrieved if such retrieval is strictly necessary for the query. Since large volumes of data are not often required for most user queries, it does not usually take long for these sub-selections to be retrieved from the data sources and to be combined to form the required data sets. Sometimes one must combine large data sets or the joins are complex. Even in such more complex situations, the data virtualization platform offers various execution plans so that the answer can be derived in a way that is smart and efficient. For example, the Denodo Platform has the following techniques available: The platform automatically activates these techniques when necessary. Moreover, you can actually see how the platform deals with this which technique is used when and can adjust where needed. In actual practice we see that query performance in this case is the same as with a physical data warehouse. Practical Example I will demonstrate this by way of a practical example. We will compare the performance of a logical data warehouse and a physical data warehouse when the same queries are run. Both data warehouses contain the same volume of data, only the architecture differs. The physical data warehouse consists of a single database in which all data â€" that is retrieved from the source systems and transformed â€" is stored. The logical data warehouse, on the other hand, supplies data via a data virtualization layer. This is just as much as in the three separate databases of the logical data warehouse. The same queries are then run on both data warehouses physical and logical for performance measurement purposes. When we compare the results, we notice that the performance between the two solutions does not differ to a significant extent: Depending on the relevant query, the data virtualization platform determines the most efficient method for rapid retrieval and joining of the required data. Although it is possible to influence the method that is used, it seems superfluous in actual practice. Data Caching Apart from the methods for performance optimization, already mentioned, the data virtualization platform also offers the data caching option. Since this option allows for data to be stored once again, it is

actually contradictory to the basic principles of data virtualization. For example, if an application cannot be consulted by BI applications during office hours, or if the query does not process the data source fast enough. In such cases, caching can be activated instead of the long-standing method of replicating data to a physical data warehouse. Caching can be set for a specific period of time or for a specific sub-selection of data. Without having to make concessions in terms of performance, the utilization of data virtualization affords you flexibility, real-time availability of data, a much simpler architecture, management simplicity and a substantially higher supply speed. In other words, stop the endless ETL process and determine the best approach per data source: The current generation of data virtualization platforms, such as Denodo, provides all the required functionality in one integrated platform. The data warehouse then becomes a multi-platform environment. The traditional, relational data warehouse will continue to exist, but only for the data that absolutely requires this approach. Other data can be linked much faster and easier by way of data virtualization. Without any loss of performance! Here is the substantiation for the figures in this post.

Chapter 2 : Management Data Warehouse | Microsoft Docs

Data Warehousing > Data Warehouse Design > Performance Tuning. Task Description. There are three major areas where a data warehousing system can use a little performance tuning.

This platform-as-a service PaaS offering provides independent compute and storage scaling on demand. Several common loading options are briefly described, but the main focus is the PolyBase technology, the preferred and fastest loading method for ingesting data into SQL Data Warehouse. Introduction Whether you are building a data mart or a data warehouse, the three fundamentals you must implement are an extraction process, a transformation process, and a loading processâ€”also known as extract, transform, and load ETL. When working with smaller workloads, the general rule from the perspective of performance and scalability is to perform transformations before loading the data. In the era of big data, however, as data sizes and volumes continue to increase, processes may encounter bottlenecks from difficult-to-scale integration and transformation layers. As workloads grow, the design paradigm is shifting. Transformations are moving to the compute resource, and workloads are distributed across multiple compute resources. In the distributed world, we call this massively parallel processing MPP , and the order of these processes differs. With SQL Data Warehouse, you can scale out your compute resources as you need them on demand to maximize power and performance of your heavier workload processes. However, we still need to load the data before we can transform. After the DSQL plan has been generated, for each subsequent step, the Control node sends the command to run in each of the compute resources. The Compute nodes are the worker nodes. They run the commands given to them from the Control node. As you scale out your compute resources by adding DWUs , you increase the number of Compute nodes. Within the Control node and in each of the Compute resources, the Data Movement Service DMS component handles the movement of data between nodesâ€”whether between the Compute nodes themselves or from Compute nodes to the Control node. DMS also includes the PolyBase technology. Network and data locality The first considerations for loading data are source-data locality and network bandwidth, utilization, and predictability of the path to the SQL Data Warehouse destination. Depending on where the data originates, network bandwidth will play a major part in your loading performance. For source data residing on your premises, network throughput performance and predictability can be enhanced with a service such as Azure Express Route. Otherwise, you must consider the current average bandwidth, utilization, predictability, and maximum capabilities of your current public Internet-facing, source-to-destination route. Note Express Route routes your data through a dedicated connection to Azure without passing through the public Internet. ExpressRoute connections offer more reliability, faster speeds, lower latencies, and higher security than typical Internet connections. For more information, see Express Route. PolyBase is by far the fastest and most scalable SQL Data Warehouse loading method to date, so we recommend it as your default loading mechanism. PolyBase can load data from gzip, zlib and Snappy compressed files. Data transfers between SQL Data Warehouse and an external resource PolyBase data loading is not limited by the Control node, and so as you scale out your DWU, your data transfer throughput also increases. By mapping the external files as external tables in SQL Data Warehouse, the data files can be accessed using standard Transact-SQL commandsâ€”that is, the external tables can be referenced as standard tables in your Transact-SQL queries. Copying data into storage The general load process begins with migrating your data into Azure Blob Storage.

