

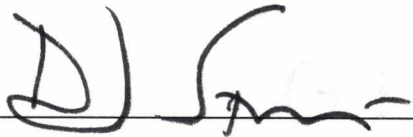
To the Graduate Council:

I am submitting herewith a thesis written by Paul L. Bowen entitled "The Effects of Different Levels of Normalization on Data Entry and Data Entry Controls: An Experimental Evaluation." I have examined the final copy of this thesis for form and content and recommend that it be accepted in partial fulfillment of the requirements for the degree of Master of Science, with a major in Computer Science.



Dr. Michael D. Vose, Major Professor

We have read this thesis
and recommend its acceptance:



Accepted for the Council:



Associate Vice Chancellor and
Dean of The Graduate School

**THE EFFECTS OF DIFFERENT LEVELS OF NORMALIZATION
ON DATA ENTRY AND DATA ENTRY CONTROLS:
AN EXPERIMENTAL EVALUATION**

A Thesis

Presented for the

Master of Science

Degree

The University of Tennessee, Knoxville

Paul L. Bowen

December 1995

Copyright © Paul L. Bowen

All rights reserved

DEDICATION

In honor of
Dorothy F. Bowen and
Doris L. Driver

ACKNOWLEDGEMENTS

I wish to express appreciation to the members of my thesis committee. I am very grateful to Dr. Michael Vose, my thesis chair, for his many contributions to my academic career. I extend my thanks to Dr. David Straight for his assistance with numerous aspects of my computer science program and to Dr. Frank Guess for his encouragement of this and other research projects.

I am indebted to my colleagues and friends Professor Faye Borthick, Jon Heales, Fiona Rohde, and Professor Ron Weber for their comments on various iterations of the thesis. My gratitude goes to the many people at the Oak Ridge National Laboratory who stirred my interest in data quality, especially Juan Lovin and Randy Haese.

I wish to convey my appreciation to some of the other people who have helped make this project possible. I am particularly thankful to Dorothy Bowen for helping me with many of the administrative tasks required to complete this thesis. My appreciation goes to Doris Driver for her encouragement. Most of all, I thank God for the many blessings He has heaped upon me.

ABSTRACT

End users are using relational database software resident on their personal workstations to create their own tables and update an increasing volume of data. The value of these databases depends on the accuracy of the data they contain. Unfortunately, end users may not have the training or expertise to normalize the data structures or implement input controls so that data accuracy is maximized. This thesis reports the results of an experiment that examined the effects of normalization level and input controls on data entry errors. The experiment also examined the effects of information systems background on the ability of end users to specify data structures and input controls.

To test the five hypotheses proposed in the thesis, eighty participants were randomly assigned to one of three data structures: not normalized, first normal form, and third normal form. The first hypothesis proposed an inverse relationship between normalization level and the proportion of errors users enter into a database. Hypothesis two anticipated a direct relationship between normalization level and data entry speed. The third hypothesis stated that data validation procedures reduce the proportion of errors users enter into a database. Hypothesis four proposed that users with more extensive information systems backgrounds would identify more input controls. The fifth hypothesis stated that users would request more data structure changes for less normalized databases. The results of the experiment provided support for all five hypotheses.

Although normalization improves data accuracy, well designed input controls appear to provide as great if not greater improvements to data accuracy. Even the combination of high levels of normalization and stringent input controls, however, would not prevent all data entry errors. One implication of the results is that assistance and training in creating normalized data structures and designing input controls should be provided to end users who are creating and maintaining their own tables. Another implication is that additional quality control procedures are needed to ensure data accuracy.

TABLE OF CONTENTS

SECTION	PAGE
1. Introduction	1
2. Relationship of Normalization to Accuracy, Complexity, and Input Controls	4
2.1. <i>Data Accuracy</i>	4
2.2. <i>Normalization</i>	5
2.3. <i>Query Complexity</i>	7
2.4. <i>Input Controls</i>	8
3. Example	10
3.1. <i>¬NF vs 1NF</i>	10
3.2. <i>1NF vs 3NF</i>	13
3.3. <i>Inserting Data</i>	16
3.4. <i>Input Controls</i>	19
4. Hypotheses	22
4.1. <i>Accuracy</i>	22
4.2. <i>Input Controls</i>	23
5. Methodology	26
5.1. <i>Description of Experiment</i>	26
5.2. <i>Data Coding</i>	28
6. Results	29
6.1. <i>Summary Data</i>	29
6.2. <i>Data Entry Accuracy and Efficiency</i>	31
6.3. <i>Validation Controls and Data Accuracy</i>	33
6.4. <i>Data Entry Controls and Data Structure Recommendations</i>	33
7. Discussion	37
8. Contributions, Limitations, and Future Research	39
REFERENCES	41

APPENDICES	47
APPENDIX A <i>Data Structures</i>	48
A.1 Not Normalized ($\neg$ NF)	48
A.2 First Normal Form (1NF)	49
A.3 Third Normal Form (3NF)	50
APPENDIX B <i>Demographic Information</i>	51
APPENDIX C <i>Instructions to Participants</i>	53
Instructions - A	53
Instructions - B	56
Instructions - C	59
APPENDIX D <i>SQL to Create Tables and Insert Initial Purchase</i> <i>Order Data</i>	62
APPENDIX E <i>Potential Errors and Possible Controls</i>	69
APPENDIX F <i>Figures</i>	73
VITA	80

LIST OF TABLES

TABLE	PAGE
1. <i>Summary of Experimental Results</i>	30
2. <i>Effect of Normalization Level on Attribute Error Percentage</i>	32
3. <i>Effect of Normalization Level, Data Entry Time, and IS Background on Attributes Entered per Minute</i>	34
4. <i>Effect of IS Background on Recommended Data Entry Controls</i>	36
5. <i>Effect of Normalization Level and IS Background on Recommended Data Structure Changes</i>	36

LIST OF FIGURES

FIGURE	PAGE
1. <i>Example Purchase Order</i>	73
2. <i>Example Purchase Order Data - Existing</i>	74
3. <i>Example Purchase Order Data - For Input By Participants</i>	78

1. Introduction

As client-server applications based on relational database management system software become more pervasive, end users are creating more tables and updating an increasing volume of data.¹ They, and others inside and outside their organization, use these systems to make important decisions. The value of these systems depends on the accuracy of the data and the ease with which it can be extracted. To ensure data accuracy and to make data readily accessible, information system (IS) professionals structure data into normalized relations, create data entry and retrieval screens, and install extensive validation controls. Although these activities are assumed to promote data accuracy and accessibility, the contributions of each of these activities to data accuracy goals has not been determined empirically. Empirical evidence is needed because end users may not perform these activities as well as IS professionals. Empirical evidence would help determine the importance of providing training and assistance and the relative emphasis that should be given to each of the activities.

Because normalization reduces the likelihood of update anomalies, the data in normalized relations are thought to be more accurate than the data in unnormalized relations (Kent 1983). Increasing levels of normalization also reduce dependencies of various types. Reducing dependencies promotes data integrity (e.g., Date 1990, 525&559) by making it easier to write correct commands for

¹ For example, project engineers or management accountants might use a PC based database management system such as Access or Paradox to track budgeted and actual costs for activities, job orders, contracts, or work groups.

updating the relations, increasing the efficiency and effectiveness of data entry controls, and reducing data redundancies. These benefits are some of the reasons database researchers advocate normalizing databases to third normal form or beyond (Date 1990, 532). Normalizing relations, however, increases the number of tables in a database (Date 1990, 552-553), which means that queries may become more complex because additional joins may be needed. The tradeoff is between reducing update anomalies through higher levels of normalization but potentially increasing query complexity through the need for more join operations. When database designers choose to use lower levels of normalization to reduce query complexity, input controls can mitigate the anomalies typically associated with these less normalized relations.

Although many researchers have analytically and normatively investigated normalization, little empirical research has been conducted on the actual effects of various levels of normalization on either data accuracy or input controls. This paper reports the results of an experiment that examines data entry experience with three sets of data structures: not normalized ($\neg$ NF), first normal form (1NF), and third normal form (3NF). This study examined the effects of different levels of normalization on data entry accuracy and speed, and the effects of validation controls on data accuracy. It also examined the relationship of information systems training and experience to the ability to specify appropriate data entry controls. The results indicate that higher normal forms reduce error rates and increase data entry speed, that validation controls significantly improve data accuracy, and that

people with more extensive IS backgrounds recommend more input controls and data structure changes. The results also show that people using less normalized data are more likely than people using more highly normalized data to recommend data structure changes.

The next section of the paper reviews research on data accuracy, normalization, query complexity, and input controls. The third section provides an example of different levels of normalization for a purchase order database and shows how lower levels of normalization can lead to data errors. The hypotheses investigated in this paper are developed in the fourth section. The fifth section describes the methodology that was used to collect, measure, and analyze the data. The sixth and seventh sections present and discuss the results. The eighth section identifies some of the contributions and limitations of the study.

2. Relationship of Normalization to Accuracy, Complexity, and Input Controls

2.1. *Data Accuracy*

Information economics research has demonstrated that accuracy is a major determinant of the value of an information system (Feltham 1972; Itami 1977; Hilton 1979, 1981; Hilton et al. 1981). Unfortunately, empirical evidence suggests that organizations experience data accuracy problems (Neter and Loebbecke 1975; Kinney 1979; Ramage et al. 1979; Johnson et al. 1981; Hylas and Ashton 1982; Morey 1982; Ham et al. 1985; Willingham and Wright 1985; Kreutzfeldt and Wallace 1986; Laudon 1986; Icerman and Hillison 1991). Therefore, decision makers using information systems with lower data accuracy will be expected to make more decision errors than if they use systems with higher accuracy. For example, errors in a database may increase the probability that military officers could make incorrect decisions (Dillard 1992).

The concern for data quality is reflected in the attempts to model, improve, and define it. Several researchers have developed quantitative models to assist auditors, controllers, and information system developers evaluate, improve, and manage data accuracy (Yu and Neter 1973; Cushing 1974; Hamlen 1980; Ballou and Pazer 1985a, 1985b; Morey 1982; Paradice and Fuerst 1991; Bowen 1992). These models provide valuable assistance for identifying, evaluating, monitoring, and managing input controls. Other researchers have advocated using rule-based integrity constraints (Parsaye and Chignell 1993), integrity analysis (Svanks 1988), total data quality management (Wang and Kon 1993), statistical process control

(Huh et al. 1990; Redman 1994), and table level control procedures (Bowen, Schneider, and Fields 1995) to improve the quality of incoming and accumulated data. Some researchers have begun to define, model, and measure the quality of data, including accuracy, accumulated in information systems (Dillard, 1992; Wang et al. 1992; Wang and Reddy 1993; Wang et al. 1993a, 1993b; Wand and Wang 1995; Fox, Levitin, and Redman 1994). See Wang, Storey, and Firth (1995) for an extensive survey of data quality research.

2.2. Normalization

Codd (1970, 381) introduced normalization in his seminal paper on relational databases and described it as a "simple elimination procedure." A number of authors (e.g., Kent 1983; Date 1990) provide definitions of the various levels of normalization and examples of how various levels of normalization can be achieved.² Date (1990, 533) stated that a relation (i.e., a table or file) is in 1NF if all underlying simple domains³ contain atomic values only. Kent (1983) interpreted 1NF as requiring that all occurrences of a record must contain the same number of fields. He viewed 1NF as excluding repeating fields and groups. Date (1990, 536) declared that a relation is in 2NF if, in addition to satisfying 1NF,

² Because the experiment only used 1NF, 2NF, and 3NF, the definitions of Boyce-Codd normal form (BCNF), fourth normal form (4NF), and fifth normal form (5NF) are not presented in this paper.

