Implementing a Database and Work Station Operating System at a Regional Craft Brewery

By Matthew Swihart

Full Sail Brewing Company, Columbia, Oregon, USA.
Paper presented at the World Brewing Congress 2000, Orlando, Florida, USA.

ABSTRACT

The art and science of beer manufacture necessitates the accumulation of a wealth of information. In modern brewing prehistory, the Brewster likely recorded very little information related to the beer process or individual batch observations. Oral descriptions of the process and anecdotes of successes and failure were the only means of conveying the appropriate methodology one needed to make beer. As the guild and trades of brewing gave way to the scientific method, documentation escalated. Each developmental stage of brewing has brought a host of new items to quantify and record. In the modern production environment, beer quality can be optimized with tools associated with the efficient and rapid management of production data. Readily storing, sorting and further manipulating production information can assist brewing decisions to react to the changing environments of production.

Over the last 30 months, Full Sail Brewing Company has implemented a relational database system to ease the strain of the quantity of data recorded. Work stations, positioned in the key areas of quality control, brewing, laboratory, filtration, and packaging, have allowed individual departments to input real time data at each step of beer production. With the combination of user friendly input screens and rapid data acquisition, each department has immediate access to any beer’s specific status. The database tracks any amount of blending of product from raw materials through packaged beer. This operating system has dramatically reduced paperwork and the total man-hours associated with documentation. Immediate feedback of items such as brewhouse efficiency, plant yield, and microbiology provide a more rapid response to quality assurance issues. Analysis of last year’s quality and efficiency specifications has been used to established performance targets for each department involved with production. Employee compensation has been directly tied to brewery performance.

Keywords: relational database, quality assurance, work station, software

INTRODUCTION

The art and science of beer manufacture necessitates the accumulation of a wealth of information. In brewing prehistory, the Brewster likely recorded very little information regarding a batch of beer produced. Oral descriptions of the process and anecdotes of successes and failure were the only means of conveying the appropriate methodology one needed to make beer. As the guild and trades of brewing gave way to the scientific
method, documentation escalated. Each developmental stage of brewing has brought a host of new items to quantify and record. In the modern production environment, beer quality can be optimized with tools associated with the efficient and rapid management of production data. Readily storing, sorting and further manipulating production information can assist brewing decisions to react to the changing environments of production.

Over the last 30 months, our brewery has implemented a relational database system to ease the strain of the mounds of data we were tracking. Up to this point, paperwork had been generated at each stage of the production process. Brewers kept detail brew logs, filtration recorded any movement of beer, packaging recorded efficiencies, the laboratory tracked sanitation effectiveness, and production planning provided detailed records for the appropriate government agencies. At each of these stages, several documents were involved. Unfortunately, many of the documents contained duplicate information, potentially erroneous. For these reasons, a sensible central documentation and disseminating network was needed to streamline the paperwork, and retrieve information more rapidly. The brewery utilized a relational database system with accompanying development software to eliminate duplicate information and dramatically reduce paperwork. The system’s sorting and querying language allows rapid retrieval of information and can display this information in a variety of formats. The single database then uses a system of workstations associated with each stage of production. The workstations are customized for each user, with specific screens, summaries, and options available for various departments. Summaries include: brewhouse efficiency, beer yield tracking, material loss, packaging efficiency, packaging line downtime analysis by equipment, governmental production summary (BATF report), microbiological report, final beer QC analysis, and on-hand cellar and raw material inventories. Through the use of the database system, production efficiencies have improved and the man-hours associated with quality assurance tracking have been reduced.

**THE NEED FOR BETTER TRACKING OF BEER PRODUCTION**

Prior to implementing the new database system, brewing and packaging information was tracked using a series of standard forms associated with each stage of production. (Table 1). At the close of production each week, month, or year, each department’s documentation was stapled together and filed in a series of binders throughout the production year. To generate production reports for governmental documentation, each department would generate a production report and the production planner would collate all the information on raw material usage, plant yield, and general production barreleage.

