# Pulling Designs into an Enterprise Database

Rhodes Hileman
**B. E. Meyers & Co., Inc.**

## Overview

We are going to look at a design data management system for electronic engineering in a startup company. Typically this system is used in the first three to five years of a new enterprise. After that it would be superseded by a real ERP system. The system is based on SQL so data migration is straight forward.

There are some differences between the needs of an MRP for manufacturing and such a system for engineering. This system is built to provide the appropriate flexibility for engineering, while presenting also clean and "rigid" output to manufacturing, but it is not designed for manufacturing inventory management, although its functions could be so extended.

## History

Electronic design startups need to manage their design data in a way that presents unambiguous fabrication and assembly documents to manufacturing. The mission of the present system was/is to capture the engineers' intentions accurately and completely, and to reuse prior (known good) work as much as possible, keeping the count of part types low, and delivering usable information to the purchasing and manufacturing teams.

The first step was to create a part naming protocol. We adapted an existing protocol, with a long track record, to our present needs, mostly by shedding an eight character limit for sixteen, and taking account of DX's reserved characters.

Once parts had reliable names, we created a new parts library that EEs never add to; they just draw from it. The library had no legacy parts, only parts created under the new rules. When new parts are needed, the EEs ask for them and pass us datasheets; we build the parts and decals. Their older libraries are available to them, but all finished designs draw exclusively from the new library.

The next step was to give all components unique part numbers for use in inventory and purchasing. This was done using a common seven digit system consisting of three digits of class code and four of base number.

Research demonstrated long ago that numbers longer than seven digits are subject to increased error rates. The three digit class code is meaningful and not unique. The four digit base number is unique but not meaningful; it is just pulled from a serial set of numbers.

In the four years since the introduction of this part numbering scheme, with about 50 electronic products, we have used about 870 base numbers, and the rate of new number creation has slowed to about ten per month. So 9999 distinct part types is plenty for the first decade of an electronics business.

**Database structure**

There are four basic tables in the system (fig. 1).

The top level is the "products" table (fig. 2 – next page). Every electronic product has a product ID number (the primary key) and a corresponding form known as the "product page". This is the fundamental index of design data for that product. Revision control is managed on this page. When an assembly package is generated, a PDF "snapshot" of this page is included in the package to show the state of all documents at the time of that release.

The "product page" shows the location, on a company server, of specifications, block diagrams, schematics, PCB designs, fabrication files (gerbers, PDFs, a CAM350 file, and an IPC netlist), assembly files (drawings, BOMs, special instructions, stencils), and embedded and diagnostic software for the product.

The product page lists all relevant ECOs and provides hot links to them. The current revision levels of all documents are gathered in a block. The EE responsible for the design, and the PCB designer are listed. The various names and numbers for the product (in different departments) and the larger systems to which it belongs are all listed here.

This is the place where any work should start. Certainly an assembler or technician beginning work on a design should look here first to be sure he/she is starting with current documentation.