³ Domains are sets of values, all of the same type.

every nonkey attribute is fully dependent on the primary key.⁴ Kent (1983) explained that 2NF is violated when a nonkey field is a fact about a subset of the primary key, i.e., violations of 2NF are relevant only when the primary key consists of multiple attributes. Date (1990, 539) specified that a relation was in 3NF if, in addition to satisfying 2NF, every nonkey attribute is nontransitively dependent on the primary key. Transitive dependencies occur when an attribute's value depends on the value of an attribute other than the primary key. Kent (1983) expressed the requirements of 3NF as requiring that every attribute be either part of the primary key or provide a single-valued fact about the entire primary key.

A common theme in achieving higher levels of normalization is the decomposition of a table into two or more smaller tables (where smaller is defined as fewer attributes). Date (1986, 109) asserts that each level of normalization is generally more desirable than the levels below it. In particular, decompositions into higher levels of normal forms are desirable because data can be repeated in sets of relations that only satisfy lower normal forms (Fagin 1977, 264). That is, the reduction of insert, update, and delete anomalies is one of the most important purposes of normalization (Codd 1974; Kent 1983). Even in his seminal paper, Codd (1970, 387) recognized the problem of maintaining data in a consistent state and anticipated that this problem would become more serious as more data were integrated into common data banks. Hence, the connection between normalization

⁴ The primary key for a table is the attribute or set of attributes that uniquely define each record.

and data accuracy was recognized when the relational model was first articulated. It has remained a concern of researchers throughout the continuing development of the relational model.

Tables that are not normalized may contain redundant data that increase the likelihood of inconsistencies and non-atomic data that impair the ability to impose more stringent input controls. Tables that only satisfy first normal form may contain redundancies that can lead to update problems (Date 1990, p 534). Tables that satisfy third but not fourth or fifth normal form can lead to different and more subtle update problems (Date 1990, 549). Increasing the level of normalization reduces update anomalies and data inconsistencies (Kent 1983) thereby increasing data accuracy.

2.3. Query Complexity

Tables that are not normalized, i.e., that do not satisfy the atomicity⁵ constraint or that contain repeating fields, can make queries more complex and the data more difficult to use. Normalizing the tables to at least first normal form (1NF) eliminates problems caused by lack of atomicity and by repeating fields. Increasing the level of normalization, however, increases the number of tables (e.g., Date 1990, 535&537, Kent 1983). Although the data structure of individual tables in lower normal forms is more complex than data in individual tables that satisfy

⁵ The atomicity constraint requires that each field (attribute or column) contain only one piece of data. E.g., a single address field that contained both city and state data violates the atomicity constraint.

higher normal forms (Date 1990, 535), the increased number of tables that result from higher normal forms makes many user queries more complex.

Codd (1974) stated that databases are established for the benefit of end users, not for application programmers. Higher levels of normalization could be undesirable if the fragmentation associated with these higher levels increases query complexity and reduces users' ability to obtain the information they need.⁶ Hence, there is potentially a conflict between the benefits of data accuracy and the costs of query complexity that may result from higher levels of normalization.

2.4. Input Controls

Data input controls are used to identify errors before the data are used to update the relations (Weber 1988, 414). These controls include field checks and record checks (Weber 1988, 416). Field checks include tests for format, data type, range, set membership, and master file reference. Record checks test the interrelationships between fields, e.g., a reasonableness check that could verify that employees' salaries are within the ranges prescribed for their job classifications. Regardless of normalization issues, many field and record checks should be present to help ensure data integrity. Application programs for highly normalized relations

⁶ Higher levels of normalization may also have other negative effects. For example, higher levels of normalization may be detrimental to database performance due to I/O and other considerations (e.g., Tsai 1988, 451; Shasta 1992, 92). The relational model itself is criticized by some database researchers. For example, Chen (1976, 9) observes that the relational model can achieve databases with high degrees of data independence, but may lose important semantic information about the real world that the databases are modelling. These issues are beyond the scope of this paper.

will need fewer field and record checks, however, because higher levels of normalization automatically include the controls that must be programmed explicitly into applications that update unnormalized or less normalized relations.

3. Example

This section discusses and illustrates the impact of unnormalized form ($\neg$ NF), first normal form (1NF), and third normal form (3NF) on the structure of the data, query complexity, data quality, and input controls. Although the effects of query complexity are not measured by the experiment described in this paper, the increased complexity of queries using highly normalized tables helps explain why persons who create tables, especially end users, may not design data structures that conform to high levels of normalization. Purchase order data will be used to illustrate some of the impacts of these three different levels of normalization. Figure 1⁷ presents an example purchase order. Appendix A contains possible data structures for each of the three normalization levels.

3.1. $\neg$ NF vs 1NF

The difference between $\neg$ NF and 1NF data structures is that $\neg$ NF data structures contain repeating fields or exhibit lack of atomicity. Converting $\neg$ NF data structures to 1NF removes repeating fields and ensures that all fields contain only atomic values. Relative to 1NF, $\neg$ NF data structures can often increase query complexity.

The presence of either repeating fields or lack of atomicity indicates that a data structure fails to satisfy 1NF, i.e., is not normalized. Repeating fields exist

⁷ All figures may be found in Appendix F.

when more than one attribute contains the same type of data. An attribute violates atomicity if it contains more than one element of data. Appendix A.1 contains a possible data structure for purchase order data that will be used to illustrate these problems. Repeating fields in Appendix A.1 are those groups of attributes that are associated with individual items, i.e., $ITEMNO_i$, $DESC_i$, $UPRICE_i$, QTY_i , $UMEAS_i$, $i = 1, 2$. The attribute $VCSZ$ violates atomicity because it contains three (city, state, and zip code) rather than one unit of information. The attribute $TERMS$ violates atomicity because it contains two units of information: discount percent and number of days to which the discount applies.

Eliminating repeating fields and lack of atomicity problems will produce data structures that satisfy first normal form. Repeating fields can be eliminated by creating separate tables for each group of repeating data. The newly created tables must also contain the necessary attributes to link, i.e., join, the data back to the associated data remaining in the original table.⁸ In this example, repeating fields are eliminated by creating the *POITMI* table (see Appendix A.2) that contains the repeating attributes from the *PO* table in Appendix A.1 plus the *PONO* attribute. The presence of *PONO* in both the *POI* and *POITMI* tables in Appendix A.2 permits the data in these two tables to be joined when required to satisfy user queries. The lack of atomicity problem is eliminated by creating a separate attribute for each unit of information, i.e., $VCSZ$ in Appendix A.1 becomes $VCITY$,

⁸ The attributes used to join the newly created tables back to the original table are called foreign keys.

VSTATE, and VZIP in Appendix A.2 and TERMS in Appendix A.1 becomes TERMDISC and TERMDAYS in Appendix A.2.

Repeating fields increase query complexity because all instances of the repeating data elements must be specified in the query, i.e., any of the repeating fields could contain the data of interest. For example, suppose a user wants to find all orders that include item number 31542. The query⁹ would be:

```
select *  
from PO  
where ITEMNO1 = 31542  
      or ITEMNO2 = 31542;
```

If item number data in the *PO* table were in more than two repeating fields, the query of the $\rightarrow$ NF data structure would become correspondingly more difficult and complex. If, however, the data conformed to 1NF as shown in Appendix A.2, the query would be:

```
select *  
from POITM1  
where ITEMNO = 31542;
```

Lack of atomicity can also cause queries of unnormalized data to become more complex. To select records using a non-atomic attribute requires that users make assumptions about the internal format of the attribute and that these assumptions are valid for all records in the table. For example, the assumptions necessary for a query of the *PO* table that displays all purchase orders with

⁹ In the queries, attribute names will appear in upper case letters; table names in upper case, italicized letters; and SQL reserved words will appear in lower case, italicized letters.

discounts greater than 3% must make assumptions about the TERMS attribute, e.g., that the discount percent precedes the slash, that the text that precedes the slash can be converted to a numeric value, and whether the value is expressed as a percentage (e.g., 3) or as a decimal value (e.g., 0.03). Furthermore, the relational database management system, RDBMS, must contain substring functions, and the person writing the query must understand how to use these functions. To display purchase orders with greater than 3% discounts from the $\rightarrow$ NF data structure in Appendix A.1, the query would be:

```
select *  
from PO  
where to_number(substr(TERM,1,instr(TERM,'/')-1)) > 0.03;
```

If the data conformed to 1NF as shown in Appendix A.2, the query would be:

```
select *  
from PO1  
where TERMDISC > 0.03;
```

3.2. 1NF vs 3NF

The difference between 1NF and 3NF data structures is that 1NF data structures may contain partial dependencies, transient dependencies, or both. Partial dependencies occur when the primary key consists of more than one attribute and the value of a non-primary key attribute is uniquely determined by the values of a proper subset of the attributes in the primary key. Transient dependencies occur when the value of a non-primary key attribute is uniquely determined by the value of another non-primary key attribute. Converting 1NF data structures to 3NF

removes both partial and transient dependencies. Data structures that satisfy 3NF, however, can *increase* query complexity relative to data structures that only satisfy 1NF.

Eliminating partial dependencies (violations of 2NF) and transient dependencies (violations of 3NF) will produce data structures that satisfy 3NF. Because the primary key for the *POI* table in Appendix A.2 consists of a single attribute, PONO, the *POI* table does not violate 2NF. Because DESC and UMEAS are dependent on ITEMNO alone rather than the entire primary key,¹⁰ the *POITM1* table in Appendix A.2 does violate 2NF. This partial dependency problem can be eliminated by moving the attributes DESC and UMEAS to a separate table, i.e., to *ITMDESC3* in Appendix A.3, along with the attribute required to join the data to the remaining *POITM3* data, i.e., ITEMNO.¹¹ Because each nonkey attribute in *POITM3* and *ITMDESC3* depend solely on the primary keys of these tables, these tables also satisfy 3NF.

In the *POI* table in Appendix A.2, the vendor attributes (VNAME, VSTREET, VSTATE, and VZIP) depend on VNO, i.e., these attributes are transiently dependent on PONO via VNO. Because zip codes¹² uniquely determine cities and states, VCITY and VSTATE are transiently dependent on VNO

¹⁰ The primary key in *POITM1* is the concatenation of PONO and ITEMNO.

¹¹ This distribution of attributes between *POITM3* and *ITMDESC3* assumes that the value of UPRICE can change, i.e., that UPRICE depends on both PONO and ITEMNO. This assumption is equivalent to saying that UPRICE can change over time.

¹² Post codes in Australia are equivalent to zip codes in the United States.

via VZIP.¹³ These transient dependencies can be eliminated by creating two new tables: *VENDOR3* and *ZIP3*. The *VENDOR3* table contains the attributes *VNAME*, *VSTREET*, and *VZIP* along with *VNO* to join *VENDOR3* to *PO3*. The *ZIP3* table contains the attributes *VCITY* and *VSTATE* along with *VZIP* to join *ZIP3* with *VENDOR3*. The resulting tables in Appendix A.3 do not contain repeating fields or non-atomic attributes (i.e., do not violate 1NF), partial dependencies (i.e., do not violate 2NF), nor transient dependencies (i.e., do not violate 3NF), i.e., the tables are in 3NF.

An advantage of 1NF data structures relative to 3NF data structures is that they reduce the number of joins required to collect the data needed to satisfy many user queries. Reducing the expected number of joins has both human and machine benefits, i.e., it reduces complexity for users as well as memory and processing time requirements. To illustrate, using the 1NF data structure in Appendix A.2, the command:

```
select *  
from PO1, POITM1  
where PO1.PONO = 'KA-21979'  
and PO1.PONO = POITM1.PONO;
```

would display all the purchase order information shown in Figure 1 with the exception of the organization's own name and address. To display the same data using the 3NF data structure in Appendix A.3, the command would be:

¹³ Because different cities can have streets with the same name, *VSTREET* does not depend on *VCITY* and, hence, does not depend on *VZIP*.

```

select *
from PO3, POITM3, ITMDESC3, VENDOR3, ZIP3
where PO3.PONO = 'KA-21979'
      and PO3.PONO = POITM3.PONO
      and POITM3.ITEMNO = ITMDESC3.ITEMNO
      and PO3.VNO = VENDOR3.VNO
      and VENDOR3.VZIP = ZIP3.VZIP;

```

For this illustration, the query for the 1NF data structure contains fewer terms and conditions, i.e., is less complex than the query for the 3NF data structure. In addition, for queries involving data from more than one table, the user must remember the semantics of the relationships between the tables. The complexity and difficulty of learning, remembering, and exploiting these relationships increases as the number of tables increase, i.e., as the level of normalization increases.

