

Seeding For Data Growth

Here are five cost-conscious ways to ensure that your new data warehouse can scale up to meet business needs.

By Mark Hall

April 15, 2002

Computerworld - Even 7-foot-1-inch basketball player Shaquille O'Neal started small. But with an ideal genetic design and proper ongoing nourishment, he grew to become the powerful central force of a dominant organization.

Just like your data warehouse.

Corporate databases usually start small, too, but as more vital business data is poured into them - everything from customer transactions to pricing structures - they can get big quickly. And as more users query that data, the potential for a system slowdown grows. If databases aren't tuned for scalability at the design stage, relevant data could be excluded, forcing users to draw conclusions from incomplete data.

Making the right choices upfront can give a data warehouse the roots to handle dramatic growth. Here are five cost-conscious strategies to achieve scalability in a new database design.

1. Know your business needs The first step is to figure out what you're dealing with. "Two factors stand out: the size of the database you're starting with, and the number of users accessing it," says [Phil Isensee](#), director of central computing at Oregon State University in Corvallis.

Isensee says it's important to know the kinds of queries users are more likely to make, because that helps you construct indexes in the database. You should also survey your application base, because some applications generate far more data than others, he says.

2. Streamline your data After evaluating your business demands, "normalize all your data," says Mike Schmitz, a consultant at [High Performance Data Warehousing](#) in Bend, Ore.

That means that if you have 20 data warehouse sources of customer information, you should set up pointers inside those sources to replace all redundant data, such as addresses and billing codes. "Normalization buys you less redundant data, less index space, and is a key factor for growth," Schmitz says.

3. Set up data partitions Database partitioning is the next key design feature of a scalable data warehouse, says Schmitz. For example, partitions can be used to manage data in units of time. One partition can be based on one day, another on seven days and still another on 30 days, matching partitions to business operations. This approach is particularly useful for updating an existing warehouse. "When you load data into a day's partition

and index it, you can update the entire warehouse in microseconds," Schmitz says.

4. Choose systems with power The underlying technology is also critical when scalability is crucial to a data warehouse, says Dan Vesset, an analyst at [IDC](#) in Framingham, Mass. "Not every software package scales equally well," he says.

For large-scale data warehouses of up to hundreds of terabytes, Oracle Corp.'s 8i and 9i and IBM's DB2 databases are the best choices and should be run on mainframes or massive Unix servers such as Sun Microsystems Inc.'s Enterprise 10000, Schmitz says.

For maximum scalability, you'll probably adopt symmetrical multiprocessing (SMP) systems. SMP systems eclipsed massively parallel processing computers as the preferred architecture in 2000, according to Waltham, Mass.-based research firm [Winter Corp.](#), which reports that 55% of the world's largest multiterabyte databases run on SMP systems.

"SMP is the way to go," says Isensee. "You don't want one user's query tying up all the resources."

5. Consider outsourcing Elizabeth Koehler, manager of financial planning and analysis at [CBS MarketWatch.com Inc.](#) in San Francisco, says that several years ago, her company was faced with a software upgrade to a more powerful version of its data analytics package and the increased hardware and support costs that went with it. To help control costs and keep pace with its ever-expanding data, the firm outsourced its data warehouse operations.

MarketWatch switched from [Accrue Software Inc.](#) in Fremont, Calif., to outsourcer [digiMine Inc.](#) in Bellevue, Wash., in September 2000. If it hadn't, says Koehler, she's not sure her analysts would have received their daily reports when, one year later, the financial Web site experienced all-time high traffic loads after the Sept. 11 attacks sent stock markets reeling.