

SQL Server 2000 Data Warehousing

I have been fascinated by data warehousing for many years, ever since Microsoft introduced CUBE and ROLLUP functions in SQL Server 6.5, enabling a T-SQL programmer to aggregate values into very simple result sets that could be used to analyse data trends. I also enjoyed telling great stories – or more likely urban myths – about the sales of beer and nappies being intrinsically linked on a Friday night as men were ordered to do the shopping for the weekend and took the opportunity to stock up on alcohol. This trend was, of course, discovered by analysing or mining the store sales data in a data warehouse and serves to prove the commercial benefit of just such a system, or so the story goes...

A data warehouse is a read only repository of subject-oriented data structured to enable efficient querying and analysis. Data is taken from operational systems and fed into the data warehouse via a data cleansing or scrubbing process. Once inside the data warehouse the data is placed into tables linked in such a way so that pre-defined queries execute quickly and ad hoc queries run as speedily as possible.

Performance is critical to a data warehouse, as is the accuracy of the returned data, as a failure in either aspect will lead to immediate disillusionment within the users.

Data Warehouse Data Structures

Relational databases typically use a cost based optimiser. If a query contains multiple table joins or links the optimiser will examine each combination of table joins in turn to determine the best execution plan.

For example, a query such as, “Show me all of the sales of beer brand X in all Kwik-E-Mart stores for the past 2 weeks excluding London” would probably touch a products table, a stores table, a sales table and a regions table in a conventional database schema.

This can lead to many combinations being examined - for example a 4-table join has a total of 4 factorial (24) possible join combinations, a 10 table join an amazing 3,628,800 possible join combinations!! The query optimiser in SQL Server 2000 has been tuned to try and overcome this multi join problem but the issue still exists and is subject to the limitations of the data structure.

Within a data warehouse environment when a wide variety of queries can be executed the designer needs to try and pre-empt the user by reducing these table joins. This is achieved by using a star or snowflake schema, amalgamating commonly used data into a single table.

Data merging to produce a star schema is a useful design tool when the following conditions are met:

- Tables share a common key
- Data from the tables is used together on a frequent basis
- Data insertion, if appropriate, is the same across the tables

In fact there is a slight difference between the star and snowflake structures as the snowflake structure has the surrounding dimension tables in a more normalised form.

OLAP, MOLAP, ROLAP and HOLAP

OLAP (On-line Analytical Processing) is technology used to build and maintain data in a multidimensional format such as a cube. The source data is often stored in an underlying relational database in traditional rows and columns, and the cube built on top of that.

MOLAP is the name given to conventional cube or multidimensional OLAP structures. This offers fast query performance, as the data is pre-built in the cube format but often requires mass storage to store the aggregations – data explosion as it is called.

ROLAP is seen to be a more scalable solution, and enables the data to remain in the SQL Server tables but a skeletal cube structure to be built to house the aggregations.

HOLAP is the use of a hybrid storage structure, that is data that combines both ROLAP and MOLAP data. A good example of a hybrid solution is the use of regular reports that analyse geographical data but occasionally need to drill down into product detail data. The geographical data is placed into a MOLAP cube and the product data placed into a ROLAP cube.

SLOWLAP is what you get when you do it wrong. OK, sorry about that but I always have to get that pun in when talking about OLAP.

MOLAP? HOLAP? ROLAP? Which one do I use?

| | MOLAP | ROLAP | HOLAP |
|------------|---|---|---|
| Advantages | <ul style="list-style-type: none"> • Data navigation, slicing and analysis is quicker as the dimensions have all been pre-calculated | <ul style="list-style-type: none"> • Data resides in the original data source, allowing changes to be reflected quicker • Cubes are | <ul style="list-style-type: none"> • Best of both worlds, as the structure can be tuned to the business requirements for optimised performance |

| | and stored | quicker to process | |
|---------------|--|---|--|
| Disadvantages | <ul style="list-style-type: none"> • Cube may take a while to process initially • Cube needs to be managed on a regular basis as new data comes in to the Data Warehouse | <ul style="list-style-type: none"> • Data navigation may be slower | <ul style="list-style-type: none"> • Administration and management may be cumbersome. |

The cube is the central object in a multidimensional database containing dimensions and measures – dimensions being derived from underlying tables and columns and measures being the quantitative data derived from the columns. Dimensions should in any case be distinct categories added to the cube. Measures are more normally time periods or geographical based metrics often contained in a hierarchical structure, for example hours roll into days, which roll into weeks ...

The cube can hold a number of aggregations that can dramatically improve the efficiency and response time of a query. The scope of the aggregations can be massive, and the designer of a data warehouse needs to offset performance against storage space.

Data Mining and SQL Server Analysis Services

New in SQL Server 2000 are a whole bunch of data mining tools. Remember the nappy and beer story? Well that nugget of information was mined from a cube using tools similar to Analysis Services, in other words tools that can detect trends and points of data that are of interest.

When building a data mining model a set of training data needs to be collected that is based on accurate data from previous activities. For example, if you were selling SQL Server training courses you collect the historical demographic data of course attendees and run that through your model to ensure that the results were as expected.

Once the training data has been assembled the appropriate algorithm needs to be chosen. There are two data mining algorithms used in Analysis Services, both of which are based on statistical theories that have been around for a number of years;

- Decision trees represent the data classification questions as nodes on a tree or branch like structure. The predictive nature of the algorithm is

based on the training data set influencing where the node is located and the depth of the node in the structure.

- Clustering or the expectation method, in that it groups data into clusters or neighbourhoods of similar predictable characteristics. In many instances the clusters are counter intuitive and obscure, but that is the whole point of data mining!

The stored data mining model is known as a model node which contains detailed information such as probabilities, attributes and data descriptions. Within Analysis Services is a data model browsing tool that visualises the model content into something most people would understand.

Some of the real power of data warehousing is being uncovered as more and more organisations build web sites that track a user's site journey and purchasing habits. Previous articles have covered Commerce Serve 2000, the Microsoft .NET server product designed to run commercial web sites. When installed Commerce Server builds a comprehensive data warehouse on the underlying SQL Server, which in turn can be used to analyse a range of user activities. If you haven't been asked to build a data warehouse yet, the chances are you will as the demand for business intelligence increases. Hopefully this article has wetted your appetite to explore data warehousing further.