3.3. *Inserting Data*

Most end users enter data via data entry screens created by IS personnel or created by themselves using a screen generation facility in the database management system. Although end users would seldom use SQL insert statements, examining these statements is important for several reasons. First, the examination reveals some of the inherent differences in insert statements for different levels of normalization. These differences are important if batch programs must be written to update the data, e.g., transfers of corporate or departmental data from a mainframe or mini-computer to an end user's PC. Second, the examination illustrates the differences in the amount of data that must be input and stored for data structures in each of the normalization levels. Increased amounts of data

increase complexity, increase storage requirements and, in some instances, may degrade response times for both updates and queries. Third, the examination makes it easier to identify the controls needed for the different levels of normalization and to identify redundant data.

The SQL statement to insert the data in Figure 1 to the 1NF data structure is:

```
insert into PO values('KA-21979','01-SEP-94','45321','Diddledo Distributors','9821 Koala Parkway','Koolarooilly, Queensland 4398','Prepaid Air','2/20','14-SEP-94','92140','File Powder',89.50,25,'CA','31542','Country Ham',65.60,320,'EA');
```

The SQL statements to insert the data in Figure 1 to the 1NF data structure are:

```
insert into PO1 values('KA-21979','01-SEP-94','45321','Diddledo Distributors','9821 Koala Parkway','Koolarooilly','Queensland','4398','Prepaid Air',2,20,'14-SEP-94');  
insert into POITM1 values('KA-21979','92140','File Powder',89.50,25,'CA');  
insert into POITM1 values('KA-21979','31542','Country Ham',65.60,320,'EA');
```

The SQL statements to insert the data in Figure 1 to the 3NF data structure are:

```
insert into PO3 values('KA-21979','01-SEP-94','45321','Prepaid Air',2,20,'14-SEP-94');  
insert into POITM3 values('KA-21979','92140',89.50,25);  
insert into POITM3 values('KA-21979','31542',65.60,320);  
insert into ITMDESC3 values('92140','File Powder','CA');  
insert into ITMDESC3 values('31542','Country Ham','EA');  
insert into VENDOR3 values('45321','Diddledo Distributors','9821 Koala Parkway','4398');  
insert into ZIP3 values('4398','Koolarooilly','Queensland');
```

Although the 1NF data structure requires three rather than one insert statement, each of these insert statements are shorter, i.e., less complex. The reduced complexity of the 1NF statements would become more apparent if the number of repeating groups were increased, e.g., from two to ten. Seven insert statements are shown for the 3NF data structure. Not only are these statements shorter than those for either the $\neg$ NF or 1NF data structures but, if neither the vendor nor the items were new, only the first three insert statements would be required, i.e., the last four insert statements would not be entered.

Relative to $\neg$ NF, the 1NF data structure would save the space allocated for repeated groups for which no item was ordered, e.g., if only one item was ordered on the purchase order. The 1NF data structure would also save space relative to the $\neg$ NF data structure if more items were ordered than repeating groups, e.g., if three items were needed then another entire record must be created for the third item. The only additional storage requirements of 1NF relative to $\neg$ NF is that the purchase order number (PONO) must be entered into each *POITMI* record so that item data can be joined to purchase order data in *POI*.

Relative to 1NF, the 3NF data structure would save the space used for redundant data, e.g., vendor data for vendors that received more than one purchase order. The additional storage requirements of 3NF relative to 1NF are the attributes used to join the data together, e.g., the vendor number (VNO) in *VENDOR3* is required to link vendor data to the purchase order data in *PO3*.

3.4. Input Controls

Each of the data structures need a different set of input controls. Because data structures that only conform to lower levels of normalization require more controls than data structures that satisfy higher levels of normalization, the discussion begins by identifying controls needed by the 3NF data structure and then identifies additional controls needed by the 1NF and $\neg$ NF data structures. The controls identified in this example are not exhaustive, i.e., additional types of controls may be useful for these or other data structures.

The 3NF data structure would need data type controls, referential integrity checks, completeness checks, range checks, and template (format) checks. Data type controls verify that the data entered are of the appropriate type, e.g., that *PO3.PODATE*¹⁴ contains a date and *POITM3.UPRICE* contains a number. Referential integrity checks ensure that entries in one table have matching values in another table, e.g., that all *PO3.VNO* values have corresponding values in the *VENDOR3* table and that *VENDOR3.ZIP* values have corresponding entries in the *ZIP3* table. Completeness checks confirm that an attribute has a value, e.g., that *ITMDESC3.DESC* contains something other than a null value. Range checks determine that the value entered is within an acceptable range, e.g., $0 \leq PO3.TERMDISC \leq 5$. Template checks verify that the value entered conforms to an acceptable format, e.g., that each occurrence of *PO3.PONO* begins with two

¹⁴ SQL uses the construction *tablename.attributename* to uniquely identify data locations.

upper case letters, has a dash as the third character, and ends with five digits.

Relative to 3NF, the 1NF data structure would need additional input controls. These controls are needed to preserve the consistency of redundant data that can be present in 1NF but not in 3NF. For example, control procedures would be needed to ensure that vendor data is the same for all purchase orders sent to the same vendor. If a vendor changed street locations, the value of *PO1.VSTREET* should change in all records belonging to that vendor.

Relative to 1NF, the $\neg$ NF data structure would need additional controls. These controls are needed to ensure the data quality of repeating fields and of attributes that exhibit lack of atomicity. Controls for repeating fields should include completeness controls for groups of attributes, e.g., if *PO.ITEMNO2* did not contain a value then neither should any of the other attributes associated with the second item. If, however, *PO.ITEMNO2* did contain a value then the other attributes associated with the second item should also contain values. Controls for attributes that lack atomicity should include template checks that would not be required otherwise, e.g., a template check should verify that all values of *PO.TERMS* contain a number followed by a slash and another number.

The controls needed by 3NF data structures are often easy to implement with functions automatically provided by many RDBMSs. The additional controls needed by 1NF and $\neg$ NF typically require customized procedures and may consume more processing time. Hence, implementing the input controls needed by 1NF and

→NF data structures will incur more costs and may impair response times relative to equally effective controls for 3NF data structures.

4. Hypotheses

4.1. *Accuracy*

Normalized databases eliminate repeated fields, restrict attributes to atomic values, reduce data redundancy, and contain simpler tables. Eliminating repeating fields reduces the likelihood of duplicate data, makes queries of the repeated data easier, and eliminates the need for special records to hold repeating data that did not fit in the number of repetitions allocated. Permitting only atomic values means that attribute data types can be more restrictive, e.g., numeric instead of character. Most DBMSs automatically enforce data type restrictions.

Reducing data redundancy reduces the chances for errors. If a data value only exists in one tuple in one table, then it only has to be changed in that one place. If the value exists in several tables and/or tuples, it must be changed in all of these locations. Using simpler tables makes individual tables easier to control and to analyze for errors. In some cases, simpler tables mean that less elaborate but more precise data input controls can be used. For example, creating different tables for different types of employees may permit range checks with smaller standard deviations. Hence, individually simpler tables should reduce data errors. Hence, increasing the level of normalization should reduce errors, i.e., users will input a higher proportion of errors into databases that only conform to the lower normal forms than into databases that conform to the higher normal forms, or equivalently:

H1: There is an inverse relationship between the level of normalization and the proportion of errors users enter into a database.

The logical coherence and simplicity of tables that satisfy higher levels of normalization¹⁵ should increase data entry efficiency. Hence, users will input a higher number of attributes into databases that satisfy higher normal forms than into databases that conform to lower normal forms, i.e.:

H2: There is a direct relationship between the level of normalization and the speed that users enter data.

4.2. Input Controls

The primary methods organizations use to reduce data errors are installing data entry controls in the programs that are used for initial data capture and capturing data in less error-prone ways. Data entry controls may be implemented via specifying the appropriate data type (e.g., numeric, date, or time), selecting acceptable values from a list, implementing pre- or post-triggers on fields, and executing procedures before committing a tuple or set of tuples to one or more tables. These controls can explicitly test for or prevent possible errors caused by lack of normalization such as duplicate data and format and range violations, i.e.,

¹⁵ I.e., ignoring data redundancy considerations.

controls can compensate for problems caused by lack of normalization or lower levels of normalization. In addition, data entry controls can test for potential errors not affected by the level of normalization, i.e., validation controls should improve data accuracy regardless of the level of normalization. Consequently:

H3: Data validation procedures reduce the proportion of errors users enter into databases.

Because preventing errors, i.e., preserving data accuracy, increases the value of the system, users should suggest improved controls to reduce errors whether these errors arise from lack of normalization or from other sources. Greater experience and training make people more aware of the need for controls and of the controls required to prevent specific types of errors. Therefore, there is a direct relationship between IS background and identification of the appropriate input controls required to reduce specific errors, i.e.:

H4: Users with more extensive IS backgrounds will request more specific input controls than users with less IS experience and training.

Because lower levels of normalization increase the complexity of data structures, users inputting data into these structures will be more likely to experience frustration than users inputting data into the simpler data structures associated with higher normal forms. This frustration provides incentives to request

changes to unnormalized or less normalized data structures. Consequently, there is an inverse relationship between the level of normalization and the amount of requested changes to the data structure, i.e.:

H5: Users will request more data structure changes for databases that only conform to the lower normal forms than for databases that conform to higher normal forms.

5. Methodology

5.1. *Description of Experiment*

Eighty advanced undergraduate and masters level Commerce and Information Technology students participated in the experiment. Prior to the experiment, all participants received training in SQL, forms, and controls. Demographic and information system experience and training data were collected from all participants.¹⁶ The grade point average (GPA) was obtained from the participants' academic records.¹⁷ GPA data for participants who had recently transferred from other institutions were not used because the data were not included on the participant's transcript or because the previous institution's system was not comparable.

The experiment used a posttest-only control group design (Campbell and Stanley 1963, 25-27).¹⁸ Based their IS background,¹⁹ participants were randomly assigned to one of three treatment groups corresponding to one of three data

¹⁶ See Appendix B.

¹⁷ The GPA is on a seven point scale. The self reported GPA information provided on the demographic data questionnaire (Appendix B) was *not* used in the analysis.

¹⁸ This is one of three true experimental designs discussed by Campbell and Stanley (1963). In general, they prefer this experimental design over the other two: the pretest-posttest control group design and the Solomon four -group design (1963, 26).

¹⁹ Based on the information provided by each participant, a knowledgeable IS person (not the author) ranked each participant in descending order, i.e., the person considered to have the most extensive IS background was ranked number 80, the person with the next most extensive background was ranked number 79, etc. The order of the three treatments was then randomized, with the order being: ¬NF, 3NF, and 1NF. Participants were assigned according to their IS background ranking, i.e., 80 to ¬NF, 79 to 3NF, 78 to 1NF, 77 to 1NF, 76 to 3NF, 75 to ¬NF, 74 to ¬NF, etc. This method of randomization was intended to make the IS experience and training of the groups as equal as possible.

structures: not normalized ($\neg$ NF), first normal form (1NF), or third normal form (3NF).²⁰ The experiment was conducted in 10 sessions over three days in a computer lab that contained nine personal workstations.²¹ Except for the components that related to the data structure to which they had been assigned, each participant received the same set of instructions.²² After using their assigned workstation to log in to their individual areas on the server, each participant started a script that recorded all their interactions with the computer. Next, they entered a one line command to execute one of three preexisting command files. The command file automatically created a set of tables in their own user area and inserted eight correct purchase orders.²³ Participants used standard SQL insert and update commands to enter data about four additional purchase orders.²⁴ Because well designed input screens and controls can compensate for the effects of different levels of normalization, SQL insert statements were used to focus the experiment on normalization rather than on the design and implementation of input screens. After entering the data and querying the tables, participants composed a

²⁰ See Appendix A.

