

Reducing Complexity in an Accounting System by Using the REA Data Model to Design a Relational Database

Ian Adamson*

Department of Accounting, Brock University, St. Catharine's, Ontario, Canada

Relational databases are used extensively in today's business environment, from simple small enterprises using Microsoft Access to large corporations using Oracle or Microsoft SQL Server. A database must be efficiently designed in order to avoid errors and data redundancies. Poorly designed databases can be very complex, difficult to maintain and prone to anomalies. In order to avoid this situation, an effective design methodology should be used.

Designing a relational database involves developing a detailed data model that can then be used to create the database. Historically software engineers have used the Entity Relationship Model (ERM), a graphical tool that represents the logical relationships between entities. Entities are typically shown as rectangles with the connections between the entities portraying the relationships. In a logical sense, entities are the equivalent of grammatical nouns, such as employees, inventory, payables, or fixed assets, with the characteristics of the entity defined by means of its properties called attributes. Relationships are the equivalent of verbs or associations, such as the act of purchasing or selling, or an employee being a member of a department.

On implementation of the database, the entities become relations or tables while the relationships reveal the logical connection between the tables. Such connections can be one-to-one, one-to-many, or many-to-many. For example the figure below show a one-to-many relationship between Customer and Sale entities (Figure 1).

This shows that a Customer entity can have many sales but a sale belongs only to one customer. The diamond shape shows the relationship has been identified as R which could also be the verb 'have'. For a number of relationships in an ERM they can be shown as R_1 to R_n . On implementation in a relational database both Customer and Sale become tables.

How are the relationships established? Both Customer and Sale have to have a unique identifier in order to identify each instance of the entity class. Such identifiers are called primary keys and usually are numeric. The primary key for Customer would be Customer Number and for Sale, the Invoice Number. The one-to-many relational connection would be implemented with the Sale entity having Customer Number as the foreign key. This example can be shown by the following (Figure 2). It can be seen that Customer one has two invoices while customer two has only one invoice. Primary Keys are shown as PK while the Foreign Key is shown as FK.

In order to construct an ERM of a system the entities or nouns are first identified and then the relationship constructed. Consider simple sales and corresponding cash receipts transactions. Possible entities, but not all, are shown in the following (Table 1).

Possible relationships are shown in the following ERM as identified by R_1 to R_{18} . Each of the relationships would have to be implemented

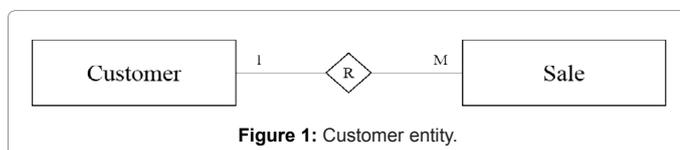


Figure 1: Customer entity.

by primary keys and foreign keys. It can also be seen that each of the clerical personnel are Employees, requiring only on entity class of Employee of which each member is an instance (Figure 3).

All together there are ten entity classes and eighteen relationships for this simple scenario. In a full accounting database system the number of tables and relationships would be very large resulting in a degree of complexity that would be significant. This complexity could result in multiple data redundancies and duplication of tables. By looking at the Resource Event Agent Data Model it is possible to significantly reduce such complexity.

The Resources, Events, Agents (REA) Model

Hruby shows that REA is a business process modeling methodology that is closer to business reality than any other known alternative. An increasing number of business analysts have found that the models they develop become better when they have REA in mind. REA systems are usually modeled as relational databases, though this is not necessarily a requirement, typically using entity-relationship diagrams.

The REA model derives its name from the three distinct categories of entities resources, events, and agents [1,2].

- Resources are acquired and used by the company and have economic value to the organization.
- Events are typically business activities in which the company

Customer	
Employee	Sales Clerk
Sales Order	
Sales Invoice	
Inventory	
Accounts Receivable	
Employee Clerk	Accounts Receivable
Cash	
Cash Receipts	
Employee	Cash Receipts Clerk
Cash Receipts Listing	
General Ledger	
Employee	Accounting Clerk

