**When to use aggregate tables**

MicroStrategy uses optimized SQL to query the relational database directly to answer users’ questions. Users can ask any question that is supported by the data in their warehouse and then analyze the results until they find a precise answer.

The disadvantage to this relational OLAP (ROLAP) methodology is that accessing huge fact tables can be potentially time-consuming. Multidimensional OLAP (MOLAP) is sometimes considered by some to be the answer to this problem. However, MOLAP is not scalable for large projects because of the difficulty of maintaining every possible combination of aggregates as the number of attributes and the amount of data increases. MicroStrategy’s solution is the use of aggregate tables to provide quicker access to frequently-accessed data while still retaining the power to answer any user query.

Aggregate tables are advantageous because they:

|  |  |  |
| --- | --- | --- |
|  | • | Reduce input/output, CPU, RAM, and swapping requirements |

|  |  |  |
| --- | --- | --- |
|  | • | Eliminate the need to perform dynamic calculations |

|  |  |  |
| --- | --- | --- |
|  | • | Decrease the number of physical disk reads and the number of records that must be read to satisfy a query |

|  |  |  |
| --- | --- | --- |
|  | • | Minimize the amount of data that must be aggregated and sorted at run time |

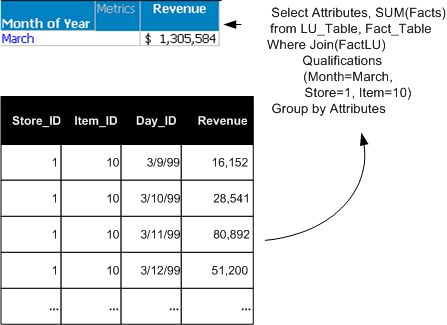
|  |  |  |
| --- | --- | --- |
|  | • | Move time-intensive calculations with complicated logic or significant computations into a batch routine from dynamic SQL executed at report run time |

In summary, the MicroStrategy SQL Engine, in combination with aggregate tables and caching, can produce results at about the same speed as MOLAP. This combined solution allows questions to be answered on the fly and is also scalable for large databases.

**Aggregation versus pre-aggregation**

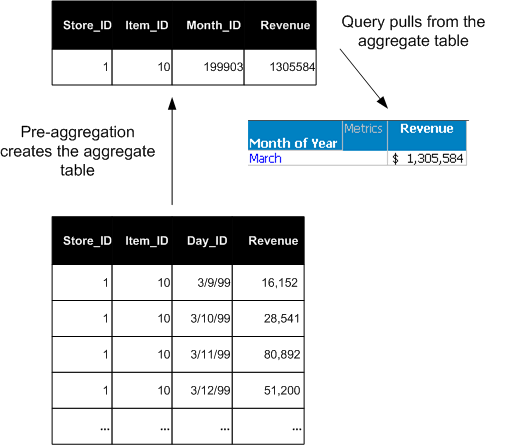
Whenever the display level of data on a report must differ from the level at which the data is initially captured, aggregation, that is, the rolling up of data, must occur. By default, aggregation occurs dynamically with a SQL statement at report run-time.

For example, sales data is stored by day in a fact table. A report requesting month-level data is executed. The daily values from the fact table are selected, sorted, and added to produce the monthly totals, as shown below.



Aggregation can also be completed before reports are executed; the results of the aggregation are stored in an aggregate table. This process is called pre-aggregation. You can build these pre-aggregated—or aggregate—tables as part of the ETL process. If sales data is frequently requested at the month level, as in the previous example, an aggregate table with the sales data rolled up to the month level is useful.

Pre-aggregation eliminates the reading, sorting, and calculation of data from many database rows in a large, lower-level fact table at run time, as shown in the following example.



If the daily sales fact table is the lowest-level fact table and contains atomic-level data, it is referred to as a base table. In these terms, an aggregate table is any fact table whose data is derived by aggregating data from an existing base table.

**Degree of aggregation**

While MOLAP can provide fast performance when it answers a question, it requires a completely aggregated schema to answer most questions. That is, every possible combination of aggregate associations must be generated when the multidimensional cube is built. This ensures that all possible questions can be answered. This scenario becomes very difficult to maintain as the number of attributes and the amount of data increase, and therefore is not very scalable.

In a ROLAP environment, the degree of aggregation can be as dense or as sparse as is appropriate for your users. A densely aggregated warehouse has a large number of aggregate tables while a sparsely aggregated warehouse has fewer. Sparse aggregation refers to the fact that a given project only requires as many aggregate fact tables as is useful to its users.

ROLAP, therefore, provides much greater flexibility than MOLAP. Only the aggregate combinations that you determine are beneficial must be created. That is, if the aggregate table is useful in answering frequently-asked queries, its presence provides a response as fast as a MOLAP system can provide. However, if a certain aggregate combination is rarely or never used, the space in the RDBMS does not need to be consumed and the resources to build that table during the batch process do not need to be used.

Not every attribute level or hierarchy intersection is suitable for pre-aggregation. Build aggregate tables only if they can benefit users, since the creation and maintenance of aggregate tables requires additional work by the database administrator. Also, do not waste database space for tables that will not be used.

Consider the following factors when deciding whether to create aggregate tables: