

Learning from Errors as a Pedagogic Approach for Reaching a Higher Conceptual Level in Database Modeling

Adi Katz^(✉) and Ronit Shmallo

Sami Shamoon College of Engineering (SCE), Ashdod, Israel
{adis, ronitsl}@sce.ac.il

Abstract. We apply a pedagogic approach named learning from errors (LFE) to the area of relational database modeling. Database modeling is a complex cognitive process characterized by a high level of element interactivity. Finding an appropriate pedagogy to teach database modeling is a challenge for information systems educators. One of the challenges that practitioners meet is the need to help database students shift between different levels of abstraction. We metaphorically treat the LFE approach as a bridge over the gulf of abstraction levels. Errors have a powerful potential in education, to encourage students to move to a deeper processing level of the course material. We use Rasmussen's three levels of human performance model to explain and demonstrate the promising potential of the LFE approach in database modeling.

Keywords: Conceptual modelling · Databases · Learning from errors · Cognitive conflict · Levels of abstraction

1 Introduction

Teaching conceptual database modelling is a challenge to information systems (IS) educators. We apply a pedagogic approach named “learning from errors” (LFE) to the area of relational database design, to more effectively deliver and instruct the cognitively complex material of database modeling. The approach was already applied successfully in the areas of mathematics [1], physics [2] and computer science [3] education but not yet applied in the area of database design.

We recently examined the difficulties that students encounter in the activity of conceptual modelling by analyzing their solutions to a database course exercise, in the form of a textual scenario, and mapped their errors into categories [4]. Our plan is to design learning lessons, class and homework activities that utilize these errors for the learning process. We will then use our learning activities in an experiment, in which we intent to compare between the traditional database teaching approach and a combination of the traditional approach with the LFE approach to test the effectiveness of the latter.

In the current paper, our intention is to explain why the LFE approach that was already found to be beneficial in different educational areas is particularly powerful in the process of learning database conceptual modeling. We use Rasmussen's three levels

of human performance model [5] as a theoretic framework, and we stress the importance of making transitions across abstraction levels in training database students. To demonstrate the promising potential of the LFE approach in database conceptual modeling activities, we present two examples; each is part of a different class exercise. The exercises we use for demonstration are of a type called “erroneous examples” [1, 6]. Of course, the literature describes other activities that utilize errors for the learning process such as “self-explanations” [7], “mistaken statements” and comparison of students’ solutions to those of an expert [2].

2 Teaching Database Modelling as a Challenge to IS Educators

In database courses, students learn various activities that are related to defining, creating and manipulating databases. In each activity, database designers are required to observe data at three different levels of abstraction: internal (physical), conceptual (central/logical) and external (user view). For example, in database definition, data defining language (DDL) is divided into physical, central and view DDL. We focus on the conceptual level, in which a relational schema is created. A database schema includes a list of entities, attributes, relationships, user operations, database semantic constraints (business rules), and interrelation referential integrity constraints which are used to maintain consistency of reference among records from different relations [8]. Different levels of abstraction are also reflected in database course material, ranging from the theory level (e.g. learning the notion of a reference integrity constraint) to the practical level (e.g. writing a DDL code such as “on delete cascade” to ensure that a reference integrity constraint would not be violated). Educators have a twofold challenge in teaching relational database design concepts; they need to deliver the theory of relational databases and also provide students with practical skills to perform effectively in real life [9]. In our years of experience in teaching the database course, there is a gulf between theoretical concepts and their meaning at the concrete level of an organization’s needs, demands and constraints.

In a typical database course, many elements are taught. Among them are: relational model principles, entity relationship (ER) model, key types, SQL, normalization rules, and optimization. Database modeling is a process characterized by a high level of element interactivity, since the different topics are understood and learned with reference to other topics, and cannot be considered independently. One of the challenges of database design instructors is dealing with the fact that the high-element interactivity material covers various activities that are related to the different levels of abstraction.