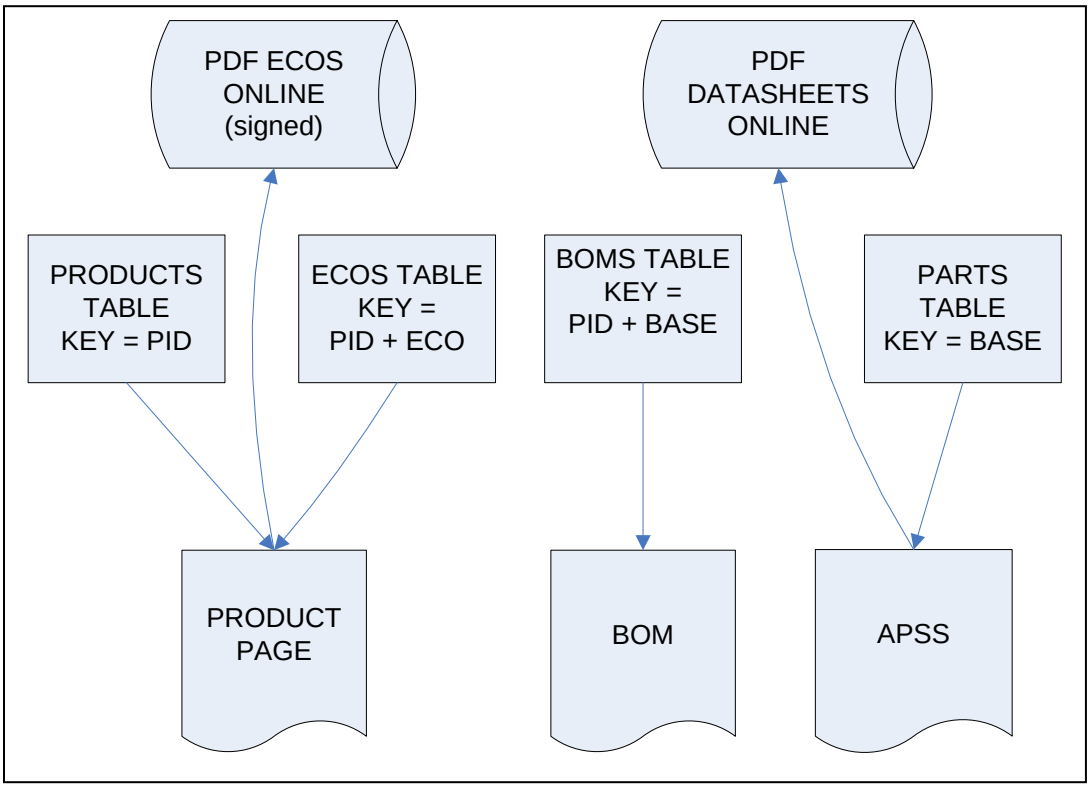


**Figure 1 - Tables, PDF links, and report forms**

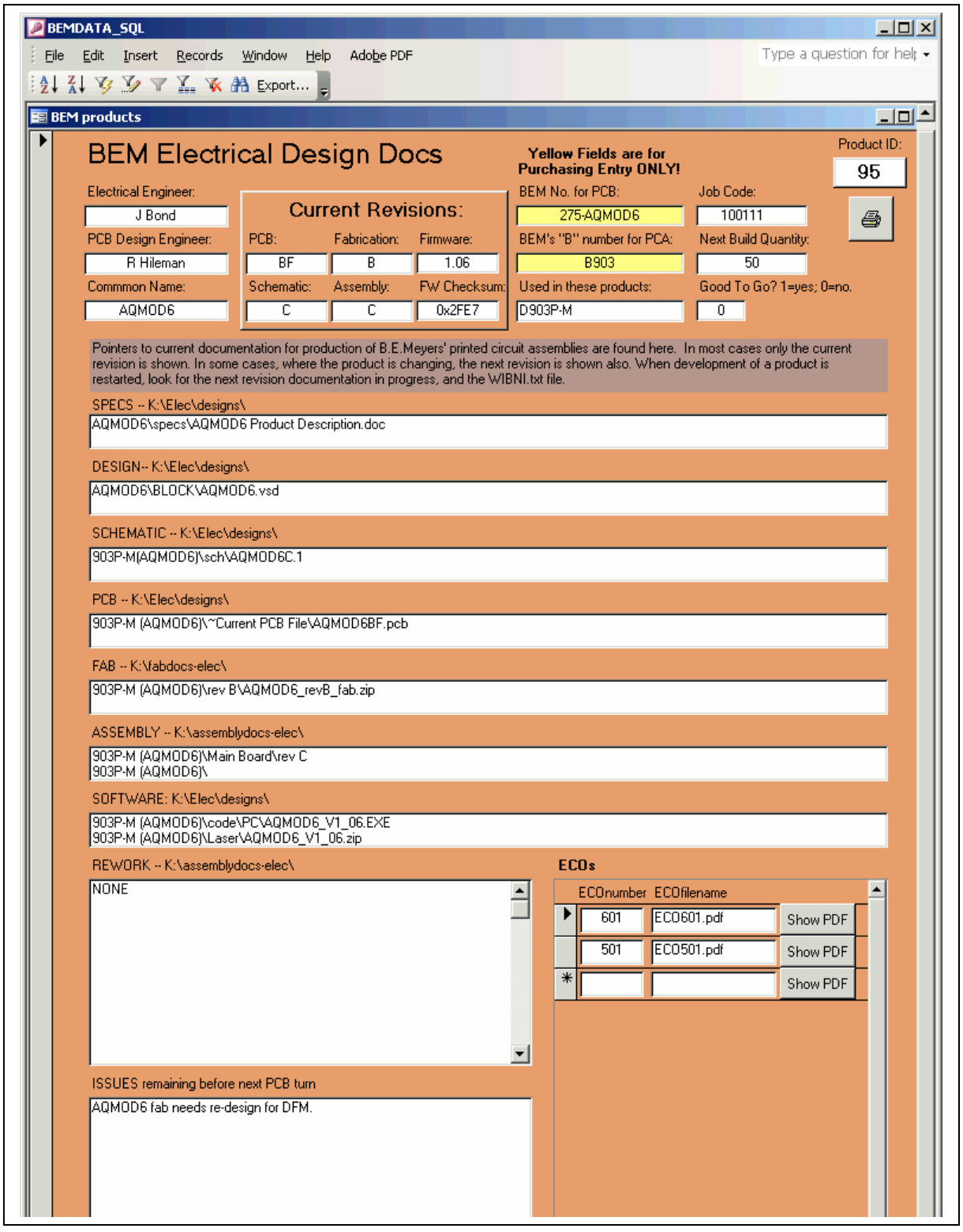


**Figure 2 - Product page**

The bottom level of the system is the "parts" table where the APSS (Approved Parts Specification Sheet) resides for every component. The primary key for this table is the base number. This is the only table in our system that DxDesigner sees.

