

Data Base Accounting 101a

by Lincoln Stoller, Ph.D.

April 1, 1994

Copyright © 1994, Braided Matrix, Inc.

What is accounting?

The goal of accounting is to improve businesses through a better understanding of revenues and expenses. Accounting information historically has played a major role in understanding both tactical and strategic quantitative issues. This includes evaluating business performance over time, understanding the relationship between sales and prices, and even deciding on an opportune time to expand business operations. This article puts accounting systems in perspective and lays out their fundamental relational data structure.

What does a RDB accounting system have to do?

An accounting application must support the goals of accounting by providing data for decision making. It must also adhere to the following generally accepted accounting principles:

- i) **Consistency:** the same information must be treated the same way over time. The exception is when changing situations cause similar treatments to be misleading.
- ii) **Materiality:** all events that might influence the judgment of an informed reader must be recorded.
- iii) **Full disclosure:** reports should include everything a user would need to know to answer relevant questions.

What DB structures are required for accounting?

The data structures required depend on whether you are implementing a partial or complete accounting system. Partial systems are the result of adding accounting functions to a system initially designed for another purpose. Although partial accounting systems are common, there is no set rule on how to design them. The best way to retrofit an existing structure is different in every case.

The alternative is to design accounting functions into the system from the start. This doesn't mean that you write a complete accounting system, but that you should begin by planning for one. There are certain major components that can be left undeveloped, but other core structures and functions must be included at

the outset. Here I consider the structure known as the general ledger, required in every accounting system.

The basic accounting model divides a business into accounts. A separate account exists for every aspect of the business being tracked. Accounts can be defined to track almost any aspect of business; the collection of all accounts is called the “chart of accounts.” The chart of accounts is to a business what a file structure is to a data base. Setting up the chart of accounts is one of the important jobs that usually requires the advice of an accountant.

All accounts begin with a zero balance. Whenever something of value enters or leaves an area of the business it is recorded as a transfer of assets from one account to another. This transfer, called a transaction, adds to the balance in one account and subtracts from the balance in the other. Clearly, if all balances start at zero and a transaction causes one to go up while another goes down, then account balances will be both positive and negative.

As an example, when a customer purchases something in cash, a transaction is entered raising the balance in the cash account and lowering the balance in the sales account. A positive balance in the cash account means that there is money in the bank. A negative balance in the sales account means goods and/or services are being sold. The more negative the balance of the sales account, the more that is being sold.

One of the confusing conventions in accounting is that sometimes growing negative balances indicate certain quantities are increasing, as in the amount sold in the case of sales accounts. Of course it is essential that everyone agree on which accounts have balances that increase as assets go in, and which accounts have balances that decrease. The accounting field has resolved this with a set of conventions dividing accounts into different types and specifying how each type is handled.

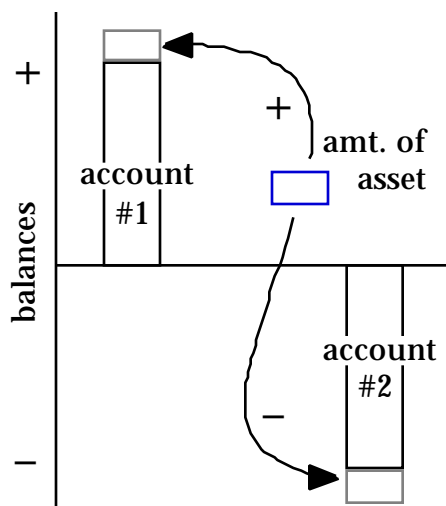


Figure 1: Every increase in one account is offset by a decrease in another account's balance.

The essential thing to remember is that assets move from account to account — they are never created or destroyed — and every transfer is recorded with a transaction. This tracking of the origin and destination of assets is the basis of double-entry accounting. The concept generalizes to transactions with more than one source or destination, though the system is still called a double-entry system.

The simplest system contains only accounts and transactions. The data structure consists of an accounts file and a transactions file. Each account should have:

- i) an ID number,
- ii) a name,
- iii) a balance representing its accumulated assets.