The contents of long term memory are sophisticated cognitive structures known as schemas that make up our knowledge base [10]. Complex schemas consist of huge arrays of interrelated elements [11]. High-element interactivity material is difficult to understand. Indeed, the elements can be learned individually, but they cannot be fully understood until all aspects and their interactions are processed simultaneously. Intrinsic cognitive load is affected by element interactivity and extraneous cognitive load is affected by instructional design [12]. This is why instructional approaches intended to reduce cognitive load are primarily effective when element interactivity is

high. There is an additional type of cognitive load, referred to as germane or effective cognitive load, which is also influenced by the instructional designer, since he or she determines the learning activities and the way in which information is presented. Whereas extraneous cognitive load interferes with learning, germane cognitive load enhances learning, since cognitive resources are being devoted to schema acquisition and automation [12, 13]. A simultaneous processing of all essential elements must occur eventually despite the high-intrinsic cognitive load because it is only then that understanding takes place [13]. At the following sections, we explain and demonstrate how the LFE approach creates a germane or effective cognitive load that enhances simultaneous processing of all essential elements in database modeling.

3 Learning from Errors in Educating Database Modeling

There are previous pedagogic works in database modeling with attempts to deal with the challenge of effectively delivering theory and practice and bridging the gulf between different levels of abstraction. Such interesting attempts are the integrated spiral approach [14] and the cognitive apprenticeship based approach [9]. We offer an alternative approach called *learning from errors* (LFE) as an aid for tying elements that are highly integrated, but belong to different levels of abstraction. Empirical results have shown that LFE promotes the learning process [6, 7], and the approach was already applied successfully in mathematics [1], physics [2] and computer science [3]. Errors are often treated negatively, but our approach takes advantage of errors and utilizes them in educating relational database modeling as a bridge to transfer between different levels of abstraction. We claim that in the area of database design, the LFE approach may push students to a cognitive level in which connections between different elements are created to form viable mental models of database conceptualization.

Errors may trigger cognitive conflicts or dissonances, which, in turn, yield a process of reflection and critical thinking [1]. Learning activities should raise cognitive conflicts, since the point at which they arise, are the ones to yield the recognition of the source of the error [3]. A cognitive conflict driven learning approach encourages students to engage with the learning materials and motivate them to construct appropriate and viable mental models [15]. Erroneous situations fail to conform what needs to be, and therefore errors are subjectively experienced as conflicts between what the learner believes ought to be true, and what he or she perceives to be the case [16]). We claim that the LFE approach bolsters germane (effective) cognitive load, because it creates motivation to explore the learning material in more depth, in order to solve a cognitive conflict created by the deliberate errors presented. This exploratory process promotes new insights, and then knowledge in the long-term memory is updated to a schema with connections between the different elements that are related to database modeling and design.

The difference between a professional and a novice is that the latter hasn't acquired the schemas of an expert. Learning requires a change in the schematic structures of long term memory. During the learning process, as the learner becomes increasingly familiar with the material, performance progresses from clumsy, slow and difficult to smooth and effortless [10]. The transition between performance levels is in line with Rasmussen's model [5] of human performance, as we explain in the following section.

4 Rasmussen's Three Level Model of Human Performance

Rasmussen distinguishes three categories of human behavior: skill-, rule-, and knowledge-based [5]. We follow Rasmussen's three-level framework to explain the promising potential of the LFE approach in pushing students to a higher level of learning process of conceptual modeling. We do so by demonstrating the categories of human behavior on database conceptual modeling activities. All levels are expressed in learning database design; however, most interesting is the potential of the LFE approach to shift students from the rule-based to the higher knowledge-based level of learning, in which complex schemas of huge arrays of interrelated elements in the long-term memory are formed, transformed, reorganized and updated.

4.1 Skill-Based Behavior

The skill-based behavior represents sensory-motor actions. For example, when students listen to a teacher in a database course class, and during so they write in their notebooks, they unconsciously and rapidly move the hand that holds the pen horizontally across the notebook lines and they move their hand vertically to the beginning of the next line each time they notice that the point of the pen is about to reach the edge of the page. This is a highly integrated pattern of behavior, done with almost no conscious attention or control. Also, translating the sounds of the teacher's utterances to a graphic textual display in the notebook is also a smooth, automated and coordinated behavior. In both examples, the students' senses are only directed towards the environmental aspects that are needed to update and orient their internal map and their activities are a sequence of skilled acts which are composed from a large repertoire of automated sub-routines.