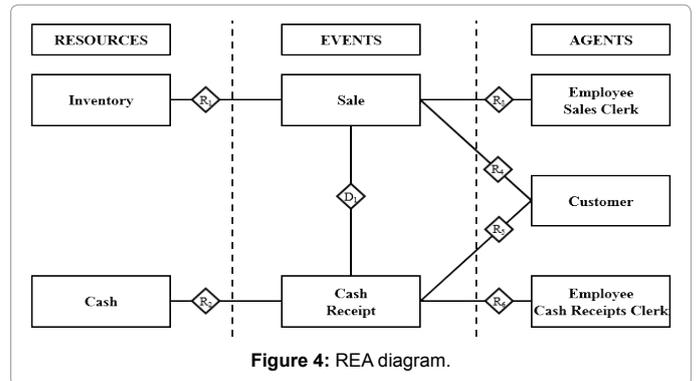
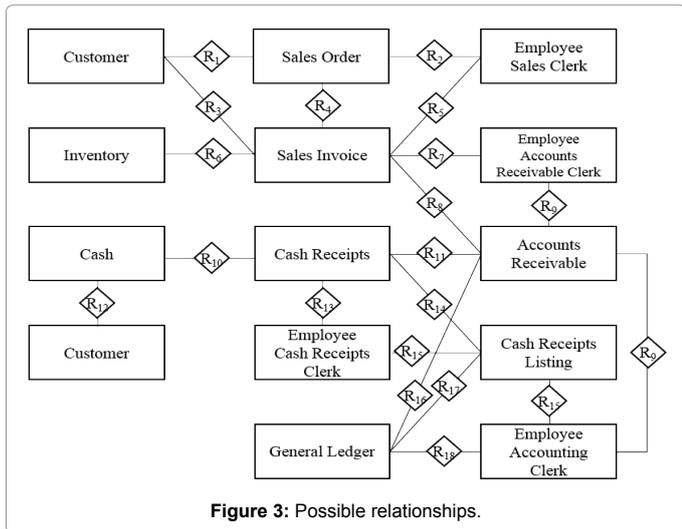
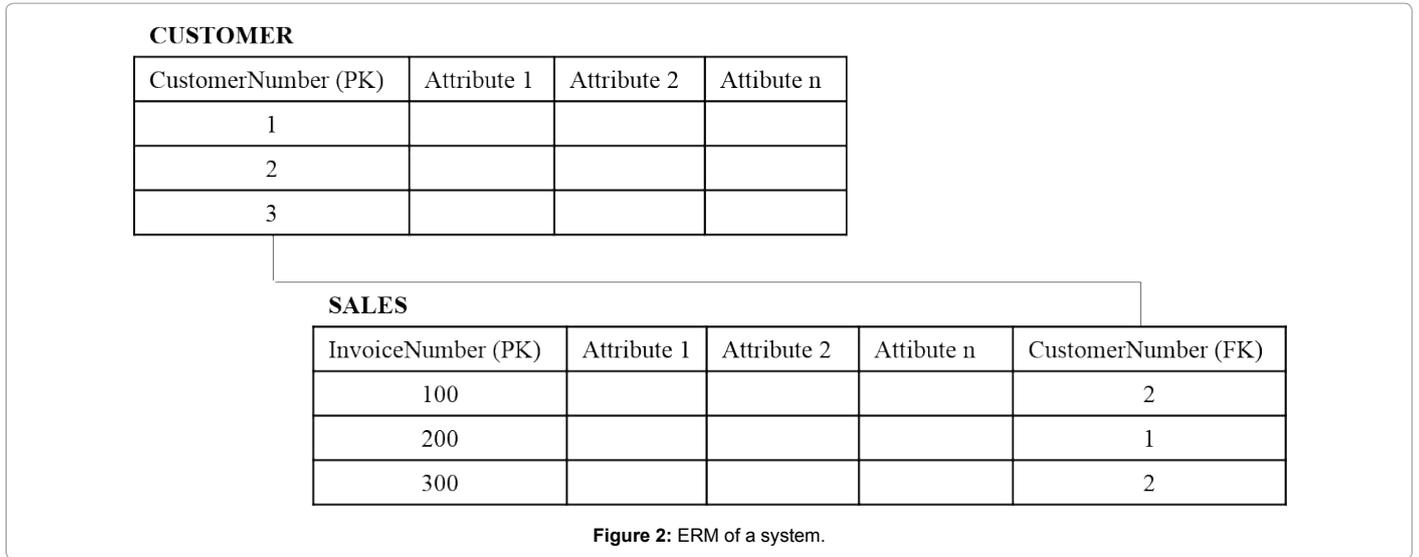
Table 1: Possible entities.

*Corresponding author: Ian Adamson, Associate Professor, Department of Accounting, Brock University, St. Catharine's, Ontario, Canada, Tel: (905) 688-5550; Fax: (905) 688-9779; E-mail: iadamson@brocku.ca

Received May 12, 2017; Accepted May 19, 2017; Published May 26, 2017

Citation: Adamson I (2017) Reducing Complexity in an Accounting System by Using the REA Data Model to Design a Relational Database. J Account Mark 6: 237. doi: 10.4172/2168-9601.1000237

Copyright: © 2017 Adamson I. This is an open-access article distributed under the terms of the Creative Commons Attribution License, which permits unrestricted use, distribution, and reproduction in any medium, provided the original author and source are credited.