The simplest transaction has two components. If we hard-code two components for each transaction we can store all the transaction information in a single file. Each transaction record will then have:

- i) an ID number,
- ii) a description,
- iii) a date of entry,
- iv) a dollar amount representing the value of the asset being transferred,
- v) the ID of the account the asset is coming from,
- vi) the ID of the account the asset is going to.

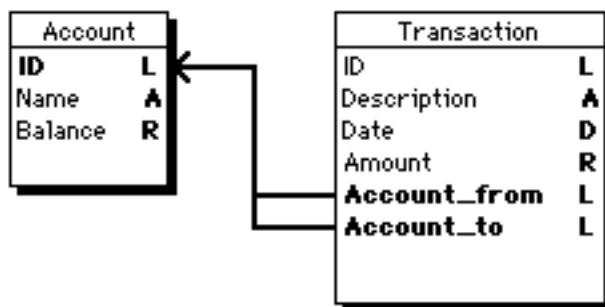


Figure 2: A very simple core data structure.

There are many ways to improve on this structure, some of which I'll consider in subsequent articles. But the structure is sufficient to illustrate the main features of an accounting system.

The posting process and the general ledger

One of the most important processes in accounting is that of posting transactions. Accounting entries have the unusual property that after a certain point in time they cannot be modified. This is the point when they are “posted.”

In paper-based systems each transaction is entered twice. The first is in what is called a journal, of which there are several. Each journal records entries related to different parts of the business. The same information is later copied to a centralized list of account balances. The central list, which integrates all the information spread over the journals, is called the general ledger.

Posting is the process of updating the general ledger to reflect the journal entries. Once posted, the journal entries cannot be modified. Posting is central to maintaining accuracy in reporting.

There are numerous ways to handle posting. In the simplest system every transaction is posted as it is entered. That is, account balances are updated as soon as transactions are entered and the transactions, once entered, cannot be modified. When handled in this way, the accounts file, shown in figure 2, plays the role of the general ledger.

What can an accounting system do?

People have realized that computers have too much potential to be limited to small, segregated applications. There is a virtual rush to develop more comprehensive systems. These same forces are pushing accounting systems from “back-office” applications to complete management systems, what I’ll call enterprise systems.

For most businesses, enterprise systems cannot be achieved with off-the-shelf software. Such systems need to access all relevant business information. This means that whatever else the system does, it must support accounting. Figure 3 shows some of the many areas that depend on accounting information. The next generation of expanded business systems will have accounting systems at its heart.

Areas Affected by Accounting		
<i>Goods and Services</i>	<i>Presales</i>	<i>Management</i>
Customer Support	R & D	Executive Planning
Services	Marketing	Bookkeeping
Payroll	Advertising	Office Management
Personnel	Bids & Quotes	Tax planning
Inventory		Debt obligations
Sales		Investments
Purchasing		Legal planning

Figure 3: Accounting systems tie together all aspects of a business.

What's in it for you?

In many cases opportunities for developing accounting systems depend on clients adding features to existing systems. Adding partial accounting services, such as billing or job costing, to an existing database is a sensible short term approach but will lead to problems if the goal is a complete accounting system. Comprehensive systems can not be designed piece-meal.

A better approach is to start new systems on an accounting foundation; this is the bottom-up approach. But accounting applications are large — easily spanning 8 megabytes or more — so building on an accounting system is only really possible if you license the source code to an existing system. Such source code packages are supplied by various companies, mine included, and each has a slightly different focus. Starting with the source code for an already complete accounting system guarantees you a path through the accounting jungle.

Enterprise systems require more analysis, design, programming, and support. They also cost a lot more. On the other hand, they can offer revolutionary levels of control and efficiency. James McConathay, president of MicroNet Technology, a leading hard disk manufacturer, told me he didn't think MicroNet could even operate without its accounting system.

The move to enterprise systems perfectly suits the skills and tools of 4D developers. While building enterprise systems is not for everyone, as a 4D developer you are in a great position to catch a wave that offers the highest rewards.