4.2 Rule-Based Behavior

At the next level of rule-based behavior, students learn and store relational database related rules. For example, referring to Codd's normalization rules [17], they are instructed to follow a set of defined rules, in order to meet normal forms (NF). They learn how to apply these rules for decomposing existing relations (*design by analysis* approach, [8]) or for constructing relation schemas according to given textual scenarios (*design by synthesis* approach, [8]). Usually they are taught how to use these rules according to identified functional and multivariate dependencies between attributes. Then, existing relations or textual scenarios will release these stored know-how rules. In most exercises related to normalization, students are required to apply the rule that fits, according to similar situations or examples already taught. Figure 1 demonstrates an erroneous example of a "students" relation with arrows that express dependencies between attributes. The inclusion of all address attributes in the relation violates the 3rd normal form (3NF), because it allows a transitive dependency of zip code on the other address attributes. The arrows pattern serves as a *sign*, which is the perceived environmental information at the rule-based level [5], for identifying a transparent

dependency that violates 3NF, and for activating predetermined actions of decomposing the relation according to the learned solution. Actually, even if the relation's attributes were totally meaningless or independent of a given scenario (e.g. A, B, C... G), the pattern of 3NF violation would still be apparent and easily identified. At this level of processing, the rule is recalled from memory after pattern recognition occurs.

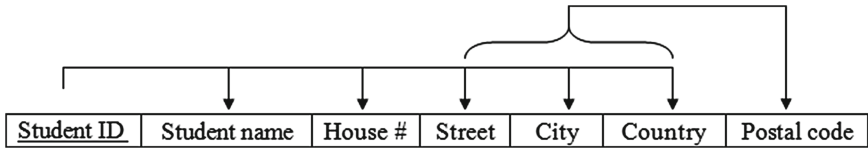


Fig. 1. Example for dependencies between attributes that violate 3NF

4.3 Knowledge-Based Behavior

At the highest level in terms of attention resources consumption, are knowledge-based behaviors, in which students are required to functionally understand and analyze the environmental information. Following the previous NF example, at this level students understand the importance of the rules, and the meaning or consequences of violating them.

As aforementioned, errors may trigger cognitive conflicts that force students to deeply understand concepts, and relations between concepts. Erroneous examples are a type of LFE activities that have a high potential for raising cognitive conflicts. In the case of teaching Normalization, there are usually various exercises for identification of NF rules. They usually present relations that violate these rules, and students are asked to repair the violations by adding and omitting attributes from the given relations, and by correcting erroneous keys. In most cases, erroneous examples indeed serve as situations with a potential to recall stored rules in order to repair accordingly. In many cases, the cognitive conflict accompanied by erroneous examples will motivate students to move to a higher conceptual level and then different plans are considered. For demonstration, we refer to the normalization example of the erroneous relation in Fig. 1. Presenting the “students” relation should raise the question of whether it is better to repair the relation or decide on de-normalization, an explicit violating of the 3NF rule. The student needs to predict the different effects that are related to the different possible solutions. In favor of violating the 3NF rule, considerations that are raised may be the user's perspective (it is more natural to see all address components together), and the system's performance (keeping all attributes in one relation spares JOIN actions, that would be required for queries). In favor of repairing the violation, is the consideration of saving storage space (violating 3NF wastes storage space because of data repetitions). The process of considering different consequences pushes the students to cognitively form connections between the elements, and to understand that in this case there is a tradeoff type of interaction between them. The conceptual level of abstraction, (a normalization rule) is tied to aspects related to internal abstraction level considerations (storage space and system performance) and to external level considerations (user view).