In the area of quality control, the biggest obstacle was keeping track of any blended beer. As a philosophy at Full Sail Brewing, batch autonomy is preserved the majority of the time. We blend product on a need basis for consistency issues related to color, apparent extract, original extract, or efficient capacity utilization. (Figure 1). Tracking a brew that had been divided into two or more tanks became quite problematic. When reviewing production information, we could not specifically generate exact wort or raw materials residing within a given bottle of beer on a specific production date. If a beer had a microbiological concern, the only way to track the beers or tanks involved was

![Diagram](image)

**FIGURE 1**

In any given package, raw materials and product may be blended.
to sort the production binders by hand and draw a flowchart of a beer's progression- a laborious an inexact process. For these reasons, a convenient database to allow each step of production to input QA and tracking information was developed.

**TABLE 1**

<table>
<thead>
<tr>
<th>Documentation Area</th>
<th>Pre-database manhours per week</th>
<th>Post-database manhours per week</th>
</tr>
</thead>
<tbody>
<tr>
<td>Raw Material Analysis</td>
<td>0.25</td>
<td>0.17</td>
</tr>
<tr>
<td>Wort Production</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Filtration</td>
<td>0.75</td>
<td>0.38</td>
</tr>
<tr>
<td>Fermentation</td>
<td>2.0</td>
<td>2.0</td>
</tr>
<tr>
<td>Yeast Handling</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Production- Bottling &amp; Kegging</td>
<td>1.0</td>
<td>1.0</td>
</tr>
<tr>
<td>O.A. &amp; Lab Data-Bottling &amp; Kegging</td>
<td>1.33</td>
<td>1.33</td>
</tr>
<tr>
<td>Microbiology</td>
<td>2.0</td>
<td>1.0</td>
</tr>
<tr>
<td>Brewing Summary and Efficiency Reports</td>
<td>3.0</td>
<td>0.17</td>
</tr>
<tr>
<td>Weekly Lab Reports</td>
<td>2.0</td>
<td>0.17</td>
</tr>
<tr>
<td>Documentation for Government Reporting</td>
<td>2.0</td>
<td>0.25</td>
</tr>
<tr>
<td>Total Manhours per Week</td>
<td>17.3 hours per week</td>
<td>9.47 hours per week</td>
</tr>
</tbody>
</table>

**RELATIONAL DATABASES**

Relational databases are the modern result of data storage evolution. By definition, any large amount of information that is stored is a database: a phone book would be a simple example. In the modern world, most large amounts of data are stored electronically. The system software that supports data (input, retrieval, sort, calculate) would be known as a database management system [1.3-5]. Of course in the brewing world, the desired stored information contains batch numbers, cell counts, extracts, maturation times- any host of information of interest to the brewmaster. The advent of computer storage increased our ability to store such information, but it also maximized our ability to store unnecessary or duplicative information. Since computers have become more commonplace, and information storage more prolific, the need to streamline and economize the storage of information became a necessity. Relational databases represent a breakthrough in data storage to maximize efficient database manipulation.

The relational database structure was introduced to the computing world by Dr. Edgar F. Codd in 1970 [2]. Codd suggested that databases be separated into a series of tables of unique information. In a brewery, all malt suppliers, and malts available might be listed in a malt table. Hops suppliers would be stored in a separate table. Each of these tables and the way they relate to one another was then further characterized by a series of rules Codd developed and simplified by the mid 1980s. These rules of normalization center on the efficient storage of information and are necessary to avoid storing duplicate information [3, 50-52]. Fortunately, there are a host of software applications that are designed to help database designers set up their information in an efficient manner. A few currently available relational database management systems are listed in Appendix I. While the author implemented our database system utilizing Microsoft's ACCESS, it is not the author's intention to formally promote any specific software system.

**TABLE DESIGN: THE HALLMARK OF AN EFFICIENT RELATIONAL DATABASE**

Tables are used in relational databases to store the lists of information. For an optimized relational database design, Codd suggested a series of rules which are listed in Table 2. These rules suggest separating information into related tables and using a unique field in each table to identify any given record. For our implementation, the brewing process was analyzed from raw material arrival through production and on to final product.

**TABLE 2**

Codd suggested several rules to efficiently structure information in a relational database.

