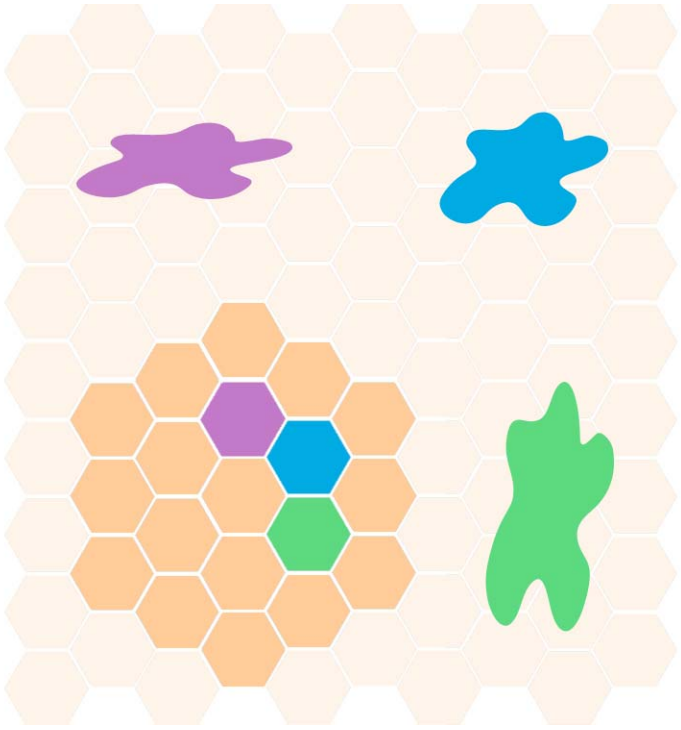
# Equal Format Databases:

## Eliminating Costly Database Changes



A White Paper

August 2008

Making a change to your business can be costly, especially when it includes updating your database and software. The cost of researching your needs, planning and implementing the update, and training employees could negate the benefits you had hoped to realize in the first place.

Now consider the possibility of a database design that doesn't require expensive changes on a continuing basis; a database that operates independently of its content and can handle your changing needs, even as your business evolves and grows. This database design, called an *equal format database*, is already a reality.

In this paper, we will discuss the struggles we currently have using traditional relational database designs, and how our needs can be met with a new type of design that can change how software manipulates data.

## The Cost of Information

Information technology is critical to your business and has a strong impact on your ability to be effective on the job. We all expect to be able to retrieve and analyze data quickly and efficiently, but sometimes that's hard to do when data is buried under layers of old systems and requires the use of time-consuming workarounds to retrieve.

Without a truly flexible system, companies can miss out on opportunities for change and growth. Without the continuity of a flexible design, businesses are not able to upgrade critical systems without risking access to current processes and historical business data. With the arrival of the equal format database, however, it's time to consider how our traditional system designs might be hindering growth, and whether the time is right to move on to the next generation of database methodology.

**Data isolation leads to inefficiencies.** Each software process you add into your company tends to come with its own database. Figure 1 shows a typical predicament. Each system is working with data that is kept separate from the data in the other systems. Traditionally, this disjunction could be solved with custom-built databases, integration systems, or point-to-point translations, but is usually solved by using people to fill the necessary gaps.

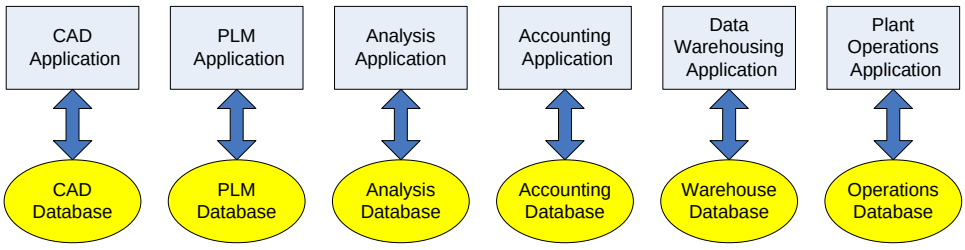



Figure 1

**Paying the price for a custom-built database.** Your database system was designed and built for specific needs. The best products were chosen and integrated into your business. But now, yesterday's solutions are not flexible enough for today's requirements, and now you need changes. Unfortunately, since the system was custom designed, custom work is required to update it.