²¹ These machines were IBM compatible PCs running a terminal emulation package. The participants used these machines to connect to a Sun SPARCserver 20 running the Oracle RDBMS.

²² See Appendix C. In addition to the instructions in Appendix C, the participants received a copy of the appropriate data structure from Appendix A and the four purchase orders shown in Figure 3.

²³ See Figure 2 and Appendix D.

²⁴ See Figure 3.

memorandum that listed the controls²⁵ that they recommended be implemented for the data structure to which they were assigned. Each participant emailed their memorandum and script file to the coordinator of the experiment.

The session of the experiment lasted two hours. During the first hour the participants read the instructions (5-10 minutes) and entered data (50-55 minutes). At the end of one hour the participants were requested to stop entering data. Participants spent the second hour querying the data to detect anomalies (10-20 minutes), writing the controls memorandum (15-30 minutes), and sending email (2-5 minutes). Almost all participants used the full two hours. Performance on the experiment comprised five percent of each participant's grade in the subject.

5.2. Data Coding

Errors were determined by comparing the data in the participants' databases with the data on the paper input forms (see Figure 3). Recommendations by the participants were classified as specific controls, data structure changes, or general controls. Specific controls and data structure changes were controls that related directly to the data the participants entered. General controls were those that could relate to any database, i.e., general controls did not provide evidence of the participant's ability to apply appropriate controls to the specific situation.

²⁵ Appendix E lists the potential errors in the four additional purchase orders and, for each error, the possible controls that could prevent or detect it.

6. Results

6.1. *Summary Data*

Table 1 presents a summary of the results of the experiment. Useful data was not always available from all respondents for all measures due to participants' failure to follow directions and due to email problems.²⁶ The attribute error percent reports only errors made in attribute values, i.e., it does not include missing values or duplicate records. Examining only attribute errors is necessary to make comparisons of errors in all three data structures equitable. All errors were transcription or transposition errors. The attribute error percent with controls calculates the error rate if stringent data entry controls had been in place, i.e., these errors could not have been prevented by a well-designed information system that included all possible master file, range, format, template, and data type checks. The first of these two percentages, base tables assumed to exist, assumes data already exists for vendor and item description data. The second of these percentages, if attributes entered into base tables, assumes that all possible controls exist but that the data is being entered into the master files, e.g., into the tables for vendor and item description data.

²⁶ The size and number of the log files exceeded the disk space available for email. Some participants deleted their files before they could be contacted and requested to retransmit them.

TABLE 1
Summary of Experimental Results

Measurement	-NF	1NF	3NF
Data Entry Time			
average	57.0370	49.8636	55.0000
standard deviation	16.3582	11.2827	11.8423
count	27	22	25
Attributes Entered			
average	61.0000	78.9200	82.7308
standard deviation	34.1869	31.5568	24.5193
count	27	25	26
Attributes Entered per Minute			
average	1.1374	1.8538	1.5359
standard deviation	0.7020	0.8884	0.4883
count	27	22	25
Attribute Error Percent			
average	0.0451	0.0236	0.0170
standard deviation	0.0710	0.0440	0.0170
count	23	24	26
Attribute Error Percent with Controls: Base Tables Assumed to Exist			
average	0.0237	0.0081	0.0056
standard deviation	0.0594	0.0228	0.0084
count	23	24	26
Attribute Error Percent with Controls: If Attributes Entered into Base Tables			
average	0.0272	0.0102	0.0080
standard deviation	0.0590	0.0230	0.0084
count	23	24	26
Specific Controls Recommended			
average	6.8958	8.7885	7.4815
standard deviation	4.6881	7.4229	5.3653
count	24	26	27
Data Structure Changes Recommended			
average	1.4167	0.5000	0.2593
standard deviation	1.3123	0.7468	0.5833
count	24	26	27

6.2. Data Entry Accuracy and Efficiency

Hypothesis 1 suggested that an inverse relationship exists between the level of normalization and the proportion of errors users enter into a database. The results of an analysis of variance²⁷ (ANOVA) indicate a positive relationship between higher levels of normalization and more accurate attribute values, $p\text{-value} = 0.0257$, one-tail test.²⁸ See Table 2.

Hypothesis 2 asserted that there is a direct relationship between the level of normalization and the speed that users enter data. Because participants were using SQL statements to enter data directly into relational tables, participants with stronger IS backgrounds should be able to enter data more quickly regardless of which data structure they are updating. Although the keyboard proficiency of each participant was not measured directly, time spent entering data is a measure of typing proficiency. Less time spent entering data indicates greater typing proficiency. The results of an analysis of covariance (ANCOVA) with data entry time and IS background as covariants indicate a positive relationship between higher levels of normalization and data entry speed, $p\text{-value} = 0.0198$, one-tail

²⁷ Montgomery (1984) presents ANOVA, ANCOVA, and regression using both mathematical and pragmatic terminology. See especially pages 44-59, 110-115, 399-440, and 475-495.

²⁸ These results are influenced by two outliers with standardized residuals of 4.88 and 3.27. If these two observations are removed from the analysis, the significance of the results do not meet the traditional $p\text{-value}$ of 0.05, however, no theoretical reason exists for deleting the two observations. A nonparametric Kruskal-Wallis test of hypothesis one with the two observations included yielded a $p\text{-value}$ of 0.0938, one-tail test.

TABLE 2
Effect of Normalization Level on Attribute Error Percentage

Source	DF	Mean Square	F Value	Pr > F	Parameter Estimate
Model	1	0.00947	3.93	0.0257	
Error	71	0.00241			
Normalization	1	0.00947	3.93	0.0257	-0.01392
$R^2 = 0.0525$					

test.²⁹ The results also confirm the positive relationship between IS background and data entry speed, $p\text{-value} = 0.0290$, one-tail test.³⁰ See Table 3.

6.3. *Validation Controls and Data Accuracy*

Hypothesis 3 stated that data validation procedures reduce the proportion of errors users enter into databases. The results of a paired-comparison t-test substantiate the expected positive relationship between validation controls and data accuracy: mean = 0.0159, standard error = 0.0039, $T = 4.0268$, and $p\text{-value} < 0.0001$.

6.4. *Data Entry Controls and Data Structure Recommendations*

Hypothesis 4 suggested that users with more extensive IS backgrounds will request more specific input controls than users with less IS experience and training. The results of regression confirm the expected positive relationship between IS

²⁹ These results are influenced by two outliers with standardized residuals of -3.39 and 3.26. If these two observations are removed from the analysis, the significance of the results do not meet the traditional $p\text{-value}$ of 0.05, however, no theoretical reason exists for deleting the two observations. A nonparametric Kruskal-Wallis test of hypothesis two with the two observations included yielded a $p\text{-value}$ of 0.0995, one-tail test.

³⁰ Recall that the participant deemed to have the most extensive experience and training in information systems received a ranking of 80 and the participant with the least extensive IS background received a ranking of 1. Hence a negative coefficient for IS background indicates a positive relationship between IS background and the number of attributes entered per minute.

TABLE 3
*Effect of Normalization Level, Data Entry Time, and IS Background on
 Attributes Entered per Minute*

Source	DF	Mean Square	F Value	Pr > F	Parameter Estimate
Model	3	4.80974	11.81	0.0001	
Error	70	0.40742			
Normalization	1	1.79178	4.40	0.0198	0.18654
Time	1	8.34343	20.48	0.0001	-0.02503
IS Background	1	1.51491	3.72	0.0290	0.06504
$R^2 = 0.34$					

background and specific data entry controls, $p\text{-value} = 0.0041$, one-tail test.³¹ See Table 4.

Hypothesis 5 suggested that users will request more data structure changes for databases that only conform to the lower normal forms than for databases that conform to higher normal forms. Hypothesis 4 implied that people with more extensive IS experience and training are more likely to recommend appropriate data structure changes than people with less extensive IS backgrounds. Regression results confirm the expected relationship between normalization level and requested data structure changes, $p\text{-value} = 0.0001$, one tailed test.³² The relationship between IS background and requested data structure changes, $p\text{-value} = 0.0068$, one tailed test, provides further support for hypothesis 4. See Table 5.

³¹ Analysis of the individual error terms revealed one outlier with a standardized residual of 4.94. No theoretical reason exists for removing this observation from the analysis. Furthermore, if this observation is removed from the analysis, the significance of the results still satisfy the traditional criteria of $p\text{-value} < 0.05$.

³² Analysis of the individual error terms did not reveal any outliers. Furthermore, a nonparametric Kruskal-Wallis test yielded a $p\text{-value} < 0.0001$.

TABLE 4
Effect of IS Background on Recommended Data Entry Controls

Source	DF	Mean Square	F Value	Pr > F	Parameter Estimate
Model	1	250.47267	7.41	0.0041	
Error	75	33.79110			
IS Background	1	250.47267	7.41	0.0041	0.07846
$R^2 = 0.09$					

TABLE 5
Effect of Normalization Level and IS Background on Recommended Data Structure Changes

Source	DF	Mean Square	F Value	Pr > F	Parameter Estimate
Model	2	10.99377	13.20	0.0001	
Error	74	0.83300			
Normalization	1	15.61597	18.75	0.0001	-0.55477
IS Background	1	5.34082	6.41	0.0068	0.01147
$R^2 = 0.26$					

7. Discussion

The results support the association of higher levels of normalization with improved data accuracy and data entry speed. This association accentuates the need to normalize or increase the level of normalization of databases intended for end user access that were not normalized due to performance³³ or other considerations. This association also implies that higher levels of normalization would improve data quality in end user tables.

The results show that people with more extensive IS backgrounds are better able to identify appropriate data entry controls and data structure changes. This association implies that end users creating their own tables should receive training in normalization and data entry controls. It also implies that IS professionals should assist end users who develop tables that will be used by multiple users or for important decisions.

An analysis of the accuracy of data entered with and without controls revealed two important points. First, although normalization does improve data accuracy, well-designed input controls yield as great if not greater improvements to data quality than higher levels of normalization. Second, even the most stringent set of data entry controls possible would not have detected or prevented some of the data errors. These two observations support the current emphasis on ensuring

³³ Higher forms of normalization are detrimental to database performance due to I/O and other considerations (e.g., Tsai 1988, 451; Shasta 1992, 92)

data quality through stringent data entry controls but indicate that data entry controls alone are not adequate to guarantee data accuracy.

8. Contributions, Limitations, and Future Research

The research provided empirical evidence about the effect of various levels of normalization on data entry and data entry controls. The results indicate that higher normal forms increase data accuracy and data entry efficiency, that validation controls produce significantly more accurate data, and that more extensive IS experience and training are required to recommend the appropriate input controls and data structures. The results also imply that less normalized data structures are more likely to frustrate people inputting data and cause them to recommend changing the structures to conform with higher levels of normalization.

A limitation of this study is that it did not collect data over a long period from actual end-users interacting with databases they use in their job-related activities. The participants have received training in IT and business related subjects, however, and thus they are probably typical of end-users that develop their own systems. Furthermore, the participants may have more training and greater understanding of relational databases, SQL, and controls than many end-users who must enter data and create their own set of data entry controls as part of their job-related activities.

Future research could investigate the relationship of normalization and data errors in organizations' databases. Such research is potentially valuable not only for specific insights into the relationship between normalization and data errors but for general insights into the magnitude and nature of data errors in the databases organizations currently used to make decisions and produce reports. Such research

requires the development of procedures for identifying errors. Furthermore, this research may experience difficulties comparing databases between organizations due to differences in organizational structures or in the competence and motivations of the data owners. Future research could also investigate the relationship of normalization and user satisfaction in organizations' databases. Comparability between databases may be problematical for this research as well.