**Table Design Criteria:**

1. Separate information into unique tables, all fields should relate to the table's objective
2. Fields should be broken down into their most simple elements
3. One field in each table should be able to identify all the fields in its row or record
4. That unique field, the primary key, cannot be have duplicate records at any time
5. Do not store any fields that calculate values based on other fields in the same record

Table 3 shows a listing of the tables used to differentiate our process into unique segments. Tables 4 and 5, show specific examples of the unique tables “Hop Table” and “Malt Table”, and within those tables, the unique field identifiers or primary keys, “HopID” and “MaltID”, have been used to uniquely describe each record in each respective table. Per Codd's rules, there should not exist any duplicate values for the “HopID” or “MaltID” primary keys.

In addition to the unique tables for each production step, intermediary tables (“Hop Brew”, “Pale Brew”, and “Specialty Brew”) were created to track multiple material additions to single worts. In this way, an unlimited amount of hop additions or
TABLE 3
Breakdown all process information into simple and unique tables that describe and document each step of production.

<table>
<thead>
<tr>
<th>Table Name</th>
<th>Unique Field Identifier</th>
<th>Some relevant field data recorded</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pale Malt</td>
<td>Pale ID</td>
<td>Shipping date, malt analysis numbers, lbs delivered ...</td>
</tr>
<tr>
<td>Hops</td>
<td>Hop ID</td>
<td>Variety, supplier, alpha acids, season ...</td>
</tr>
<tr>
<td>Specialty Malt</td>
<td>Specialty ID</td>
<td>Malt type, color, supplier, extract as is, moisture ...</td>
</tr>
<tr>
<td>Wort</td>
<td>Wort ID</td>
<td>Production data, brand number, strike temp, wort color, key production times, pH, hop brew ID, pale brew ID, specialty brew ID, Lab ID</td>
</tr>
<tr>
<td>Hop Brew</td>
<td>Hop Brew ID</td>
<td>Wort ID, hop ID, lbs of hops used</td>
</tr>
<tr>
<td>Pale Brew</td>
<td>Pale Brew ID</td>
<td>Wort ID, pale ID, lbs of pale malt used</td>
</tr>
<tr>
<td>Specialty Brew</td>
<td>Specialty Brew ID</td>
<td>Wort ID, specialty ID, lbs of specialty used</td>
</tr>
<tr>
<td>Brand</td>
<td>Brand ID</td>
<td>Brand description</td>
</tr>
<tr>
<td>Beer</td>
<td>Beer ID</td>
<td>Beer ID, Wort ID (table to connect all worts into one beer)</td>
</tr>
<tr>
<td>Extract</td>
<td>Extract ID</td>
<td>Date &amp; Time, Extract, Temperature, Beer ID, Pressure, Lab ID</td>
</tr>
<tr>
<td>Filter</td>
<td>Filter ID</td>
<td>Date, dissolved oxygen upstream and downstream, clarity, volume, Lab ID ...</td>
</tr>
<tr>
<td>Blend</td>
<td>Blend ID</td>
<td>Wort ID, Lab ID (this table tracks any and all blending of any beer)</td>
</tr>
<tr>
<td>Keg</td>
<td>Keg ID</td>
<td>Date, quantity, package type (1/2 bbl, e.g.), D.O., Lab ID ...</td>
</tr>
<tr>
<td>Bottle</td>
<td>Bottle ID</td>
<td>Date, quantity, package type, D.O., Lab ID ...</td>
</tr>
<tr>
<td>Blend QA</td>
<td>Blend ID</td>
<td>Beer color, apparent extract, pH, clarity, Lab ID ...</td>
</tr>
<tr>
<td>Lab</td>
<td>Lab ID</td>
<td>Date, Growth ID</td>
</tr>
<tr>
<td>Growth</td>
<td>Growth ID</td>
<td>Date, colony shape, quantity, color, morphology, gram positive or negative, ... (In the growth table, Growth ID = 1 represents no growth)</td>
</tr>
</tbody>
</table>

Efficient table design also suggests limiting any calculations in data storage areas. If a brewmaster wishes to calculate any values based on information recorded in the fields, those values must not be stored as a separate field identity. Assume one wished to know the extract efficiency of the lauter tun, and already recorded are the values of As/ls laboratory malt extract, various malt masses, kettle extract and kettle volume. By recording the extract efficiency as a separate field identity, the database designer is duplicating information already available in the existing fields. For this reason, calculations are not stored, but run on an as needed basis at the time of application. Most database management systems support running calculations as queries-a topic discussed later.

TABLES 4 AND 5
Separate Information into Simple and Unique Tables.