**Losing time and revenue by sticking with the old and familiar.** You've outgrown your database system. It doesn't meet your needs anymore, and you're spending too much time and money working around its limitations. Every day you become less efficient and move further away from realizing the full use of your stored data. Even if you can afford the work, it can take too long to implement.

For example, perhaps your company has relied on a much-updated legacy system to manage your core business, health insurance. You need to target new market segments, but due to the limits on your database, you cannot extend the definition of customers. Now you are losing revenue as you spend time deciding how to address the issue. Should you continue to patch the system, or start all over with something new? Either way, time and money are at stake.

**Relying on your database for future needs.** Even when current changes are manageable, your relational database system has other issues: Decisions you make today may not fit your needs tomorrow, so unfortunately you may find yourself facing the same issues again soon. Each change has side effects. Ask yourself, "Is your historical data retrievable using your present applications? Can you retrieve your data from even 10 years ago?" Surprisingly, many companies would have to answer, "No." When you store information today, you need a format that will remain relevant and usable forever. Traditional database designs do not naturally lead to active data continuity and seamless archival capabilities.

**Connecting your databases can cause gridlock.** If your database system synchronizes data from multiple software processes, it is intended to enhance your overall productivity. However, the greater the connections between multiple databases and applications, the harder it is to make changes without disturbing all users. Even a minor software upgrade can cause major disruptions to many people.

**Slow system change can lose you customers.** Your customers are frequently affected by your database system, whether it's through your website, while talking with a representative on the telephone, or using other types of communication. When your database isn't keeping up with the speed of changes in your business, everyone suffers, including your customers and your reputation.

## An Equal Format Database is the Answer

An *equal format database* is a database design style that allows you to store and retrieve all data in one table design set, regardless of the content and logical structure. It is a uniform table design that can be built using any database management system, replacing traditional table design that is structurally customized to the data. Let's explore why this would be useful.

Traditional relational database designs claim to be biased for speed in a demanding environment, but these claims are based on uniformity in data and operations. To add more variation, more columns must be added. To track information changes, more columns must be added. It is difficult to predict all of the columns that might be needed. Adding more columns leaves more empty cells, wasting space. To address this, the process of normalization is used to create more tables. Adding more tables creates software complexity. These are design problems inherent in databases with position-based context definition.

All of the issues that we've discussed so far are inherent in traditional database designs. These designs were based on decisions based on criteria which have changed over the decades. Speed and space limitations are always relevant, but the improved equipment, products, and designs of today provide a different basis for decisions that can lead to a better architecture overall. If we re-examine our goals and needs, what kind of design would we do? Wouldn't we improve our situation by removing the tight connection between a database structure and its content?

As previously shown in Figure 1, traditional design methods lead to a different database for each application. To share data, point-to-point translations move data between databases or applications. Figure 2 shows how the number of translations grows with the square of the number of points ($N^2 - N$). Each arrow represents two translations – one each way. If N=6, then the number of translations is (36 – 6) = 30. If N=10, then the number of translations grows to (100-10) = 90.

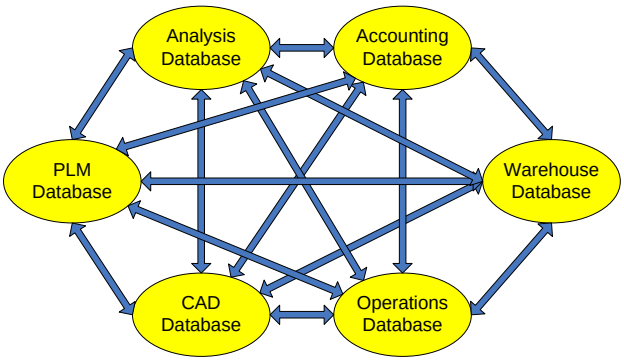



Figure 2

Due to high costs, the number of point-to-point solutions that can be created is limited. While a central database should be a better solution (as shown in Figure 3), the separate business and logical models of each database are difficult and expensive to combine and coordinate. The ability to integrate better makes single-vendor suppliers more attractive, but at the loss of product choice. When custom solutions are used, the difficulty of upgrading becomes a dominant factor.

