Database design is a complex task for end users and novice designers. User views are an excellent input source for database design. This research investigates the effect of certain characteristics of user views on designer performance in the logical database design task. The study examines the extent to which performance of nonexpert designers engaged in normalizing user views and eliminating derived relationships can be predicted by examining the structure of user views. Two constructs that capture the complexity of a user view are defined - degree of nesting and derivation span. User views which varied on these parameters were included in a case. Subjects, who were enrolled in a database class and had learned the view decomposition approach, were asked to conduct logical database design of the case. Each view was graded using a predefined scheme. For each subject, there was a set of scores obtained for each user view. A multiple regression analysis using scores as the dependent variable indicated that both independent variables - degree of nesting and derivation span - significantly explained designer performance in normalizing the user views. Further, subjects had considerable difficulty in detecting derived relationships. The findings suggest that the degree of nesting and derivation distance cause complexity in database design tasks and lower designer performance.

Usability studies in conceptual and logical database design have generally considered four classes of variables: data model, human, task, and performance (Batra and Srinivasan, 1992). None of the usability studies have focused on the task. Generally, the data models are carefully chosen, subjects are classified based on cognitive variables or skill, and performance is meticulously gauged by univariate or multivariate measures together with a grading scheme. However, task is usually designed arbitrarily and as long as the task bears resemblance to the criterion setting, it is considered acceptable. Characteristics of the task are rarely varied systematically. This results in a narrow interpretation of user performance.

An important task variable is task complexity. If task complexity is high, user performance is low. Task complexity in the later stages of systems development, e.g., coding, has been addressed by software engineering literature. However, similar metrics have not been proposed for tasks at the analysis and design stages. Wrigley and Dexter (1991) did provide a metric for estimating the size of systems requirements based on the number of entities, relationships, input events, and output events. This metric is aggregate and may be useful for estimating costs, lines of code, and number of programmers required for the project. However, if one needs to gauge the cognitive difficulty of modeling of individual user view, one must examine the characteristics of user views. Such a metric will indicate what kind of user views pose difficulty for nonexpert designers.

The purpose of this paper is to estimate the complexity of user views based on their characteristics. User views, which are forms and reports employed in an application, are one of the most reliable inputs for conducting database analysis and design. This study used the relational data model (Codd, 1970), which is currently the most popular model for database systems in a study involving novice designers. In relational databases, the logical design is generally based on normalization, which attempts to minimize redundancy and avoid update anomalies. When the input is user views, that is, forms and reports used in an application, the use of normalization leads to decomposition of user views into normalized relations. The process of modelling user views by using normalization is called view decomposition. A user view seems conducive to effective relational design since it can be written as an unnormalized table which can then be normalized. Figure 1 shows an example of a user view.
Although the tabular structure of a user view is helpful in designing a relational database, certain characteristics can make a user view complex. This study defines and considers two characteristics of a user view - degree of nesting and derivation span - to determine the extent to which these variables explain designer performance. Nesting implies set data, which is similar to the notion of multivalued data and can result in complex relationships. Also, user views that house attributes belonging to distantly related objects can be difficult to model since the nature of relationship may be obscured by intermediate entities and relationships. Derivation span is a measure of this distance. A high derivation span can not only lead to difficulty in normalizing a user view, it is also possible that the derived relationships are erroneously retained in the integrated logical data model.

The study reported in this paper considers user views as the presentation format and investigates performance of non-expert designers engaged in a logical database design task using the view decomposition methodology. Specifically, the study determines the extent to which certain characteristics of user views can explain the variation in designer performance. The next section discusses empirical literature pertinent to the area of conceptual and logical database design. The third section presents the view decomposition approach and introduces the notion of degree of nesting and derivation span. The fourth section describes the methodology. The fifth section presents and discusses the results. The sixth section discusses the implications of the findings. The final section summarizes the findings and provides suggestions for future research.