Chapter 3 : Performance Tuning in Data Warehousing

The management data warehouse is a relational database that contains the data that is collected from a server that is a data collection target. This data is used to generate the reports for the System Data collection sets, and can also be used to create custom reports.

When and why to put what data where and how. Data must be available 24x7, and many business users demand that data supporting decision making be accessible within hoursâ€"in some cases, minutes or even secondsâ€"of when an event occurs. Organizations also realize that the same data needs to be utilized by many different processes and thus many different workload profiles. Ideally, organizations should keep detail data at the lowest form possible in a functionally neutral data model. This enables the business community to ask any number of questions from a wide range of perspectives and processes. The basic premise is that you can always aggregate detail data, but you can never decompose summary data. This does not imply that you should never use summary tables; rather, it means that you should not replace detail data with only summary data. If the desired analysis requires detail data, using only summary tables will fail no matter the technical justification. Implementers and users must work together to understand the business requirements and what drives them; then they need to use the least intrusive process possible to meet those requirements. The primary reason for building summary tables, adding indexes, enforcing priority and denormalizing data is to increase performance. If you could use any amount of data to ask any question and get an instant response, you would never need to build summaries or indexes. These structures function as a workaround to other limitations. Their very existence requires more space, data management and time between the occurrence of an event and the ability to take action. The question is how to balance providing higher performance with minimizing data replication and management. Query frequency and performance consistency are also important considerations. With a summary table, queries executed in approximately six seconds instead of four minutes, a difference of 2, minutes of processing time. Even factoring in the minutes required each week to maintain the summary table, the resulting savings of 2, minutes per week clearly justified making a summary table. Over time, this company found that usage shifted from resolving many queries with summary data to matching a majority of queries against detail data. With fewer queries benefiting from a summary table, it was simply dropped without affecting other processes. Optimizing performance The process of optimizing the data management environment should not be undertaken without understanding the consequences. When optimizing performance, the starting point is having a clear grasp of two aspects of the system. First, what is the unaided performance? The database may be powerful enough that executing the query against the tables without additional indexes or denormalization gives the required response time. This raises the second aspect. An organization may indicate a major problem with the data warehouse is that the response time on reports is not fast enough. However, when asked what the user would do if the reports were faster, the answer is: What business outcome is changed because the response time goes from one minute to 10 seconds? If that question cannot be answered, then stop all your efforts until you can justify the expense of added optimizations. If it can be answered, the IT group should follow a step-by-step approach to balance cost of performance with benefit of analytic output. However, it has a relatively higher cost. So before proceeding, make sure the anticipated business value will outweigh the cost of the additional data movement and management required to keep the extracted data in agreement with detail data. Explore, expand and export Explore the business usage of the answer sets and validate that the change in business outcome will drive the expected revenue increase or cost decrease, depending on the application. Are the business users willing to stand behind the expectations, and does the enhanced performance justify the cost? Expand the current platform. Is the performance requirement so critical that new capacity is warranted? Export the data from the main data warehouse to an application-specific platform. In this situation, a dedicated environment tuned specifically for its application will provide much more control over the individual application. Keep in mind all of the cost of the duplicated data, added time lag to action, and the cost of a new platform and software environment that will need to be managed and supported. Justifying the steps Taking these seven steps requires understanding

the cost involved with each step and the benefits derived from doing so. It also requires making decisions that support long- and short-term needs. In some cases you may create summary tables or add denormalized data models that you will drop eventually as the functions evolve over time. This is acceptable as long as eliminating the tables does not cause interruptions or massive application changes. One way to ensure this is to refrain from using the summary or denormalized tables as the input to more downstream applications whenever possible. When applying the seven steps, perform cost-benefit analysis for each proposed step, and include physical aspects such as disk space, resources to manage the structure, and lost opportunities due to time delays to maintain the process. Improvements may be seen in: Query performance and opportunity gained from faster response User concurrency rates.