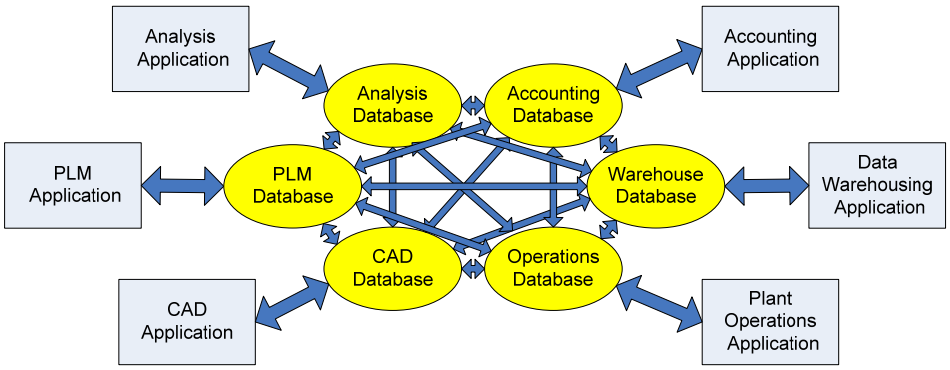



Figure 3

Each of these databases might be composed of hundreds of tables. Instead of thousands of custom table designs, we propose a design that uses several tables to hold data without needing to change for new content or structure. This type of design is inherently flexible because it does not need to change as the data type changes. Figure 4 shows how data can move easily from an application to the database and then to a different application.

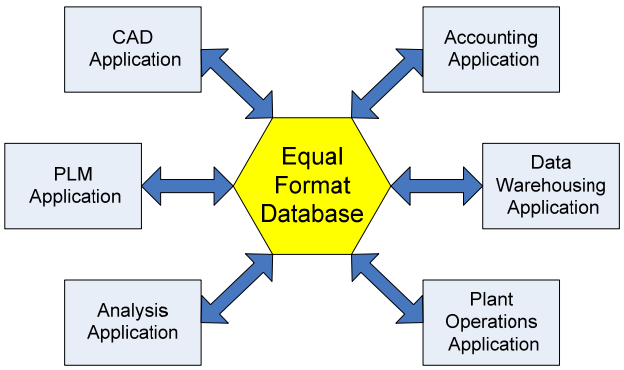



Figure 4

Equal format databases can replace any combination of database systems and subsystems. The design style can be applied to relational databases, but it can also be applied to other structures. Because there is one consistent design for all data, the data can flow across systems, allowing a truly modular approach.

Figure 5 shows how you can move data from one system to another without changing its logical structure or content - easing the distribution of data. Legacy systems can be wrapped in an equal format interface, or can easily be translated thanks to the adaptive capability of the equal format medium.