The APSS form (fig. 4 - next page) shows the name used for the part type in the schematic library, the name of the PCB decal, class code and base number, our description line, the description line used by the primary distributor, the manufacturer and their PN, the distributor and their PN, up to two alternate sources, and crosses, and various fields descriptive of the part.

A pushbutton link to the PDF datasheet on our server is included, and the file name is shown. A where-used list is presented, with reference designators and PID numbers. Other fields show what quantity levels to buy, a judgment call usually based on the extent of the where-used list and the price of the component. Comments and special requirements have their own memo fields.

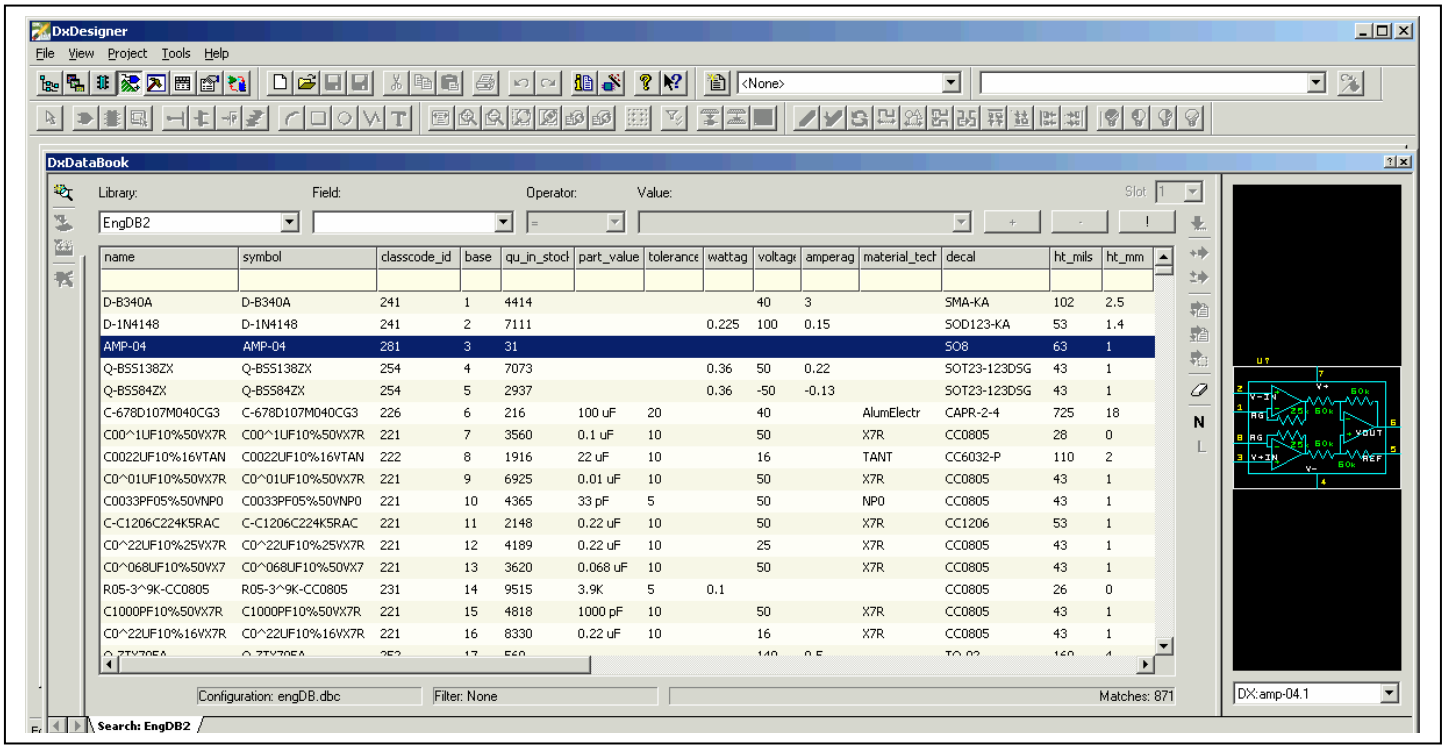


**Figure 3 - DxDatabook view of the Parts table**

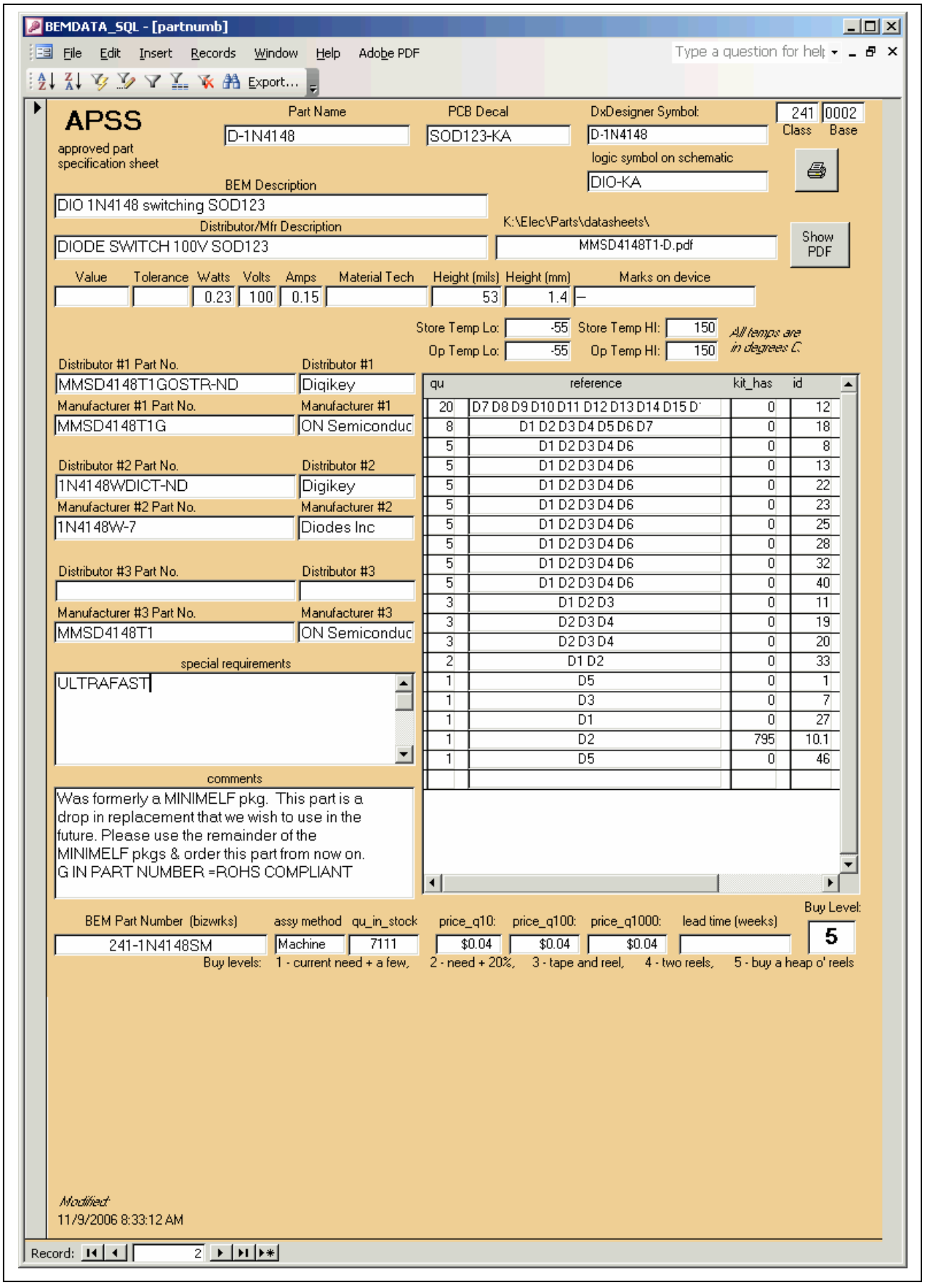


**Figure 4 - Approved Parts Specification Sheet**

Next is the BOMs table. The primary key to this table is a "unique" combination of two keys: the PID number and the base number. Once you have selected a PID, a BOM can be printed for that product.

The BOM report from MS Access (fig. 5) will show in its header the revision levels for schematic, fab, assembly, the PID number, the product name, and other manufacturing numbers for it.

Besides a generous selection of fields from the "parts" table, some special columns show reference designators for parts that are to be stuffed, a "DNP" column for parts to be bought but not stuffed yet, and a "NoBuy" column for parts that exist in the design data that should or could not be purchased such as pad patterns for wires, or components not part of this configuration. The BOM, like the APSS, also shows whether the part is suited to machine placement or is to be stuffed by hand.