Chapter 4 : Measuring the Data Warehouse â€" Enterprise Information Management Institute

If it is, please let us know via a Review Reviews August 21, - Noob Sorry for incorrect question. In this case may I change my question - Could creating and deleting tables many times per day during years damage data dictionary? If yes how we can restore it? Followup August 23, - 2: From an Oracle perspective, DDL is something you do rarely, with perhaps "truncate" being the exception. Recreate Tables August 22, - This is highly uncommon or at least IMO this is not what you want to do if you design an application which works well in Oracle. My other bet on recreating the tables is on temporary tables. In Oracle this is fundamentally different: Temporary Tables are created once like any other heap table. The structure of them is permanent and visible to all sessions, and you reference them in your code like you would reference any other heap table. The data in them however is visible only to your session, it is temporary and cleaned up for you on commit, or at the end of your session. So if you write a process which loads some data into a temporary table for whatever reason, in many other RDBMS you would 1. However; step 1 in Oracle is not usually done during a process which fills the temp table but initially when you install the application. Step 3 is omitted completely of course. You simply use the temporary tables like you would use a heap table. This most certainly is compeletely unnecessary. Of course this could only be the start - as already said: You could gather dictionary stats and see if that helps. Maybe you have the possibility to trace a session you know will recreate some tables and see where the time is spent. But you have to pinpoint exactly the reason maybe your DBA already has and just tells you his findings. Noob Thank you for answer. I absolutely understand that recreating thousands table per day is bad approach for Oracle. But I want to find out could it irreplaceably damage data dictionary. For example we stopped recreating tables at AM and collect statistic on data dictionary. If you say yes - recreating thousands tables can irreplaceably damage data dictionary it means every simple DB user with rights to create and delete table can destroy Oracle DBâ€¦ If you say no â€" we will try to find another reason of our problem. Followup August 24, - 1: But its a large load on the system, and depending on what options you have installed, lots of triggers and hence extra work may be firing whenever you issue DDL. To Noob August 23, - 2: So my answer would be: But to be clear: This will be your next move

DOWNLOAD PDF DATA WAREHOUSE PERFORMANCE

Chapter 5 : The Seven Steps of Data Warehouse Performance Optimization - Teradata Magazine

Data Warehouse performance and maintenance - Download Document Indexes If you have millions of rows of records in one fact table, it might take a long time to generate a report.

How to Improve Data Warehouse Performance and Maintenance Posted on by Hide Suzuki Data warehousing has become an important technology even for small to mid-size companies for data analysis. One of the issues we encounter with data warehouses is performance, since we combine a large amount of data from multiple data sources. There are several steps we can take to maximize data warehouse performance: Separate server for BI data warehouse: Your ERP server has a lot of activities going on constantly, so it is recommended that you have a separate server for BI, this way you have a dedicated server which can produce a better performance. Limit the amount of data in the data warehouse for the last say years. You rarely need to do any reporting using such old data. See Below Database Statistics Very often we hear from our customers that reports that used to run very quickly now take a lot longer. And they claim that there has been no change in their system. How can this happen? One of the reasons is that SQL server is not processing all the commands in the most efficient way. When you run a report, your reporting tool sends a set of SQL commands to SQL server and process them in the order they are received. Just like there are many different solutions to the same math problem, there are different ways for SQL Server to process the same commands, and some are more efficient than others. How does SQL server determine how to process them? They make decisions based on database statistics, which is not updated automatically by default. Attached document shows step by step process of how to create and schedule a database maintenance plan. This should be a standard practice for all the databases, not just data warehouses. Data Warehouse performance and maintenance â€" Download Document Indexes If you have millions of rows of records in one fact table, it might take a long time to generate a report. One of the things you can do is to create indexes. Below are two reasons why indexes help speed up report generation time. One reason is that generally fact tables hold many columns that are not useful in reporting. But they still take up some space in the hard disk. When you create an index, basically you are creating a copy of the table, only with the data fields you need for your reports. This is a time consuming process for a large table. When you create an index, you are sorting data based on columns you choose. If you create an index sorted by say timeperiod and Account, then SQL does not have to go through the entire table. This is just like when you look for specific data in an Excel file. You can add or delete dimensions and also add comments or UDF fields if necessary.