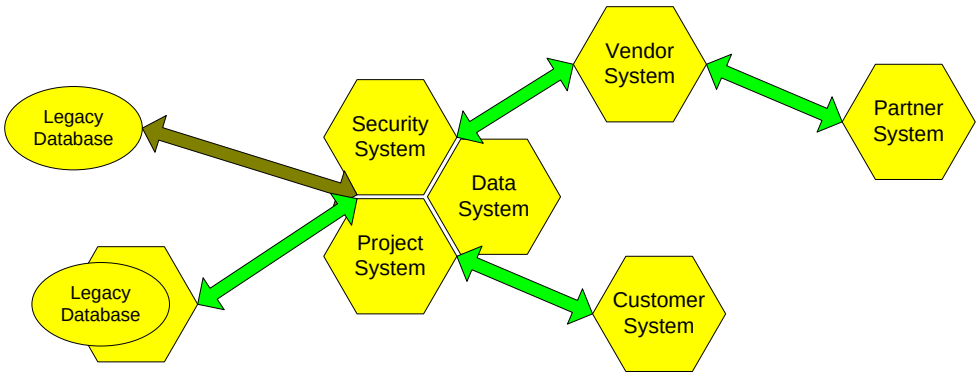



Figure 5

Now that you no longer need to spend time maintaining a variety of databases, your development time can be spent on each application's business logic, greatly reducing project time and cost. Because the methods for working with data become more uniform, a common learning curve will add more benefits.

Consider the merging of two medical databases, each with tens of dozens of databases. Each system carries separate identification data. One system labels a disease "Kidney Stones", while another system labels it "Renal Calculi". Instead, it may simply call it "Stones" with "Gall Stones" in the same grouping. (Differences such as this don't always map on a one-to-one basis.)

A traditional database merger will require substantial redesign, remapping, and recoding, over several years. With equal format databases, both systems can flow together, and logical relationships can now be established between the data. No remodeling is forced on the applications, which are only remapped to the equal format container. This can remove substantial portions of the design, development, and implementation phases. Developers are loosely coupled, allowing efficient and cooperative work.

## The Organization of Equal Format Databases

Consider how data is stored in a computer. Data is stored as a series of numbers, such as {14, 32, 44, 44, … }. But what do these numbers mean? Are they pixel colors, text characters, sound levels, or properties? **Context gives meaning** to a data value.

Figure 6 shows the traditional design of entity tables in a relational database. The value "42" is given context through implicit association by its position in the "Part123" row and in the "Count" column.

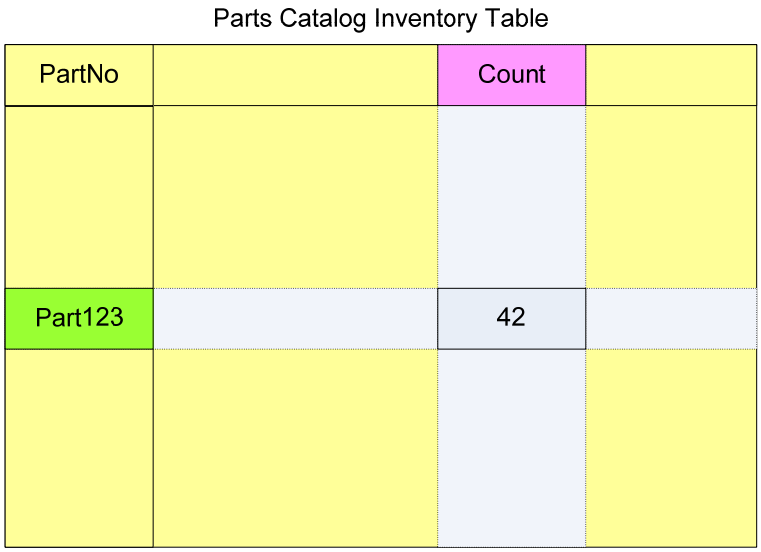



Figure 6

To create equal format databases, we must **stop using table position to indicate context**. Figure 7 shows that by adopting alternative methods of encoding context, such as attaching context tags to the value, we can create an atom of equal format data. Databases designed to hold atoms of equal format data are **defined as equal format databases**.