REFERENCES

REFERENCES

- American Institute of Certified Public Accountants (AICPA). *Statement on Auditing Standards No. 53: The Auditor's Responsibility to Detect and Report Errors and Irregularities*. New York: AICPA, 1988a.
- . *Statement on Auditing Standards No. 55: Consideration of the Internal Control Structure in a Financial Statement Audit*. New York: AICPA, 1988b.
- Anderson, K. M.; D. C. Dumphy; and P. W. F. Wilson. "Management of Data Quality in a Long-Term Epidemiologic Study: The Framingham Heart Study." in *Data Quality Control: Theory and Pragmatics*. Liepens and Uppuluri, eds. (1990): 57-68.
- Bailey, J. E. and S. W. Pearson. "Development of a Tool For Measuring and Analyzing Computer User Satisfaction." *Management Science*. (May 1983): 530-45.
- Ballou, D. P. and H. L. Pazer. "Modeling Data and Process Quality in Multi-Input, Multi-Output Information Systems." *Management Science*. (February 1985): 150-62.
- . "Cost/Quality Tradeoffs for Control Procedures in Information Systems." *Omega: International Journal of Management Science*. (Vol. 15, No. 6, 1985): 509-21.
- Beynon-Davies, P. *Relational Database Design*. Oxford: Blackwell Scientific Publications, 1992.
- Blackwell, D. "Equivalent Comparisons of Experiments." *Annals of Mathematical Statistics*. (June 1953): 265-73.
- Bowen, P. L. *Managing Data Quality in Accounting Information Systems: A Stochastic Clearing System Approach*. Unpublished Ph.D. dissertation, The University of Tennessee 1992.
- , G. P. Schneider; and K. T. Fields. "Managing Data Quality in Client-Server Environments." *Information Systems Audit and Control Journal* (Vol. IV 1995): 28-35.
- Campbell, D. J. "Task Complexity: A Review and Analysis." *Academy of Management Review*. (Vol. 13, No. 1, 1988): 40-52.
- Campbell, D. T., and J. C. Stanley. *Experimental and Quasi-Experimental Designs For Research*. Boston: Houghton Mifflin Company, 1963.
- Chen, P. P. "The Entity-Relationship Model - Toward a Unified View of Data." *ACM Transactions on Database Systems*. (March 1976): 9-36.

- Coakley, J. R., and J. K. Loebbecke. "The Expectations of Accounting Errors in Medium-Sized Manufacturing Firms." *Advances in Accounting*. (Vol. 2, 1985): 411-35.
- Codd, E. F. "A Relational Model of Data for Large Shared Data Banks." *Communications of the ACM*. (June 1970): 377-87.
- . "Recent Investigations in Relational Data Base Systems." *Information Processing* 74. (1974): 1017-87.
- Committee of Sponsoring Organizations of the Treadway Commission (CSOTC). *Internal Control -- Integrated Framework*. Exposure Draft. New York: Committee of Sponsoring Organizations of the Treadway Commission, 1991.
- Cushing, B. E. "A Mathematical Approach to the Analysis and Design of Internal Control Systems." *The Accounting Review*. (January 1974): 24-41.
- Date, C. J. *Relational Database: Selected Writings*. Reading, Massachusetts: Addison-Wesley Publishing Company, 1990.
- Date, C. J. *An Introduction to Database Systems*. 5th ed. Reading, Massachusetts: Addison-Wesley Publishing Company, 1990.
- Dillard, R. A. "Using Data Quality Measures in Decision Making Algorithms." *IEEE Expert*. (December 1992): 63-72.
- Fagin, R. "Multivalued Dependencies and a New Normal Form for Relational Databases." *ACM Transactions on Database Systems*. (September 1977): 262-278.
- Feltham, G. A. *Information Evaluation*. Sarasota, Florida: American Accounting Association, 1972.
- Flournoy, N., and L. B. Hearne. "Quality Control for a Shared Multidisciplinary Database." in *Data Quality Control: Theory and Pragmatics*. Liepens and Uppuluri, eds. (1990): 43-56.
- Fox, C.; A. Levitin; and T. Redman. "The Notion of Data and its Quality Dimensions." *Information Processing & Management*. (Vol. 30, No. 1, 1994): 9-19.
- Ham, J.; D. Losell; and W. Smieliauskas. "An Empirical Study of Error Characteristics in Accounting Populations." *The Accounting Review*. (July 1985): 387-406.
- Hamlen, S. S. "A Chance-Constrained Mixed Integer Programming Model for Internal Control Design." *The Accounting Review*. (October 1980): 578-93.
- Hilton, R. W. "The Determinants of Information System Value: An Illustrative Analysis." *Journal of Accounting Research*. (Autumn 1979): 411-35.

- _____. "The Determinants of Information System Value: Synthesizing Some General Results." *Management Science*. (January 1981): 57-64.
- _____; R. J. Swieringa; and R. E. Hoskin. "Perception of Accuracy as a Determinant of Information Value." *Journal of Accounting Research*. (Spring 1981): 86-108.
- Huh, Y. U.; F. R. Keller; T. C. Redman; and A. R. Watkins. "Data Quality." *Information and Software Technology*. (October 1990): 559-65.
- Hylas, R. E., and R. H. Ashton. "Audit Detection of Financial Statement Errors." *The Accounting Review*. (October 1982): 751-65.
- Iceman, R. C., and W. A. Hillison. "Disposition of Audit-Detected Errors: Some Evidence on Evaluative Materiality." *Auditing: A Journal of Practice & Theory*. (Spring 1991): 22-34.
- Institute of Internal Auditors Research Foundation (IIARF). *Systems Auditability and Control*. Altamonte Springs, Florida: The Institute of Internal Auditors Research Foundation, 1991.
- I/S Analyzer. "Guiding Distributed Systems." (November 1987): 1-12.
- Itami, H. *Adaptive Behavior: Management Control and Information Analysis*. Sarasota, Fla.: American Accounting Association, 1977.
- Ives, B.; M. H. Olson; and J. J. Baroudi. "The Measurement of User Information Satisfaction." *Communications of the ACM*. (October 1983): 785-93.
- Jang, Y., H. B. Kon, and R. Y. Wang. "A Data Consumer-based Approach to Supporting Data Quality Judgment." *Proceedings of the Second Annual Workshop on Information Technology and Systems*. Dallas, Texas. 1992.
- Johnson, J. R.; R. A. Leitch; and J. Neter. "Characteristics of Errors in Accounts Receivable and Inventory Audits." *The Accounting Review*. (April 1981): 270-93.
- Kent, W. "A Simple Guide To Five Normal Forms In Relational Database Theory." *Communications of the ACM*. (February 1983): 120-25.
- Kinney, W. R., Jr. "The Predictive Power of Limited Information in Preliminary Analytical Review: An Empirical Study." *Journal of Accounting Research*. (Supplement 1979): 148-65.
- Kreutzfeldt, R. W., and W. A. Wallace. "Error Characteristics in Audit Populations: Their Profile and Relationship to Environmental Factors." *Auditing: A Journal of Practice & Theory*. (Fall 1986): 20-43.

- Laudon, K. C. "Data Quality and Due Process in Large Interorganizational Record Systems." *Communications of the ACM*. (January 1986): 4-11.
- LePage, N. J. "Data Quality Control at United States Fidelity and Guaranty Company." in *Data Quality Control: Theory and Pragmatics*. Liepens and Uppuluri, eds. (1990): 25-42.
- Liepens, G. E. "Reflections on Validation and Quality Assessment of FPC Form 4 Data." in *Data Quality Control: Theory and Pragmatics*. Liepens and Uppuluri, eds. (1990): 19-24.
- Liepens, G. E., and V. R. R. Uppuluri, eds. *Data Quality Control: Theory and Pragmatics*. New York: Marcel Dekker, 1990.
- Loebl, A. S. "Accuracy and Relevance and the Quality of Data." in *Data Quality Control: Theory and Pragmatics*. Liepens and Uppuluri, eds. (1990): 105-144.
- Montgomery, D. C. *Design and Analysis of Experiments*. 2nd ed. New York: John Wiley and Sons, 1984.
- Morey, R. C. "Estimating and Improving the Quality of Information in a MIS." *Communications of the ACM*. (May 1982): 337-42.
- Neter, J., and J. K. Loebbecke. *Behavior of Major Statistical Estimators in Sampling Audit Populations*. New York: AICPA, 1975.
- Paradice, D. B. and W. L. Fuerst. "An MIS Data Quality Methodology Based on Optimal Error Detection." *The Journal of Information Systems*. (Spring 1991): 48-66.
- Parsaye, K. and M. Chignell. "Data Quality Control with Smart Databases." *AI Expert*. (May 1993): 22-27.
- Ramage, J. G.; A. M. Krieger; and L. L. Spero. "An Empirical Study of Error Characteristics in Audit Populations." *Journal of Accounting Research*. (Supplement 1979): 72-102.
- Redman, T. C. "Data Quality for Telecommunications." *IEEE Journal On Selected Areas In Communications*. (February 1994): 306-12.
- Shasta, D. E. *Database Tuning: A Principled Approach*. Englewood Cliffs, New Jersey: Prentice-Hall, 1992.
- Specht, P. H. "Job Characteristics as Indicants of CBIS Data Requirements." *MIS Quarterly*. (September 1986): 270-86.

- Stratton, W. O. "Accounting Systems: The Reliability Approach to Internal Control Evaluation." *Decision Sciences*. (January 1981): 51-67.
- Svanks, M. I. "Integrity Analysis: Methods for Automating Data Quality Assurance." *Information and Software Technology*. (December 1988): 595-605.
- Tsai, A.Y.H. *Database Systems: Management and Use*. Scarborough, Ontario: Prentice-Hall Canada, 1988.
- Vasarhelyi, M. A., and F. B. Halper. "The Continuous Audit of Online Systems." *Auditing: A Journal of Practice & Theory*. (Spring 1991): 110-25.
- Wand, Y., and R. Y. Wang. "Anchoring Data Quality Dimensions in Ontological Foundations." Forthcoming, *Communications of the ACM* (1995).
- Wang, R. Y., ed. *Information Technology in Action: Trends and Perspectives*. Englewood Cliffs, New Jersey, 1993.
- _____, and H. B. Kon. "Toward Total Data Quality Management." in *Information Technology in Action: Trends and Perspectives*. R. Y. Wang ed. (1993): 179-97.
- _____, and M. P. Reedy. "Quality Data Objects." Working paper TDQM-92-06, MIT TDQM Research Program, E53-320, 50 Memorial Drive, Cambridge, Ma. 02139. (1993)
- _____, V. C. Storey, and C. P. Firth. "A Framework for Analysis of Data Quality Research." *IEEE Transactions on Knowledge and Data Engineering*. (August 1995): 623-39.
- Weber, R. *EDP Auditing*. 2nd ed. New York: McGraw-Hill Book Company, 1988.
- Willingham, J. J., and W. F. Wright. "Financial Statement Errors and Internal Control Judgments." *Auditing: A Journal of Practice & Theory*. (Fall 1985): 57-70.
- Wood, R. E. "Task Complexity: Definition of the Construct." *Organizational Behavior and Human Decision Processes*. (Vol. 37, 1986): 60-82.
- Yu, S. and J. Neter. "A Stochastic Model of the Internal Control System." *Journal of Accounting Research*. (Autumn 1973): 273-95.