Chapter 6 : Data Warehousing Tuning

Today we are excited to announce that Azure SQL Data Warehouse has set new performance benchmarks for cloud data warehousing by delivering at least 2x faster query performance compared to before.

Follow the Getting Started Guide Amazon Redshift is a fast, scalable data warehouse that makes it simple and cost-effective to analyze all your data across your data warehouse and data lake. Redshift delivers ten times faster performance than other data warehouses by using machine learning, massively parallel query execution, and columnar storage on high-performance disk. You can setup and deploy a new data warehouse in minutes, and run queries across petabytes of data in your Redshift data warehouse, and exabytes of data in your data lake built on Amazon S3. To create your first Amazon Redshift data warehouse, follow our Getting Started Guide and get the most out of your experience. Contact us to request support for your proof-of-concept or evaluation. Benefits Faster Performance Amazon Redshift delivers 10x better performance than other data warehouses. It uses machine learning, a massively parallel architecture, compute-optimized hardware, and result set caching to deliver high throughput and sub-second response times. With Redshift, you spend less time waiting, and more time gaining insights from your data. Easy to set up, deploy, and manage Amazon Redshift is simple to use, enabling you to deploy a new data warehouse in minutes. Redshift automates most of the common administrative tasks to manage, monitor, and scale your data warehouse. This helps you break free from the complexities of managing on-premises data warehouses. There are no upfront costs with Redshift, and you only pay for what you use. Scale quickly to meet your needs Amazon Redshift enables you to scale from querying gigabytes to exabytes of data across your Redshift data warehouse and Amazon S3 data lake. Quickly analyze any size of data in S3 with no loading or ETL required, and easily resize your Redshift cluster with just a few clicks on the console or a simple API call. With Redshift, you can scale up or down as your needs change. Query your data lake Amazon Redshift extends your data warehouse to your data lake to help you gain unique insights that you could not get by querying independent data silos. You can directly query open data formats stored in Amazon S3 with Redshift Spectrum, a feature of Redshift, without the need for unnecessary data movement. This enables you to analyze data across your data warehouse and data lake, together, with a single service. Secure Amazon Redshift runs mission critical workloads for large financial services, healthcare, retail, and government organizations. How it works Featured customers Amazon Redshift powers the largest number of data warehousing deployments in the cloud for business, real-time, and predictive analyses. Finra uses Amazon Redshift to analyze billions of transactions daily. Yelp uses Amazon Redshift to analyze massive amounts of clickstream and log data. Equinox Fitness Club closes their customer journey loop by analyzing clickstream data with Amazon Redshift. Use cases Accelerate all your analytics workloads You can use Amazon Redshift to get sub-second results for reports, dashboards, and interactive analysis, and to get fast results for complex queries on any scale of data. Redshift uses machine learning, massively parallel execution, columnar storage on high-performance disk, and result caching to give you the performance you need on any mix of workloads. Unified data warehouse and data lake You can use Amazon Redshift to run queries across your data warehouse and data lake to unlock insights that you would not be able to obtain by querying independent data silos. Redshift Spectrum can directly query open file formats in Amazon S3 and data in Redshift in a single query, without the need or delay of loading the S3 data. This gives you the freedom to store data where you want, in the format you want, and get fast query results at any scale. Modernize your on-premises data warehouse You can modernize your on-premises data warehouse to a fast, scalable, easy to manage, and cost-effective cloud data warehouse running on Amazon Redshift. Redshift automates most of the common administrative tasks to setup, scale, manage, and maintain your data warehouse. By handling these time-consuming and labor-intensive tasks, you can focus more on your data and your analytics.

Chapter 7 : What is Azure SQL Data Warehouse? | Microsoft Docs

SQL Data Warehouse stores data into relational tables with columnar storage. This format significantly reduces the data storage costs, and improves query performance. Once data is stored in SQL Data Warehouse, you can run analytics at massive scale.