Resource	Inventory
Resource	Cash
Event	Sales
Event	Cash Receipt
Agent	Customer
Agent	Sales Clerk
Agent	Cash Receipts Clerk

Table 2: The Sales Transaction using the REA Data Model.

engages and about which management wishes to gather data for planning or control purposes; and

- Agents include people and organizations who participate in events. Management also needs gather data on these entities for planning, control, and evaluation purposes

As a rule, central to an REA model is as a pair of events associated with an exchange relationship. The structure of this relationship is based upon the duality principle. The REA duality principle: An event generating an outflow of resources must be paired in duality relationship with an event generating an inflow of resources, and vice-versa. These corresponding events are also called give and take events. Therefore an exchange transaction is basically a pair of dual events that links the act of giving, the decrement event, with the act of taking, the increment event. If there is no duality then the transaction is meaningless, with the exception of a transaction such as a charitable donation where there would be an inflow of a resource such as cash, but no outflow, thus no duality.

One of these events usually corresponds to a resource being given away, while the other corresponds to a resource being received. For example, an event resulting in the receipt or inflow of inventory must

be related to an event of equal monetary value loss or outflow. In the terms of a sales transaction, one event would be the sale transaction where goods are given up and the other would be cash receipt, where cash is received or gained.

The Sales Transaction using the REA Data Model

In sales/cash receipts transactions the resources are typically the inventory item that is being sold and the cash received from the customer. There are two events, the sale and the cash receipt, that have a duality relationship. The agents in this scenario are the Sales Clerk, the Customer, and the Accounting Clerk (Table 2). The REA diagram can be seen in the following (Figure 4).

Notice that there are no journals or ledgers. Such entities are an artifact of the traditional accounting system and cannot be classified as resources, events, or agents. The REA Model eliminates accounting structures that are not necessary in the electronic era. These entities disappear since they can be generated in real time using record detail in

an REA system. Consider an example of producing an Aged Accounts Receivable Listing using the following (Figure 5).

From this data it is conceivable to have an application program, written in a language such as PLSQL, C++ or even COBOL, to construct the listing. Simple partial pseudo code for a program is shown below (Figure 6).

Any reports or listings can be generated in a similar manner whenever they are needed, not just at month or year end.

REA Model Benefits

The REA Model uses an events-based approach looking at all the possible events that can occur. These events can then be processed to give the user their required output, for example, an Accounts Receivable Aged Trial Balance, which has not been pre-defined. The report is only produced when the user accesses the specific events defined for that report [3].

An Entity-Relationship diagramming approach, a more traditional methodology, is a view-based approach requiring that all potential entities are included in the relational model, which would include the Accounts Receivable Aged Trial Balance and other similar reports. This

Traditional Model	REA Model
10 Entities	6 Entities
18 Relationships	7 Relationships

Table 3: An Entity-Relationship diagramming approach.

can result in an incomprehensible diagram for complex real-world relational database applications.

In examining the REA Model of Sales/Cash Receipts it is evident that there fewer entities and relationships (Table 3).

The result would be reduced data redundancy, a condition created within a database where the same data item is held in two separate tables. The traditional model has four tables, Sales Order, Accounts Receivable, Cash Receipts Listing, and the General Ledger which contain data that are already present in other tables such as sales invoice and Cash Receipts. This is an increase in data redundancy. Any changes to Sales Invoice and Cash Receipts requires corresponding changes to theses four tables. This can result in insert, update, and deletion anomalies and data inconsistencies within the database, a situation prone to error.

In the traditional model adjustments usually have to made to sub-ledgers in order to balance to the general ledger, or vice-versa. In the REA Data Model this is not required. All transactions are entered events and the sub-journals and general ledger are created from these events. Thus the sub-ledgers will balance to the general ledger and not adjusting entries for balancing would be required.

Thus the REA Model results in a more simplified relational data base which significantly reduces the chance of data duplication, data redundancy, and as a result, a data base that requires less maintenance and is less prone to error.

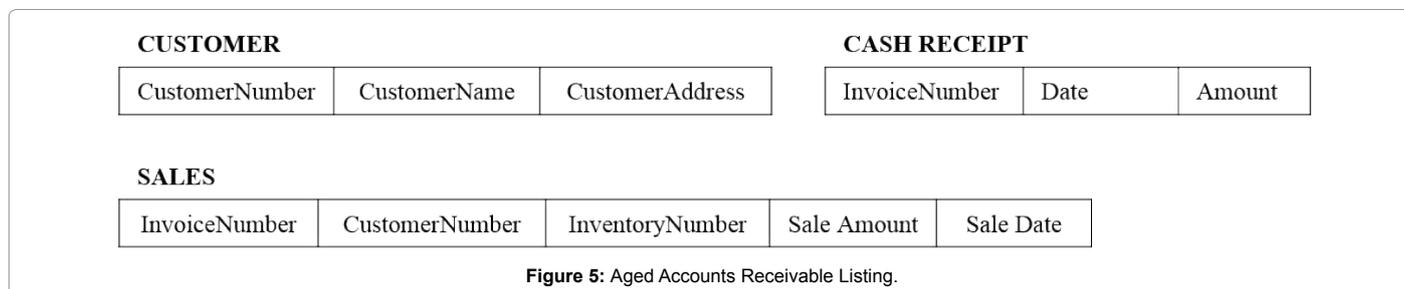


Figure 5: Aged Accounts Receivable Listing.