TABLE 4
Hop Table

<table>
<thead>
<tr>
<th>Hop Number</th>
<th>Variety</th>
<th>Alpha Acid</th>
<th>Supplier</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Cascade</td>
<td>6%</td>
<td>A</td>
</tr>
<tr>
<td>2</td>
<td>Saaz</td>
<td>2.4%</td>
<td>A</td>
</tr>
<tr>
<td>3</td>
<td>East Kent Golding</td>
<td>5.3%</td>
<td>B</td>
</tr>
<tr>
<td>4</td>
<td>Hallertau</td>
<td>3.4%</td>
<td>B</td>
</tr>
</tbody>
</table>

TABLE 5
Malt Table

<table>
<thead>
<tr>
<th>Malt Number</th>
<th>Variety</th>
<th>Moisture</th>
<th>Color</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>Dextrin</td>
<td>5.1%</td>
<td>10</td>
</tr>
<tr>
<td>2</td>
<td>Caramel 1</td>
<td>4.6%</td>
<td>26</td>
</tr>
<tr>
<td>3</td>
<td>Caramel 2</td>
<td>4.2%</td>
<td>78</td>
</tr>
<tr>
<td>4</td>
<td>High Roast</td>
<td>1.8%</td>
<td>600</td>
</tr>
</tbody>
</table>

LINKING THE TABLES TOGETHER

Once the tables are designed, a method for the tables to talk to one another must be created. Linking allows the small independent and unique tables to act as one giant table of information. Each specific software package managing the database will have various ways to link the table information together. The linking method will involve a field in one table that is tied or related to a field in another table. The most efficient table and
linking designs link a primary key in one table to another table’s respective field. For instance, within our wort table, there exists a field known as “brand ID”. The “brand ID” field is linked to the unique primary key of the brand table called “brand 10”. Through this method, the wort table now contains not only all the information related to wort production, but links all the relevant information related to a certain brand of beer. It does this without having to duplicate that information multiple times, when many brands of the same type are brewed and recorded in the wort table. As a good database design practice, linked fields will often have the exact same name for easy clarification.

The type of link between the many tables can also be specified in various database languages. Within the logic of link types, tables can be connected when the linked fields have identical values, an inner join, or when a table has only one value represented, known as an outer join. In our application, all links were inner joins. For a full discussion of join applications in a brewery environment, please reference Takacs’ MBAA article, [1, 416-418].

THE BLENDING SCHEMA

In order to facilitate any number of blending operations, an intermediate table was created to record the beer records associated with one another. Within the new blend table, a field was created known as “blend ID”. Each time a beer is moved, whether racking or filtering, a new blend ID is created and added to the blend table. The beer ID’s of the racked beer are carried into the table and duplicated along with the blend ID as needed. If only one beer is moved into an empty tank, only one record is created in the blend table to record the new blend ID and the beer ID that was racked. However, if the destination tank still contained beer, several beer records would be recorded into the blend table. The first with the new blend ID and racked beer ID, the second record would contain the new blend ID, and the beer IDs of the resident beer already in the tank. Ratios of beer ID are also calculated at each step to quantity the relative amount of each beer ID in the entire blend. Appendix II lists the subroutine flow chart used to add records to the blend table.

THE WORK STATION ENVIRONMENT

Once the essential spine of the database had been created, Figure 2, individual workstations were set up to allow each department to access their respective data. All the main data tables were stored on the brewery’s central network server. Utilizing the LAN (local area network) applications of our network, we placed personal computers in the key areas of brewing, filtration, laboratory, packaging, and quality control. Once established, the database software system, in our case Microsoft ACCESS, was installed at each station. Only the needed forms and queries necessary for each department were placed at the local computer. The individual departments would access and input their production information, with the data being permanently stored on our main server. For instance, the filtration operator would view screens only pertaining and accessing data within the filtration, blend, and tank tables. Filter QA data, dissolved oxygen, clarity, barrel age, could be stored within the appropriate tables and reports could be generated at any interval to display filtration volume, QA information, yield calculations, and anything else of interest to the filtration department. The screens are designed within the database operating system and act as a user-friendly interface to the data tables. It is not necessary for the work station users to view the raw data entry. A typical filtration screen and associated data table are shown in Figure 3. The filtration operator opens the filtration screen, chooses the tanks he/she will filter beer to and from, enters the volume filtered, dissolved oxygen in bright tank, and clicks an action button. The button activates the internal codes within the database management system to update the necessary inventory tables and create the needed blend records within the blend table.