Queries are often very complex and involve aggregations. For OLAP systems, response time is an effectiveness measure. OLAP databases store aggregated, historical data in multi-dimensional schemas usually star schemas. OLAP systems typically have data latency of a few hours, as opposed to data marts, where latency is expected to be closer to one day. The OLAP approach is used to analyze multidimensional data from multiple sources and perspectives. The three basic operations in OLAP are: OLTP systems emphasize very fast query processing and maintaining data integrity in multi-access environments. For OLTP systems, effectiveness is measured by the number of transactions per second. OLTP databases contain detailed and current data. The schema used to store transactional databases is the entity model usually 3NF. Predictive analytics is about finding and quantifying hidden patterns in the data using complex mathematical models that can be used to predict future outcomes. Predictive analysis is different from OLAP in that OLAP focuses on historical data analysis and is reactive in nature, while predictive analysis focuses on the future. These systems are also used for customer relationship management CRM. History[edit] The concept of data warehousing dates back to the late s [11] when IBM researchers Barry Devlin and Paul Murphy developed the "business data warehouse". In essence, the data warehousing concept was intended to provide an architectural model for the flow of data from operational systems to decision support environments. The concept attempted to address the various problems associated with this flow, mainly the high costs associated with it. In the absence of a data warehousing architecture, an enormous amount of redundancy was required to support multiple decision support environments. In larger corporations, it was typical for multiple decision support environments to operate independently. Though each environment served different users, they often required much of the same stored data. The process of gathering, cleaning and integrating data from various sources, usually from long-term existing operational systems usually referred to as legacy systems , was typically in part replicated for each environment. Moreover, the operational systems were frequently reexamined as new decision support requirements emerged. Often new requirements necessitated gathering, cleaning and integrating new data from " data marts " that was tailored for ready access by users. Key developments in early years of data warehousing were: Textual disambiguation applies context to raw text and reformats the raw text and context into a standard data base format. Once raw text is passed through textual disambiguation, it can easily and efficiently be accessed and analyzed by standard business intelligence technology. Textual disambiguation is accomplished through the execution of textual ETL. Textual disambiguation is useful wherever raw text is found, such as in documents, Hadoop, email, and so forth. Facts[edit] A fact is a value or measurement, which represents a fact about the managed entity or system. Facts, as reported by the reporting entity, are said to be at raw level. These are called aggregates or summaries or aggregated facts. For instance, if there are three BTS in a city, then the facts above can be aggregated from the BTS to the city level in the network dimension. In a dimensional approach , transaction data are partitioned into "facts", which are generally numeric transaction data, and " dimensions ", which are the reference information that gives context to the facts. For example, a sales transaction can be broken up into facts such as the number of products ordered and the total price paid for the products, and into dimensions such as order date, customer name, product number, order ship-to and bill-to locations, and salesperson responsible for receiving the order. A key advantage of a dimensional approach is that the data warehouse is easier for the user to understand and to use. Also, the retrieval of data from the data warehouse tends to operate very quickly. Another advantage offered by dimensional model is that it does not involve a relational database every time. Thus, this type of modeling technique is very useful for end-user queries in data warehouse. The model of facts and dimensions can also be understood as data cube. Where the dimensions are the categorical coordinates in a multi-dimensional cube, while the fact is a value corresponding to the coordinates. The main disadvantages of the dimensional

approach are the following: To maintain the integrity of facts and dimensions, loading the data warehouse with data from different operational systems is complicated. It is difficult to modify the data warehouse structure if the organization adopting the dimensional approach changes the way in which it does business. Normalized approach[edit] In the normalized approach, the data in the data warehouse are stored following, to a degree, database normalization rules. Tables are grouped together by subject areas that reflect general data categories e. The normalized structure divides data into entities, which creates several tables in a relational database. When applied in large enterprises the result is dozens of tables that are linked together by a web of joins. Furthermore, each of the created entities is converted into separate physical tables when the database is implemented Kimball, Ralph Some disadvantages of this approach are that, because of the number of tables involved, it can be difficult for users to join data from different sources into meaningful information and to access the information without a precise understanding of the sources of data and of the data structure of the data warehouse. Both normalized and dimensional models can be represented in entity-relationship diagrams as both contain joined relational tables. The difference between the two models is the degree of normalization also known as Normal Forms. These approaches are not mutually exclusive, and there are other approaches. Dimensional approaches can involve normalizing data to a degree Kimball, Ralph In Information-Driven Business, [18] Robert Hillard proposes an approach to comparing the two approaches based on the information needs of the business problem. The technique shows that normalized models hold far more information than their dimensional equivalents even when the same fields are used in both models but this extra information comes at the cost of usability. The technique measures information quantity in terms of information entropy and usability in terms of the Small Worlds data transformation measure. Please help improve this article by adding citations to reliable sources. Unsourced material may be challenged and removed. July Bottom-up design[edit] In the bottom-up approach, data marts are first created to provide reporting and analytical capabilities for specific business processes. These data marts can then be integrated to create a comprehensive data warehouse. The data warehouse bus architecture is primarily an implementation of "the bus", a collection of conformed dimensions and conformed facts , which are dimensions that are shared in a specific way between facts in two or more data marts. Dimensional data marts containing data needed for specific business processes or specific departments are created from the data warehouse. Legacy systems feeding the warehouse often include customer relationship management and enterprise resource planning , generating large amounts of data. To consolidate these various data models, and facilitate the extract transform load process, data warehouses often make use of an operational data store , the information from which is parsed into the actual DW. To reduce data redundancy, larger systems often store the data in a normalized way. Data marts for specific reports can then be built on top of the data warehouse. A hybrid DW database is kept on third normal form to eliminate data redundancy. A normal relational database, however, is not efficient for business intelligence reports where dimensional modelling is prevalent. Small data marts can shop for data from the consolidated warehouse and use the filtered, specific data for the fact tables and dimensions required. The DW provides a single source of information from which the data marts can read, providing a wide range of business information. The hybrid architecture allows a DW to be replaced with a master data management repository where operational, not static information could reside. The data vault modeling components follow hub and spokes architecture. This modeling style is a hybrid design, consisting of the best practices from both third normal form and star schema. The data vault model is not a true third normal form, and breaks some of its rules, but it is a top-down architecture with a bottom up design. The data vault model is geared to be strictly a data warehouse. It is not geared to be end-user accessible, which when built, still requires the use of a data mart or star schema based release area for business purposes. Data warehouse characteristics[edit] There are basic features that define the data in the data warehouse that include subject orientation, data integration, time-variant, nonvolatile data, and data granularity. Subject-Oriented[edit] Unlike the operational systems, the data in the data warehouse revolves around subjects of the enterprise database normalization. Subject orientation can be really useful for decision making. Gathering the required objects is called subject oriented. Integrated[edit] The data found within the data warehouse is integrated. Since it comes from several operational systems, all inconsistencies must be removed. Consistencies include