APPENDICES

APPENDIX A

Data Structures

A.1 Not Normalized (¬NF)

<u>Attribute Name</u>	<u>Type</u>	<u>Definition</u>	<u>Comment</u>
<u>PO Table</u>			
PONO	Char(8)	Purchase order number	
PODATE	Date	Date of purchase order	
VNO	Char(5)	Vendor number	
VNAME	Char(30)	Vendor name	
VSTREET	Char(30)	Vendor's street address	
VCSZ	Char(30)	Vendor's city, state, & zip	Violates atomicity
TRANS	Char(15)	How transportation paid	
TERMS	Char(5)	Terms of payment	Violates atomicity
DUEDATE	Date	Date due or date wanted	
ITEMNO1	Char(6)	Item number of first item	Repeating fields -
DESC1	Char(30)	Description of ITEMNO1	violate 1NF
UPRICE1	Number	Unit price of ITEMNO1	
QTY1	Number	Quantity ordered of ITEMNO1	
UMEAS1	Char(5)	Unit of measure of ITEMNO1	
ITEMNO2	Char(6)	Item number of the second item	
DESC2	Char(30)	Description of ITEMNO2	
UPRICE2	Number	Unit price of ITEMNO2	
QTY2	Number	Quantity ordered of ITEMNO2	
UMEAS2	Char(5)	Unit of measure of ITEMNO2	

A.2 First Normal Form (1NF)

<u>Attribute Name</u>	<u>Type</u>	<u>Definition</u>
<u>PO1 Table</u>		
PONO	Char(8)	Purchase order number
PODATE	Date	Date of purchase order
VNO	Char(5)	Vendor number
VNAME	Char(30)	Vendor name
VSTREET	Char(30)	Vendor's street address
VCITY	Char(15)	Vendor's city
VSTATE	Char(15)	Vendor's state
VZIP	Char(10)	Vendor's zip or post code
TRANS	Char(15)	How transportation paid
TERMDISC	Number	Terms of payment - discount percent
TERMDAYS	Number	Terms of payment - number of days discount applies
DUEDATE	Date	Date due or date wanted
<u>POITM1 Table</u>		
PONO	Char(8)	Purchase order number
ITEMNO	Char(6)	Item number of item
DESC	Char(30)	Description of ITEMNO
UPRICE	Number	Unit price of ITEMNO
QTY	Number	Quantity ordered of ITEMNO
UMEAS	Char(5)	Unit of measure of ITEMNO

A.3 Third Normal Form (3NF)

<u>Attribute Name</u>	<u>Type</u>	<u>Definition</u>
<u>PO3 Table</u>		
PONO	Char(8)	Purchase order number
PODATE	Date	Date of purchase order
VNO	Char(5)	Vendor number
TRANS	Char(15)	How transportation paid
TERMDISC	Number	Terms of payment - discount percent
TERMDAYS	Number	Terms of payment - number of days discount applies
DUEDATE	Date	Date due or date wanted
<u>POITM3 Table</u>		
PONO	Char(8)	Purchase order number
ITEMNO	Char(6)	Item number of item
UPRICE	Number	Unit price of ITEMNO
QTY	Number	Quantity ordered of ITEMNO
<u>ITMDESC3 Table</u>		
ITEMNO	Char(6)	Item number of item
DESC	Char(30)	Description of ITEMNO
UMEAS	Char(5)	Unit of measure of ITEMNO
<u>VENDOR3 Table</u>		
VNO	Char(5)	Vendor number
VNAME	Char(30)	Vendor name
VSTREET	Char(30)	Vendor's street address
VZIP	Char(10)	Vendor's zip or post code
<u>ZIP3 Table</u>		
VZIP	Char(10)	Vendor's zip or post code
VCITY	Char(15)	Vendor's city
VSTATE	Char(15)	Vendor's state

APPENDIX B
Demographic Information

Please supply the following information.

Name: _____ Student Number: _____

Tutorial Group: _____ Date of birth: (day/month/year) __/__/__ Gender: m f

IS/IT/CS Subjects Taken (most recent first):

1. Subject No: _____ Grade Received: _____ Date (semester/year): _____
Title: _____ Institution (e.g., UQ): _____
2. Subject No: _____ Grade Received: _____ Date (semester/year): _____
Title: _____ Institution (e.g., UQ): _____
3. Subject No: _____ Grade Received: _____ Date (semester/year): _____
Title: _____ Institution (e.g., UQ): _____
4. Subject No: _____ Grade Received: _____ Date (semester/year): _____
Title: _____ Institution (e.g., UQ): _____
5. Subject No: _____ Grade Received: _____ Date (semester/year): _____
Title: _____ Institution (e.g., UQ): _____
6. Subject No: _____ Grade Received: _____ Date (semester/year): _____
Title: _____ Institution (e.g., UQ): _____
7. Subject No: _____ Grade Received: _____ Date (semester/year): _____
Title: _____ Institution (e.g., UQ): _____
8. Subject No: _____ Grade Received: _____ Date (semester/year): _____
Title: _____ Institution (e.g., UQ): _____

Work Experience (most recent first):

1. Start date: ____/____/____ End date: ____/____/____
Title or position: _____
2. Start date: ____/____/____ End date: ____/____/____
Title or position: _____
3. Start date: ____/____/____ End date: ____/____/____
Title or position: _____

Education:

Overall GPA: ____ GPA in IT/IS/CS subjects: ____

Degree:

Commerce IT CS Other (please enter): _____

Instructions to Participants

Instructions - A

Purposes

Some of the purposes of this exercise are:

1. To let you practice designing input controls for a specific situation.
2. To ensure that you can use SQL to execute a command file, to insert data, and to display data.
3. To ensure that you can use email.

Overview of Tasks

1. Copy files containing data structures and initial data
2. Use SQL to create the tables and load the initial data
3. Use SQL to insert new records (the data structure and four new POs are attached)
4. Use SQL to compare previous and new data
5. Send a memo via email listing the specific input controls you recommend implementing for the system
6. Send your log files
7. Complete the survey

Read the entire set of instructions *before* beginning Task 1.

Task 1 - Copy files containing data structures and initial data

- a. To record your session, log into your area on valinor and type:
- ```
script your username
pwd
```

This command puts everything you type in a file with a timestamp at the top of the file to indicate when you started.

- b. To copy the SQL commands to create the tables and insert the initial data, type:
- ```
cp /staff/bowen/labqza.sql .          /* the full stop indicates to copy the file to
                                         your directory */
```

Task 2 - Use SQL to create the tables and load the initial data

Start sqlplus and type

```

spool your_username.lst          /* to record your SQL commands */
@labqza.sql                     /* to create the tables and load the initial
                                data */
@/staff/bowen/date.sql          /* to record the date and time */

```

Task 3 - Use SQL to insert new records

Use the INSERT command to insert the new data on the 4 new POs into the appropriate tables. See the attached data structure on page 3 so you know the data type of each attribute and, hence, the appropriate way to enter data values for each field. The four new POs are on pages 4-7. **Note:** The new POs contain intentional errors to prompt you to recommend certain controls. After inserting the four new purchase orders, type

```
@/staff/bowen/date.sql          /* to the record the date and time */
```

Task 4 - Use SQL to compare previous and new data

Use SQL SELECT commands to compare the initial data and the data you entered. Search diligently for potential problems and inconsistencies with the data. Task 5 requires you to design specific controls to prevent or detect these problems. Assume the initial data were correct. After you have finished looking at the data, type

```
@/staff/bowen/date.sql          /* to the record the date and time */
spool off                       /* to close the SQL file with your
                                commands */
exit                            /* to exit SQL */
```

Task 5 - Recommend controls

Based on potential problems you discovered when performing task 4 and your knowledge about input controls, compose and send a memo to pbowen@lorien.commerce.uq.edu.au explaining the input controls you recommend implementing for the system. Please use the following major headings.

1. General Input Controls
2. Field Checks
3. Record Checks
4. Data Structure Changes
5. Other Input Controls

Option 5a (recommended)

You may wish to use pine to compose and send the memo. To start this email utility, at the operating system prompt, type

```
pine
```

To compose a message in Pine, select the C option within Pine, then enter

"pbowen@lorien.commerce.uq.edu.au" in the "To:" field, and "Lab Quiz 1" in the "Subject:" field. Type your memo in the message text part of the screen. Press ^X (Control-X) to send the message.

Option 5b

Alternately, you can compose your memo regarding the recommended input controls using "vi" and send the file in the same manner for sending log files, e.g., vi *your username.memo*, compose the memo, and mail pbowen@lorien.commerce.uq.edu.au <*filename*.

Task 6 - Transmit log files

At the operating system prompt, type

date

/* to record the date and time */

exit

/* to close the log file */

Send your log files to pbowen@lorien.commerce.uq.edu.au by typing

mail pbowen@lorien.commerce.uq.edu.au <filename

You created two of these files: *your username.lst* and *your username* with no extension.

The instructors will use these files as the basis for marking your quiz.

Task 7 - Complete the survey

Ask the person monitoring the session for a survey. Complete and return the survey before leaving the lab.

Instructions - B

Purposes

Some of the purposes of this exercise are:

1. To let you practice designing input controls for a specific situation.
2. To ensure that you can use SQL to execute a command file, to insert data, and to display data.
3. To ensure that you can use email.

Overview of Tasks

1. Copy files containing data structures and initial data
2. Use SQL to create the tables and load the initial data
3. Use SQL to insert new records (the data structure and four new POs are attached)
4. Use SQL to compare previous and new data
5. Send a memo via email listing the specific input controls you recommend implementing for the system
6. Send your log files
7. Complete the survey

Read the entire set of instructions *before* beginning Task 1.

Task 1 - Copy files containing data structures and initial data

- a. To record your session, log into your area on valinor and type:

```
script your_username  
pwd
```

This command puts everything you type in a file with a timestamp at the top of the file to indicate when you started.

- b. To copy the SQL commands to create the tables and insert the initial data, type:

```
cp /staff/bowen/labqzb.sql .      /* the full stop indicates to copy the file to  
                                  your directory */
```

Task 2 - Use SQL to create the tables and load the initial data

Start sqlplus and type

```
spool your_username.lst          /* to record your SQL commands */  
@labqzb.sql                      /* to create the tables and load the initial  
                                data */  
@/staff/bowen/date.sql          /* to the record the date and time */
```

Task 3 - Use SQL to insert new records

Use the INSERT command to insert the new data on the 4 new POs into the appropriate tables. See the attached data structure on page 3 so you know the data type of each attribute and, hence, the appropriate way to enter data values for each field. The four new POs are on pages 4-7. **Note:** The new POs contain intentional errors to prompt you to recommend certain controls. After inserting the four new purchase orders, type

```
@/staff/bowen/date.sql          /* to the record the date and time */
```

Task 4 - Use SQL to compare previous and new data

Use SQL SELECT commands to compare the initial data and the data you entered. Search diligently for potential problems and inconsistencies with the data. Task 5 requires you to design specific controls to prevent or detect these problems. Assume the initial data were correct. After you have finished looking at the data, type

```
@/staff/bowen/date.sql          /* to the record the date and time */
spool off                       /* to close the SQL file with your
                                commands */
exit                            /* to exit SQL */
```

Task 5 - Recommend controls

Based on potential problems you discovered when performing task 4 and your knowledge about input controls, compose and send a memo to pbowen@lorien.commerce.uq.edu.au explaining the input controls you recommend implementing for the system. Please use the following major headings.

1. General Input Controls
2. Field Checks
3. Record Checks
4. Data Structure Changes
5. Other Input Controls

Option 5a (recommended)

You may wish to use pine to compose and send the memo. To start this email utility, at the operating system prompt, type

```
pine
```

To compose a message in Pine, select the C option within Pine, then enter "pbowen@lorien.commerce.uq.edu.au" in the "To:" field, and "Lab Quiz 1" in the "Subject:" field. Type your memo in the message text part of the screen. Press ^X (Control-X) to send the message.

Option 5b

Alternately, you can compose your memo regarding the recommended input controls using "vi" and send the file in the same manner for sending log files, e.g., vi *your username.memo*, compose the memo, and mail pbowen@lorien.commerce.uq.edu.au < *filename*.

