

# Database Design ROI

Paul Nielsen, SQL Server MVP

On the surface, IT costs are easily calculated as software costs (licenses and development costs) and operations costs (equipment, contracts, and staff). Buried several layers down in an application's project plan is the database design line item. A cost in terms of project time has been assigned to the database design and when the task is complete, the project will move on. However, the quality of the database design will continue to affect several other IT costs for the duration of the database's life cycle.

Specifically, the quality of the database design initially impacts development costs and in the long term will impact software maintenance and development costs, as well as application performance and scalability which effects hardware costs, IT operational costs and application availability, and eventually drives extensibility and flexibility. Because the database design significantly affects so many other costs, excellence in database design is more than merely an academic exercise, in fact, the return on investment (ROI) is very positive.

While this white paper can't place specific numbers to a particular project, it does highlight areas where the database design significantly impacts IT costs so you can estimate the ROI or risk associated with the database design of your project.

The resulting excellence in database design can be briefly described as a design that meets the requirements as simply as possible with a physical database schema that maximizes the database platform.

## **Database Design Impacts Initial Development Costs**

The primary software features of a data-centric application are determined by the database design.

- If the database design fails to support the requirements the project can fail. Before the database is determined to be the cause of the difficulty, project teams can easily extend the project costs to several times a reasonable cost. When the poor database design is recognized, several layers of the project must be completely overhauled to remedy the problem further compounding the development costs.

*I was asked to optimize a database for a material specification project that was about 2 years and 1 million dollars over budget and having trouble completing the last 10% of the requirements. In analyzing the database I determined the flaw was that the schema could not possibly support the all the requirements. With a new database schema the project was completed in 6 months.*

- A more common problem is a database design that is overly complex or difficult. This can extend the development costs as developers create unnecessary code to workaround the difficulties of the database.

*I have personally witnessed a project that should have been 3-5 man months grow into a team of 18 developers and 1.5 years of development that can be directly traced to an overly complex database scheme.*

- Overly complex databases also tend to have an excessive number of tables which generate additional code for the data access layer rather than the right number of tables to best model the business problem. This extends the development costs with every code iteration.
- Poorly designed systems are more difficult to document which results in increased documentation time and cost and often incomplete or incorrect documentation.

### **Database Design Impacts the Organization's Effectiveness**

The database design significantly impacts the quality and usability of the data.

- A database design that is not properly normalized will introduce data update anomalies and data errors. This type of error can show up as inconveniences such as incorrect or multiple conflicting addresses for a client, or the inability to delete duplicate data. More serious errors can cause data to incorrectly link to other data, such as medical records linked to patient data. The most likely result of this failure is regular data errors and increased manual data correction and monitoring costs by staff.
- A poorly designed database may place the entire organization at risk due to the incomplete or incorrect information. An overly complex or poorly designed database is not only more difficult to fully understand but requires more complicated SQL code to extract the correct result from a SQL query.

In contrast, a well designed database is more obvious to the developer and helps avoid coding errors.

The actual cost of incorrect data depends on the application and may range from incorrect customer information, or a misguided product R&D project, to loss of life or limb.

*For example, I've personally seen a database design that required extensive left outer joins to properly extract data combined with code written by application developers who did not fully understand the various types of SQL joins. The organizational cost was that several reports did not contain every client or every transaction.*

### **Database Design Impacts Maintenance Development Costs**

Several studies have proven that the true cost of software is not the initial development but the ongoing maintenance costs.

- A poor database design causes developers to code with difficult workarounds which are in turn difficult to decipher and maintain. This can easily triple maintenance development costs. It also increases the ramp-up time of new staff members longer because they have to learn the database and the code workarounds.

### **Database Design Impact Performance and Scalability**

The database design is the foundation of application performance, or a root cause of poor performance.

There are several factors that influence performance. Optimization Theory demonstrates there's an inherent dependency between these factors. To summarize Optimization Theory, database design enables set-based code, which enables efficient indexing, which reduces transaction duration and improves concurrency. The root cause of good performance therefore, is an excellent database design.

- A badly designed database will require more complex SQL code which will perform slowly.
- Developers will tend to revert to iterative cursors when faced with a complex problem, therefore a poorly designed database will cause more cursors which perform significantly slower than set-based code.
- It is more difficult to properly analyze the indexing requirements for a poorly designed database. This can result in redundant indexes or missing indexes, both of which can impact performance by several magnitudes.
- The compounded problems of overly complex SQL, cursors, and poor indexing all increase transaction time which in turn further degrades performance.
- Poor performance increases costs by reducing the hardware lifecycle, increasing hardware costs (i.e. throwing more hardware at the problem to get acceptable performance), and reducing the effectiveness of every user.

### **Database Design Impact Long Term Extensibility and Flexibility**

A broader, organizational driven analysis reveals that the true cost of IT is its ability to morph functionality to meet the organization's current needs either enables or hinders the organization's ability to move quickly with new initiatives. Of the sacred "ilities" of IT – e.g. availability, scalability – *extensibility*, or the ability to flexibility meet, or even thrive in the face of rapidly changing requirements is the determining factor in the success of an organization.

- A poorly designed database is more difficult to extend. This lack of agility tends to spawn additional small databases designed to add features to the application or fill departmental needs and may drive the organization to out-source functionality.

- Non-agile databases prevent IT from responding to new business requirements in a timely manor, which costs opportunity for the organization.

*One organization is still using a poorly designed database that was originally developing in the 60's. Without a well-defined data abstraction layer and a numerous work-arounds in code, they feel it's impossible to migrate to a better database. The organizational cost is that several new requirements go unanswered or are handled at the department level using MS Access.*

Based on my career optimizing databases, I estimate that every addition man-day spent on database design saves one man-month of initial development costs, and two man-months of post implementation costs. In addition, every iteration of database design reduces the chance the project will fail by 25%.

As with any construction project the foundation determines the viability of the project and either limits or enables other features. Smart CIOs invest in database design because they know excellence in database design is the foundation of any data-centric application.

---

Paul Nielsen is a Microsoft MVP, PASS Board of Director and CIO, author of the SQL Server Bible series (Wiley) and currently writing Database Design in Action (Manning). He was the subject matter expert (SME) for the Microsoft official Course 2784, Optimizing Queries with SQL Server 2005. He leads his courses on database design and optimization and his website is [www.SQLServerBible.com](http://www.SQLServerBible.com).

© 2006 Paul Nielsen - Permission granted to copy or distribute this white paper in entirety and without alteration.