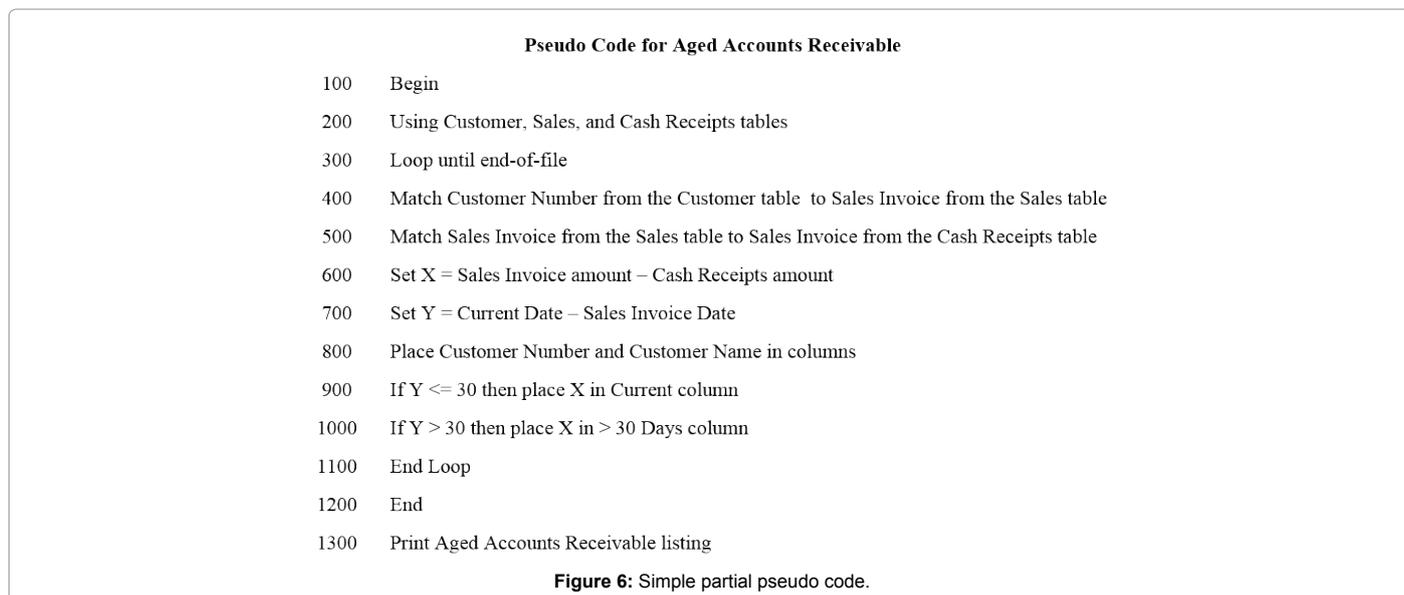


Figure 6: Simple partial pseudo code.

References

1. Hruby P (2006) Model-Driven Design Using Business Patterns. Springer.
2. McCarthy WE (1982) The REA Accounting Model: A Generalized Framework for Accounting Systems in a Shared Data Environment. The Accounting Review.
3. Peter C (1976) The Entity-Relationship Model--Toward a Unified View of Data. ACM Transactions on Database Systems.

Citation: Adamson I (2017) Reducing Complexity in an Accounting System by Using the REA Data Model to Design a Relational Database. J Account Mark 6: 237. doi: [10.4172/2168-9601.1000237](https://doi.org/10.4172/2168-9601.1000237)

OMICS International: Open Access Publication Benefits & Features

Unique features:

- Increased global visibility of articles through worldwide distribution and indexing
- Showcasing recent research output in a timely and updated manner
- Special issues on the current trends of scientific research

Special features:

- 700+ Open Access Journals
- 50,000+ editorial team
- Rapid review process
- Quality and quick editorial, review and publication processing
- Indexing at major indexing services
- Sharing Option: Social Networking Enabled
- Authors, Reviewers and Editors rewarded with online Scientific Credits
- Better discount for your subsequent articles

Submit your manuscript at: <http://www.omicsonline.org/submission>