naming conventions, measurement of variables, encoding structures, physical attributes of data, and so forth. Time-variant[edit] While operational systems reflect current values as they support day-to-day operations, data warehouse data represents data over a long time horizon up to 10 years which means it stores historical data. It is mainly meant for data mining and forecasting, If a user is searching for a buying pattern of a specific customer, the user needs to look at data on the current and past purchases. The user may start looking at the total sale units of a product in an entire region. Then the user looks at the states in that region. Finally, they may examine the individual stores in a certain state. Therefore, typically, the analysis starts at a higher level and moves down to lower levels of details. The hardware utilized, software created and data resources specifically required for the correct functionality of a data warehouse are the main components of the data warehouse architecture. All data warehouses have multiple phases in which the requirements of the organization are modified and fine tuned. Fully normalized database designs that is, those satisfying all Codd rules often result in information from a business transaction being stored in dozens to hundreds of tables. Relational databases are efficient at managing the relationships between these tables. To improve performance, older data are usually periodically purged from operational systems. Data warehouses are optimized for analytic access patterns. Unlike operational systems which maintain a snapshot of the business, data warehouses generally maintain an infinite history which is implemented through ETL processes that periodically migrate data from the operational systems over to the data warehouse. Evolution in organization use[edit] These terms refer to the level of sophistication of a data warehouse: Offline operational data warehouse Data warehouses in this stage of evolution are updated on a regular time cycle usually daily, weekly or monthly from the operational systems and the data is stored in an integrated reporting-oriented data Offline data warehouse Data warehouses at this stage are updated from data in the operational systems on a regular basis and the data warehouse data are stored in a data structure designed to facilitate reporting. On time data warehouse Online Integrated Data Warehousing represent the real time Data warehouses stage data in the warehouse is updated for every transaction performed on the source data Integrated data warehouse These data warehouses assemble data from different areas of business, so users can look up the information they need across other systems.

Chapter 8 : Azure SQL Data Warehouse loading patterns and strategies | SQL Server Customer Advisory T

SQL Data Warehouse customers can now expect significant query performance improvements, including the ability to run concurrent queries, and to store an unlimited amount of columnar data, empowering you to run your largest and most complex analytics workloads.