Task 6 - Transmit log files

At the operating system prompt, type

date	/* to record the date and time */
exit	/* to close the log file */

Send your log files to pbowen@lorien.commerce.uq.edu.au by typing

mail pbowen@lorien.commerce.uq.edu.au <filename

You created two of these files: *your username.lst* and *your username* with no extension.

The instructors will use these files as the basis for marking your quiz.

Task 7 - Complete the survey

Ask the person monitoring the session for a survey. Complete and return the survey before leaving the lab.

Instructions - C

Purposes

Some of the purposes of this exercise are:

1. To let you practice designing input controls for a specific situation.
2. To ensure that you can use SQL to execute a command file, to insert data, and to display data.
3. To ensure that you can use email.

Overview of Tasks

1. Copy files containing data structures and initial data
2. Use SQL to create the tables and load the initial data
3. Use SQL to insert new records (the data structure and four new POs are attached)
4. Use SQL to compare previous and new data
5. Send a memo via email listing the specific input controls you recommend implementing for the system
6. Send your log files
7. Complete the survey

Read the entire set of instructions *before* beginning Task 1.

Task 1 - Copy files containing data structures and initial data

- a. To record your session, log into your area on valinor and type:

```
script your username
pwd
```

This command puts everything you type in a file with a timestamp at the top of the file to indicate when you started.

- b. To copy the SQL commands to create the tables and insert the initial data, type:

```
cp /staff/bowen/labqzc.sql .      /* the full stop indicates to copy the file to
                                   your directory */
```

Task 2 - Use SQL to create the tables and load the initial data

Start sqlplus and type

```
spool your username.lst          /* to record your SQL commands */
@labqzc.sql                      /* to create the tables and load the initial
                                   data */
@/staff/bowen/date.sql          /* to the record the date and time */
```

Task 3 - Use SQL to insert new records

Use the INSERT command to insert the new data on the 4 new POs into the appropriate tables. See the attached data structure on page 3 so you know the data type of each attribute and, hence, the appropriate way to enter data values for each field. The four new POs are on pages 4-7. **Note:** The new POs contain intentional errors to prompt you to recommend certain controls. After inserting the four new purchase orders, type

```
@/staff/bowen/date.sql          /* to the record the date and time */
```

Task 4 - Use SQL to compare previous and new data

Use SQL SELECT commands to compare the initial data and the data you entered. Search diligently for potential problems and inconsistencies with the data. Task 5 requires you to design specific controls to prevent or detect these problems. Assume the initial data were correct. After you have finished looking at the data, type

```
@/staff/bowen/date.sql          /* to the record the date and time */
spool off                       /* to close the SQL file with your
                                commands */
exit                            /* to exit SQL */
```

Task 5 - Recommend controls

Based on potential problems you discovered when performing task 4 and your knowledge about input controls, compose and send a memo to pbowen@lorien.commerce.uq.edu.au explaining the input controls you recommend implementing for the system. Please use the following major headings.

1. General Input Controls
2. Field Checks
3. Record Checks
4. Data Structure Changes
5. Other Input Controls

Option 5a (recommended)

You may wish to use pine to compose and send the memo. To start this email utility, at the operating system prompt, type

```
pine
```

To compose a message in Pine, select the C option within Pine, then enter "pbowen@lorien.commerce.uq.edu.au" in the "To:" field, and "Lab Quiz 1" in the "Subject:" field. Type your memo in the message text part of the screen. Press ^X (Control-X) to send the message.

Option 5b

Alternately, you can compose your memo regarding the recommended input controls using "vi" and send the file in the same manner for sending log files, e.g., vi *your username.memo*, compose the memo, and mail pbowen@lorien.commerce.uq.edu.au <*filename*.

Task 6 - Transmit log files

At the operating system prompt, type

date

/* to record the date and time */

exit

/* to close the log file */

Send your log files to pbowen@lorien.commerce.uq.edu.au by typing

mail pbowen@lorien.commerce.uq.edu.au < *filename*

You created two of these files: *your username.lst* and *your username* with no extension.

The instructors will use these files as the basis for marking your quiz.

Task 7 - Complete the survey

Ask the person monitoring the session for a survey. Complete and return the survey before leaving the lab.

APPENDIX D

SQL to Create Tables and Insert Initial Purchase Order Data

```
/* FILE: labqza.sql */  
/*      Final Version  */
```

```
drop table poa;
```

```
/* Create Table Statements */
```

```
create table poa(pono char(8),  
                podate date,  
                vno char(5),  
                vname char(30),  
                vstreet char(30),  
                vcspc char(30),  
                trans char(15),  
                terms char(5),  
                duedate date,  
                itemno1 char(6),  
                idesc1 char(30),  
                uprice1 number,  
                qty1 number,  
                umeas1 char(5),  
                itemno2 char(6),  
                idesc2 char(30),  
                uprice2 number,  
                qty2 number,  
                umeas2 char(5));
```

```
/* Insert Statements */
```

```
insert into poa values('TA-21971','02-MAY-95','45321','Koowang Office  
Supplies','9812 Koala Parkway','Karanga Queensland 4938','Prepaid','2/20',  
'05-MAY-95','92140','Notepads',24.50,3,'DZ','31542','Pens',16.50,6,'DZ');
```

```
insert into poa values('TA-21972','02-MAY-95','45321','Koowang Office  
Supplies','9812 Koala Parkway','Karanga Queensland 4938','Prepaid','2/20',  
'05-MAY-95','92520','2 Ring Binders',4.50,15,'EA','31342','File Folders',  
27.25,2,'BX');
```

insert into poa values('TA-21973','03-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga Queensland 4938','Prepaid','2/20','08-MAY-95','92264','Staplers',16.50,3,'EA','11590','Copier Paper',92.75,15,'BX');

insert into poa values('TA-21974','04-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga Queensland 4938','Prepaid','2/20','09-MAY-95','78601','Toner',94.30,4,'EA',null,null,null,null,null);

insert into poa values('TA-21975','05-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga Queensland 4938','Prepaid','2/20','10-MAY-95','92140','Notepads',24.50,4,'DZ','31542','Pens',16.50,3,'DZ');

insert into poa values('TA-21976','05-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga Queensland 4938','Prepaid','2/20','11-MAY-95','92520','2 Ring Binders',4.50,10,'EA','31342','File Folders',27.25,3,'BX');

insert into poa values('TA-21977','08-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga Queensland 4938','Prepaid','2/20','12-MAY-95','11590','Copier Paper',92.75,24,'BX','08526','Liquid Paper',41.25,1,'DZ');

insert into poa values('TA-21978','08-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga Queensland 4938','Prepaid','2/20','12-MAY-95','78601','Toner',94.30,3,'EA',null,null,null,null,null);

```
/* FILE: labqzb.sql */
/*      Final Version */
```

```
drop table pob;
drop table poitmb;
create table pob(pono char(8),
                 podate date,
                 vno char(8),
                 vname char(30),
                 vstreet char(30),
                 vcity char(15),
                 vstate char(15),
                 vpc char(4),
                 trans char(15),
                 termdisc number,
                 termdays number,
                 duedate date);
```

```
create table poitmb(pono char(8),
                   itemno char(6),
                   idesc char(30),
                   uprice number,
                   qty number,
                   umeas char(5));
```

```
/* Insert Statements */
```

```
insert into pob values('TA-21971','02-MAY-95','45321','Koowang Office
Supplies','9812 Koala Parkway','Karanga','Queensland','4938','Prepaid',
2,20,'05-MAY-95');
```

```
insert into poitmb values('TA-21971','92140','Notepads',24.50,3,'DZ');
insert into poitmb values('TA-21971','31542','Pens',16.50,6,'DZ');
```

```
insert into pob values('TA-21972','02-MAY-95','45321','Koowang Office
Supplies','9812 Koala Parkway','Karanga','Queensland','4938','Prepaid',
2,20,'05-MAY-95');
```

```
insert into poitmb values('TA-21972','92520','2 Ring Binders',4.50,15,'EA');
insert into poitmb values('TA-21972','31342','File Folders',27.25,2,'BX');
```

insert into pob values('TA-21973','03-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga','Queensland','4938','Prepaid',2,20,'08-MAY-95');

insert into poitmb values('TA-21973','92264','Staplers',16.50,3,'EA');
insert into poitmb values('TA-21973','11590','Copier Paper',92.75,15,'BX');

insert into pob values('TA-21974','04-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga','Queensland','4938','Prepaid',2,20,'09-MAY-95');

insert into poitmb values('TA-21974','78601','Toner',94.30,4,'EA');

insert into pob values('TA-21975','05-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga','Queensland','4938','Prepaid',2,20,'10-MAY-95');

insert into poitmb values('TA-21975','92140','Notepads',24.50,4,'DZ');
insert into poitmb values('TA-21975','31542','Pens',16.50,3,'DZ');

insert into pob values('TA-21976','05-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga','Queensland','4938','Prepaid',2,20,'11-MAY-95');

insert into poitmb values('TA-21976','92520','2 Ring Binders',4.50,10,'EA');
insert into poitmb values('TA-21976','31342','File Folders',27.25,3,'BX');

insert into pob values('TA-21977','08-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga','Queensland','4938','Prepaid',2,20,'12-MAY-95');

insert into poitmb values('TA-21977','11590','Copier Paper',92.75,24,'BX');
insert into poitmb values('TA-21977','08526','Liquid Paper',41.25,1,'DZ');

insert into pob values('TA-21978','08-MAY-95','45321','Koowang Office Supplies','9812 Koala Parkway','Karanga','Queensland','4938','Prepaid',2,20,'12-MAY-95');

insert into poitmb values('TA-21978','78601','Toner',94.30,3,'EA');

```
/* FILE: labqzc.sql */  
/*      Final Version */
```

```
drop table poc;  
drop table poitmnc;  
drop table itmdesc;  
drop table vendorc;  
drop table pcodec;
```

```
create table poc(pono char(8),  
                podate date,  
                vno char(5),  
                trans char(15),  
                termdisc number,  
                termdays number,  
                duedate date);
```

```
create table poitmnc(pono char(8),  
                    itemno char(6),  
                    uprice number,  
                    qty number);
```

```
create table itmdesc(itemno char(6),  
                    idesc char(30),  
                    umeas char(5));
```

```
create table vendorc(vno char(5),  
                    vname char(30),  
                    vstreet char(30),  
                    vpc char(4));
```

```
create table vpcc(vpc char(4),  
                 vcity char(15),  
                 vstate char(15));
```

```
/* Insert Statements */
```

```
insert into vendorc values('45321','Koowang Office Supplies','9812 Koala  
Parkway','4938');
```

```

insert into vpcc values('4938','Karanga','Queensland');
insert into itmdesc values('92140','Notepads','DZ');
insert into itmdesc values('31542','Pens','DZ');
insert into itmdesc values('92520','2 Ring Binders','EA');
insert into itmdesc values('31342','File Folders','BX');
insert into itmdesc values('92264','Staplers','EA');
insert into itmdesc values('11590','Copier Paper','BX');
insert into itmdesc values('78601','Toner','EA');
insert into itmdesc values('08526','Liquid Paper','DZ');

insert into poc
values('TA-21971','02-MAY-95','45321','Prepaid',2,20,'05-MAY-95');

insert into poitmc values('TA-21971','92140',24.50,3);
insert into poitmc values('TA-21971','31542',16.50,6);

insert into poc
values('TA-21972','02-MAY-95','45321','Prepaid',2,20,'05-MAY-95');

insert into poitmc values('TA-21972','92520',4.50,15);
insert into poitmc values('TA-21972','31342',27.25,2);

insert into poc
values('TA-21973','03-MAY-95','45321','Prepaid',2,20,'08-MAY-95');

insert into poitmc values('TA-21973','92264',16.50,3);
insert into poitmc values('TA-21973','11590',92.75,15);

insert into poc
values('TA-21974','04-MAY-95','45321','Prepaid',2,20,'09-MAY-95');

insert into poitmc values('TA-21974','78601',94.30,4);

insert into poc
values('TA-21975','05-MAY-95','45321','Prepaid',2,20,'10-MAY-95');

insert into poitmc values('TA-21975','92140',24.50,4);
insert into poitmc values('TA-21975','31542',16.50,3);

insert into poc
values('TA-21976','05-MAY-95','45321','Prepaid',2,20,'11-MAY-95');

