

PowerOLAP™
for the
Introductory User

White Paper



Copyright © 2007 All Rights Reserved



PowerOLAP™ for the Introductory User

PowerOLAP™ represents a milestone in the evolution of Business Intelligence technology. Like any significant evolutionary event, PowerOLAP combines the best features of what came before it with new capabilities. Most important, PowerOLAP enables end-users to reach through seamlessly to pre-existing databases to perform sophisticated, real-time planning, analysis and reporting tasks. In addition, PowerOLAP employs Excel and the Web as a front end, connecting users throughout an organization with underlying data sources via the tools they know best—the familiar spreadsheet and Web browser—direct to their desktops.

In order to appreciate PowerOLAP's unique functionality, we need to take an overview of how data typically moves through an organization, from the manner in which it is recorded, through its storage in and retrieval from a database, to the moment when it is accessed by an end-user. These are important considerations because they impact upon what knowledge workers ultimately can or cannot do at their desks.

It's All Relational: Going Back to the Database

Every business has a system, if not several systems, for collecting and storing its core operational data. The entry point for data may be at someone's desk (keyed in at a terminal), or through an automated collection and transmission system that will record each *transaction*. Examples of transactions are entries in an accounting system, reservations for an airline seat or a hotel room, and credit card purchases of clothes in a department store. Once a transaction is completed, the necessary data is sent back for storage to a database. For example, the data sent to a database from a sales register might include date, store location, item identification number, credit card number, purchase price, among other particulars.

The overall system for the collection and storage of data is known as *the database management system*. In the past, companies often developed their own individualized systems, but eventually the *relational model* was acknowledged as the best database management system for capturing, validating, and storing large volumes of individual transactions, the core operational data of the business. Since the 1970s, companies have made huge investments in relational database management systems (RDBMS).

Database management systems also allow for the retrieval and processing of stored data. It should be noted, though, that the internal design of the database affects how quickly and flexibly data can be retrieved and processed. Imagine the case of a company needing to verify a single transaction, or wanting simply to total the purchases of an item at one location during a fixed period of time. RDBMSs are particularly amenable to this kind of *on-line transaction processing (OLTP)*. However, the static physical design of relational databases—data is stored in two-dimensional tables, with each entry being a row in the table—impedes more complicated data analysis. The sheer volume of data in the database also makes analytical activities very time-consuming, if not unrealistically complex.



There do exist classes of “front end” tools that work well for data retrieval and provide formatting and report-writing options, and even some analytical functionality, to technically savvy end-users. The more advanced of these query/report writer tools, as they are called, employ graphical user interfaces (GUIs: point-and click boxes, for example) to automate the task of writing the Standard Query Language (SQL) used to access relational databases. The significance of this direct access to core operational data—a key feature which, as we shall see, PowerOLAP incorporates—should not be overlooked: it empowers end-users to individualize (within the parameters of what is functionally possible) and to schedule their own querying and reporting, without the need to re-stage data, and without anyone else’s help.

The weakness of query/report writer tools is that their capabilities are bound by the underlying structure of the database. Although the “translation” of a request may be simplified through the use of GUIs, the subsequent querying/reporting will still be executed according to the database’s “grammar” (SQL statements). Again, the relational structure of the database stymies sophisticated analysis.

[It is worth noting here, too, that we have only addressed the capability for reporting: *transaction-processing systems have even less functional flexibility in accommodating “write-back” systems* (e.g., for budgeting/forecasting), which is why organizations typically default to creating overtaxed, multi-spreadsheet applications.]

The Power of PowerOLAP (1): Modeling From a Multidimensional Database

What is lacking in both in-built RDBMS tools and front-end tools is the ability to create data models that reflect the actual business dimensions of an organization. In order to work up data in a meaningful way, analysts need to prepare models that can quickly and accurately compare and contrast all manner of “multidimensional” data: for example, actual sales of a particular product, over time, within particular regions as compared with the budget plan. Analysts must be able to create “what if” scenarios within their models: by how much will the total cost of a product change if a less costly production material is substituted in the manufacturing process? And if the price of the product is lowered, and sales rise by ten percent as a result, by how much will revenue and profit totals be affected? These types of analyses, which today’s competitive conditions require be done, and then shared among many users—in real time, if at all possible—cannot be performed on data as it exists in relational database systems.

A new database model has evolved out of the necessity to perform these kinds of sophisticated analyses: the *multidimensional model*. Multidimensional databases can accommodate *on-line analytical processing (OLAP)*—the ability to organize data hierarchically in multiple dimensions for end-users to quickly create, analyze, and calculate complex data relationships.