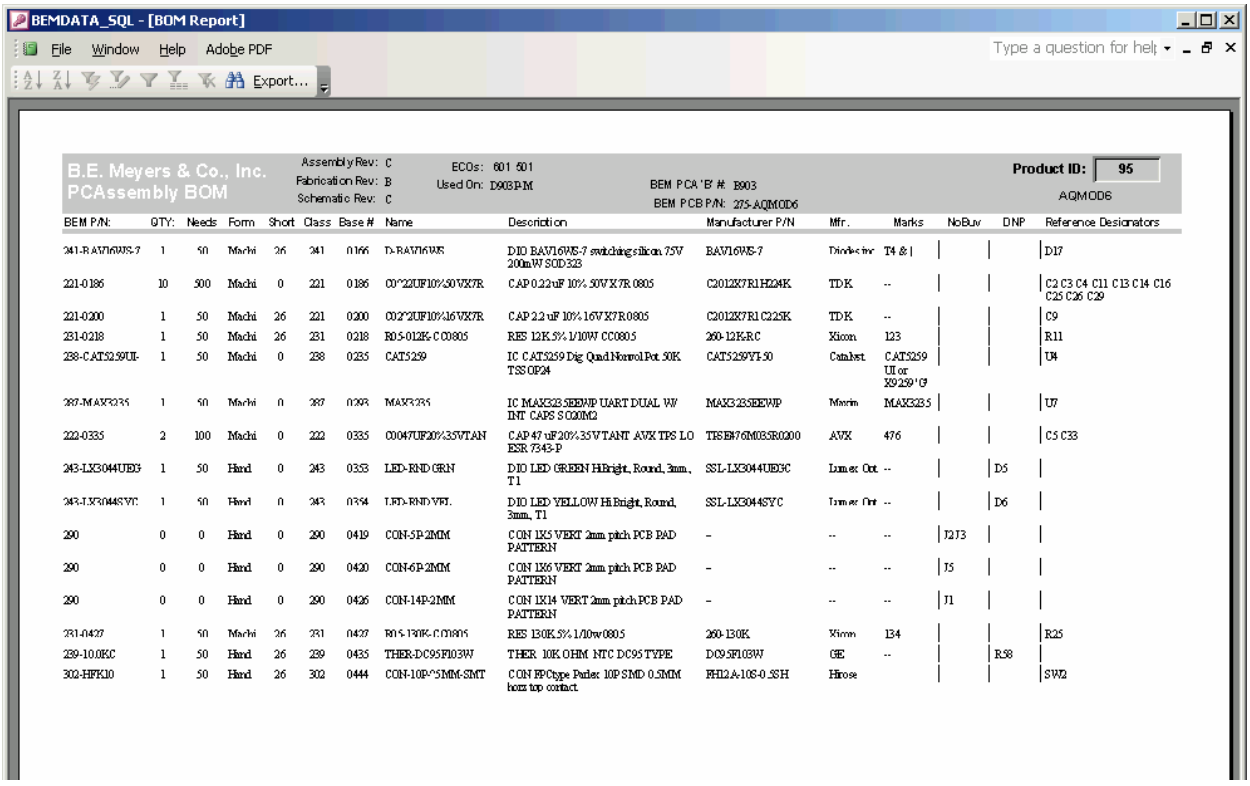


**Figure 5 - Bill of Materials report**

The last of the basic four tables in this system is the "ECOs" table. Again a "unique" key is made from two fields, the PID number and the ECO number. Some ECOs refer to more than one product. This table (in Access) automatically generates the name of the ECO PDF file. The ECO list in the product page provides a link to those PDFs.

Another trick from the "parts" table is “bag labels” (fig. 6). This is a report form that prints on Avery address label sheets. It prints a basic set of fields for each component by drawing from the “parts” table. The list of components can be selected for a particular BOM by filtering for a PID. This makes a handy tool for pulling a kit, or an entire label set, maybe for bin boxes, could be printed.

Inventory and purchasing use another report from the BOMs table, the “open items report” (fig. 7 - next page), to easily assess the components still needed by a kit. This requires that the next build quantity is entered into the product page, to drive demand, and that the “kit has” column is filled in as the kit is pulled. This serves some limited MRP functionality.

There are several other tables in the system which are not demonstrated here: buycards, purchase orders, vendors, manufacturers, representatives, Corrective Action Preventive Action (CAPA), non-conforming product (NCP), test data, product models, and a serial numbers table with UID label and RFID label printing capabilities.