According to Rasmussen [5], there can be varying degrees of training for a person that is in a task depending on variations and disturbances. Constructively using errors in class and homework activities create more opportunities for cognitive conflict when learning initially and when practicing. The cognitive conflicts may motivate the students to move to a deeper learning process. In our view, Rasmussen’s theoretical perspective is related to the conception of hierarchy of processing stages, which vary in depth. Greater depth implies a greater degree of semantic or cognitive analysis. Trace persistence in our memory is a function of depth of analysis, with deeper levels of analysis associated with more elaborate, longer lasting, and stronger traces [18]. Compared to the rule-based level, the knowledge-based is higher in processing depth. Educators can and should manipulate and influence the learning process to further elevate the students from simply rehearsing rules to a deeper analysis and a more elaborative level of semantics. Errors have a positive potential in education, being a source of critical and creative thinking, serving as a lever for a deeper learning process [1]. We believe that the powerful potential of errors exists in many topics and areas of learning, including database modeling. The LFE approach constructively uses errors and capitalizes on them as a departure point for an inquiry about the nature of database and about the various concepts and aspects related to database design.

5 Errors as means for Shifting Between Levels of Abstraction

In database modeling training, modeling aids are required to facilitate an easy transition from one level of abstraction to another, and to allow a view of multiple levels simultaneously [19]. We see how errors and therefore the various LFE activities can serve as such modeling aids.

Error or fault events are identified with reference to intended states, normal functions, or other variants of propose or meaning. *Causes* of improper functions depend upon changes in the physical world, and are explained “bottom-up” while *reasons* for proper function are derived “top-down” from the functional purpose. The difference between causes and reasons shows different levels of an abstraction hierarchy [5]. We claim that erroneous events will force the student to consider the functions of a database system at several levels, and that they will have to go through different information flow paths, top-down and bottom-up. Therefore, errors have a powerful potential in education, since they encourage students to get on a metaphorical bridge over the gulf of abstraction levels in database modeling and design, and to transition between levels.

In the following example appearing in Fig. 2, we demonstrate how an erroneous database schema (a partial schema, taken from an ‘erroneous example’ exercise type) can serve as a bridge over the gulf of abstraction levels, and how in particular it encourages students to think of the hierarchical nature of a database schema, and about the meaning of referential constraints. Figure 2 shows a partial solution for an online flower shop scenario with diagrammatic displays of foreign key-primary key (FK-PK) relations. We deliberately added two erroneous FK-PK relations and deliberately omitted a necessary FK-PK relation. In “Bouquet Order Details”, the attributes ID, Catalog Num, and Date should be defined as a FK in reference to “Bouquet Orders”,

but instead, attribute ID is defined as a FK in reference to “Customers” and Catalog Num is separately defined as a FK in reference to “Bouquets”. The omission of the FK-PK relations between “Bouquet Order Details” and “Bouquet Orders” enables abnormal data entries such as the two records appearing in “Bouquet Order Details”; the first showing a bouquet (456) along with a customer ID (111) who did not actually order it, and the second showing a bouquet order that did not occur at the inserted date (4/20/2015). In other words, records in “Bouquet Order Details” refer to none existing records in “Bouquet Orders”. These inconsistent records presented at the physical world are expected to raise a cognitive conflict that will encourage an inspection, which with proper and effective teacher guidance would lead to thoughts about the meaning of imposing referential integrity constraints, and about the hierarchic nature of a database schema. In the process of finding a solution for the deviation from the valid state of system integrity, students will be encouraged to find the cause for the spurious records, a referential integrity constraint not specified between “Bouquet Order Details” and “Bouquet Orders”, and the explanation at this level will be “bottom-up”. A discussion about the hierarchical structure of a database schema for proper function will be derived “top-down”, with thoughts about the meaning or semantics of the schema and about the gradient transition from parent to child levels of the hierarchy: starting from entities (things) in the real-world that are clear and straightforward and gradually going down towards relations that represent entities that are relatively more abstract and more complex to understand. Discussing relationships between entities may open another discussion about relationship cardinality ratios (1:1, 1: N, M: N), and tying these types to the gradient shift from clear and tangible entities to entities that are more abstract and complex. Referring to the example in Fig. 2, the entities are graduating from top to bottom: since customers and bouquets are in a M: N relationship, but cannot be directly connected in a relational database, “Customers” and “Bouquets” are both parents of “Bouquet Orders”. “Bouquet Orders” in turn should be the parent of “Bouquet Order Details”. “Customers” and “Bouquets” are tangible and more easily understood than “Bouquet Orders”, and the same can be said respectively about “Bouquet Orders” and “Bouquet Order Details”.