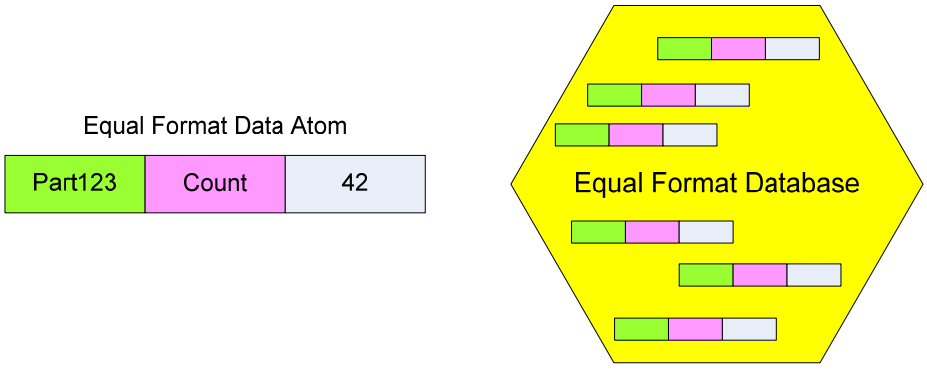



Figure 7

## Types of Equal Format Databases

The lack of flexibility in traditional relational database designs has encouraged the creation of alternative database technologies, each with varying degrees of success. Though they are all flexible, lack of a unifying principle has kept them from integrating with each other and with relational databases. Using equal format design concepts, they can work together and achieve the full measure of their potential. The following is a summary of types, with their strengths and weaknesses:

1) Object Databases. The mapping of data to object models has been most useful with simulation and engineering modeling, application objects, and other object programming technologies. Its inherent hierarchical organization leads to different organizational standards for different industries. After almost two decades of use, it has never achieved the mainstream use anticipated. The newer Entity Models are the latest incarnation of this methodology.
2) Abstract Middleware. There have been a number of products, often organized around object systems, with the purpose of merging (or upgrading) legacy databases, or for serializing object data. These middleware products create a mathematical mapping to a least-common denominator database. Often, it is difficult to read the data outside of the proprietary application. If the middleware company goes out of business, the data format becomes a dead end, which has made companies wary in purchasing middleware. For the last two decades, these competing interests have prevented this industry from growing, though the offerings of middleware by large RDBMS vendors have offered some stability, they still offer no integration or distribution between systems.
3) XML and Text Storage. Text has long been a favorite method of flexible storage. It has archival and cross-platform qualities that are hard to beat. It can be used to store data in any format, but it is not efficient in space or time, especially for uniform data and binary data. Its hierarchical nature, and syntactic flexibility leads to slowness in scanning data. It has many standards built around many vocabularies.
4) Semantic Databases. The mapping of data to semantic models has long been used for many specialized tasks, such as natural and artificial language translations, linguistic techniques, mathematical analyses, etc. Over the last several years, the Semantic Web has been created by the semantic tagging of web page content, and various databases are being constructed to hold these tags. Their methods are limited to the tagging of textual information and knowledgebase construction.
5) Semantic-Relational Databases Semantic techniques in relational databases are available, and are efficient in space and time. These semantic-relational databases have the greatest commercial promise for implementing all of the capabilities available with equal format databases. They offer storage of all data – whether commercial accounting, warehousing, semantic, and otherwise. Semantic-relation structures can be built on any vendor's database server, are flexible and have speeds comparable to traditional relational database designs. They are new to the market.

# Equal Format Database Business Scenarios

The equal format database design can replace any other database design. The following examples show situations in which you might want to make the switch.

**Business system operations.** Your company has decided to upgrade its antiquated database and software in preparation for upcoming business needs. Following standard design practices, your software integrator determines that the new system will have two hundred tables, take three years to design, develop, and incorporate, and cost $50 million. Using equal format databases, however, the software integrator can use tools developed on previous work to cut time and costs in half. And future changes will be even less expensive.

**Business mergers.** Your company has merged with another company, and you are trying to bring the systems together. With equal format databases, you can reduce translation time, and the data can be combined and related cooperatively. You can maintain previous business functionality, and move your company to new processes in a controlled manner.

**Website catalogs.** Your company wants its customers to be able to order items from your website. Since you sell a variety of items, your database needs are large. For example, the DVDs need to have information about writers, directors, actors, plot, running time, and more; the lawn mowers have engine size, drive train, fuel type, and other information. Traditional relational design would require merging of separate databases for each product line, and it is expensive to add new product lines. With an equal format database, all of your categories of goods would be covered with much less effort and expense.