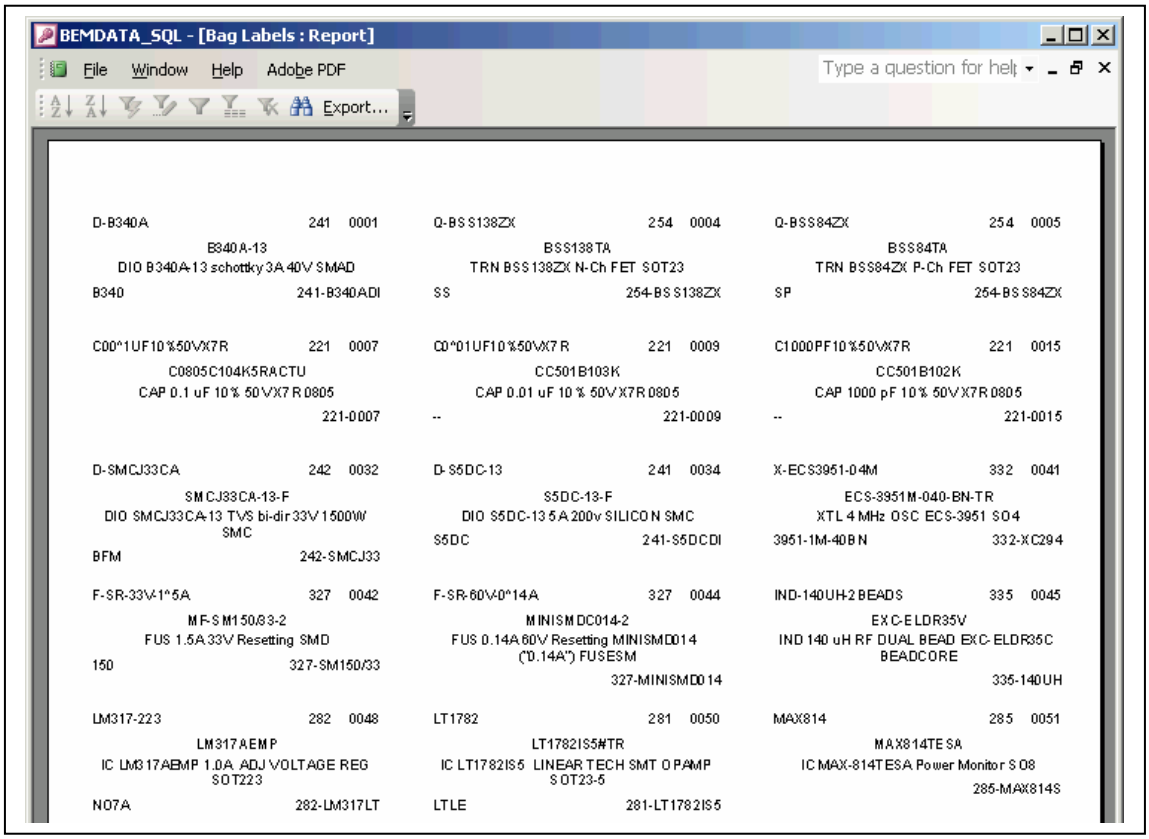


**Figure 6 - Bag Labels for a Kit.**

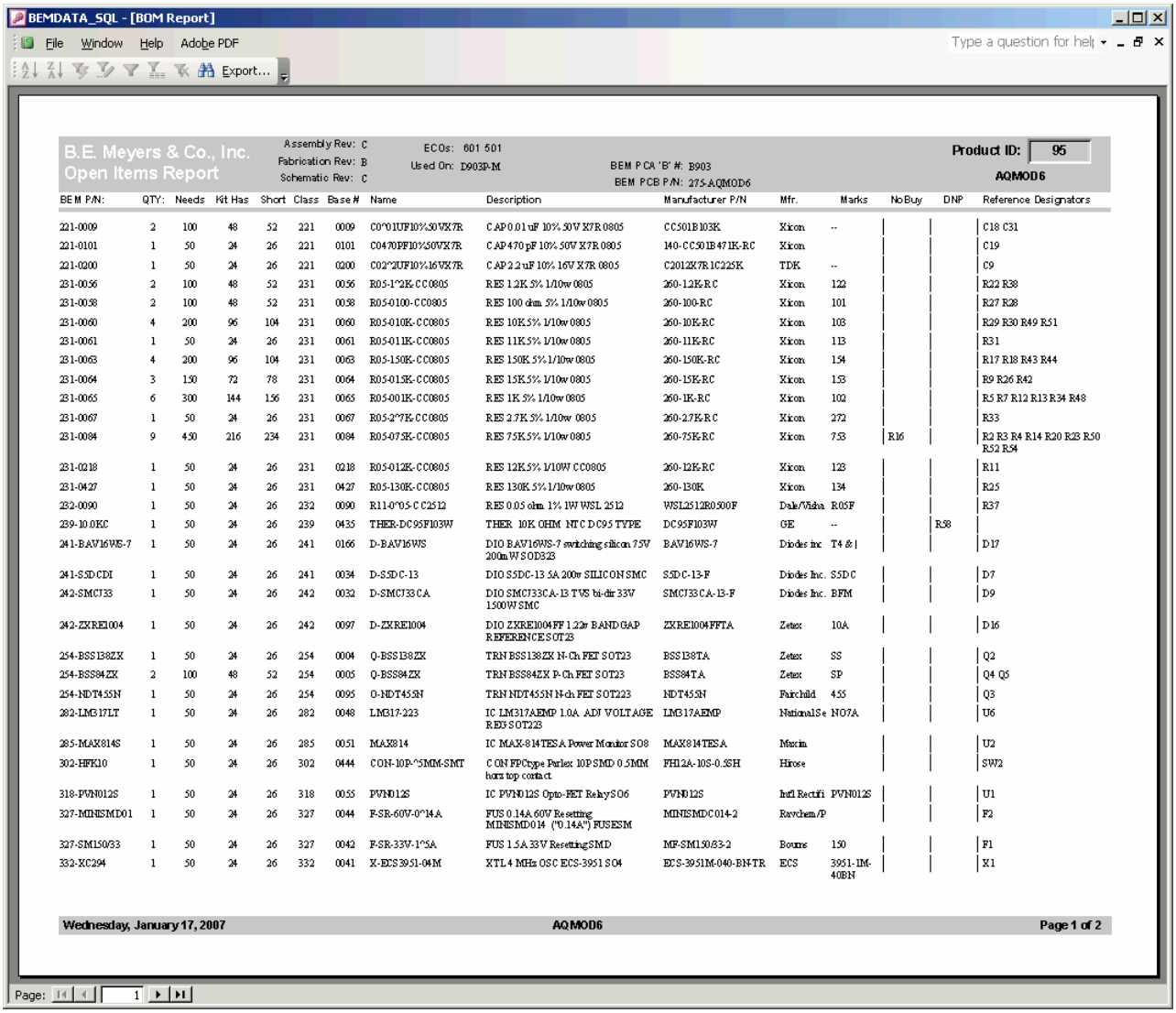


**Figure 7 - Shortage Report from Inventory to Purchasing, a.k.a. "Open Items"**

## Operation

Using the manufacturer's datasheet, we enter the parts in the database first, where all the particulars are coordinated on one page, the APSS. Then we build the library items. Once all the new parts are created, we open up, or create, the schematic and drag and drop the components from DX Databook (fig. 3 - page 4) onto the schematic page in DX Designer. As soon as we can create a parts list and PCB net list without errors, the schematic is printed and reviewed by the EE, and the process iterates. When we freeze the schematic, we import its parts list to the BOMs table using a PID number (product ID) assigned previously.

At this point a preliminary BOM report can be printed, the buyers can get to work on the long lead parts, and PCB design in PADS starts too. We need to discover if the BOM has some hope of fitting in the mechanical envelope provided. Both tasks are important feedback to the EE: 1) his design is buyable and 2) it fits. Or not. In either case it's good to find out early. From this point forward all changes are done by ECOs.

A set of cards are fabricated and a few assemblies are built up by hand in-house. These cards are programmed and tested and the flow of ECOs begins. When the EE and the test tech are happy, we freeze the design and send a larger kit out for the assembly of somewhere between one and ten panels of fabs.