![FIGURE 2](image-url) Link tables together to fully enable the relational database structure. This schematic represents a simplified linking of raw pale malt through any amount of blending to final bottled product.
information at hand. A good database management system must
the information. A query at the filtration station would ask what
filter tables, sort and order, and perform calculations on any
allow for quick and easy retrieval of any contained information.
mate value in information is the rapid sorting and ordering of the
given number of fields 14,4211.

VOLUMES OF BEER ARE ON HAND THROUGHOUT THE BREWERY AND THE STA-
locally at the PC and reflect the needs of the individual needing
ING TO SEVERAL MATHEMATICAL DATABASE THEORIES. IN THE MANY MOD-
easy to use and have a wealth of help information entrained in
ERN MANAGEMENT SYSTEMS, THESE QUERY LANGUAGES ARE OFTEN VERY

The packaging manager has
queries to generate daily efficiency reports of the bottling and
keg line operation. The laboratory has queries to search for any
non-standard microbiological growth throughout the plant and
to chart that growth in specific equipment. Brewing generates
queries to calculate brewing cycle times, wort color deviations,
and extract efficiencies. All these queries are only run when
accessed at the workstations. They analyze, sort, and filter the
data within the tables and display the results. They do not gen-
are any excess data to be stored within the system. They are
run when needed, and when stored locally, do not represent any
additional load on the efficiency of the database system.

These queries are often structured into nicely formatted
reports for ease of viewing to the work station user. Once for-
matted, the reports can be immediately accessed at any time in
production. Such rapid retrieval of information resulted in sig-
nificant man-hour reductions in documentation, Table 1. Each
department spends about the same amount of time in document-
whether recording the information on a piece of paper or
on the terminal, but the rapid assessment and analysis of the
stored data is where the database’s efficiency can be realized,
with a total documentation time reduced by 45%. Of enormous
benefit is the ability to write queries on the fly. A brewmaster
suddenly faced with a production challenge, e.g. reduced extract
efficiencies or increased lauter tun cycles, may quickly write a
query to view betaglucan levels in malt shipments from differ-
ent suppliers over the course of the year. Such a query would
only require a few minutes of time to the initiated, and result in
a strong tool for production management. Using these queries,
each department has analyzed their own individual performance
efficiencies over the 18 months since full database installation.
These analysis have allowed us to establish performance targets
for the coming year. At each workstation, the department can
view their team and individual production performance as a use-
ful feedback and interactive management technique. The self-
established performance targets were tied into merit increases
for the first time in January 2000. See Table 6.

<table>
<thead>
<tr>
<th>Merit Increase Point structure- 20 points possible</th>
<th>Example Performance Target</th>
</tr>
</thead>
<tbody>
<tr>
<td>Team Quality Goal – 10 points</td>
<td>• Decrease Standard Deviation in extract efficiency and apparent extract in final beer by 1% - 10 points</td>
</tr>
<tr>
<td></td>
<td>• 0.5% decrease – 8 points</td>
</tr>
<tr>
<td></td>
<td>• 0% - 6 points</td>
</tr>
<tr>
<td></td>
<td>• &gt;0 – 0 points</td>
</tr>
<tr>
<td>Team Efficiency Goal – 10 points</td>
<td>• Increase bright beer yield by 2% - 10 points</td>
</tr>
<tr>
<td></td>
<td>• Increase yield by 1% - 8 points</td>
</tr>
<tr>
<td></td>
<td>• 0% increase – 6 points</td>
</tr>
<tr>
<td></td>
<td>• decrease in bright beer yield – 0 points</td>
</tr>
</tbody>
</table>

TABLE 6
Performance targets have been established by analyzing
database data over the last 18 months. These targets have
been tied into employee merit increases for the 2000 produc-
tion year.

FIGURE 3
Screens are designed to allow friendly input into the data
tables. Action buttons on the screens activate commands to
update relevant tank inventory volumes.