**Catalog data distribution.** Every time you update your catalog, you add new attributes that require programming changes. With equal format databases, this work is greatly reduced or eliminated. You can even create a global data distribution system that can deliver catalog information to your customers in an easy-to-use format.

**Product lifecycle management scenario.** Lifecycle management systems allow you to collect, manage, and track your data throughout the lifecycle of a product or project, but they are expensive and are made to work with specific products. Equal format databases provide an alternative infrastructure that can be tailored to your business practices and choice of products.

**Data integration.** Your engineering design team is using several software products to do its work. The team is spending a lot of time comparing the information in the separate databases to verify that they all carry the same design state. With an equal format database, you can create a core database, or create an integration system to read data from all of your databases, allowing any authorized team member to view the data. This keeps the engineering team data synchronized and reduces design errors.

**Data warehousing and business intelligence.** Your operations database collects data efficiently for current operations, but to analyze this information, you have it extracted, transformed, and loaded into custom designed data warehouses. With equal format databases, you do not have to change the format of the warehouse as your needs evolve, and your analysis tools can work with a uniform data design which can be related back to the original data.

**Data archiving.** Your database was custom designed and continues to meet your needs. But will the data still be accessible in the future? Equal format databases will allow your data to be accessed beyond the lifespan of the original database system design, providing an archival format more accessible than the original format.

## The Benefits of Equal Format Databases

Equal format databases provide a single design that can hold any and all of your data – past, present, and future. Such flexibility offers many benefits for your investment.

- **Simpler implementation.** Equal format databases remove a costly portion of any database work – the table design and the data management processes. This accounts for a substantial portion of the work and time, not only in creation but in the synchronization needed in its behalf. For simple products, as much as 80 percent of a project can be organized around its data management processes.

- **Change-free structure.** When you store data in an equal format database, you can change the style and content of your data without reprogramming the database system. If your forms and reports are designed to take advantage of this, you might not even need to change any software – you can just add the new data. And if you want to re-organize your data, you can do it without changing the table design, allowing co-existence of generations of data. Now you can spend your software budget where you really need it, such as adding application capabilities.

- **Point-in-Time Data Storage.** Now that data is atomized, you can track data with better precision, including point-in-time data storage. Updates and Deletes can be done nondestructively, providing the capability of selecting data from any point-in-time and speeding data manipulation. Less synchronization and triggering are needed.

- **All-in-one storage.** Anything that you used to store in other formats can now be stored in an equal format database. Traditional storage methods such as normalized relational tables, data warehouse fact tables, XML storage, hierarchical structures or object-relational structures can be replaced. Even point-to-point solutions and extract-transform-load solutions become less complex or unnecessary.

- **Easy data transport.** Structural differences are no longer the barrier. Data flows more easily and securely when you use equal format. You can extend your system across your enterprise, including your suppliers and your customers.

- **Global data distribution.** You can distribute catalog and sales information in equal format databases. Delivering data to your customers and dealers in a uniform manner makes most design changes unnecessary when the data format changes. Detailed information can be shared in a vendor-neutral format, providing data distribution, management, analysis, and persistence across your enterprise. Varying dimensional units can co-exist on a per-item basis, allowing global product distribution support.

- **Better reusability and easier learning curve.** You can improve the learning curve of database users when skills and knowledge are reused and previous work can be applied. Connect your software to an equal format database, and users who are already familiar with it are ready to go.

## Bringing Equal Format Database Design to Your Company

Traditional database designs are technological dead ends, eventually draining the resources of businesses with constant changes. We all recognize the limitations, yet we haven't moved to change because there hasn't been a better alternative until now. Well, times have changed and with them have come the equal format database, allowing you to store data in a format that is accessible by any database management application or participating software product. Flexibility will rule, and there will be less need for upgrades and less time waiting for your software to catch up to your business needs. With an equal format database design, you can always access your data, even when your company's needs change and grow.