In another more practical level of abstraction is the discussion about writing the DDL to specify only valid constraints. First, there is a need to add the DDL expression to respectively enforce the missing constraint. This addition involves a decision about the option that would deal best with a violation caused by deleting or updating a bouquet order (reject, cascade, set default or set null). Then, a question should be raised whether to remove or to keep the constraints defining “Bouquet order details” as a referencing relation to both “Customers” and “Bouquets”. The discussion regarding this question would lead to understanding of the system’s actions (checks) that occur each time a record is inserted to the referencing (child) relation, “Bouquet Order Details”, and each time a record is deleted from the referenced (father) relations, “Customers” or “Bouquets”. A hierarchical structure should include ‘father-child’ relations, but ‘grandfather-grandchild’ relations are redundant since there is an implicit FK relationship from a child to his grandfather through the father. This redundancy adds unnecessarily system checks of compliance with the defined referential integrity constraints. Therefore, this is an opportunity to also understand the practical consequences of (system performance) of incorrect semantics (conceptual schema).

Using an erroneous example exercise, we demonstrated how constructively incorporating errors in class and homework assignments can trigger a simultaneous processing of interrelated elements that belong to different levels of abstraction. This processing would eventually lead to the essential high level of understanding of conceptual database modeling.

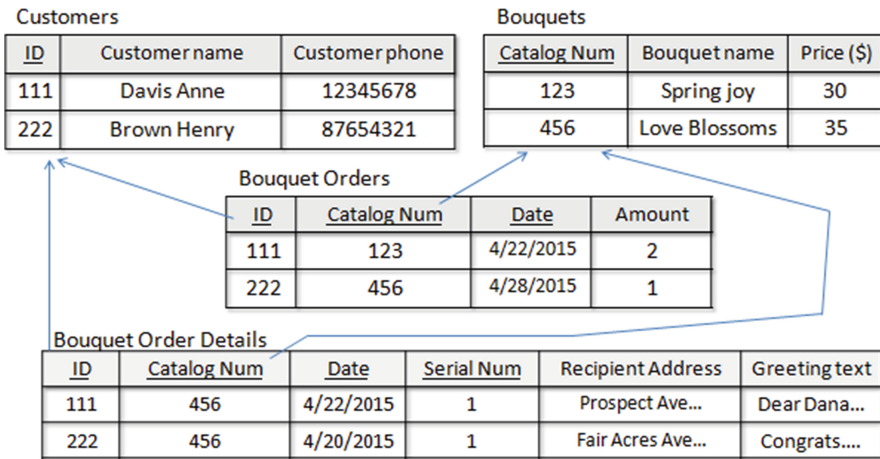


Fig. 2. A partial schema of a flower shop scenario to demonstrate erroneous example exercises

6 Conclusions: Future Directions and Instructional Implications

We introduced LFE as a new pedagogic approach to the area of database design. LFE has already been successfully applied in several areas, but not yet applied in the area of relational database conceptual modelling. We explained why a constructive use of errors is powerful in the process of training database conceptual modeling using Rasmussen’s human performance model and focusing on the ability of errors to help database students make transitions across the different levels of abstraction.

We intend to empirically test whether the LFE approach is effective in educating conceptual modelling of relational databases. Our inquiry has three key phases:

1. Mapping errors of students, driven from solutions to scenario tasks. We already mapped students’ errors using solutions to a scenario task in a form of a textual description (scenario) of an organization with certain needs and constraints. The students were required to identify relations, draw FK-PK relationships between relations and correctly apply database normalization rules. We analyzed the solutions and mapped the errors into categories and sub-categories [4].
2. Designing an educational program with learning activities that utilizes the errors found in the mapping phase. We are currently developing a number of class and homework assignments that focus on detecting errors, learning their consequences

and solving them. The assignments involve ideas from previous work such as “erroneous examples” [6] and “self-explanations” [7].