For the assembly data package ("assembly.zip"), the schematic is printed to PDF, as are the BOM, assembly drawing, PID page, rework notes, and relevant ECOs.

These files are on our servers with read/write access limited to two database administrators, and read/only access for technicians, assemblers, and buyers. Engineers still have read/write to their schematics. All design documents -- schematics, PADS PCB job files, gerbers, and the assembly zip package, -- reside within a standardized structure online. They are all controlled by limited directory access to establish accountability.

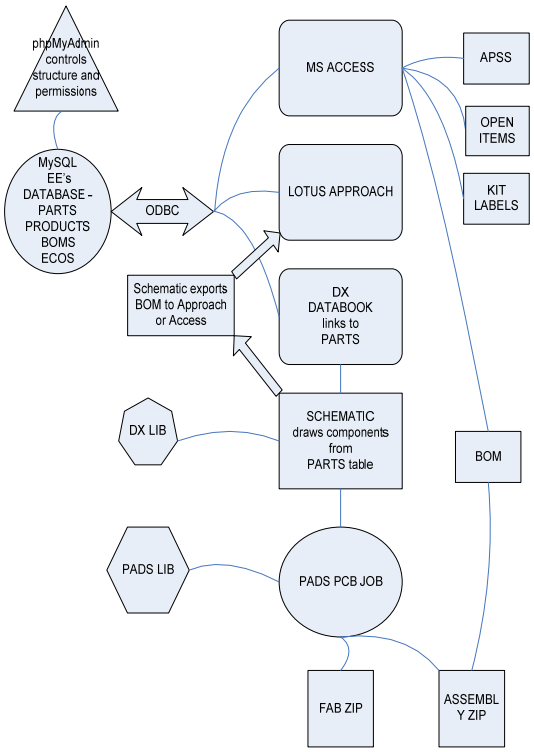


**Figure 8 - System structure**

## Administration

The MySQL database we use is administered using phpMyAdmin. We use MySQL version 4.0.20a-nt and phpMyAdmin version 2.8.1. We are using Windows IIS web server and running php on a mostly dedicated Win server which also houses the MS Access file and the Lotus Approach file. We all have shortcuts to these same two front end files though we limit editing them to the database administrators. Everyone has either Access or Approach, or both, on their local machines.

phpMyAdmin is a great tool (fig. 9). It provides for MySQL database administration through a web browser, using php running on a web server. It is password protected. Administrators can create and configure new tables, new users and permissions, new databases here. Exporting and importing the entire database, or just the structure, or just the data, is easily and quickly done from this tool. The resulting export file is a readable text file and surprisingly small.