```

```
insert into poitmc values('TA-21976','92520',4.50,10);
insert into poitmc values('TA-21976','31342',27.25,3);
```

```
insert into poc
values('TA-21977','08-MAY-95','45321','Prepaid',2,20,'12-MAY-95');
```

```
insert into poitmc values('TA-21977','11590',92.75,24);
insert into poitmc values('TA-21977','08526',41.25,1);
```

```
insert into poc
values('TA-21978','08-MAY-95','45321','Prepaid',2,20,'12-MAY-95');
```

```
insert into poitmc values('TA-21978','78601',94.30,3);
```

APPENDIX E
Potential Errors and Possible Controls

General <i>Stimulus</i> Data entry via format of the SQL <i>insert</i> statements	<i>Potential Error</i> Various	<i>Possible Control</i> Input form that matches the PO
Field Checks <i>Stimulus</i> TA-21979	<i>Potential Error</i>	<i>Possible Control</i>
Due Date	Falls on a Sunday	1. Check for day of week (reasonableness)
Unit Price (92140)	Transcription	1. Look up price in a vendor- item table (reference file check) 2. Compare price to a previous order for the same item (reasonableness)
Quantity (92140)	Transcription	1. Compare quantity to previous orders for the same item (reasonableness)
Unit Price (31542)	Transposition	1. Look up price in a vendor- item table (reference file check) 2. Compare price to a previous order for the same item (reasonableness)
TA-21980 Terms	Transcription	1. Look up terms in the vendor table 2. Compare terms to terms on a previous order with the same vendor
Item No (92250)	Transposition	1. Look up item in a vendor-item or inventory table (reference file check) 2. Add a check digit to the item number

Item No (31542)	Transcription	1. Look up item in a vendor-item or inventory table (reference file check) 2. Add a check digit to the item number 3. Automatically display the description associated with the item number input (see #1)
Unit Price (31542)	Transposition	1. Look up price in a vendor-item table (reference file check) 2. Compare price to a previous order for the same item (reasonableness)
TA-21981 Vendor Number	Transcription	1. Look up vendor numbers in the vendor table 2. Add a check digit to the vendor number 3. Perform a format/template check and report errors for non-numeric bytes
Order Date	Transcription	1. Order date should be less than or equal to today's date
Terms	Transcription	1. Range check on discount percent
Due Date	Transcription	1. Use date type attribute
Unit Meas (92264)	Transcription	1. Look up unit of measure in a vendor-item or inventory table (reference file check)
Item No (115590)	Transcription	1. Look up item in a vendor-item or inventory table (reference file check) 2. Add a check digit to the item number 3. Format/template check for extra (or missing) bytes
Unit Price (08526)	Transcription	1. Look up price in a vendor-item table (reference file check) 2. Compare price to a previous order for the same item (reasonableness)

AT-51982			
Order Number	Transcription		1. The system should automatically generate the next order number 2. The system should check the reasonableness of a new order number based on recently entered order numbers
Due Date	Transcription		1. The due date should not precede today's date
Unit Measure (both)	Transcription		1. Look up unit of measure in a vendor-item or inventory table (reference file check)
Description (second)	Missing data		1. Look up unit of measure in a vendor-item or inventory table (reference file check) 2. Do not permit missing data (null input) for the description field
Record Checks			
TA-21981			
Order Date	Transcription		1. Compare Order Date with previous and next record where previous and next record relate to a sequential set of order numbers
Due Date	Transcription		1. The difference between the order date and the due date should be similar to that on similar orders to the same vendor or for similar products.
AT-51982			
Due Date	Transcription		1. The due date should be greater than the order date. 2. The difference between the order date and the due date should be similar to that on similar orders to the same vendor or for similar products.

Data Structure Changes

TA-21979

Street Address

Transposition in street
number

1. Create a separate vendor table ($\neg$ NF and 1NF)
2. Separate attribute for street number

TA-21980

Terms

Transcription

1. Separate terms into discount percent and discount days ($\neg$ NF)
2. Store discount percent and discount days in a vendor table ($\neg$ NF and 1NF)

TA-21981

Vendor Number

Transcription

1. Create a separate vendor table ($\neg$ NF and 1NF)

Terms

Transcription

1. Separate terms into discount percent and discount days ($\neg$ NF)
2. Store discount percent and discount days in a vendor table ($\neg$ NF and 1NF)

Third Item (08526)

Lack of space

1. Create separate PO and POITM tables ($\neg$ NF)

Other

AT-51982

Items

Duplicate

1. Define the primary key for POITM as the concatenation of order number and item number

APPENDIX F

Figures

PURCHASE ORDER Dixieland Foods 341 Tallokas Road Corvester, Georgia 30393 USA				
To: Diddledo Distributors 45321 9821 Koala Parkway Koolarooilly, Queensland 4398 Australia			Date: 1 September 1994 Order Number: KA-21979	
Transportation: Prepaid Air			Terms: 2/20	Due Date: 14 Sep 94
Item No	Description	Unit Price	Quantity	Unit Meas
92140	Filé Powder	89.50	25	CA
31542	Country Ham	65.60	320	EA

FIG. 1 - *Example Purchase Order*

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolarooilly, Queensland 4321				
To: Koowang Office Supplies 45321 9812 Koala Parkway Karanga, Queensland 4938			Date: 2 May 1995 Order Number: TA-21971	
Transportation: Prepaid		Terms: 2/20	Due Date: 5 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
92140	Notepads	24.50	3	DZ
31542	Pens	16.50	6	DZ

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolarooilly, Queensland 4321				
To: Koowang Office Supplies 45321 9812 Koala Parkway Karanga, Queensland 4938			Date: 2 May 1995 Order Number: TA-21972	
Transportation: Prepaid		Terms: 2/20	Due Date: 5 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
92520	2 Ring Binders	4.50	15	EA
31342	File Folders	27.25	2	BX

FIG. 2 - Example Purchase Order Data - Existing

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolarooilly, Queensland 4321				
To: Koowang Office Supplies 45321 9812 Koala Parkway Karanga, Queensland 4938			Date: 3 May 1995 Order Number: TA-21973	
Transportation: Prepaid		Terms: 2/20	Due Date: 8 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
92264	Staplers	16.50	3	EA
11590	Copier Paper	92.75	15	BX

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolarooilly, Queensland 4321				
To: Koowang Office Supplies 45321 9812 Koala Parkway Karanga, Queensland 4938			Date: 4 May 1995 Order Number: TA-21974	
Transportation: Prepaid		Terms: 2/20	Due Date: 9 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
78601	Toner	94.30	4	EA

FIG. 2 (cont.) - Example Purchase Order Data - Existing

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolarooilly, Queensland 4321				
To: Koowang Office Supplies 45321 9821 Koala Parkway Karanga, Queensland 4938			Date: 5 May 1995 Order Number: TA-21975	
Transportation: Prepaid		Terms: 2/20	Due Date: 10 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
92140	Notepads	24.50	4	DZ
31542	Pens	16.50	3	DZ

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolarooilly, Queensland 4321				
To: Koowang Office Supplies 45321 9812 Koala Parkway Karanga, Queensland 4938			Date: 5 May 1995 Order Number: TA-21976	
Transportation: Prepaid		Terms: 2/20	Due Date: 11 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
92520	2 Ring Binders	4.50	10	EA
31342	File Folders	27.25	3	BX

FIG. 2 (cont.) - Example Purchase Order Data - Existing

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolarooilly, Queensland 4321				
To: Koowang Office Supplies 43521 9812 Koala Parkway Karanga, Queensland 4938			Date: 8 May 1995 Order Number: TA-21977	
Transportation: Prepaid		Terms: 2/20	Due Date: 12 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
11590	Copier Paper	92.75	24	BX
08526	Liquid Paper	41.25	1	DZ

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolarooilly, Queensland 4321				
To: Koowang Office Supplies 45321 9812 Koala Parkway Karanga, Queensland 4938			Date: 8 May 1995 Order Number: TA-21978	
Transportation: Prepaid		Terms: 2/20	Due Date: 12 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
78601	Toner	94.30	3	EA

FIG. 2 (cont.) - Example Purchase Order Data - Existing

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolaroopilly, Queensland 4321				
To: Koowang Office Supplies 45321 9821 Koala Parkway Karanga, Queensland 4938			Date: 9 May 1995 Order Number: TA-21979	
Transportation: Prepaid		Terms: 2/20	Due Date: 14 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
92140	Notepads	44.50	25	DZ
31542	Pens	15.60	10	DZ

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolaroopilly, Queensland 4321				
To: Koowang Office Supplies 45321 9812 Koala Parkway Karanga, Queensland 4938			Date: 10 May 1995 Order Number: TA-21980	
Transportation: Prepaid		Terms: 3/30	Due Date: 16 May 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
92250	2 Ring Binders	4.50	18	EA
31542	File Folders	22.75	12	BX

FIG. 3 - Example Purchase Order Data - For Input By Participants

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolaroopilly, Queensland 4321				
To: Koowang Office Supplies 4B521 9812 Koala Parkway Karanga, Queensland 4938			Date: 21 May 1995 Order Number: TA-21981	
Transportation: Prepaid		Terms: 7/20	Due Date: 31 June 1995	
Item No	Description	Unit Price	Quantity	Unit Meas
92264	Staplers	16.50	5	DZ
115590	Copier Paper	92.75	20	BX
08526	Liquid Paper	11.25	2	DZ

PURCHASE ORDER Diddledo Distributors 341 Doonah Road Doolaroopilly, Queensland 4321				
To: Koowang Office Supplies 45321 9812 Koala Parkway Karanga, Queensland 4938			Date: 15 May 1995 Order Number: AT-51982	
Transportation: Prepaid		Terms: 2/20	Due Date: 19 May 1994	
Item No	Description	Unit Price	Quantity	Unit Meas
78601	Toner	94.30	2	CA
78601		94.30	3	CA

FIG. 3 (cont.) - Example Purchase Order Data - For Input By Subjects

VITA

Paul L. Bowen was born to Helen Duboise Bowen and W. Paul Bowen, Jr. on June 9, 1951. He attended Blue Grass Elementary School in Knox County, Tennessee. Paul graduated from The McCallie School, Chattanooga, Tennessee, in June 1969. He received a Bachelor of Science in Industrial Management from the Georgia Institute of Technology in March 1973. Paul earned a Master of Business Administration with concentrations in Finance and Management Science from The University of Tennessee in March 1976. He completed the requirements for a Master of Accountancy from The University of Tennessee in June 1990. Paul received a Doctor of Philosophy in Business Administration with a concentration in Accounting and a support area in Computer Science from The University of Tennessee in December 1992. He received a Master of Science in Computer Science from The University of Tennessee in December 1995.

Paul worked for Valley Fidelity Bank from June 1973 to September 1975 and for the Bank of Knoxville from June 1976 to October 1980. Paul worked as a systems analyst and project manager for the Oak Ridge National Laboratory from October 1980 to August 1988. He taught accounting information systems for the Department of Accounting & Business Law at The University of Tennessee from January 1988 to August 1992 and for the School of Accountancy at Auburn University from September 1992 to June 1993. Since June 1993 he has taught various information systems subjects for the Department of Commerce at The University of Queensland.

Paul is a Certified Public Accountant and a member of Beta Alpha Psi, Beta Gamma Sigma, the American Accounting Association, the American Statistical Association, the Association for Computing Machinery, the Association for Information Systems, the Australian Computer Society, the Institute for Operations Research and the Management Sciences (INFORMS), and the Institute of Electrical and Electronics Engineers.