3. Conducting an experimental study that compares a group of students who will learn conceptual modelling through the traditional database teaching approach (a control group) to a group of students who will be exposed to a learning process that would have a combination of the traditional teaching approach with the LFE approach (an experimental group). The distinct groups belong to the same academic institution that has two separate campuses. The database course in both campuses is taught at the same semester, by the same lecturer, with the same background material, and so would be the exam questions. The results of the comparison will enable us to reach a conclusion regarding the effectiveness of the LFE approach in educating database modelling. We expect to find higher-quality solutions in the experimental group of students, and also a higher level of student satisfaction from the learning process.

If the LFE approach proves itself to be effective, it could be used as a guide for designing both database educational programs, as well as a guide for designing computerized learning supporting tools for an effective learning process that will lead to high performance in the task of database modeling.

References

1. Borasi, R.: *Reconceiving Mathematics Instruction: A Focus on Errors*. Ablex Pub, Norwood (1996)
2. Yerushalmi, E., Polingher, C.: Guiding students to learn from mistakes. *Phys. Educ.* **41**(6), 532–538 (2006)
3. Ginat, D., Shmallo, R.: Constructive use of errors in teaching CS1. In: *Proceedings of the 44th ACM Technical Symposium on Computer Science Education*, pp. 353–358 (2013)
4. Katz, A., Shmallo, R.: Improving relational data modeling through learning from errors. In: *Proceedings of the IADIS Multi Conference of Computer Science and Information Systems MCCSIS, Theory and Practice in Modern Computing TPMC*, pp. 198–202. IADIS Press, Las Palmas de Gran Canaria, Spain (2015)
5. Rasmussen, J.: Skills, rules, and knowledge; signals, signs, and symbols, and other distinctions in human performance models. *IEEE Trans. Syst. Man Cybern.* **3**, 257–266 (1983)
6. Grobe, C.S., Renkl, A.: Finding and fixing errors in worked examples: can this foster learning outcomes? *Learn. Instr.* **17**(6), 612–634 (2007)
7. Siegler, R., Chen, Z.: Differentiation and integration: guiding principles for analyzing cognitive change. *Dev. Sci.* **11**(4), 433–448 (2008)
8. Elmasri, R., Navathe, S.: *Fundamentals of Database Systems*, 6th edn. Addison-Wesley, Boston (2011)
9. Al-Dmour, A.: A cognitive apprenticeship based approach to teaching relational database analysis and design. *J. Inf. Comput. Sci.* **7**, 2495–2502 (2010)
10. Sweller, J.: Cognitive load during problem solving: effects on learning. *Cogn. Sci.* **12**, 257–285 (1988)
11. Artino Jr., A.R.: Cognitive load theory and the role of learner experience: an abbreviated review for educational practitioners. *Aace J.* **16**(4), 425–439 (2008)

12. Sweller, J., van Merriënboer, J.J.G., Paas, F.: Cognitive architecture and instructional design. *Educ. Psychol. Rev.* **10**, 251–296 (1998)
13. Paas, F., Renkl, A., Sweller, J.: Cognitive load theory and instructional design: recent developments. *Educ. Psychol.* **38**(1), 1–4 (2003)
14. Watson, R.T.: The essential skills of data modeling. *J. Inf. Syst. Educ.* **17**(1), 39–41 (2006)
15. Ma, L., Ferguson, J., Roper, M., Ross, I., Wood, M.: Improving the mental models held by novice programmers using cognitive conflict and Jeliot visualisations. *ACM SIGCSE Bull.* **41**(3), 166–170 (2009)
16. Ohlsson, S.: Learning from performance errors. *Psychol. Rev.* **103**(2), 241 (1996)
17. Codd, E.F.: Further normalization of the data base relational model. In: Rustin, R.J. (ed.) *Data Base Systems: Courant Computer Science Symposia Series 6*. Prentice-Hall, Englewood Cliffs (1972)
18. Craik, F.I., Lockhart, R.S.: Levels of processing: a framework for memory research. *J. Verbal Learn. Verbal Behav.* **11**(6), 671–684 (1972)
19. Srinivasan, A., Te’eni, D.: Modeling as constrained problem solving: an empirical study of the data modeling process. *Manag. Sci.* **41**(3), 419–434 (1995)