By Sid Adelman Running a data warehouse without the advantage of metrics is like trying to navigate a ship without a chart, compass, or sextant. Without metrics, we have no way of knowing if we have delivered a data warehouse that anyone could consider to be successful. We would have no idea about response time, machine utilization, availability, user satisfaction or the quality of the data in the warehouse. This column suggests metrics that are appropriate to the data warehouse, recommends standards, sometimes represented as service level agreements, suggests who should be responsible for measuring, who should be responsible for taking action to correct situations that are out of compliance with the standards, and recommends how to represent the results of the measurements to management. Conformance to Measures of Success Most projects have explicit or implicit measures of success and most of these can be measured. The measurements of a data warehouse will determine if it was or was not a success. Types of Metrics There are a number of types of metrics that are relevant for understanding how well we are doing. Usageâ€" Usage tells us if the data warehouse is being used, to what extent and by whom. Our metrics may show that our goal was met. Metrics on usage are often an eye opener as you discover large numbers of identified users do not use the system or use it only sporadically. Performanceâ€" Performance is usually reflected in response time. You will want to know what percentage of the queries ran longer than, for example, 10 minutes, how long they actually ran, and you will want to know which departments experienced the long run times. Measurements sometimes uncover poor performing queries that are the result of users not understanding the ramifications of some of their actions. The realization that a field is very frequently being summarized could lead to the creation of a summary table. As it becomes apparent that another field is frequently being accessed, or that tables are being joined on a specific key, the DBA would build an index for those fields. ETL performance is particularly important for large and very large databases. The time to perform the processes of transformation, cleansing, aggregations, and loads can sometimes exceed the available window for the ETL process. The metrics on each of these processes can help determine the source of the problem and help direct where to focus the resources. Performance metrics can anticipate problems and provide the diagnostic capability that suggests remediation before a performance problem rears its ugly head. Availabilityâ€" Availability is the percentage of time the system can be accessed by the users during scheduled hours. Most organizations do not have the same availability requirements for their data warehouse as they do for their operational systems. Operations will sometimes not give the data warehouse environment the right level of attention and this results in poor availability. In addition, the ETL process aborting or not completing on time can also impact availability. Resource utilizationâ€" This would include the number of machine cycles, memory usage and the accesses to the disk. The information about the percentage of disk utilization should result in a better distribution and partitioning of the data on the array of disks or should point out the necessity of purchasing additional disk. Sometimes these surveys uncover misunderstandings in the way the system was intended to be used. User satisfaction surveys typically include questions on the use of the access and analysis tool, data quality, availability, response time, and support. Data qualityâ€" Some aspects of the quality of the data can be automatically quantified and should include the percentage of values that are outside of the valid values, the percentage of fields that are missing, non-unique data, data that is the wrong data type, data that is outside of the acceptable ranges, and data that violates business rules. Metrics that have a bearing on the quality of data would include record counts, the number of distinct values, the number of records with null values, the number of records within a certain range, the number of records with a certain distinct value, and the rate of change of the data. Management is always surprised and dismayed when they discover just how dirty their data is. The data quality metric will serve as a barometer of the constant data quality improvement that should be a part of every data warehouse initiative. Dormant dataâ€" Dormant data is data that is never, ever

accessed. Loading this unused data night after night is expensive, consumes disk space, wastes energy, and may reduce the likelihood that the system will be available to the users by 8: Dormant data is a total waste to the organization and is the albatross that will weigh down the budget, extend ETL time, and tax the skills of the DBA staff. Dormant data exists either because the requirements gathering process was lax or because the users were unable to sufficiently articulate their requirements resulting in loading useless data for fear that the users may possibly need it in the future. Dormant data remains because there may be no tools in place to even let the DBAs know that the data is not being accessed. Use of tools which tools are being used and at what level â€" Since many data warehouse installations have multiple business intelligence tools, this metric should tell us which tools are used and to what extent. Most organizations have anticipated which tools will be used by which departments and by which category of users, e. There may be a misunderstanding of the tools and their purpose or the profiling may have been incorrect. An organization may be purchasing software when it already has seats available and does not need to purchase any additional software. The organization may be paying maintenance for software sitting on the shelf. In both cases, metrics could identify opportunities to save money. Costsâ€" This is what the data warehouse costs, both on initial installation and as an ongoing expense. The value of measuring the costs, and comparing those costs to the anticipated costs budget will give the organization the information it needs to better anticipate costs and then to help determine if a project is cost justified before it actually gets implemented. Ongoing costs are often grossly underestimated. When anticipating costs, consider the total cost of ownership that would include items such as servers, software licenses, support staff, and so on. Some of the intangibles do not lend themselves to quantification but it is important to identify them whenever possible. The organization must measure the benefits to determine if the anticipated benefits were achieved. This knowledge will help, along with the information gleaned from measuring actual costs, in adding to the enterprise knowledge of anticipating the ROI of each project and in setting priorities. Security conformanceâ€" What security violations have been detected, where are the violations coming from and have there been any breaches? This information will help plug security holes and will also provide a level of comfort to upper management as they worry about security exposures. Service Level Agreements SLAs SLAs are written agreements between the business â€" the folks who will be using the data warehouse â€" and IT â€" the people who are responsible for building and providing the data warehouse infrastructure. The SLAs will identify your goals and the metrics will tell you if these goals have been met. Metrics supporting the SLAs may include: Availability Response time Response to problems SLAs let you know which of the user requirements are being met. SLAs also serve to hold user expectations in check. Some organizations have groups dedicated to performance and these are the folks who normally have the primary responsibility. Measurement is usually not a full-time job but it is a job that cannot be forgotten or denied. If there are serious performance problems or availability problems, near real-time awareness is paramount so that the responsible persons can be alerted and effective actions can be taken. Means to Measure A number of the data warehouse products have built-in abilities to capture and to report on the system. These metrics can be accessed and delivered in a form with some effort that is meaningful to the technical people, to the users, and to management. In addition there are add-on products that supplement in various areas such as data quality and performance. Quite often, organizations have these means of measuring but are either unaware of their existence, or no one has been assigned to execute the measurements and to then take action on their results. Use of Measurements We need to measure because the data warehouse should always be considered a work in progress. None have gone in without the need to make changes. There is always opportunity to enhance the data warehouse and, in fact, without enhancement, a data warehouse would rarely meet any specified measure of success. The process should always be to measure, identify problems and opportunities and take appropriate action to solve the problems and exploit the opportunities. Chargebacks are rarely welcomed no one wants to have additional costs assigned to his or her department but for those organizations that do charge back the use of their systems to the departments that employ them, metrics are critical to an equitable distribution of costs. This becomes even more important when money is transferred from one organization to another. Reporting results to management Management wants to know how things are going. Are the users using the system, are they happy, and are they achieving the benefits they were