Literature Survey

Conceptual and logical database design tasks are important but difficult tasks. Conceptual design cannot be entirely performed by automatic tools, and the designer has full responsibility for the process of understanding and transforming requirements into conceptual schemas (Batini, Ceri and Navathe, 1992). This also applies to logical design if not preceded by conceptual design, that is, if the logical design stems directly from user requirements. If the designer is a novice, the design can be error-prone (Storey, Thompson, and Ram, 1995). Schiffman, Meile and Igbaria (1992) report that end-user programmers use the largest number of software and perform the largest number of tasks. It can be expected that such end-users also conduct database design for their applications. However, they are likely to be novices and, probably, commit a number of mistakes.

Many usability studies have been conducted to investigate designer performance in conceptual and logical design. These studies have typically compared designer performance between two or more data models. Due to the popularity of relational DBMS, these studies have invariably considered the relational data model as one of the treatments. Past studies have focused on data models used, but not on the experimental task itself. Consequently, little is known about the aspects of task that affect the ability of a designer to correctly model an application. A brief survey of the salient studies along with the tasks employed follows.

Brosey and Shneiderman (1978) compared relational and hierarchical models using instance diagrams. Durding, Becker and Gould (1977) presented subjects with a group of words and asked them to organize the words as a hierarchy, list or network. Juhn and Naumann (1985) focused on the user validation task and presented subjects with a database using different data representations. Shoval and Even-Chaime (1987) employed the relational data model to compare normalization with the information analysis approach using data flow diagrams. Batra, Hoffer and Bostrom (1990) presented subjects with a text description of the application and compared the relational model with the extended entity relationship (EER) model. A similar task representation was used by Batra and Kirs (1993) who compared the ER model (Chen, 1976) with data aggregation, and by Bock and Ryan (1993), who compared the EER model with object oriented data model. Hoffer (1982) presented subjects with a functional description of an organization situation. Using four data representations, Palvia (1991) provided subjects with data files displayed on paper to test their understanding of the database. Jahnennpaa and Machesky (1989) provided subjects with sample output reports and found that the logical data structure (LDS) model, as compared to the relational model, led to more accurate representations. Kim and March (1995) found that, compared to extended ER, NIAM led to inferior analyst performance.

A review of past literature indicates that these studies focused on the data models but not on the task. User views were sometimes considered as the presentation format, but the studies did not relate performance to any characteristics of user views.

In the context of human computer interaction, human performance can be considered as dependent on the system, user characteristics, and task (Jenkins, 1982). Specifically, in database design, the performance is a result of interaction between the data model, designer characteristics, and task. Among the three categories of independent variables, task characteristics have not been studied. User views, which may be the most accurate source of input for database design, have rarely been employed. Specifically, the structure of user views has not been examined to determine what makes it easy or difficult. Research on the effect of task characteristics on designer performance can lead to formulating training strategies for addressing the difficulty faced by designers.

Research Problem

This paper reports a study which focused on how the task characteristics affect designer performance in the conceptual data modelling task. The task involved normalization of user
Related Content

NetCube: Fast, Approximate Database Queries Using Bayesian Networks
[www.igi-global.com/chapter/netcube-fast-approximate-database-queries/28567?camid=4v1a](www.igi-global.com/chapter/netcube-fast-approximate-database-queries/28567?camid=4v1a)

Fuzzy and Probabilistic Object Bases
[www.igi-global.com/chapter/fuzzy-probabilistic-object-bases/4807?camid=4v1a](www.igi-global.com/chapter/fuzzy-probabilistic-object-bases/4807?camid=4v1a)

Understanding the Role of Use Cases in UML: A Review and Research Agenda
[www.igi-global.com/article/understanding-role-use-cases-uml/3256?camid=4v1a](www.igi-global.com/article/understanding-role-use-cases-uml/3256?camid=4v1a)

Managing Large Rule-Bases
[www.igi-global.com/article/managing-large-rule-bases/51165?camid=4v1a](www.igi-global.com/article/managing-large-rule-bases/51165?camid=4v1a)