PowerOLAP employs a multidimensional database that enables end-users to create their own models of data relationships. The structure of a model is not a list of records (as exists in a relational database) but a “cube”, the sides of which are chosen dimensions (for example, *Products, Months, Regions, Total Sales, Budget vs. Actual*). Dimensions are composed of individual members (for example, the members of the Months dimension could be *Jan, Feb, March*, and their sum, *1st Qtr*). At the intersection of dimension members is a single cell within the cube.



In one respect, cubes modeled within a multidimensional database are similar to spreadsheets: both are arrays of cells. As in a spreadsheet, when the value of a cell in a cube is changed, values throughout the cube—and in other cubes, if relationships have been established—will also change. Whereas spreadsheets accommodate only two dimensions, cubes can contain as many dimensions as an end-user wants to include in order to create an effective business model.

A view of a multidimensional cube, referred to as a “slice view,” will actually resemble a spreadsheet grid—we usually see comparison data in such a row-and-column format, on paper or on screen. The most advanced multidimensional databases, PowerOLAP included, allow users to use spreadsheets as a front-end to further manipulate the data within a cube’s slice view—for example, to utilize the spreadsheet’s graphical and report-writing features.

The Power of PowerOLAP (2): From Relational Storage to Multidimensional Modeling via OLAP Exchange™

We have detailed how core operational data is collected and stored: most often in a relational database. We have considered the weaknesses of the in-built and front-end tools of relational databases: because of the structure of the database, they are unable to accommodate sophisticated queries, much less create the multidimensional models that analysts need to pattern complex real-world data relationships. We have examined how OLAP technology accomplishes sophisticated data analysis: by enabling end-users to model cubes within a multidimensional database. There remains a missing link in the movement of data from “there” to “here”: how does data get from its likely source, a relational database, to the OLAP stage, namely, within a multidimensional structure?

Until the advent of PowerOLAP, the most advanced OLAP products required great effort to transfer transactional data into multidimensional cubes. First it was necessary to prepare the transactional data into specialized form, aggregating and adjusting it for input; then there was the process of physically copying all this “analysis data” into the multi-user multidimensional database. All this could be a costly, time-consuming, resource-intensive procedure, requiring special programs and even a great deal of rudimentary re-keying. The end result was an independent “proprietary” multidimensional database entirely separated from a company’s transactional system, which continued to receive and store key operational data.

PowerOLAP overcomes these serious shortcomings of previous classes of OLAP products. By utilizing a highly evolved “reach-through” functionality, known as OLAP Exchange, PowerOLAP enables end-users to perform dynamic analyses on data that has been *seamlessly accessed from a relational source*. PowerOLAP, therefore, incorporates the functionality and performance of the most advanced OLAP tools into the relational database model.

Customized Integrations to the End User’s Database Systems

The value of PowerOLAP begins during installation. PowerOLAP integrations have been performed with many major applications, including the accounting systems used by most mid-sized and larger businesses. By providing this kind of integrated solution capability, the product may be installed and begin delivering dynamically connected multidimensional cubes very rapidly. This saves valuable time and consulting expenses.



The Empowerment of End-Users—and the Organization

Consider the powerful impact on end-users:

- They have a dynamic connection to the data they need for sophisticated planning, analysis and reporting —the data is entirely “within reach,” up-to-date, available for their own multidimensional cube-modeling.
- End-users can then work within a spreadsheet or Web browser to perform further data manipulations in a familiar environment. This includes “write back” capabilities to budgeting/planning models.

PowerOLAP also profoundly benefits an organization by allowing it to leverage its existing investment in a relational database system:

- There is no need to create an un-updatable independent repository for “analysis data”—instead, data enters at the transaction level, as it always has, and is available as it exists via OLAP Exchange for immediate multidimensional analysis.
- Calculations performed within PowerOLAP’s multidimensional database can be sent via OLAP Exchange to the underlying relational database, guaranteeing consistent data integrity between the two systems.

Even for those who are offline, data changes/updates are available upon reconnection to anyone through PowerOLAP’s exclusive Synchronization Server™ capabilities. PowerOLAP is an enterprise-wide tool, allowing sophisticated analyses, as well as planning data, to be shared throughout an organization nearly instantaneously.

This last point deserves emphasis: PowerOLAP not only establishes a direct connection between a transactional system and an end-user, but also connects all end-users (at least all who are qualified) throughout a global enterprise via a PowerOLAP-dedicated server.

Summary: PowerOLAP: Evolving to a “First of its Kind” Solution

PowerOLAP provides key components that make up for an all-too-often missing link in decision support technology. With PowerOLAP, end-users have the thorough functionality they need to perform sophisticated reports, analyses and plan requirements on data in *real-time* within a multidimensional environment.