ASKING QUESTIONS OF YOUR DATABASE

While a cornucopia of raw data is always intriguing, the ulti-
mate value in information is the rapid sorting and ordering of the
information at hand. A good database management system must
allow for quick and easy retrieval of any contained information.
In most database systems, there exists a searching mechanism to
filter tables, sort and order, and perform calculations on any
given number of fields 14,4211. Query languages have been written
11, 351 to allow the useful manipulation of database tables accord-
ing to several mathematical database theories. In the many mod-
ern management systems, these query languages are often very
easy to use and have a wealth of help information entrained in
their interface packages.

Queries are design specific to the user accessing the informa-
tion tables. At each of our workstations, the queries are stored
locally at the PC and reflect the needs of the individual needing
the information. A query at the filtration station would ask what
volumes of beer are on hand throughout the brewery and the sta-
tus of maturation of each beer. The packaging manager has
queries to generate daily efficiency reports of the bottling and
**TROUBLESHOOTING**

There are two major areas of development that have presented challenges to database implementation. Both areas involve the interface between the logical data structure and the human operators at each workstation. Firstly, since the database is predicated on real time data input, a major hurdle in training involved getting operators to input data commensurate with production activities. The entire structure of the database is predicated on real time data input. If a beer is filtered, it must be recorded. Upon emptying a bright tank of a brand, those brands must be recorded and the tank table updated. Problems in implementation have centered on operators unable to access their information as a direct result of another department’s lack of data input. For instance, if the packaging manager neglects to record production packaging on any given day, the bright tank may still show beer residing on the database when in reality, it does not. As a consequence, when the filtration operator racks beer and attempts to enter the filtration into the database, they find the inaccuracy of a virtual beer residing within the bright tank. The filtration cannot be recorded until the packaging manager removes the beer through production. Of even greater risk would be the filtration operator moving forward with his data input, and recording a filtration into a bright tank that has not been emptied on the database. This action would result in the creation of a blended beer that does not exist in the actual brewing environment. The result of operators not inputting data every time a beer moves can quickly snowball into a mass of incorrect data. To compensate for this potential, the database manager checks key reports each day and makes comparisons with the production the manager’s schedule. Using cellar inventory reports and reviews of daily production have alerted the database manager to errors or absences in input.

Similarly, the software to warn workstation operators of potential mistakes in input had to be developed. The interface package must contain a number of error traps that anticipate errors in beer recording. Where we have been unable to anticipate errors, we have solved with trial-by-fire. The first year of database implementation continued the older style QA recording paperwork, while, at monthly intervals, the two systems were compared for accuracy. As problems in interface structures naturally were discovered, error traps were written for each specific situation. Although this has meant a great deal of time and effort involved from the programmer, the resulting database interface is flexible in design and has morphed as needed for each workstation and operator. The resulting workstations have been tailored for specific needs and problems for each operator.

**FUTURE WORK**

The future work with our existing data structure centers on exercising the extreme flexibility of its modular structure. If one considers the blending schema as a vertebrate spine, delivering raw materials through manufacturing to a finished product, additional QA data of interest can be added at any additional stage. The QA data of interest can be added as additional tables and inserted into the spine at the key positions using the reference primary keys for each production step. Consider the example of

![Diagram of database structure]

**FIGURE 4**

Once the initial spine is created linking raw materials to finished product, additional QA items of interest can be added with additional tables.
fermentation. Already in the database exists the table for beer. All fermentations in production can be identified in the beer table with the unique field of “Beer ID”. If one is interested in adding the QA information of tracking fermentation profiles, an additional table called fermentation profile can be easily added. The fermentation profile table must include the “Beer ID” and be directly linked to the beer table and its respective primary key of “Beer ID”. Additional fields might record extract, pressure, temperature, pH, and time of reading. By recording this information and the associated “Beer ID”, a fermentation profile is documented for every beer in production. Figure 4 displays the insertion of additional data tables into the main production spine and the entire planned structure for the brewery data structure. In the next coming months, tables considered for addition include the QA tracking of pale malt analysis, fermentation profiles, shelf life stability, distributor rotation, retailer rotation, customer feedback, taste panel, and yeast generation tracking.

REFERENCES


a) Microsoft's ACCESS and SQL server
b) SyBase SQL server
c) FoxPro
d) ORACLE SQL and SQL Plus
e) InterBase

APPENDIX I
A small list of currently available relational database management systems.

APPENDIX II
Flow chart of subroutine used to create blend table records.