We can still zip the entire database onto a single floppy. Permissions can be controlled down to the field level, but we only do it down to tables. We have found that we can trust users to stay out of the fields they have no business changing. If that policy fails we can resort to field level control, but it is overhead we'd like to avoid.

In calling Acrobat Reader from MS Access for launching PDFs listed in the database, we ran into a problem with an interesting and amusing solution. Not everyone in the company has the same version of Acrobat, and the versions incorporate the version level into the directory name wherein resides the executable, so we could not call Acrobat directly from within a script in Access. Our solution was to create a batch file in a fixed server location that passed the call for the file directly to the client Windows, which uses its registry to figure out where its Acrobat is for opening the PDF file type. It is a single line batch file, to wit "Call %1". We named the file "acro.bat".

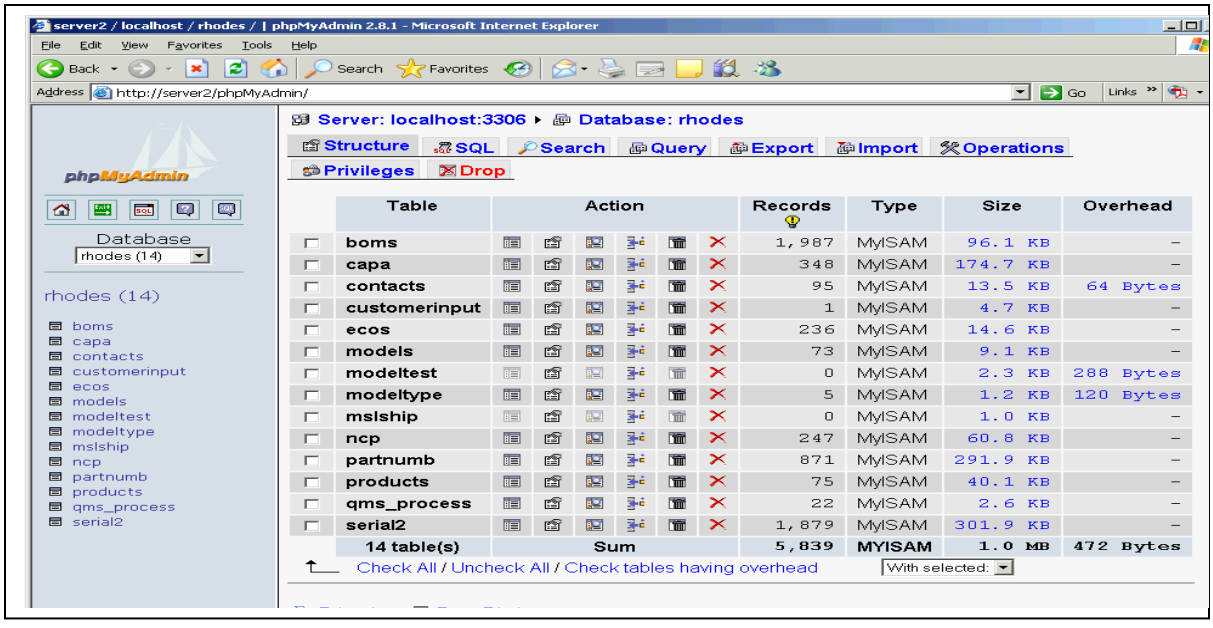


**Figure 9 - phpMyAdmin tool for administering a My SQL database.**

**Summary**

We have shown you a robust system for managing electronics design data that can be made by a database analyst in collusion with an electronics technician. Although experience with the design process is required, the creation of the system is neither costly nor terribly difficult, and you can build as you go. The system can easily migrate to the next level of enterprise software that will be implemented after a few years of company growth.

The system encourages the use of completely attributed parts in DX schematics and PADS PCB jobs, by making them easy to drag and drop from DxDatabook to DxDesigner. This tends to generate more complete design documentation.

It provides for easy reuse of parts existing in the system, and thus in inventory. This tends to limit part type count.

It reduces errors by keeping parts information in a central location that can be used by anyone in the company with appropriate SQL permissions.

It produces complete BOM reports for use by purchasing and assemblers.

It enables inventory to pull electronic kits easily, and publish shortage lists to purchasing on a “pushbutton” basis.

Most significantly, it has provided a reference system to control the revisions of all design documents.

This system enabled B E Meyers, in four years, to triple in size smoothly, and its implementation provided a foundation for the company to achieve ISO 9001 certification in 2006.