expecting? Management is usually content with monthly metric reports unless there are serious problems. In which case, management will want to be briefed more frequently on the problems, the steps that are being taken to resolve those problems, and results of the resolutions. Metrics should be reported with just the information that is of interest to each manager. A good approach is to use conventional data warehouse tools accessing a small metrics data mart. Any metrics that represent problems should be highlighted or shown in red. A dashboard is appropriate for metrics of performance and availability. Summary Each organization should identify the metrics they will need and use as they continually work to improve their own data warehouse. An understanding of the appropriate metrics, the responsibility for gathering the metrics, and the use of those metrics can make the difference between success and failure of the data warehouse project. He is a frequent contributor to journals that focus on data warehousing. He can be reached at and sidadelman aol. His web site is www.

Chapter 9 : Azure sets new performance benchmarks with SQL Data Warehouse | Blog | Microsoft Azure

The pressure is on business intelligence and data warehousing professionals to handle ever-higher data volumes and ever-more-complex queries while reducing decision latency. Follow this five-step approach to identify key business drivers, optimize system performance, guide new technology deployments.

Next Page A data warehouse keeps evolving and it is unpredictable what query the user is going to post in the future. Therefore it becomes more difficult to tune a data warehouse system. In this chapter, we will discuss how to tune the different aspects of a data warehouse such as performance, data load, queries, etc. It is very difficult to predict what query the user is going to post in the future. Business requirements change with time. Users and their profiles keep changing. The user can switch from one group to another. The data load on the warehouse also changes with time. It is necessary to specify the measures in service level agreement SLA. It is of no use trying to tune response time, if they are already better than those required. It is essential to have realistic expectations while making performance assessment. It is also essential that the users have feasible expectations. To hide the complexity of the system from the user, aggregations and views should be used. It is also possible that the user can write a query you had not tuned for. Data Load Tuning Data load is a critical part of overnight processing. Nothing else can run until data load is complete. This is the entry point into the system. Therefore it is very important to tune the data load first. In this approach, normal checks and constraints need to be performed. When the data is inserted into the table, the code will run to check for enough space to insert the data. If sufficient space is not available, then more space may have to be allocated to these tables. These checks take time to perform and are costly to CPU. The second approach is to bypass all these checks and constraints and place the data directly into the preformatted blocks. These blocks are later written to the database. It is faster than the first approach, but it can work only with whole blocks of data. This can lead to some space wastage. The third approach is that while loading the data into the table that already contains the table, we can maintain indexes. The choice between the third and the fourth approach depends on how much data is already loaded and how many indexes need to be rebuilt. Integrity Checks Integrity checking highly affects the performance of the load. Integrity checks should be applied on the source system to avoid performance degrade of data load.